

**Universidad Autónoma Metropolitana
Unidad Azcapotzalco**

**División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación**

Reporte Final

**Sistema para la administración de información en pequeños
negocios.**

Proyecto Tecnológico
Trimestre 2018-Invierno

Adriana Camal Mendoza
208243159
al208243159@alumnos.azc.uam.mx

Asesor

Mtro. Josué Figueroa González
Profesor Asociado
Departamento de Sistemas
jfgo@correo.azc.uam.mx

Coasesor

Dr. Alejandro Reyes Ortiz
Profesor Asociado
Departamento de Sistemas
jaro@correo.azc.uam.mx

11 de abril del 2018

Yo, Josué Figueroa González, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Mtro. Josué Figueroa González

Yo, Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. Alejandro Reyes Ortiz

Yo, Adriana Camal Mendoza, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Adriana Camal Mendoza

Resumen

En la actualidad los pequeños negocios muestran muchas desventajas frente a las cadenas establecidas o tiendas de conveniencia, principalmente por el uso limitado de tecnologías para su administración, es por esto que el principal objetivo de este proyecto es el diseño un sistema de información que permita a estos pequeños negocios superar algunas de estas deficiencias y así volverse más competitivos.

Para la creación de este sistema de información se hizo uso de tecnologías de desarrollo como Spring, Java Swing y el framework de Hibernate, con las que se logró un sistema informático que brinda la oportunidad de manejar la información del negocio de una forma amigable.

Las funcionalidades del sistema permiten la captura, almacenamiento y consulta de datos del área de almacén del negocio de manera sencilla, además de un sistema de ventas fácil y rápido.

Tabla de contenido

1.Introducción	1
2.Antecedentes	3
3.Justificación	6
4.Objetivos	8
5.Marco Teórico	10
5.1 Sistema de información.....	11
5.2 Bases de datos relacionales	12
5.3 Diseño de sistemas.....	12
5.4 Casos de uso	13
5.5 Spring.....	13
5.6 Java Swing	13
5.7 Hibernate.....	14
6.Desarrollo del Proyecto.....	15
6.1 Diagramas y documentación de casos de uso	18
6.2 Modelo entidad-relación de la base de datos.....	37
6.3 Diccionario de datos.....	38
6.4 Vistas	41
6.4.1 Vistas aplicación de Escritorio	41
6.4.2 Vistas aplicación Web	67
7.Resultados	77
8.Conclusiones	80
9.Referencias Bibliográficas	82
10.Anexo	85

ÍNDICE DE TABLAS Y FIGURAS

Figuras

Figura 1. Módulos que conforman el sistema.....	16
Figura 2. Casos de uso del módulo de Supervisión de Almacén y Ventas.....	18
Figura 3. Casos de uso del módulo de Registro y Consulta de Puntos de Fidelidad.....	21
Figura 4. Casos de uso del módulo Gestión de Almacén.....	23
Figura 5. Casos de uso del módulo de Ventas.....	32
Figura 6. Casos de uso del módulo Clientes.....	34
Figura 7. Entidad-Relación de la base de datos.....	37
Figura 8. Login del sistema.....	41
Figura 9. Página principal del sistema de información.....	42
Figura 10. Menú almacén.....	43
Figura 11. Menú productos.....	44
Figura12. Sección “Registrar nuevo producto”	45
Figura13. Sección “Modificar producto”	46
Figura 14. Sección “Consultar producto”	47
Figura 15. Menú proveedores.....	48
Figura 16. Sección “Nuevo proveedor”	49
Figura 17. Sección “Modificar proveedor”	50
Figura 18. Sección “Consultar proveedor”	51
Figura 19. Menú Devoluciones.....	52
Figura 20. Sección “Registrar devolución”	53
Figura 21. Sección “Modificar devolución”	54
Figura 22. Sección “Consultar devolución”	55
Figura 23. Menú Pedidos.....	56
Figura 24. Sección “Registrar pedido”	57
Figura 25. Sección “Modificar pedido”	58
Figura 26. Sección “Consultar pedido”	59
Figura 27. Menú cuentas por pagar.	60
Figura 28. Menú ventas.	61

Figura 29. Sección “Registrar ventas”	62
Figura 30. Sección “Consultar ventas”	63
Figura 31. Menú Clientes.	64
Figura 32. Sección “Registrar cliente”	65
Figura 33. Sección “Modificar cliente”	66
Figura 34. Página de inicio.	67
Figura 35. Menú cliente.	68
Figura 36. Sección “Nueva cuenta”	69
Figura 37. Sección “Login”	70
Figura 38. Sección “Puntos”	71
Figura 39. Login de administrador.....	72
Figura 40. Menú del administrador.....	73
Figura 41. Sección “Consultar ventas”	74
Figura 42. Sección “Consultar almacén”	75
Figura 43. Sección “Consultar proveedores”	76

Tablas

Tabla 1. Tablas que conforman la base de datos.....	38
Tabla 2. Proveedor.....	38
Tabla 3. Empleado.....	38
Tabla 4. Cliente.....	39
Tabla 5. Producto.....	39
Tabla 6. Venta.....	39
Tabla 7. Producto_Venta.....	40
Tabla 8. Pedido.....	40
Tabla 9. Devolución.....	40

ANEXO A.	
1.-Cliente.java	86
2.-Devolución.java.....	87
3.-Empleado.java	89
4.-Pedido.java	90
5.- Producto.java	92
6.- ProductoVenta.java	94
6.-Proveedor.java	95
8.- Venta.java	96
ANEXO B.	98
1.- Conexión.java.....	98
ANEXO C.	99
1.- Cliente.hbm.xml	99
2.- Devolución.hbm.xml	99
3.- Empleado.hbm.xml	99
4.- Pedido.hbm.xml	100
5.- Producto.hbm.xml	100
6.- ProductoVenta.hbm.xml.....	100
7.- Proveedor.hbm.xml	101
8.- Venta.hbm.xml.....	101
ANEXO D.	101
1.- DatosEstaticos.java	101
10.- OpCliente.java (Web).....	131
11.- OpVenta.java (Web).....	132
12.- OpEmpleado.java (Web)	136
13.- OpProducto.java (Web)	136
14.- OpProveedor.java (Web)	139
2.- OpCliente.java	102
3.- OpDevolución.java	104
4.- OpEmpleado.java	108
5.- OpPedido.java	112
6.- OpProducto.java	118
7.- OpProductoVenta.java.....	123
8.- OpProveedor.java	124
9.- OpVenta.java	128
ANEXO E.	140
1.- ConsultarCuentasPorPagar.java.....	140
10.- MenuDevoluciones.java.....	194
11.- MenuPedidos.java	198
12.- MenúPrincipal.java	202
13.- MenuProductos.java	205
14.- MenúProveedores.java	209
15.- MenuVentas.java	213
16.- ModificarCliente.java	216
17.- ModificarDevolución.java	224
18.- ModificarPedido.java	237
19.- ModificarProducto.java	249
2.- ConsultarDevoluciones.java.....	147
20.- ModificarProveedor.java	261

21.- NuevoProducto.java	271
22.- NuevoProveedor.java	281
23.- RegistrarCliente.java	288
24.- RegistrarDevolución.java	295
25.- RegistrarPedido.java	304
26.- RegistrarVentas.java	313
3.- ConsultarPedidos.java.....	154
4.- ConsultarProducto.java	162
5.- ConsultarProveedor.java	171
6.- ConsultarVentas.java	177
7.- Login.java	184
8.- MenuAlmacen.java	186
9.- MenuClientes.java	191
ANEXO F.....	325
1.- TextPrompt.java.....	325
ANEXO G.....	328
1.- Hibernate.cfg.xml.....	328
ANEXO H.....	328
1.- AlmacenController.java	328
2.- ClienteController.java	330
3.- ProveedorController.java.....	332
4.- VentaController.java	334
ANEXO I.....	336
1.-AlmacenFechasFrm.java	336
2.- ClienteVentaFrm.java	337
3.- VentasFechasFrm.java	338
ANEXO J.....	339
1.- controlador-servlet.xml	339
2.- web.xml.....	339
ANEXO K.....	340
1.- index.jsp	340
10.- consultarVentas.jsp.....	355
2.- menuCliente.jsp	341
3.- loginCliente.jsp	342
4.- NuevaCuenta.jsp	344
5.-consultarPuntos.jsp	346
6.- loginAdministrador.jsp.....	347
7.- menuEncargado.jsp	349
8.- consultarAlmacen.jsp.....	350
9.- consultarProveedores.jsp	353

1.Introducción

En la actualidad los sistemas de información son una pieza fundamental para el éxito de las empresas o negocios, dichos sistemas permiten optimizar y con ello agilizar el almacenamiento y consulta de la información, la cual también puede ser procesada de una manera mucho más sencilla con ayuda de tecnología computacional, que cabe mencionar, está creciendo a pasos agigantados.

De una manera más formal, un sistema de información es el conjunto de elementos o componentes que permiten capturar, almacenar, procesar y mostrar información de una manera sencilla y rápida con el fin de automatizar algunas funciones administrativas u operativas de las empresas que implementan dicho sistema.

Se considera como tienda de conveniencia a aquel establecimiento que da servicio las 24 horas del día los 365 días del año y en donde se puede encontrar desde comida rápida hasta productos de telefonía. Parte del éxito de este tipo de tiendas es la implementación de sistemas de información, porque ofrecen a los consumidores servicios adicionales que los pequeños negocios no brindan, tales como: depósitos en efectivo, recargas telefónicas, pago de teléfono, luz, gas, entre otros; pero no solo es por este tipo de servicios que los pequeños negocios siguen en desventaja, sino porque la mayoría aún realiza sus operaciones administrativas como: inventarios, ingresos y egresos de manera manual e información que es importante, se encuentra anotada en libretas o, en el mejor de los casos, en hojas de Excel.

2. Antecedentes

Sistema de Administración para Pequeñas y Medianas Empresas en México. [1]

Este sistema permite capturar información de gran utilidad para las empresas como datos de clientes, proveedores, artículos e inventarios, lo que se asemeja al proyecto que se propone. En lo que difieren es que en la presente propuesta no se contará con la opción de generar facturas, ya que está enfocado a pequeños negocios que en su mayoría no necesitan generarlas.

Sistema de Inventario y Ventas para Tlapalería con Pronóstico de Mercado. [2]

El sistema ofrece un pronóstico de mercado tomando en cuenta los reportes que son generados gracias a la captura de información de cada una de las ventas realizadas, lo cual permite saber la demanda y tendencia que tiene un producto para incrementar las ganancias; aunque el sistema que se está proponiendo no cuenta con esta funcionalidad sí ofrece una opción de lealtad para sus clientes por medio de puntos que serán acumulables en cada compra, esto ayudará a mejorar los ingresos del negocio. Ambos sistemas permitirán registrar a los clientes, tener un control de inventario, registro de ventas y de gastos, y acceso controlado al sistema con login.

Diseño de un Sistema de Información para una Pequeña Empresa de Computo. [3]

Este sistema permite capturar información específica de sus clientes para dar seguimiento a pedidos, entregas, devoluciones, garantías, cuentas por cobrar, etc., lo que permitirá mejorar el servicio al cliente y ayudará a que se incrementen las ventas; a diferencia el proyecto que se propone, está enfocado a los costos, ganancias, reportes, etc. Ambos sistemas manejarán información, que si bien no se trata de los mismos artículos sí son datos de gran importancia y esencial para el pequeño negocio.

SmartNet Admin. [4]

Es un software estándar para ser utilizado en la administración de cualquier tipo de empresa o negocio, una de las similitudes de este proyecto con dicho software es que se tienen interfaces amigables e intuitivas para facilitar su uso, también comparten un control de acceso y de permisos para saber quién ha entrado a consultar o manipular la información; sin embargo existen diferencias como el uso de funciones para control fiscal, de personal, distintas formas de pago y exportación de datos a Excel que esta propuesta no tendrá por el momento.

LemonPOS. [5]

Este es un software de código abierto enfocado a las ventas y a la gestión de productos, actividades que este proyecto también manejará, además de que ambos software podrán consultar información de otras sucursales en caso de existir, a través de una base de datos. Sin embargo, el proyecto propuesto no usará imágenes para mostrar que artículo se está vendiendo o inventariando, implementación con la que sí cuenta LemonPOS.

Care2x. [6]

Software de código abierto enfocado a la administración de hospitales, aunque el tipo de información que se captura no es la misma a la de la propuesta, sí existe una gran similitud con la forma de gestionar los datos, por ejemplo crear, editar, eliminar los datos de enfermeras, inventarios de medicamentos, consultar información de pacientes, etc. actividades que en nuestro sistema también se realizarán solo que enfocado a productos, se cuenta también con una interfaz amigable para un uso sencillo como el que se propone en el proyecto.

3. Justificación

Los pequeños negocios se encuentran en desventaja ante las tiendas de conveniencia porque no ofrecen ofertas, ni servicios adicionales y sobre todo no cuentan con el apoyo tecnológico que permite tener un mayor control de su información de manera sencilla.

La aplicación de sistemas de información en pequeños negocios es un tema de importancia porque permite a los negocios ser más competitivos.

El sistema que se propone en este proyecto brinda las herramientas necesarias para lograr una mejor organización del establecimiento y ofrecer algún beneficio de fidelización a clientes al momento del cobro, ya que, optimizando la información de inventarios, ganancias y costos se puede lograr un mejor análisis del desempeño del comercio y así buscar estrategias de venta que ayuden a mejorar las ganancias, por ejemplo, ofrecer alguna promoción, además, el sistema de fidelización ayudará a retener a los clientes de una manera innovadora para los pequeños negocios.

4. Objetivos

Objetivo General:

Implementar un sistema para automatizar las actividades de control de información en los pequeños negocios.

Objetivos Específicos:

- Diseñar e implementar un módulo para supervisar los movimientos efectuados en el almacén y ventas.
- Diseñar e implementar un módulo para el registro y consulta de puntos de fidelidad.
- Diseñar e implementar un módulo para gestionar las actividades del almacén.
- Diseñar e implementar un módulo para el registro de ventas y promociones con puntos de fidelidad.
- Diseñar e implementar un módulo para gestionar la información de los clientes.

5. Marco Teórico

5.1 Sistema de información

Los sistemas de información tienen como objetivo ayudar a la mejora del desempeño de las actividades que lleva a cabo una empresa o negocio, se han vuelto en la actualidad una herramienta que facilita la buena administración y la mejora en el suministro de información.

De manera más específica, un sistema de información es:

“Conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes para desempeñar su actividad de acuerdo a su estrategia de negocio”¹

El uso de estos sistemas conlleva ventajas y desventajas entre las cuales encontramos:

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">• Intervención y mejora en el manejo de las actividades del negocio.• Incrementa la efectividad de la operación de los negocios.• Mejora para los usuarios la calidad de la información que se maneja en el negocio• Permite la asistencia remota• Reduce errores, tiempo y recursos.• Facilita el comparar los resultados obtenidos con los objetivos que se plantee el negocio	<ul style="list-style-type: none">• El tiempo para la implementación de este sistema puede ser largo.• Pueden presentarse problemas técnicos con respecto al software y hardware utilizado cuando no se realiza un estudio adecuado.• Fallas en las funciones de implementación que no apoyen ciertas actividades del negocio• La resistencia al cambio por parte de los usuarios.

¹ Keneth C. Laudon y Jane P. Laudon. Sistema de Información Gerencial. Octava Edición. Pearson Educación. México, 2004.

5.2 Bases de datos relacionales

Con el paso del tiempo los sistemas de administración de Bases de Datos han ido cambiando, siempre en la búsqueda de la mejora, es así como se ha evolucionado a los Sistemas de Administración de Base de Datos Relacionales.

“Una base de datos relacional es un modelo organizado de entidades que posee características que tienen relaciones entre ellas.”²

Cuando una base de datos relacional está bien diseñada permite:

- Proveer información de un negocio o un proceso.
- Almacenar y recuperar información manteniendo la calidad de la misma.
- Permite la administración de datos de forma dinámica.
- El uso de diagramas Entidad-Relación facilita su construcción y comprensión.
- Proporciona una forma más sencilla de presentar los datos que es a través de tablas (una para cada relación)

5.3 Diseño de sistemas

El Diseño del Sistema de Información es una de las etapas de desarrollo de los Sistemas, con ella se busca establecer la forma en la que el sistema cumplirá los requerimientos para los cuales es creado.

“El proceso de Diseño del sistema de Información tiene como objetivo la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.”³

Esta etapa permite generar las especificaciones de construcción del sistema y la descripción técnica del mismo.

² <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/907/A5.pdf?sequence=5>

³ <http://cc.etsii.ull.es/ftp/antiguo/INGSOFG2/dsiproc.pdf>

5.4 Casos de uso

Un caso de uso es una descripción de las acciones de un sistema de manera detallada, esta herramienta permite modelar las funciones que debe realizar el sistema desde el punto de vista del usuario.

Los diagramas de casos de uso muestran de forma gráfica la interacción que existe entre los actores y el sistema. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

5.5 Spring

Spring es un framework de código abierto que permite el desarrollo de aplicaciones en una plataforma Java.

Entre sus principales características están:

- Es liviano, no requiere muchos recursos para su ejecución
- Es una aplicación open source (código fuente abierto), además no tiene costo ni requiere una licencia para su funcionamiento, esto le ha dado mucho auge y popularidad.
- Cuenta con una arquitectura basada en módulos, cada uno de los cuales se encarga de tareas independientes de la aplicación y se pueden utilizar de forma independiente.
- Si se usa Spring de forma correcta la aplicación quedará dividida en capas bien delimitadas, y con buenas prácticas de programación.

5.6 Java Swing

Java Swing es una biblioteca gráfica para Java, contiene cajas de texto, listas desplegables y tablas, permite el desarrollo de interfaces gráficas con independencia de la plataforma. Entre sus principales características están:

- Contiene una amplia variedad de componentes.
- Permite personalizar el aspecto de las interfaces.

- Arquitectura Modelo-Vista-Controlador.
- Se pueden crear escritorios virtuales
- Permite deshacer las modificaciones que se hayan realizado
- Posee menos limitaciones de plataforma

5.7 Hibernate

Hibernate es la herramienta que agiliza el mapeo de atributos entre la aplicación y la base de datos relacional mediante archivos XML que permiten establecer estas relaciones.

Entre sus principales características están:

- Productividad: elimina la mayor parte del código de persistencia
- Capacidad de mantenimiento: el énfasis se hace en la lógica del programa no en la persistencia facilitando su mantenimiento.
- Rendimiento.
- Independencia de proveedor.

6.Desarrollo del Proyecto

Se realizaron dos aplicaciones, una web y otra de escritorio, en la Figura 1 se muestran los módulos de cada una de ellas.

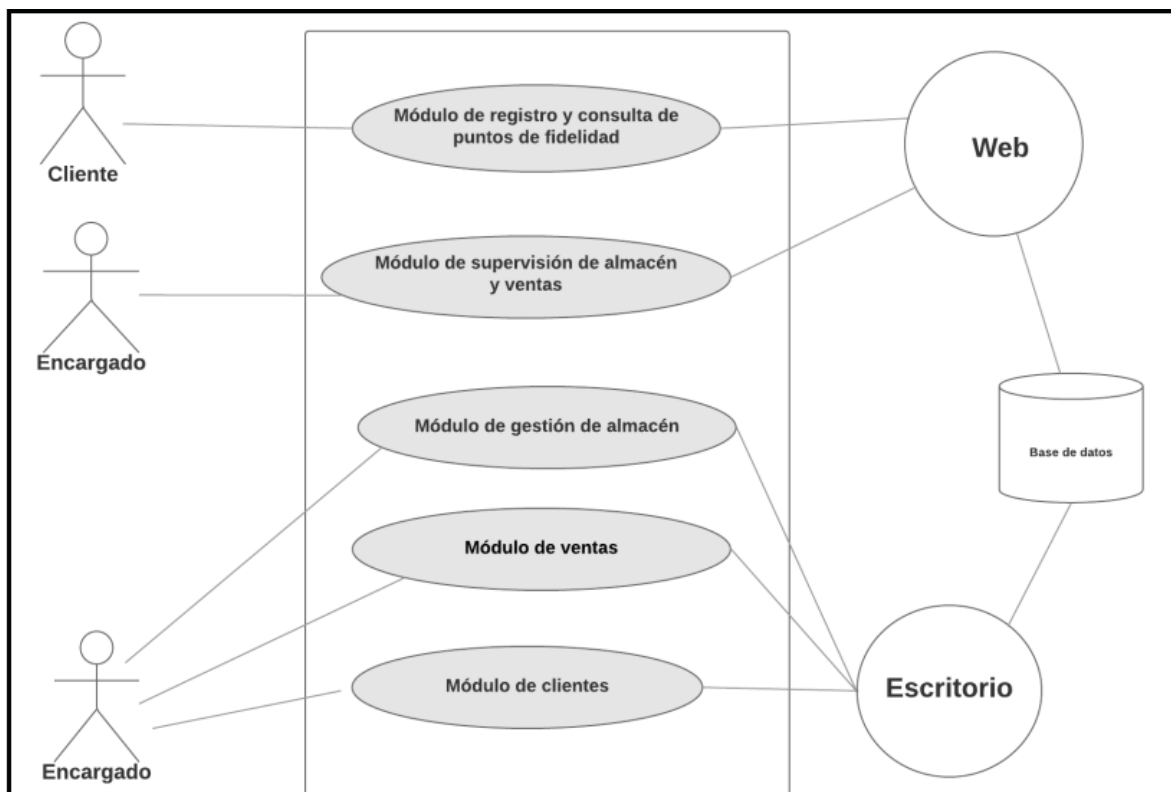


Figura 1. Módulos que conforman el sistema.

Módulo de supervisión de almacén y ventas

Permite supervisar la información de ventas y de los movimientos en el almacén como entradas y salidas de mercancía, costos, precios al público, proveedores, productos en existencia, etc., todo esto desde la aplicación web.

Módulo de registro y consulta de puntos de fidelidad

Esta sección del sistema registra los puntos de fidelidad que correspondan a un ticket de compra, de igual manera permite la consulta de la cantidad de dichos puntos, todo esto desde la aplicación web.

Módulo de gestión del almacén

Esta parte del sistema permite capturar un nuevo producto, de igual forma edita los existentes o los elimina si es necesario, también permite la gestión de información de los proveedores, devoluciones, pedidos, etc. Esta funcionalidad es parte de la aplicación de escritorio.

Módulo de ventas

Registra todos los movimientos referentes a caja como ventas, registro de clientes y realiza la aplicación de promociones con puntos de fidelidad en caso de ser necesario. Esta funcionalidad es parte de la aplicación de escritorio.

Módulo de clientes

Se encarga de registrar un nuevo cliente y asignar un número único de identificación, también permite editar a los existentes o eliminarlos de ser necesario. Esta funcionalidad es parte de la aplicación de escritorio.

6.1 Diagramas y documentación de casos de uso

La Figura 2 muestra los tres casos de uso que conforman el módulo de Supervisión de Almacén y Ventas y la interacción que hay entre el actor y el sistema.

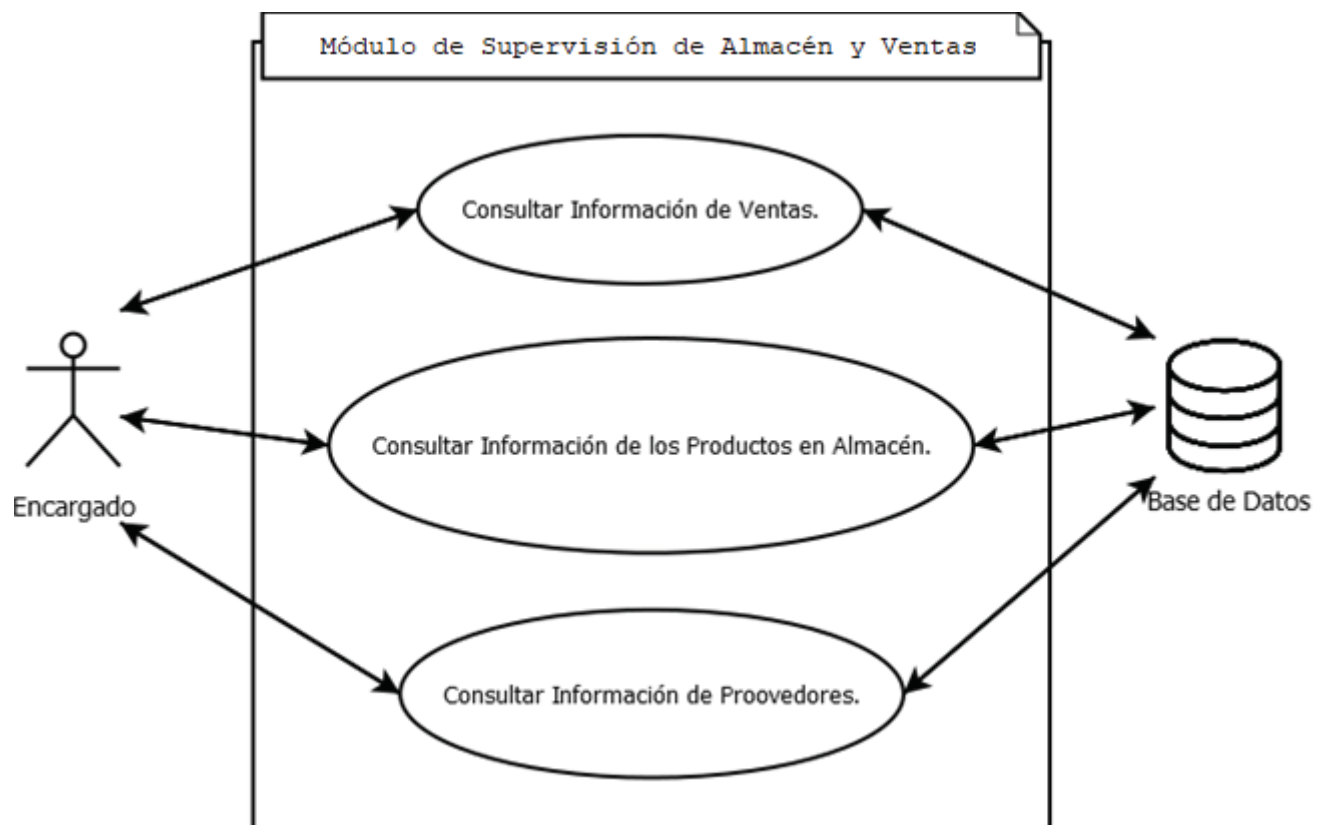


Figura 2. Casos de uso del módulo de Supervisión de Almacén y Ventas

A continuación se describe la documentación del caso de uso Consultar Información de Venta:

Caso de Uso: Consultar Información de Ventas.	
Actor: Encargado.	
Pre-Condiciones: Se debe ingresar a “Encargado”. El Encargado debe estar autenticado en el Sistema. Se debe seleccionar la sección “Ventas”.	
Flujo Normal	Alternativas
1: El Sistema muestra dos campos etiquetados como “Desde” y “Hasta” que despliegan calendarios para realizar la consulta por día, semana, mes o año.	1.1: Si se desea una consulta más específica, el Sistema muestra también la sección opcional de “Consulta Avanzada” en donde se realiza un filtrado por “Forma de Pago” y/o “Vendedor”.
2: El Encargado selecciona los criterios de consulta y oprime el botón “Consultar”.	
3: El Sistema muestra un listado con información detallada de las ventas.	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección “Ventas” oprimiendo el botón “Terminar”.	

La siguiente documentación describe el caso de uso Consultar Información de los Productos en Almacén:

Caso de Uso: Consultar Información de los Productos en Almacén.	
Actor: Encargado.	
Pre-Condiciones: Se debe ingresar a “Encargado”. El Encargado debe estar autenticado en el Sistema. Se debe seleccionar la sección “Almacén”.	
Flujo Normal	Alternativas
1: El Sistema muestra dos campos etiquetados como “Desde” y “Hasta” que despliegan calendarios para realizar la consulta de los productos por día, semana, mes o año en orden alfabético.	1.1: Si se desea una consulta más específica, el Sistema muestra también la sección opcional de “Consulta Avanzada” en donde se realiza un filtrado por “Clave”, “Código de Barras”, “Nombre de Producto” y/o “Proveedor”.
2: El Encargado selecciona los criterios de consulta y oprime el botón “Consultar”.	

<p>3: El Sistema muestra un listado con información relevante de los productos como entradas y salidas, costos de compra, costo al público, existencias, puntos asignados, etc.</p>	
<p>4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección “Almacén” oprimiendo el botón “Terminar”.</p>	

El último caso de uso de este módulo es Consultar Información de Proveedores, documentación que se describe a continuación:

<p>Caso de Uso: Consultar Información de Proveedores.</p>	
<p>Actor: Encargado.</p>	
<p>Pre-Condiciones: Se debe ingresar a “Encargado”. El Encargado debe estar autenticado en el Sistema. Se debe seleccionar la sección “Proveedores”.</p>	
<p>Flujo Normal</p>	<p>Alternativas</p>
<p>1: El Sistema muestra una lista desplegable etiquetada como “Nombre Proveedor”.</p>	
<p>2: El Encargado selecciona el nombre del proveedor del cual desea ver la información y oprime el botón “Ver”.</p>	<p>2.1: El Encargado puede seleccionar la opción “Todos los Proveedores” para visualizar el listado en orden alfabético de todos los proveedores.</p>
<p>3: El Sistema muestra información del proveedor/es como clave, nombre, dirección, teléfono, correo y contacto.</p>	
<p>4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección “Proveedores” oprimiendo el botón “Terminar”.</p>	

En la Figura 3 se pueden observar los dos casos de uso del módulo de Registro y Consulta de Puntos de Fidelidad, así como el actor y su interacción con el sistema.

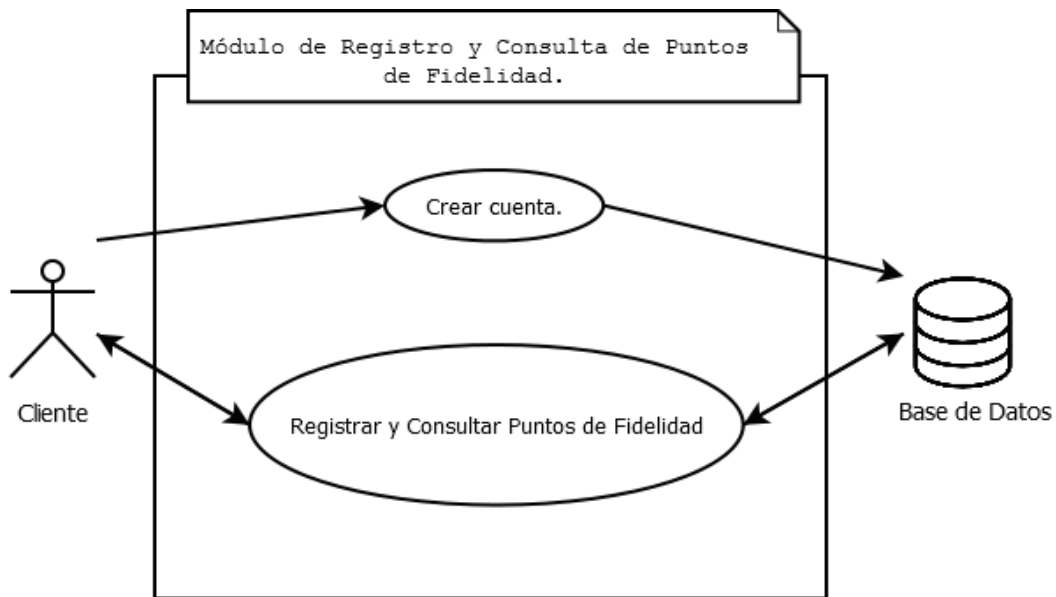


Figura 3. Casos de uso del módulo de Registro y Consulta de Puntos de Fidelidad

La documentación del caso de uso Crear Cuenta se describe detalladamente a continuación:

Caso de Uso: Crear Cuenta.	
Actor: Cliente.	
Pre-Condiciones: Se debe ingresar a "Cliente". El Cliente debe seleccionar la sección "Cuenta Nueva".	
Flujo Normal	Alternativas
1: El Sistema muestra tres cajas de texto etiquetadas como "Usuario" y dos más etiquetadas como "Clave" y "Confirmar Clave".	
2: El Cliente ingresa el nombre de usuario, clave y confirmación de clave, estos datos son los que este desee y al concluir oprime el botón "Continuar".	
3: El Sistema muestra un formulario para ingresar los datos personales del Cliente	

etiquetados como "Nombre", "Calle y Número", "Colonia", "Municipio", "Estado", "Código Postal", "Correo" y "Teléfono".	3.1: El Cliente fue registrado en el Negocio, el Sistema muestra una caja para ingresar el número de cliente asignado. 3.1.1: El Cliente captura el número asignado, el Sistema muestra el botón "Buscar" para ver su información de manera automática.
4: El Cliente ingresa los datos solicitados y al concluir oprime el botón "Crear Cuenta".	
5: El Sistema asigna un número de cliente y muestra la información que fue registrada.	5.1: Si no se asigna un número de cliente, el Sistema muestra un aviso indicando que faltan campos por llenar.
6: El Cliente oprime el botón "Terminar" para salir de la creación de la cuenta.	

La siguiente información describe la documentación del caso de uso Registrar y Consultar Puntos de Fidelidad:

Caso de Uso: Registrar y Consultar Puntos de Fidelidad.	
Actor: Cliente.	
Pre-Condiciones: Se debe ingresar a "Cliente". El Cliente debe ingresar a su cuenta. Se debe seleccionar la sección "Puntos".	
Flujo Normal	Alternativas
1: El Sistema muestra el Número y Nombre del Cliente al igual que una caja de texto etiquetada como "Número de Ticket".	1.1: Si se desea consultar información detallada de los puntos acumulados, el Sistema muestra de manera adicional el botón "Consulta de Puntos" con el cual dicho Sistema muestra fecha, número de ticket, importe y puntos acumulados.
2: El Cliente ingresa el número de ticket de la compra realizada en el Negocio y al concluir oprime el botón "Acumular Puntos".	
3: El Sistema muestra la cantidad de puntos asignados al número de ticket que fueron registrados y la cantidad de puntos acumulados en total.	
4: Se repiten los pasos 2 y 3 hasta que el Cliente oprima el botón "Terminar".	

La Figura 4 muestra los trece casos de uso que forman parte del módulo Gestión de Almacén y la interacción que hay entre el actor y el sistema.

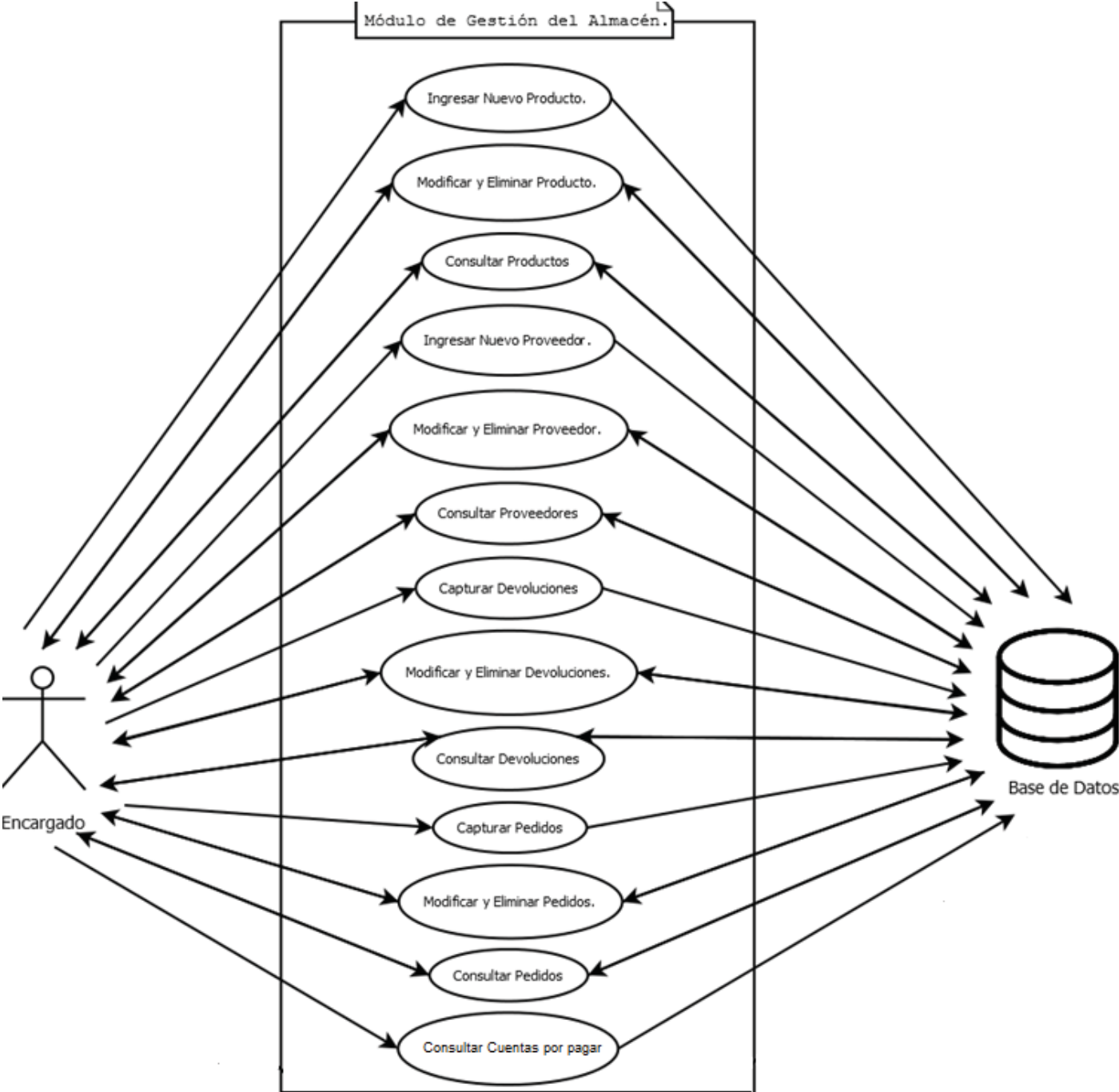


Figura 4. Casos de uso del módulo Gestión de Almacén

El caso de uso Ingresar Nuevo Producto se detalla en la siguiente documentación:

Caso de Uso: Ingresar Nuevo Producto.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Productos" y "Nuevo".	
Flujo Normal	Alternativas
1: El Sistema muestra un formulario para registrar información como "Fecha", "Código de Barras", "Nombre de Producto", "Costo de Compra", "IVA.", "Costo al Público", "Puntos Asignados", "Proveedor", "Cantidad de Ingreso" y "Clasificación".	
2: El Encargado captura la información solicitada y al terminar oprime el botón "Registrar".	
3: El Sistema muestra un aviso indicando que el registro de información ha sido satisfactoria.	3.1: Si no se captura toda la información, el Sistema muestra un aviso a un lado de la caja de texto indicando que faltan campos por llenar.
4: Se repiten los pasos 2 y 3 hasta que el Encargado sale de "Nuevo" oprimiendo el botón "Terminar".	

A continuación se encuentra la información detallada de la documentación del caso de uso Ingresar Modificar y Eliminar Producto:

Caso de Uso: Modificar y Eliminar Producto.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Productos" y "Modificar".	
Flujo Normal	Alternativas
1: El Sistema muestra el listado en orden alfabético de todos los productos registrados en el Almacén.	
2: El Encargado selecciona el producto de su interés y oprime el botón "Modificar".	2.1: Si el Encargado desea eliminar un producto de la lista, oprime el botón "Eliminar".
3: El Sistema muestra el mismo formulario que en "Nuevo Producto" pero con los campos activos para modificar la información	

que fue registrada anteriormente y que aparece en dichos campos.	
4: El Encargado captura los cambios deseados y oprime el botón "Guardar".	
5: Se repiten los pasos a partir del 2 hasta que el Encargado sale de "Modificar" oprimiendo el botón "Terminar".	

La siguiente documentación describe el caso de uso Consultar Productos:

Caso de Uso: Consultar Productos.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Productos" y "Consultar".	
Flujo Normal	Alternativas
1: El Sistema muestra dos campos etiquetados como "Desde" y "Hasta" que despliegan calendarios para realizar la consulta de los productos por día, semana, mes o año en orden alfabético.	1.1: Si se desea una consulta más específica, el Sistema muestra también la sección opcional de "Consulta Avanzada" en donde se realiza un filtrado por "Clave", "Código de Barras", "Nombre de Producto" y/o "Proveedor".
2: El Encargado selecciona los criterios de consulta y oprime el botón "Consultar".	
3: El Sistema muestra un listado con información relevante de los productos como código de barras, nombre del producto, proveedor, entrada, salida, existencia, costo de compra, costo al público, puntos asignados.	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de "Consultar" oprimiendo el botón "Terminar".	

La documentación del caso de uso Ingresar Nuevo Proveedor se describe detalladamente a continuación:

Caso de Uso: Ingresar Nuevo Proveedor.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Proveedores" y "Nuevo".	
Flujo Normal	Alternativas
1: El Sistema muestra un formulario para registrar información como "Fecha", "Nombre del Proveedor", "Dirección", "Teléfono", "Correo" y "Contacto"	
2: El Encargado captura la información solicitada y al terminar oprime el botón "Registrar".	
3: El Sistema muestra un aviso indicando que el registro de información ha sido satisfactoria.	3.1: Si no se captura toda la información, el Sistema muestra un aviso indicando que faltan campos por llenar.
4: Se repiten los pasos 2 y 3 hasta que el Encargado sale de "Nuevo" oprimiendo el botón "Terminar".	

A continuación se encuentra la información detallada de la documentación del caso de uso Ingresar Modificar y Eliminar Proveedor:

Caso de Uso: Modificar y Eliminar Proveedor.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Proveedores" y "Modificar".	
Flujo Normal	Alternativas
1: El Sistema muestra el listado en orden alfabético de todos los proveedores registrados.	
2: El Encargado selecciona el proveedor que desea editar y oprime el botón "Modificar".	2.1: Si el Encargado desea eliminar a un proveedor de la lista, oprime el botón "Eliminar".
3: El Sistema muestra el mismo formulario que en "Nuevo Proveedor" pero con los	

campos activos para modificar la información que fue registrada anteriormente y que aparece en dichos campos.	
4: El Encargado captura los cambios deseados y oprime el botón “Guardar”.	
5: Se repiten los pasos a partir del 2 hasta que el Encargado sale de “Modificar” oprimiendo el botón “Terminar”.	

La siguiente documentación describe el caso de uso Consultar Proveedores:

Caso de Uso: Consultar Proveedores.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección “Almacén”, opción “Proveedores” y “Consultar”.	
Flujo Normal	Alternativas
1: El Sistema muestra una lista desplegable etiquetada como “Nombre Proveedor”.	
2: El Encargado selecciona el nombre del proveedor del cual desea ver la información y oprime el botón “Ver”.	2.1: El Encargado selecciona la opción “Todos los Proveedores” para visualizar el listado en orden alfabético de todos los proveedores registrados.
3: El Sistema muestra información del proveedor/es como clave, fecha, nombre, dirección, teléfono, correo y contacto.	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección “Consultar” oprimiendo el botón “Terminar”.	

El caso de uso Capturar Devoluciones se detalla en la siguiente documentación:

Caso de Uso: Capturar Devoluciones.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección “Almacén”, opción “Devoluciones” y “Registrar”.	
Flujo Normal	Alternativas
1: El Sistema muestra un formulario para registrar información como “Fecha”, “Código de Barras”, “Producto”, “Proveedor”,	

“Cantidad de Productos”, “Costo por Unidad”, “Costo Devolución”, “Motivo”, “Tipo de Devolución” y “Status”	
2: El Encargado captura la información solicitada y al terminar oprime el botón “Registrar”.	
3: El Sistema muestra un aviso indicando que el registro de información ha sido satisfactoria.	3.1: Si no se captura toda la información, el Sistema muestra un aviso indicando que faltan campos por llenar.
4: Se repiten los pasos 2 y 3 hasta que el Encargado sale de “Registrar” oprimiendo el botón “Terminar”.	

La documentación del caso de uso Modificar y Eliminar Devoluciones se describe detalladamente a continuación:

Caso de Uso: Modificar y Eliminar Devoluciones.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección “Almacén”, opción “Devoluciones” y “Modificar”.	
Flujo Normal	Alternativas
1: El Sistema muestra una lista desplegable etiquetada como “Proveedor” y una tabla de información, inicialmente vacía.	
2: El Encargado selecciona el nombre del proveedor del que desea modificar la información	
3: El Sistema muestra el listado ordenado por fecha de todas las devoluciones registradas, mostrando también la clave de la devolución.	
4: El Encargado selecciona la devolución que desea editar y oprime el botón “Modificar”.	4.1: Si el Encargado desea eliminar una devolución de la lista, oprime el botón “Eliminar”.
5: El Sistema muestra el mismo formulario que en “Registrar Devoluciones” pero con los campos activos para modificar la información que fue registrada anteriormente y que aparece en dichos campos.	
6: El Encargado captura los cambios deseados y oprime el botón “Guardar”.	

7: Se repiten los pasos a partir del 2 hasta que el Encargado sale de “Modificar” oprimiendo el botón “Terminar”.	
---	--

A continuación se encuentra la información detallada de la documentación del caso de uso Consultar Devoluciones:

Caso de Uso: Consultar Devoluciones.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección “Almacén”, opción “Devoluciones” y “Consultar”.	
Flujo Normal	Alternativas
1: El Sistema muestra una lista desplegable etiquetada como “Proveedor”, la opción “Devolución” como selección por defecto y una caja de texto etiquetada como “Clave de Devolución”.	
2: El Encargado ingresa la clave de devolución de la cual desea ver la información y oprime el botón “Ver”.	2.1: El Encargado puede seleccionar la opción “Todas las Devoluciones” para visualizar el listado completo de todas las devoluciones.
3: El Sistema muestra información completa de la(s) devolución(es).	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección “Consultar” oprimiendo el botón “Terminar”.	

La siguiente información describe la documentación del caso de uso Capturar Pedidos:

Caso de Uso: Capturar Pedidos.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección “Almacén”, opción “Pedidos” y “Registrar”.	
Flujo Normal	Alternativas
1: El Sistema muestra un formulario para registrar información como “Fecha” “Código de Barras”, “Producto”, “Proveedor”,	

“Cantidad a Solicitar”, “Costo por Unidad”, “Costo Total”, y “Status”	
2: El Encargado captura la información solicitada y al terminar oprime el botón “Registrar”.	
3: El Sistema muestra un aviso indicando que el registro de información ha sido satisfactoria.	3.1: Si no se captura toda la información, el Sistema muestra un aviso indicando que faltan campos por llenar.
4: Se repiten los pasos 2 y 3 hasta que el Encargado sale de “Registrar” oprimiendo el botón “Terminar”.	

El caso de uso Modificar y Eliminar Pedidos se detalla en la siguiente documentación:

Caso de Uso: Modificar y Eliminar Pedidos.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección “Almacén”, opción “Pedidos” y “Modificar”.	
Flujo Normal	Alternativas
1: El Sistema muestra una lista desplegable etiquetada como “Proveedor” y una tabla de información, inicialmente vacía.	
2: El Encargado selecciona el nombre del proveedor del que desea modificar la información	
3: El Sistema muestra el listado ordenado por fecha de todos los pedidos que fueron registrados, mostrando también el número de pedido.	
4: El Encargado selecciona el pedido que desea editar y oprime el botón “Modificar”.	4.1: Si el Encargado desea eliminar un pedido de la lista, oprime el botón “Eliminar”.
5: El Sistema muestra el mismo formulario que en “Registrar Pedido” pero con los campos activos para modificar la información que fue registrada anteriormente y que aparece en dichos campos.	
6: El Encargado captura los cambios deseados y oprime el botón “Guardar”.	
7: Se repiten los pasos a partir del 2 hasta que el Encargado sale de “Modificar” oprimiendo el botón “Terminar”.	

La documentación del caso de uso Consultar Pedidos se describe detalladamente a continuación:

Caso de Uso: Consultar Pedidos.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Pedidos" y "Consultar".	
Flujo Normal	Alternativas
1: El Sistema muestra una lista desplegable etiquetada como "Proveedor", la opción "Pedido" como selección por defecto y una caja de texto etiquetada como "Número de Pedido".	
2: El Encargado selecciona el proveedor e ingresa el número de pedido del cual desea ver la información y oprime el botón "Ver".	2.1: El Encargado puede seleccionar la opción "Todos los Pedidos" para visualizar el listado completo de los pedidos.
3: El Sistema muestra información completa del pedido(s).	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección "Consultar" oprimiendo el botón "Terminar".	

El último caso de uso de este módulo es Consultar Cuentas por Pagar, documentación que se describe a continuación:

Caso de Uso: Consultar Cuentas por Pagar.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Almacén", opción "Cuentas por Pagar" y "Consultar".	
Flujo Normal	Alternativas
1: El Sistema muestra una lista desplegable etiquetada como "Proveedor", la opción "Cuenta" como selección por defecto y una caja de texto etiquetada como "Número de Cuenta".	
2: El Encargado selecciona el nombre del proveedor e ingresa el número de cuenta de	2.1: El Encargado puede seleccionar la opción "Todas las Cuentas" para visualizar el listado completo de todos los pedidos.

la cual desea ver la información y oprime el botón "Ver".	
3: El Sistema muestra información completa del pedido(s).	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de la sección "Consultar" oprimiendo el botón "Terminar".	

El módulo de Ventas consta de dos casos de uso que se muestran en la Figura 5, es posible observar en la misma figura la interacción del actor y el sistema en este módulo.

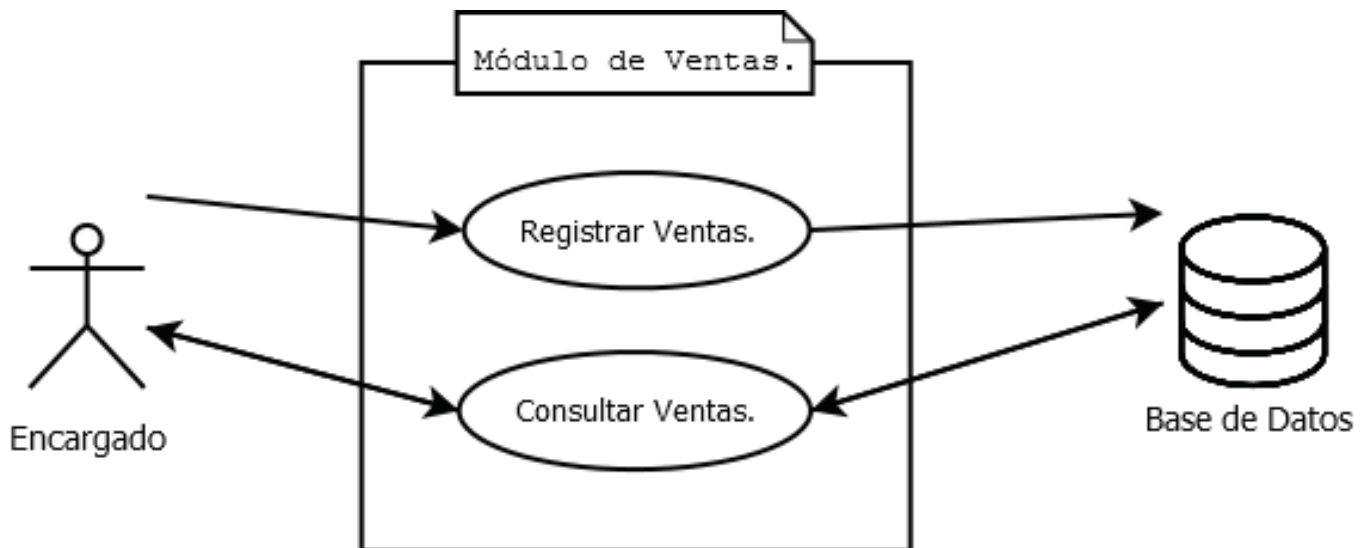


Figura 5. Casos de uso del módulo de Ventas

La siguiente información describe la documentación del caso de uso Consultar Ventas:

Caso de Uso: Consultar Ventas.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Ventas", opción "Consultar".	
Flujo Normal	Alternativas
1: El Sistema muestra dos campos etiquetados como "Desde" y "Hasta" que despliegan calendarios para realizar la consulta por día, semana, mes o año.	1.1: Si se desea una consulta más específica, el Sistema muestra también la sección opcional de "Consulta Avanzada" en donde se realiza un filtrado por "Forma de Pago" y/o "Vendedor".
2: El Encargado selecciona los criterios de consulta y oprime el botón "Consultar".	
3: El Sistema muestra un listado con información detallada de las ventas.	
4: Se repiten los pasos a partir del 2 para continuar la consulta hasta que el Encargado sale de "Consultar" oprimiendo el botón "Terminar".	

La documentación del caso de uso Registrar Ventas se describe detalladamente a continuación:

Caso de Uso: Registrar Ventas.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Ventas", opción "Caja".	
Flujo Normal	Alternativas
1: El Sistema muestra un formulario en donde "Fecha", "Clave Vendedor", "Nombre Vendedor" y "Número de Transacción" se llenan de manera automática, las secciones "Tipo de Pago" y "Cliente" están desactivadas y solo se captura "Código de Barras".	
2: El Encargado ingresa el código de barras.	
3: El Sistema muestra de forma automática, al finalizar la captura del código, "Cantidad",	

<p>“Producto”, “Costo por Unidad” “Puntos”, “Importe” y se muestra un desglose de la cantidad a pagar etiquetada de la siguiente forma: “Sub-Total”, “IVA”, “Pago con Puntos” y “Total”.</p>	
<p>4: Se repiten los pasos 2 y 3 hasta que el Encargado termina de capturar todos los productos de la compra y oprime el botón “Cobrar”.</p>	
<p>5: El Sistema activa la sección “Tipo de Pago” en donde se muestran las opciones de pago “Efectivo”, “Puntos” o “Ambos” y la sección “Cliente” en donde se muestran las opciones “Buscar Cliente” (<i>por número o nombre</i>) y “Puntos Acumulados”.</p>	
<p>6: El Encargado ingresa la información requerida para visualizar los puntos acumulados, solicitando al Cliente su número de o nombre completo, y finaliza la compra oprimiendo el botón “Pagar”.</p>	<p>6.1: Si el Cliente no tiene una cuenta asignada, el Encargado le informa que vía internet puede crear dicha cuenta para obtener un número de cliente único.</p>

La Figura 6 muestra los dos casos de uso que conforman el módulo de Clientes y la interacción que existe entre el actor y el sistema.

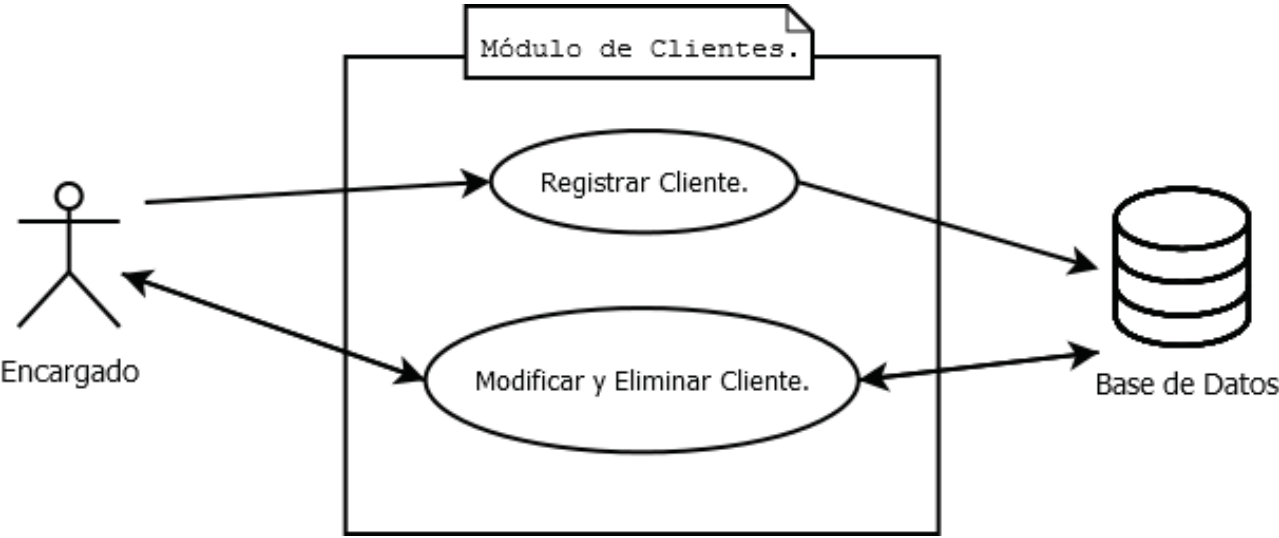


Figura 6. Casos de uso del módulo Clientes

El caso de uso Registrar Cliente se detalla en la siguiente documentación:

Caso de Uso: Registrar Cliente.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Clientes", opción "Registrar".	
Flujo Normal	Alternativas
1: El Sistema muestra un formulario para ingresar los datos personales del Cliente etiquetados como "Fecha", "Nombre", "Dirección", "Correo" y "Teléfono".	
2: El Encargado ingresa los datos solicitados y al concluir oprime el botón "Registrar".	
3: El Sistema asigna un número de cliente y muestra la información que fue registrada dando la opción para imprimir.	3.1: Si no se asigna el número de cliente, el Sistema muestra un aviso indicando que campos faltan por llenar.
4: El Encargado imprime la información oprimiendo el botón "Imprimir Comprobante" y finaliza oprimiendo el botón "Terminar" para salir de la sección "Registrar".	4.1: Se le informa al Cliente que tiene la opción de crear una cuenta, solo con el número de cliente asignado, en la página de internet para poder consultar la cantidad de puntos.

El segundo caso de uso de este módulo es Modificar y Eliminar Cliente, documentación que se describe a continuación:

Caso de Uso: Modificar y Eliminar Cliente.	
Actor: Encargado.	
Pre-Condiciones: El Encargado debe estar autenticado en el Sistema. Se debe estar en la sección "Clientes", opción "Modificar".	
Flujo Normal	Alternativas
1: El Sistema muestra dos cajas de texto etiquetadas como "Nombre del Cliente" y "Numero de Cliente".	
2: El Encargado ingresa el nombre o número del cliente que desea editar su información y oprime el botón "Modificar".	2.1: Si el Encargado desea eliminar a un cliente del sistema, oprime el botón "Eliminar".
3: El Sistema muestra el mismo formulario que en "Registrar Cliente" pero con los campos activos para modificar la información	

que fue registrada anteriormente y que aparece en dichos campos.	
4: El Encargado captura los cambios deseados y oprime el botón "Guardar".	
5: Se repiten los pasos a partir del 2 hasta que el Encargado sale de "Modificar Cliente" oprimiendo el botón "Terminar".	

6.2 Modelo entidad-relación de la base de datos

En la Figura 7 se muestra el diagrama de la entidad-relación de la base de datos, se puede observar en ella las tablas que conforman el sistema y como se relacionan:

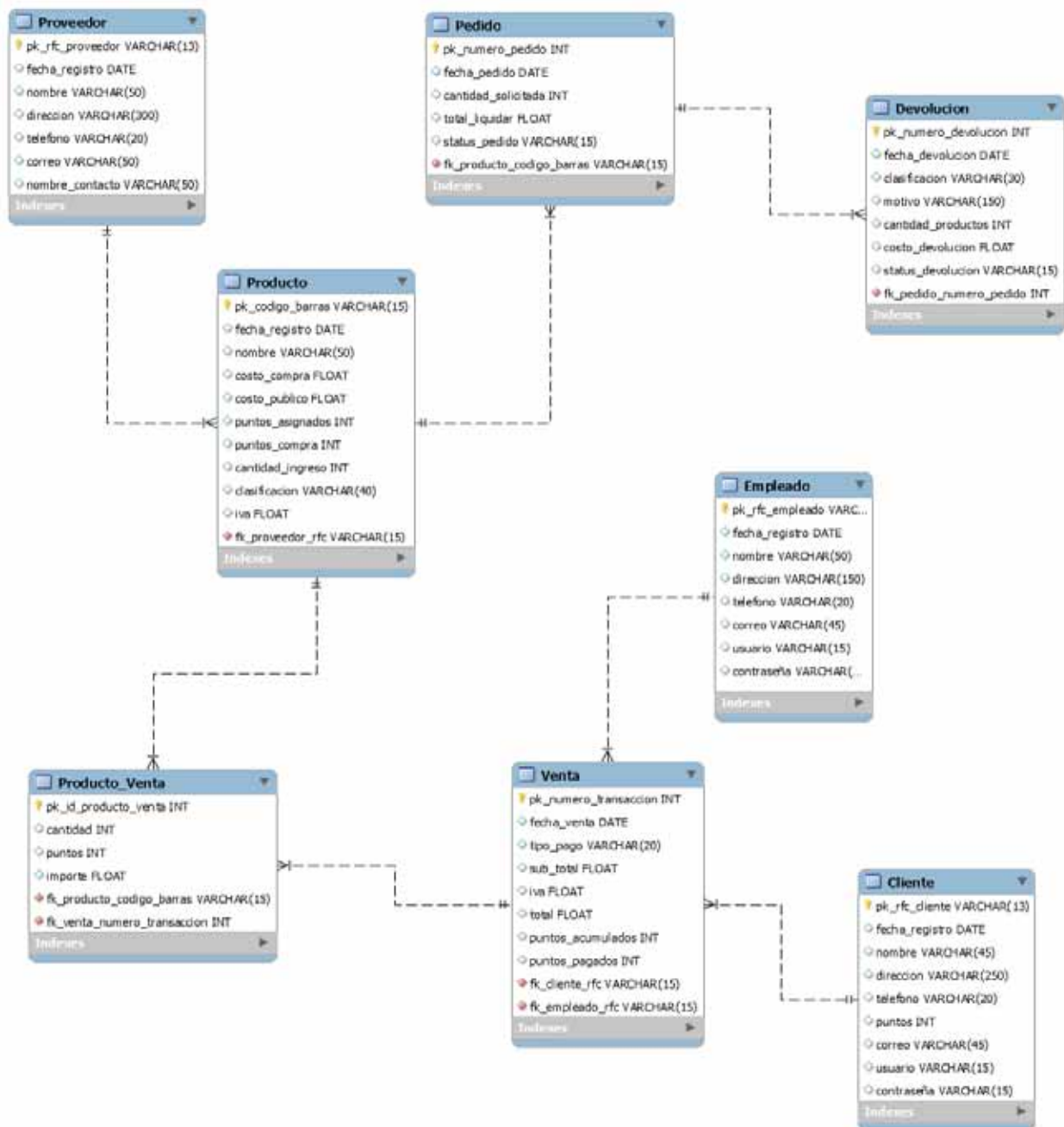


Figura 7. Diagrama Entidad-Relación de la base de datos

6.3 Diccionario de datos

En la Tabla 1 se describe la información que maneja cada una de las tablas que conforman la base de datos del sistema.

Nombre	Descripción
Proveedor	Información de los proveedores
Empleado	Información personal de los empleados
Cliente	Información personal de los clientes
Producto	Información detallada de los artículos que estarán a la venta
Venta	Información de cada venta realizada
Producto_Venta	Relación entre el producto y la venta de los artículos del negocio
Pedido	Información detallada de los pedidos que se realicen
Devolución	Información detallada de las devoluciones

Tabla 1. Tablas que conforman la base de datos

A continuación se muestran cada una de las tablas de la base de datos, así como los campos que conforman cada tabla y los tipos de datos que permiten el flujo de información necesaria para el sistema.

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_rfc_proveedor	VARCHAR	PK	RFC del proveedor
fecha_registro	DATE		Fecha de registro del proveedor
nombre	VARCHAR		Nombre completo del proveedor
dirección	VARCHAR		Dirección del proveedor
teléfono	VARCHAR		Número telefónico del proveedor
correo	VARCHAR		Correo electrónico del proveedor
nombre_contacto	VARCHAR		Nombre completo de un contacto con el proveedor

Tabla 2. Proveedor:

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_rfc_empleado	VARCHAR	PK	RFC del empleado
fecha_registro	DATE		Fecha de registro del empleado
nombre	VARCHAR		Nombre completo del empleado
dirección	VARCHAR		Dirección del empleado
teléfono	VARCHAR		Número telefónico del empleado
correo	VARCHAR		Correo electrónico del empleado
usuario	VARCHAR		Nombre asignado por el negocio para ingresar
contraseña	VARCHAR		Clave asignada por el negocio para ingresar

Tabla 3. Empleado

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_rfc_cliente	VARCHAR	PK	RFC del cliente
fecha_registro	DATE		Fecha de registro del cliente
nombre	VARCHAR		Nombre completo del cliente
dirección	VARCHAR		Dirección del cliente
teléfono	VARCHAR		Número telefónico del cliente
puntos	INT		Puntos acumulados por las compras realizadas
correo	VARCHAR		Correo electrónico del cliente
usuario	VARCHAR		Nombre elegido por el cliente para ingresar
contraseña	VARCHAR		Clave registrada por el cliente para ingresar

Tabla 4. Cliente.

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_codigo_barras	VARCHAR	PK	Identifica cada uno de los productos que están a la venta
fecha_registro	DATE		Fecha de ingreso al negocio
nombre	VARCHAR		Nombre completo del producto
costo_compra	FLOAT		Precio para el negocio
costo_publico	FLOAT		Precio para el cliente
puntos_asignados	INT		Cantidad de puntos que ofrece cada producto al cliente
puntos_compra	INT		Puntos con los que el cliente puede pagar un artículo
cantidad_ingreso	INT		Cantidad de productos que ingresan al almacén
clasificacion	VARCHAR		Departamento del negocio al que pertenece cada producto
iva	FLOAT		Impuesto que se asigna a los productos
fk_proveedor_rfc	VARCHAR	FK (Proveedor)	RFC del proveedor

Tabla 5. Producto.

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_id_producto_venta	INT	PK	Producto contenido en una venta
cantidad	INT		Cantidad de productos contenidos en una venta
puntos	INT		Cantidad de puntos que asigna cada producto de la venta
importe	FLOAT		Total a pagar por cada producto de la venta
fk_producto_codigo_barras	VARCHAR	FK (Producto)	Relación entre el código de barras y el producto que se vendió
fk_venta_numero_transaccion	INT	FK (Venta)	

Tabla 6. Venta.

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_numero_transaccion	INT	PK	Número de venta
fecha_venta	DATE		Fecha en que se realiza la venta
tipo_pago	VARCHAR		Forma en que se realizará el pago de la venta
sub_total	FLOAT		Cantidad a pagar sin el impuesto de los artículos comprados
iva	FLOAT		Impuesto que se asigna a los productos
total	FLOAT		Cantidad de dinero a pagar por la venta realizada
puntos_acumulados	INT		Total de puntos que asignó la venta
puntos_pagados	INT		Total de puntos que se utilizaron para el pago de la venta
fk_cliente_rfc	VARCHAR	FK (Cliente)	RFC del cliente
fk_empleado_rfc	VARCHAR	FK (Empleado)	RFC del empleado

Tabla 7. Producto_Venta.

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_numero_pedido	INT	PK	Número de pedido realizado
fecha_pedido	DATE		Fecha en que se realiza el pedido
cantidad_solicitada	INT		Cantidad de productos solicitados en el pedido
total_liquidar	FLOAT		Cantidad a pagar por el pedido
status_pedido	VARCHAR		Información sobre la situación actual del pedido
fk_producto_codigo_barras	VARCHAR	FK (Producto)	Relación entre el código de barras y el producto pedido

Tabla 8. Pedido.

Columna	Tipo de Dato	Tipo Llave	Descripción
pk_numero_devolucion	INT	PK	Número de devolución realizada
fecha_devolucion	DATE		Fecha en la que se realizó la devolución
clasificacion	VARCHAR		Departamento al que pertenece el artículo de la devolución
motivo	VARCHAR		Información sobre el por qué se realiza la devolución
cantidad_productos	INT		Número de productos de la devolución
costo_devolucion	FLOAT		Cantidad total de la devolución
status_devolucion	VARCHAR		Información sobre la situación actual de la devolución
fk_pedido_numero_pedido	INT	FK (Pedido)	Relación entre los artículos devueltos y el número de pedido al que pertenece

Tabla 9. Devolución.

6.4 Vistas

En las siguientes páginas se muestran las vistas de la aplicación de escritorio y de la aplicación web que conforman el sistema.

6.4.1 Vistas aplicación de Escritorio

LOGIN

La pantalla Login permite el ingreso al sistema a través de un usuario y contraseña, de esta manera se garantiza la seguridad del mismo. El código que implementa dicha pantalla se muestra en el Anexo D.4.



The image shows a screenshot of a desktop application window titled "Login". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area features a red shopping cart icon on the left and the text "Mi Negocio" in a blue, cursive font on the right. Below the logo, there are two input fields: "Usuario:" followed by a text box containing the placeholder "Escriba su Usuario", and "Clave:" followed by a text box containing the placeholder "Escriba su Password". At the bottom center, there is a green rectangular button with a circular icon of three people and the text "Ingresar".

Figura 8. Login del sistema

PÁGINA PRINCIPAL

En la Figura 9 se muestran las secciones en las cuales está dividido el sistema de información.



Figura 9. Página principal del sistema de información

MENÚ ALMÁCEN

En la Figura 10 se muestran las secciones en las que se divide el almacén y que permiten gestionar las actividades y optimizar el manejo de la información de esta área.



Figura 10. Menú almacén

MENÚ PRODUCTOS

La Figura 11 muestra el menú productos y las opciones existentes para el manejo de la información de los artículos del almacén.

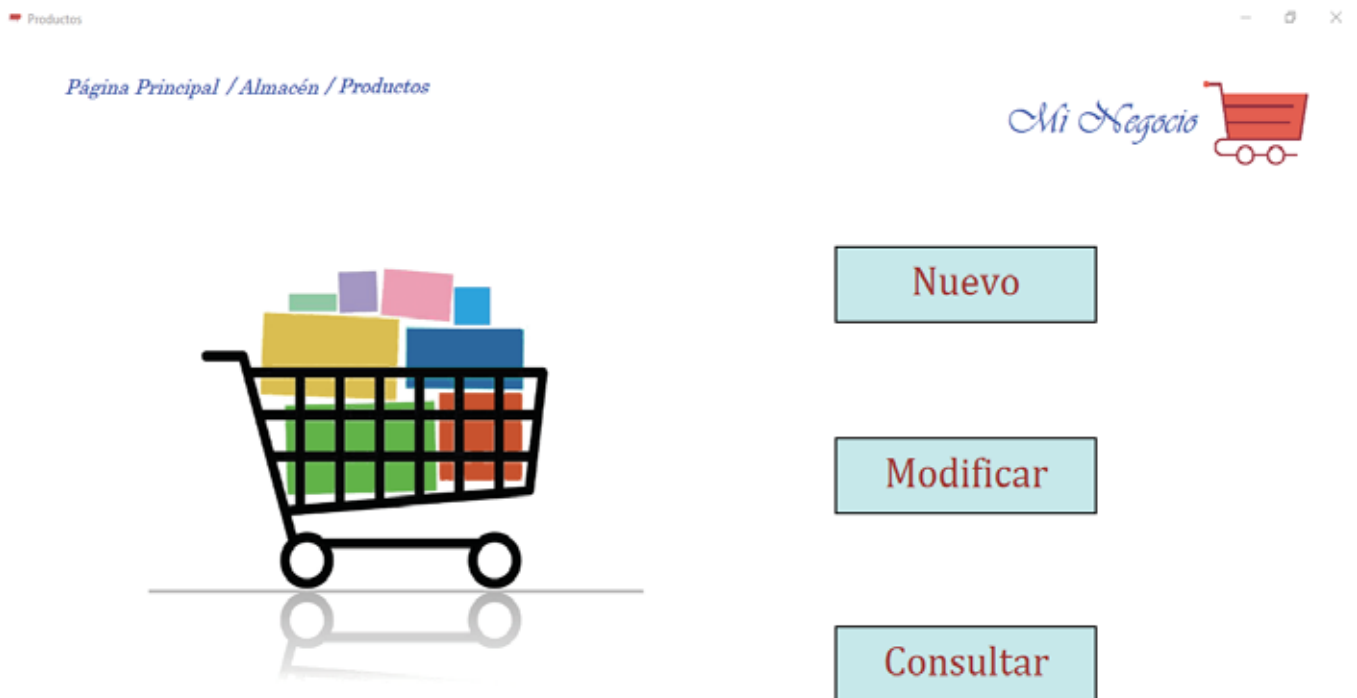


Figura 11. Menú productos

SECCIONES MENÚ PRODUCTO

En las Figuras 12, 13 y 14 se muestran las secciones que componen el menú productos, con estas secciones se busca organizar y facilitar la captura de la información de cada ingreso a esta área, se garantiza la posibilidad de cambiar datos de los productos que se registran y se hace posible encontrar los artículos existentes en el almacén. El código que permite la implementación de dichas pantallas se puede consultar en el Anexo D.6.

REGISTRAR NUEVO PRODUCTO

The screenshot shows a web browser window titled "Nuevo Producto". The breadcrumb navigation at the top reads "Página Principal / Almacén / Productos / Nuevo Producto". In the top right corner, there is a logo for "Mi Negocio" with a red shopping cart icon. Below the breadcrumb, the text "Capture la información que se solicita a continuación:" is displayed. The main form area contains the following fields and controls:

- Fecha: (with a calendar icon)
- Código de Barras:
- Nombre del Producto:
- Costo de Compra:
- Costo al Público:
- Puntos Asignados:
- Puntos Compra:
- Proveedor:
- Cantidad de Ingreso:
- Clasificación:

At the bottom of the form, there is a section for "I.V.A.:" with two radio buttons: "Agregar IVA 16%" (which is selected) and "No Agregar IVA". Below the form, there are two buttons: a red "X" button labeled "Falle" and a green checkmark button labeled "Registrar".

Figura 12. Sección "Registrar nuevo producto"

MODIFICAR PRODUCTO

The screenshot shows a web browser window titled "Modificar Producto". The breadcrumb navigation at the top left reads "Página Principal / Almacén / Productos / Modificar". The top right corner features the logo "Mi Negocio" with a shopping cart icon.

The main content area is divided into two sections:

- Lista de Productos:** A table with a header "Nombre de Productos" and a single empty row. To the right of the table are two buttons: "Modificar" (with a pencil icon) and "Eliminar" (with a trash can icon).
- Datos de los Productos:** A form with the following fields:
 - Fecha:
 - Código de Barras:
 - Nombre del Producto:
 - Costo de Compra:
 - Costo al Público:
 - Puntos Asignados:
 - Puntos Compra:
 - Proveedor:
 - Cantidad de Ingreso:
 - Clasificación:

At the bottom of the form, there are two radio buttons for "I.V.A.": "Agregar IVA 16%" and "No Agregar IVA".

At the bottom of the page, there are two large buttons: a red "Salir" button with a white 'X' icon and a green "Guardar" button with a white checkmark icon.

Figura 13. Sección "Modificar producto"

CONSULTAR PRODUCTO

The screenshot shows a web browser window titled 'Consultar Productos'. The breadcrumb navigation at the top reads 'Página Principal / Almacén / Productos / Consultar'. In the top right corner, there is a logo for 'Mi Negocio' featuring a red shopping cart icon.

The main content area contains two search panels:

- Rango de Consulta:** Includes two input fields labeled 'Desde:' and 'Hasta:', each with a calendar icon to its right.
- Consulta Avanzada:** Includes three fields: 'Código de Barras:' with a text input and a 'Buscar' button; 'Nombre del Producto:' with a text input; and 'Proveedor:' with a dropdown menu labeled 'Seleccionar un Proveedor'.

Below these panels is a green 'Consultar' button. At the bottom of the main content area is a red 'Salir' button with a white 'X' icon.

At the bottom of the browser window, a table header is visible with the following columns: CÓDIGO BARRAS, PROVEEDOR, FECHA, NOMBRE, \$ COMPRA, \$ PUBLICO, PTE. ABG., PTE. COMP., INGRESO, CLASIFICACIÓN, and IVA.

Figura 14. Sección "Consultar producto"

MENÚ PROVEEDORES

La Figura 15 muestra las tres secciones que componen el menú proveedores, con ellas se puede gestionar la información de cada proveedor.



Figura 15. Menú proveedores

SECCIONES MENÚ PROVEEDORES

En las Figuras 16, 17 y 18 se muestran las secciones que componen el menú proveedores, con ellas es posible ingresar la información correspondiente para facilitar las consultas, se posibilita el cambio de información de los proveedores antes registrados, facilitando el manejo de estos datos, además es posible observar la información registrada en el sistema de cada uno de los proveedores del negocio. El código que permite la implementación de las 3 pantallas se puede encontrar en el Anexo D.8.

NUEVO PROVEEDOR



The screenshot shows a web browser window titled "Nuevo Proveedor". The breadcrumb navigation at the top reads "Página Principal / Almacén / Proveedores / Nuevo Proveedor". In the top right corner, there is a logo that says "Mi Negocio" next to a red shopping cart icon. Below the breadcrumb, the text "Capture la información que se solicita a continuación:" is displayed. The form contains several input fields: "R.F.C." with a dropdown menu, "Fecha" with a date picker, "Nombre del Proveedor", "Dirección", "Teléfono", "Correo", and "Contacto". At the bottom of the form, there are two buttons: a red "Salir" button with a white 'X' icon and a green "Registrar" button with a white checkmark icon.

Figura 16. Sección "Nuevo proveedor"

MODIFICAR PROVEEDOR

The screenshot shows a web application window titled "Modificar Proveedor". The breadcrumb navigation at the top reads "Página Principal / Almacén / Proveedores / Modificar". In the top right corner, there is a logo for "Mi Negocio" featuring a red shopping cart icon.

The main content area is divided into two sections:

- Lista de Proveedores:** This section contains a table with the header "Nombre de Proveedores" and a single empty row. To the right of the table are two buttons: "Modificar" (with a pencil icon) and "Eliminar" (with a trash can icon).
- Datos de los Proveedores:** This section contains a form with the following fields:
 - R.F.C.:
 - Fecha:
 - Nombre del Proveedor:
 - Dirección:
 - Teléfono:
 - Correo:
 - Contacto:


At the bottom of the form, there are two buttons: a red "Salir" button with a white 'X' icon and a green "Guardar" button with a white checkmark icon.

Figura 17. Sección "Modificar proveedor"

CONSULTAR PROVEEDOR

Consultar Proveedor

Página Principal / Almacén / Proveedores / Consultar

Mi Negocio 

Consultar

Proveedor:

Consultar

RFC	FECHA REGISTRO	NOMBRE	DIRECCIÓN	TELÉFONO	CORREO	CONTACTO
-----	----------------	--------	-----------	----------	--------	----------


 Salir

Figura 18. Sección "Consultar proveedor"

MENÚ DEVOLUCIONES

En la Figura 19 se encuentra las secciones del menú devoluciones que hacen posible el manejo de la información de manera sistemática.



Figura 19. Menú Devoluciones

SECCIONES MENÚ DEVOLUCIONES

En las Figuras 20, 21 y 22 se muestran las secciones que componen el menú devoluciones, con ellas es posible capturar la información necesaria para ingresar nuevas devoluciones y gestionar así los productos existentes en el almacén, se posibilita el cambio de datos en las devoluciones realizadas, así como la consulta de las mismas. La implementación de las pantallas antes descritas se puede consultar en el Anexo D.3.

REGISTRAR DEVOLUCIÓN

The screenshot shows a web browser window titled "Registrar Devolución". The breadcrumb navigation at the top reads "Página Principal / Almacén / Devoluciones / Registrar". In the top right corner, there is a logo for "Mi Negocio" with a red shopping cart icon. The main heading of the form is "Capture la información que se solicita a continuación:". The form fields are as follows:

- Fecha: (with a calendar icon)
- Pedido:
- Código de Barras:
- Producto:
- Proveedor:
- Cantidad de Productos:
- Costo por Unidad:
- Costo de la Devolución:
- Motivo:
- Clasificación:
- Estatus:

At the bottom of the form, there are two buttons: a red "Salir" button with a white 'X' icon and a green "Registrar" button with a white checkmark icon.

Figura 20. Sección "Registrar devolución"

MODIFICAR DEVOLUCIÓN

Modificar Devoluciones

Página Principal / Almacón / Devoluciones / Modificar

Mi Negocio 

Lista de Devoluciones

Seleccione un Proveedor

FECHA	NUMERO DE DEVOLUCIÓN
-------	----------------------

Modificar

Eliminar

Datos de las Devoluciones

Fecha:

Pedido:

Código de Barras:

Producto:

Proveedor:

Cantidad de Productos:

Costo por Unidad:

Costo de la Devolución:

Motivo:

Clasificación:

Estatus:

Figura 21. Sección “Modificar devolución”

CONSULTAR DEVOLUCIÓN

The screenshot shows a web browser window titled 'Consultar Devoluciones'. The breadcrumb navigation at the top left reads 'Página Principal / Almacén / Devoluciones / Consultar'. In the top right corner, there is a logo for 'Mi Negocio' featuring a red shopping cart icon. The main content area contains a search form with the following elements:

- A dropdown menu for 'Proveedor:' with the text 'Seleccionar un Proveedor'.
- Radio buttons for '1 Devolución' and 'Todas las Devoluciones', with the latter being selected.
- A text input field for 'Devolución:' with the placeholder text 'Número de Devolución'.
- A green button labeled 'Ver'.

Below the search form is a table with the following columns: #DEVOLUCIÓN, FECHA, CÓDIGO BARRAS, PRODUCTO, PROVEEDOR, CANTIDAD, PRECIO, TOTAL, MOTIVO, CLASIFICACIÓN, and ESTATUS. The table body is currently empty. At the bottom center of the page, there is a red button with a white 'X' icon and the text 'Salir'.

Figura 22. Sección “Consultar devolución”

MENÚ PEDIDOS

En la Figura 23 muestra las secciones que integran el menú pedidos, con ellas es posible organizar de forma sistemática la información de esta actividad del negocio.

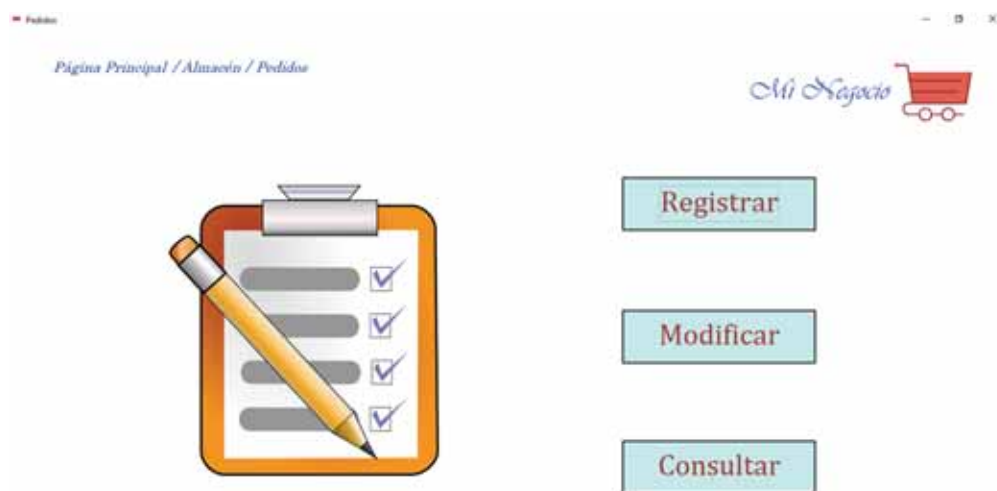


Figura 23. Menú Pedidos

SECCIONES MENÚ PEDIDOS

En las Figuras 24, 25 y 26 se muestran las secciones que componen el menú pedidos, con ellas se busca facilitar el registro, la modificación y la consulta de los pedidos realizados por el negocio y de esta manera tener un control de los mismos. Para implementar dichas pantallas se utiliza el código que se encuentra en el Anexo D.5.

REGISTRAR PEDIDO



The screenshot shows a web browser window titled "Registrar Pedido". The breadcrumb navigation at the top reads "Página Principal / Almacén / Pedidos / Registrar". In the top right corner, there is a logo for "Mi Negocio" featuring a red shopping cart icon. Below the navigation, the instruction "Capture la información que se solicita a continuación:" is displayed. The form itself is a light gray box containing the following fields: "Fecha:" with a date picker icon; "Código de Barras:" with a text input field and a small "Ingresar Código" button; "Producto:"; "Proveedor:"; "Cantidad a Solicitar:" with a text input field containing the word "Cantidad"; "Costo por Unidad:"; "Costo Total:"; and "Estatus:" with a dropdown menu showing "Seleccione un Opción". At the bottom of the form, there are two buttons: a red "Salir" button with a white 'X' icon and a green "Registrar" button with a white checkmark icon.

Figura 24. Sección "Registrar pedido"

MODIFICAR PEDIDO

Modificar Pedido

Página Principal / Almacén / Pedidos / Modificar

Mi Negocio 

Lista de Pedidos

Proveedor:

FECHA	NÚMERO DE PEDIDO
-------	------------------

 Modificar

 Eliminar

Datos de los Pedidos

Fecha:

Código de Barras:

Producto:

Proveedor:

Cantidad a Solicitar:

Costo por Unidad:

Costo Total:

Estatus:

 Salir


 Guardar

Figura 25. Sección “Modificar pedido”

CONSULTAR PEDIDO

Consultar Pedidos

Página Principal / Almacén / Pedidos / Consultar

Mi Negocio 

Consultar Pedidos

Proveedor:

1 Pedido Todos los Pedidos

Pedido:

#PEDIDO	FECHA	CÓDIGO BARRAS	PRODUCTO	PROVEEDOR	CANTIDAD	PRECIO	TOTAL	ESTATUS
---------	-------	---------------	----------	-----------	----------	--------	-------	---------

Figura 26. Sección "Consultar pedido"

MENÚ CUENTAS POR PAGAR

La Figura 27 muestra el formulario para poder consultar las cuentas por pagar en el cual se puede observar los pagos de pedidos pendientes. El código para la implementación de esta pantalla se puede encontrar en el Anexo D.5.

Cuentas por Pagar

Página Principal / Almacén / Cuentas por Pagar

Mi Negocio

Consultar Cuentas por Pagar

Proveedor:

1 Cuenta Todas las Cuentas

Pedido:

# PEDIDO	FECHA	CÓDIGO BARRAS	PRODUCTO	PROVEEDOR	CANTIDAD	PRECIO	TOTAL	ESTATUS
----------	-------	---------------	----------	-----------	----------	--------	-------	---------

Figura 27. Formulario de cuentas por pagar.

MENÚ VENTAS

En la Figura 28 se muestran las secciones en las que se divide las ventas del negocio, permite el registro y consulta de las mismas de manera sencilla.

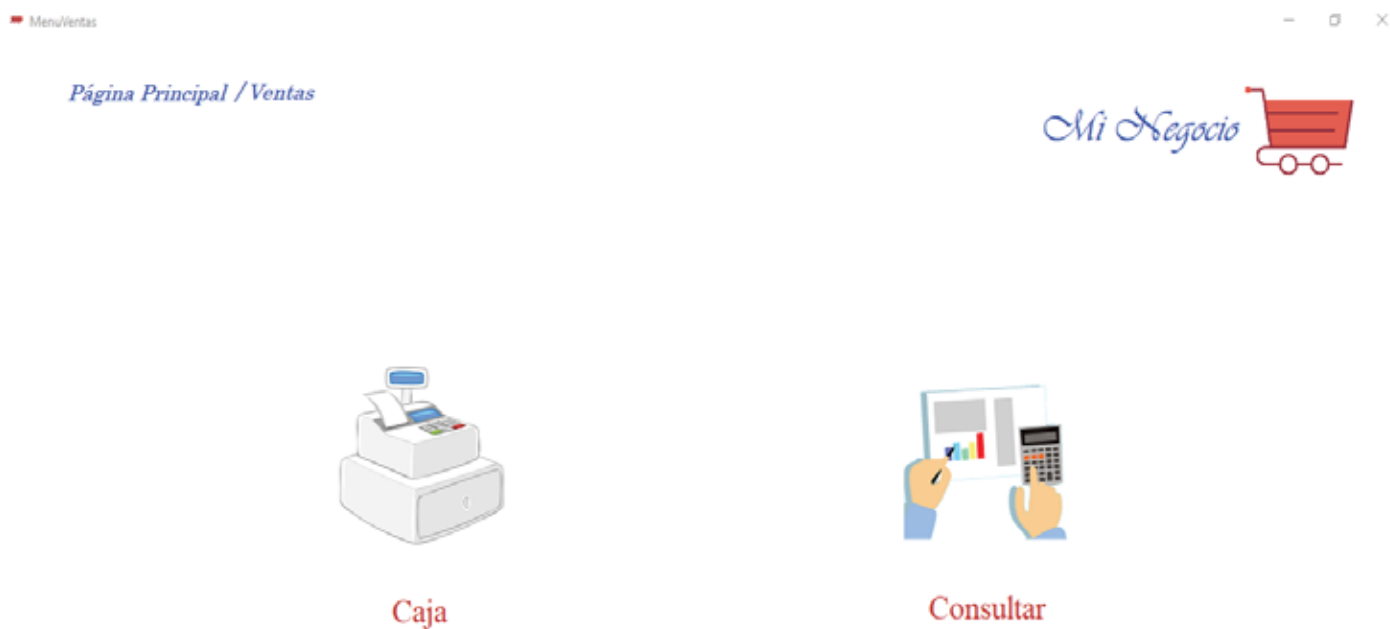


Figura 28. Menú ventas.

REGISTRAR VENTAS

La Figura 29 muestra la página de la caja que permite el registro de las ventas realizadas de forma rápida y sencilla. La implementación de esta pantalla esta compuesta por distintos métodos que se encuentran en los Anexos D.1, D.2, D.6, D.7 y D.9

Registrar Ventas

Página Principal / Ventas / Caja

Mi Negocio

Fecha: 2018-04-01 Clave Vendedor: Nombre Vendedor:

Tipo de Pago

- Efectivo
- Puntos

Cliente

Número de Cliente:

Nombre Cliente:

Puntos Acumulados:

Código de Barras:

CANTIDAD	PRODUCTO	PRECIO	PUNTO	IMPORTE EFECTIVO
----------	----------	--------	-------	------------------

Sub-total:

I.V.A.:

Pagar con Puntos:

Puntos Acumulados:

Total:

Figura 29. Sección “Registrar ventas”

CONSULTAR VENTAS

En la Figura 30 se muestra la sección de consultar ventas la cual permite al administrador o sus empleados revisar las ventas realizadas en determinado periodo. La implementación de esta pantalla se puede realizar con el código que se encuentra en el Anexo D.9.

The screenshot shows a web browser window titled 'Consultar Ventas'. The page has a breadcrumb trail 'Página Principal / Ventas / Consultar' and a logo 'Mi Negocio' with a shopping cart icon. There are two main input sections: 'Rango de Consulta' with 'Desde:' and 'Hasta:' text boxes, and 'Consulta Avanzada' with a 'Vendedor:' text box and a 'Forma de Pago:' dropdown menu. A green 'Consultar' button is centered below these sections. At the bottom, there is a red 'Salir' button with a white 'X' icon. Below the buttons is a table with the following headers: TRANSACCIÓN, FECHA, VENDEDOR, TIPO DE PAGO, IMPORTE EFECTIVO, and PUNTOS ACUMULADOS. The table body is currently empty.

Figura 30. Sección “Consultar ventas”

MENÚ CLIENTES

En la Figura 31 se muestran las secciones en las que se divide el área de clientes del negocio, esto permite registrar y organizar la información de los clientes de mejor manera.

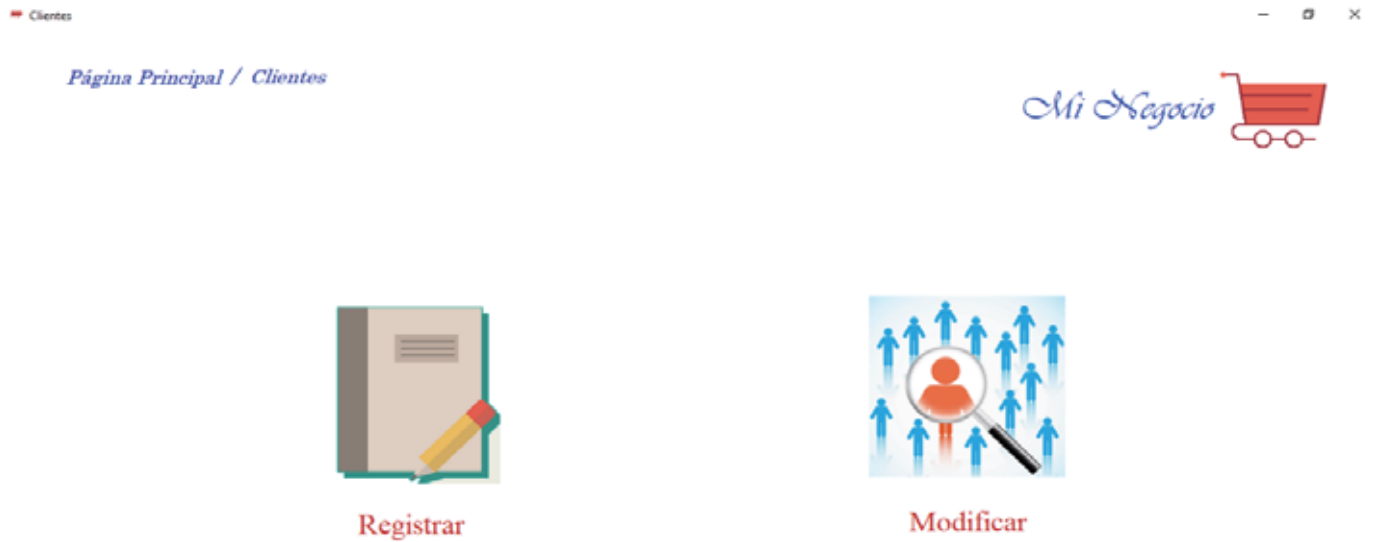


Figura 31. Menú Clientes.

SECCIONES DE MENÚ CLIENTES

En las Figuras 32 y 33 se muestran las secciones que componen el menú clientes, con ellas es posible registrar y modificar la información de los clientes del negocio, con lo que se busca que puedan obtener beneficios como acumular puntos para ser utilizados en próximas compras. Para implementar las 2 pantallas descritas anteriormente se utilizan los métodos contenidos en el Anexo D.2.

REGISTRAR CLIENTE

The screenshot shows a web browser window titled "Registrar Cliente". The breadcrumb navigation at the top left reads "Página Principal / Clientes / Registrar". The top right corner features the logo "Mi Negocio" with a red shopping cart icon. Below the navigation, the instruction "Capture la información que se solicita a continuación:" is displayed. The main form area contains several input fields: "R.F.C.:" with a placeholder "ingresa RFC", "Fecha:" with a calendar icon, "Nombre del Cliente:" with a placeholder "Ingresar Nombre", "Dirección:" with a placeholder "Ingresar Dirección", "Teléfono:" with a placeholder "(00) 55-55-55-00", and "Correo:" with a placeholder "correo@correo.com". At the bottom of the form, there are two buttons: a red "Salir" button with a white 'X' icon and a green "Registrar" button with a blue checkmark icon.

Figura 32. Sección "Registrar cliente".

MODIFICAR CLIENTE

The screenshot shows a web browser window titled "Modificar Cliente". The breadcrumb navigation at the top left reads "Página Principal / Clientes / Modificar". The logo "Mi Negocio" with a shopping cart icon is in the top right corner. The interface is divided into two main sections:

- Buscar Cliente:** A search box labeled "Número de Cliente:" with a placeholder "Ingresar RFC". Below it are two buttons: "Eliminar" (with a red pill icon) and "Modificar" (with a yellow pencil icon).
- Datos del Cliente:** A form with the following fields:
 - R.F.C.: Input field with placeholder "Ingresar RFC".
 - Fecha: Date picker.
 - Nombre del Cliente: Input field with placeholder "Ingresar nombre".
 - Dirección: Input field with placeholder "Ingresar Dirección".
 - Teléfono: Input field with placeholder "200 00-00-00-00".
 - Correo: Input field with placeholder "james@correo.com".

At the bottom of the form are two large buttons: a red "Salir" button with a white 'X' icon and a green "Guardar" button with a white checkmark icon.

Figura 33. Sección "Modificar cliente".

6.4.2 Vistas aplicación Web

En la página principal de la aplicación web, se encuentra un carrusel que muestra las promociones del negocio y permite el ingreso a clientes y administradores para consultar información.

PÁGINA DE INICIO



Figura 34. Página de inicio.

SECCIONES DE MENÚ CLIENTE


La Figura 35 muestra las secciones que conforman el menú cliente de la aplicación web, permite a los clientes seleccionar las opciones que tienen para consultar su información o registrarse en el sistema.



Figura 35. Menú cliente.

NUEVA CUENTA

En la Figura 36 se muestra la sección “Nueva Cuenta”, con ella se facilita el ingreso de los datos de los clientes nuevos, así como la búsqueda en caso de tener un pre-registro. La implementación de esta pantalla se lleva a cabo con el código encontrado en el Anexo D.10.

Mi Negocio 

[Inicio](#) / [Menú Cliente](#) / Nueva Cuenta

¿Ya cuenta con el Pre-Registro de su cuenta?

Ingrese su RFC:

Usuario:

Clave:

Fecha:

RFC:

Nombre:

Dirección:

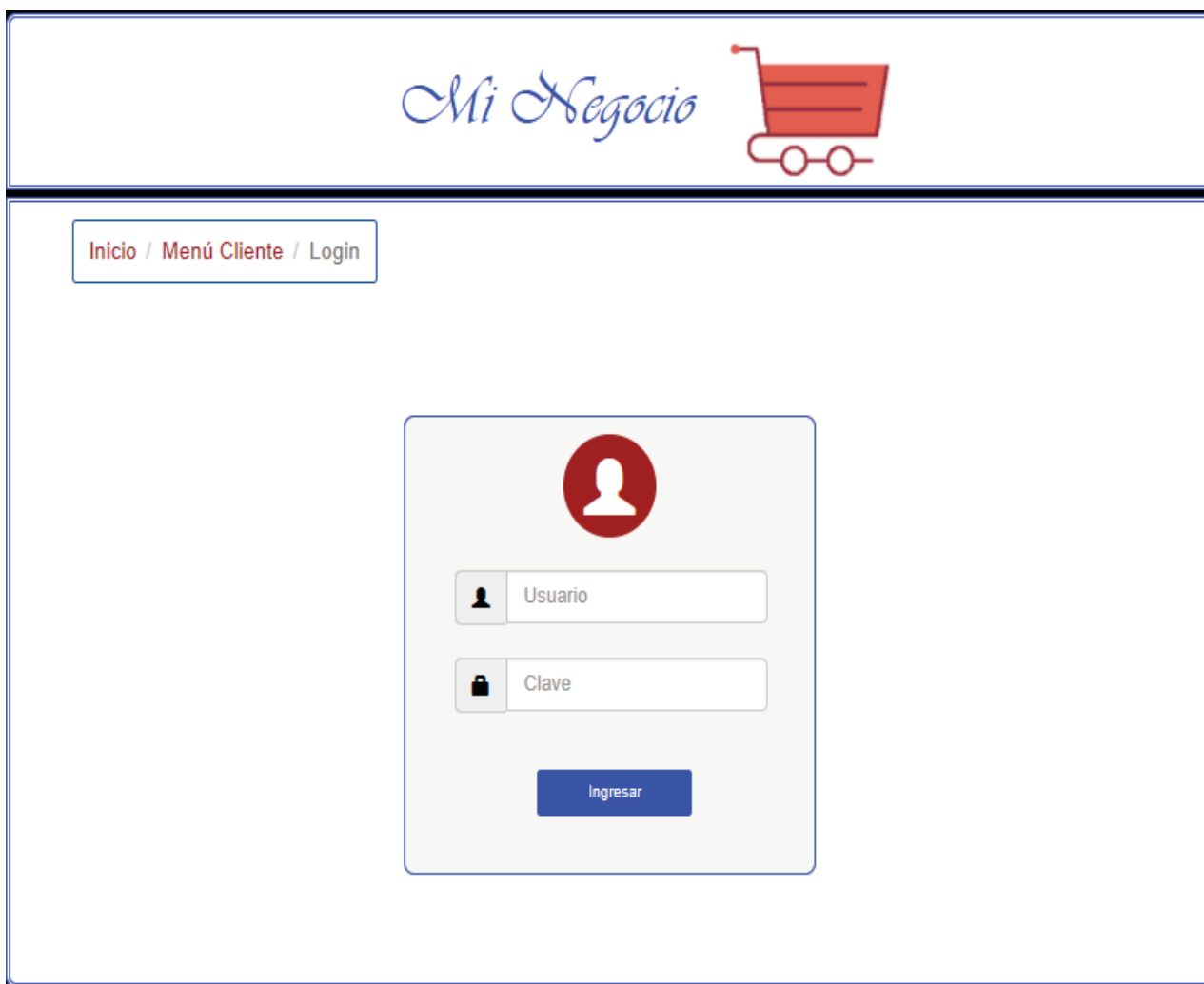
Correo:

Teléfono:


Figura 36. Sección “Nueva Cuenta”


LOGIN CLIENTE


La Figura 37 muestra la sección de Login de los clientes, con la cual se facilita la consulta de puntos acumulados por los clientes y garantiza la seguridad del sistema. El código que permite la implementación de esta pantalla se puede consultar en el Anexo D.10.




Inicio / Menú Cliente / Login

Mi Negocio 



 Usuario

 Clave

Ingresar

Figura 37. Sección "Login"

PUNTOS ACUMULADOS

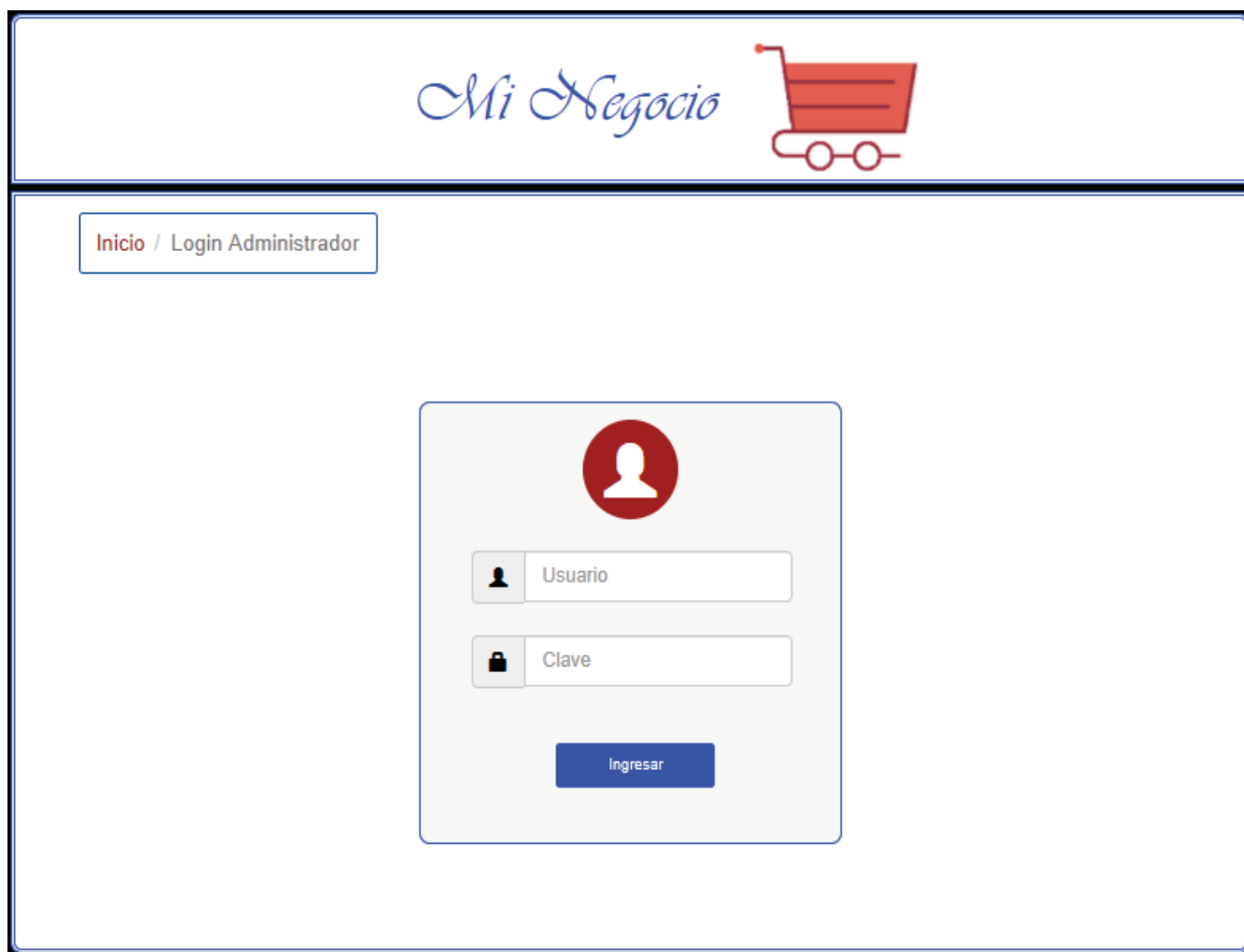
La Figura 38 muestra la sección de puntos acumulados por el cliente y las compras realizadas por este. El código para implementar dicha pantalla se encuentra en los Anexos D.10 y D.11.



Figura 38. Sección "Puntos"

LOGIN ADMINISTRADOR

La Figura 39 muestra la sección de Login del administrador, con esta sección se garantiza la seguridad del sistema para realizar consultas referentes a la administración del negocio. El código que permite la implementación de esta pantalla se puede consultar en el Anexo D.12.



The image shows a web interface for an administrator login. At the top, there is a header with the text "Mi Negocio" in a blue script font and a red shopping cart icon. Below the header, there is a navigation bar with a link "Inicio / Login Administrador". The main content area features a central login form with a red circular profile icon at the top. Below the icon are two input fields: "Usuario" (with a person icon) and "Clave" (with a lock icon). A blue "Ingresar" button is positioned below the input fields.

Figura 39. Login de administrador

MENÚ ADMINISTRADOR

En la Figura 40 se muestran las opciones existentes para la consulta de las principales actividades del negocio.




Figura 40. Menú del administrador

SECCIONES MENÚ ADMINISTRADOR

Las Figuras 41, 42 y 43 permiten mostrar la información que es posible obtener de las consultas, según los filtros de búsqueda, que puede realizar el administrador en las principales secciones del negocio que son: ventas, almacén y proveedores. Para implementar dichas pantallas se utilizan los códigos que están contenidos en los Anexos D.11, D.13 y D.14.

CONSULTAR VENTAS

Mi Negocio 

[Inicio](#) / [Menú Encargado](#) / [Ventas](#)

Rango de Consulta:

Desde:

Hasta:

Consulta Avanzada:

Forma de Pago:

Vendedor:

Venta	Fecha	Vendedor	Tipo de Pago	Sub-Total	L.V.A.	Total
-------	-------	----------	--------------	-----------	--------	-------

Figura 41. Sección “Consultar ventas”

CONSULTAR ALMACÉN



[Inicio](#) / [Menú Encargado](#) / [Almacén](#)

Rango de Consulta:

Desde:

Hasta:

Consulta Avanzada:


Código de Barras:

Proveedor:

Fecha	Código	Producto	Proveedor	Costo Compra	Costo Público	Puntos	Existencia
-------	--------	----------	-----------	--------------	---------------	--------	------------

Figura 42. Sección “Consultar almacén”

CONSULTAR PROVEEDORES

Mi Negocio 

[Inicio](#) / [Menú Encargado](#) / [Proveedores](#)

Proveedor:

RFC	Fecha	Nombre	Dirección	Teléfono	Correo	Contacto

Figura 43. Sección “Consultar proveedores”

7.Resultados

Objetivos Específicos:

- Diseñar e implementar un módulo para supervisar los movimientos efectuados en el almacén y ventas.

Este objetivo se cumplió satisfactoriamente al implementar las pantallas mostradas en las Figuras de la 39 a la 43 con las cuales se puede consultar la información de Ventas, movimientos en Almacén y ver los datos de Proveedores, todo esto de manera remota, permitiendo así al Encargado o Administrador supervisar el almacén y ventas sin tener que estar en el pequeño negocio.

- Diseñar e implementar un módulo para el registro y consulta de puntos de fidelidad.

Este objetivo se cumple a través de las pantallas implementadas que se muestran en las Figuras de la 35 a la 38, con las cuales se puede realizar el registro del Cliente y si ya se cuenta con un pre-registro se puede realizar la búsqueda de este, en el ultimo caso para concluirlo permite agregar un usuario y password definido por el Cliente y así poder visualizar las compras realizadas y la forma de pago.

- Diseñar e implementar un módulo para gestionar las actividades del almacén.

Este objetivo fue cumplido por medio de las 5 secciones que conforman la parte de Almacén de la aplicación de escritorio (“Productos”, “Proveedores”, “Pedidos”, “Devoluciones” y “Cuentas por Pagar”), dichas secciones pueden ser consultadas por medio de un menú como se muestra en la Figura 10 y cada sección contiene un sub-menú que permite elegir la acción que se desea realizar, los sub-menús se muestran en las Figuras 11, 15, 19, y 23. La implementación de las interfaces que se pueden apreciar en las Figuras 12, 13, 14, 16, 17, 18, 20, 21, 22, 24, 25, 26 y 27, permitieron el manejo de la información para cumplir con este objetivo en su totalidad.

- Diseñar e implementar un módulo para el registro de ventas y promociones con puntos de fidelidad.

Este objetivo fue cumplido por medio de las pantallas implementadas que se muestran en las Figuras 28, 29 y 30 que en conjunto permiten registrar y consultar las ventas efectuadas en el negocio, también permiten registrar los puntos acumulados por cada compra asignados a un Cliente en específico y dando la opción de pagar con puntos.

- Diseñar e implementar un módulo para gestionar la información de los clientes.

Este objetivo fue cumplido con las interfaces implementadas que se muestran en las Figuras 32 y 33 las cuales permiten realizar un pre-registro del Cliente dejando solo la opción de registrar usuario y password a través de la plicación web o en caso de ser necesario la modificación de la información de este.

8. Conclusiones

El objetivo general: “Implementar un sistema para automatizar las actividades de control de información en los pequeños negocios” Se cumple, ya que:

- Con el sistema desarrollado se logra sistematizar la información del almacén, proveedores y las ventas de un negocio, facilitando así al usuario el manejo de información de gran importancia para el mismo.
- El desarrollo de este sistema de información brinda una serie de beneficios para el negocio que lo implemente y sus clientes, esto se reflejará en buenos resultados de administración y venta, logrando así mayor ventaja competitiva.
- Este sistema puede mejorar implementando nuevas funcionalidades para su crecimiento como por ejemplo, agregar una nueva sección para administrar la información de los empleados, poder imprimir reportes de las consultas realizadas, agregar un scanner para facilitar aun mas el manejo de los productos o artículos.

9. Referencias Bibliográficas

[1] A. Soto López, “Sistema de administración para pequeñas y medianas empresas en México”, Tesis de Licenciatura, Universidad Nacional Autónoma de México, 2011. [En línea]

Disponible:[http://132.248.9.195/ptb2011/agosto/0671713/0671713_A1.pdf#search=%22sistemas de informacion%22](http://132.248.9.195/ptb2011/agosto/0671713/0671713_A1.pdf#search=%22sistemas%20de%20informacion%22)

[2] J. Aguilar Herrera, O.H. Bañuelos Hernández. “Sistema de inventario y ventas para tlapalería con pronóstico de mercado”, Reporte de Proyecto Terminal, Universidad Autónoma Metropolitana unidad Azcapotzalco, 2009. [En línea]
Disponible: <http://espartaco.azc.uam.mx/tesis/x16860.pdf>

[3] E. Barrera Romero, “Diseño de un sistema de información para una empresa pequeña de computo”, Tesis de Licenciatura, Universidad Nacional Autónoma de México, 2010. [En línea]. Disponible:
[http://132.248.9.195/ptb2010/agosto/0659970/0659970_A1.pdf#search=%22sistemas de informacion%22](http://132.248.9.195/ptb2010/agosto/0659970/0659970_A1.pdf#search=%22sistemas%20de%20informacion%22)

[4] SmartNet Admin. [En línea]. Disponible:
<http://www.smartnetadmin.com.ar/default.php>

[5] LemonPOS. [En línea]. Disponible: <http://lemonpos.sourceforge.net/>

[6] care2x. [En línea]. Disponible: <http://demo.care2x.org/index.php>

[7] J. Amaya Amaya, *Sistemas de Información Gerenciales: Hardware, Software, Redes, Internet, Diseño*, Segunda Edición. Ecoe Ediciones, 2010.

[8] M.R. González Ramírez, *Sistemas de Información para la empresa*. Publicaciones de la Universidad de Alicante, 2001.

[9] Keneth C. Laudon y Jane P. Laudon. Sistema de Información Gerencial. Octava Edición. Pearson Educación. México, 2004.

[10] J, Gutiérrez. 2008. Diagramas UML de casos de uso y de requisitos. (En línea). Consultado 04 de Abril. 2018. Formato PDF. Disponible en: http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/CasosUsoUML.pdf

[11] Vázquez Rodríguez J., “Creación de un repositorio para programas de Java utilizando la herramienta de ORM Hibernate”, Tesis profesional, Universidad de las Américas Puebla, 2006. [En línea] Disponible: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/sanchez_r_ma/

[12]<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/907/A5.pdf?sequence=5>

[13]<https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/>

[14] [https://www.ecured.cu/Swing_\(biblioteca_gr%C3%A1fica\)](https://www.ecured.cu/Swing_(biblioteca_gr%C3%A1fica))

[15] <http://cc.etsii.ull.es/ftp/antiguo/INGSOFG2/dsiproc.pdf>

[16] <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAQ1743.pdf>

[17] <http://ri.ufg.edu.sv/jspui/bitstream/11592/7213/3/005.74-C828dCapitulo%20II.pdf>

10. Anexo

ANEXO A. Código de las clases contenidas en el paquete **hibernate.beans** de la aplicación de escritorio y de la aplicación web:

1.- Cliente.java

```
package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Cliente con sus respectivos métodos
Getters y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras
Clases */

import java.sql.Date;

public class Cliente
{
    private String rfcCliente;
    private Date fechaRegistro;
    private String nombreCliente;
    private String direccion;
    private String telefono;
    private String correo;
    private Integer puntosAcumulados;
    private String usuario;
    private String contrasenia;

    /*****MÉTODOS GETTER'S Y SETTER'S *****/

    public String getRfcCliente()
    {
        return rfcCliente;
    }
    public void setRfcCliente(String rfcCliente)
    {
        this.rfcCliente = rfcCliente;
    }
    public Date getFechaRegistro()
    {
        return fechaRegistro;
    }
    public void setFechaRegistro(Date fechaRegistro)
    {
        this.fechaRegistro = fechaRegistro;
    }
    public String getNombreCliente()
    {
        return nombreCliente;
    }
    public void setNombreCliente(String nombreCliente)
    {
        this.nombreCliente = nombreCliente;
    }
    public String getDireccion()
    {
        return direccion;
    }
    public void setDireccion(String direccion)
    {
        this.direccion = direccion;
    }
    public String getTelefono()
    {
        return telefono;
    }
    public void setTelefono(String telefono)
    {

```

```

        this.telefono = telefono;
    }
    public String getCorreo()
    {
        return correo;
    }
    public void setCorreo(String correo)
    {
        this.correo = correo;
    }
    public Integer getPuntosAcumulados()
    {
        return puntosAcumulados;
    }
    public void setPuntosAcumulados(Integer puntosAcumulados)
    {
        this.puntosAcumulados = puntosAcumulados;
    }
    public String getUsuario()
    {
        return usuario;
    }
    public void setUsuario(String usuario)
    {
        this.usuario = usuario;
    }
    public String getContrasenia()
    {
        return contrasenia;
    }
    public void setContrasenia(String contrasenia)
    {
        this.contrasenia = contrasenia;
    }

    @Override
    public String toString()
    {
        String mensaje = "";

        mensaje = rfcCliente + "\n" + fechaRegistro + "\n" + nombreCliente + "\n" + direccion + "\n" +
            telefono + "\n" + correo + "\n" + puntosAcumulados + "\n" + usuario + "\n" +
contrasenia;

        return mensaje;
    }
}

```

2.- Devolución.java

```

package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Devolución con sus respectivos métodos
Getters y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras Clases
*/

import java.sql.Date;

public class Devolucion
{
    private Integer numeroDevolucion;
    private Date fechaDevolucion;
    private String clasificacion;
    private String motivo;
    private Integer cantidadProductos;
    private Float costoDevolucion;
    private String statusDevolucion;
}

```

```

private Integer pedidoNumeroPedido;

/***** MÉTODOS GETTER'S Y SETTER'S *****/

public Integer getNumeroDevolucion()
{
    return numeroDevolucion;
}
public void setNumeroDevolucion(Integer numeroDevolucion)
{
    this.numeroDevolucion = numeroDevolucion;
}
public Date getFechaDevolucion()
{
    return fechaDevolucion;
}
public void setFechaDevolucion(Date fechaDevolucion)
{
    this.fechaDevolucion = fechaDevolucion;
}
public String getClasificacion()
{
    return clasificacion;
}
public void setClasificacion(String clasificacion)
{
    this.clasificacion = clasificacion;
}
public String getMotivo()
{
    return motivo;
}
public void setMotivo(String motivo)
{
    this.motivo = motivo;
}
public Integer getCantidadProductos()
{
    return cantidadProductos;
}
public void setCantidadProductos(Integer cantidadProductos)
{
    this.cantidadProductos = cantidadProductos;
}
public Float getCostoDevolucion()
{
    return costoDevolucion;
}
public void setCostoDevolucion(Float costoDevolucion)
{
    this.costoDevolucion = costoDevolucion;
}
public String getStatusDevolucion()
{
    return statusDevolucion;
}
public void setStatusDevolucion(String statusDevolucion)
{
    this.statusDevolucion = statusDevolucion;
}
public Integer getPedidoNumeroPedido()
{
    return pedidoNumeroPedido;
}
public void setPedidoNumeroPedido(Integer pedidoNumeroPedido)
{
    this.pedidoNumeroPedido = pedidoNumeroPedido;
}

```



```

@Override
public String toString()
{
    String mensaje = "";

    mensaje = numeroDevolucion + "\n" + fechaDevolucion + "\n" + clasificacion + "\n" + motivo + "\n" +
cantidadProductos + "\n" +
        costoDevolucion + "\n" + statusDevolucion + "\n" + pedidoNumeroPedido;

    return mensaje;
}
}

```

3.- Empleado.java

```

package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Empleado con sus respectivos métodos Getters
y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras Clases */

import java.util.Date;

public class Empleado
{
    private String rfcEmpleado;
    private Date fechaRegistro;
    private String nombreEmpleado;
    private String direccion;
    private String telefono;
    private String correo;
    private String usuario;
    private String contrasenia;

    /***** MÉTODOS GETTER's Y SETTER's *****/

    public String getRfcEmpleado()
    {
        return rfcEmpleado;
    }
    public void setRfcEmpleado(String rfcEmpleado)
    {
        this.rfcEmpleado = rfcEmpleado;
    }
    public Date getFechaRegistro()
    {
        return fechaRegistro;
    }
    public void setFechaRegistro(java.sql.Date date)
    {
        this.fechaRegistro = date;
    }
    public String getNombreEmpleado()
    {
        return nombreEmpleado;
    }
    public void setNombreEmpleado(String nombreEmpleado)
    {
        this.nombreEmpleado = nombreEmpleado;
    }
    public String getDireccion()
    {
        return direccion;
    }
    public void setDireccion(String direccion)

```

```

{
    this.direccion = direccion;
}
public String getTelefono()
{
    return telefono;
}
public void setTelefono(String telefono)
{
    this.telefono = telefono;
}
public String getCorreo()
{
    return correo;
}
public void setCorreo(String correo)
{
    this.correo = correo;
}
public String getUsuario()
{
    return usuario;
}
public void setUsuario(String usuario)
{
    this.usuario = usuario;
}
public String getContrasenia()
{
    return contrasenia;
}
public void setContrasenia(String contrasenia)
{
    this.contrasenia = contrasenia;
}

@Override
public String toString()
{
    String mensaje = "";

    mensaje = rfcEmpleado + "\n" + fechaRegistro + "\n" + nombreEmpleado + "\n" + direccion + "\n" +
        telefono + "\n" + correo + "\n" + usuario + "\n" + contrasenia;

    return mensaje;
}
}

```

4.- Pedido.java

```

package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Pedido con sus respectivos métodos Getters
y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras Clases */

import java.sql.Date;

public class Pedido
{
    private Integer numeroPedido;
    private Date fechaPedido;
    private Integer cantidadSolicitada;
    private Float totalLiquidar;
    private String statusPedido;
    private String productoCodigoBarras;

    /***** MÉTODOS GETTER'S Y SETTER'S *****/
}

```

```

public Integer getNumeroPedido()
{
    return numeroPedido;
}
public void setNumeroPedido(Integer numeroPedido)
{
    this.numeroPedido = numeroPedido;
}
public Date getFechaPedido()
{
    return fechaPedido;
}
public void setFechaPedido(Date fechaPedido)
{
    this.fechaPedido = fechaPedido;
}
public Integer getCantidadSolicitada()
{
    return cantidadSolicitada;
}
public void setCantidadSolicitada(Integer cantidadSolicitada)
{
    this.cantidadSolicitada = cantidadSolicitada;
}
public Float getTotalLiquidar()
{
    return totalLiquidar;
}
public void setTotalLiquidar(Float totalLiquidar)
{
    this.totalLiquidar = totalLiquidar;
}
public String getStatusPedido()
{
    return statusPedido;
}
public void setStatusPedido(String statusPedido)
{
    this.statusPedido = statusPedido;
}
public String getProductoCodigoBarras()
{
    return productoCodigoBarras;
}
public void setProductoCodigoBarras(String productoCodigoBarras)
{
    this.productoCodigoBarras = productoCodigoBarras;
}

@Override
public String toString()
{
    String mensaje = "";

    mensaje = numeroPedido + "\n" + fechaPedido + "\n" + cantidadSolicitada + "\n" + totalLiquidar +
"\n" + statusPedido + "\n" + productoCodigoBarras;

    return mensaje;
}
}

```

5.- Producto.java

```
package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Producto con sus respectivos métodos
Getters y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras Clases
*/
import java.sql.Date;

public class Producto
{
    private String codigoBarras;
    private Date fechaRegistro;
    private String nombreProducto;
    private Float costoCompra;
    private Float costoPublico;
    private Integer puntosAsignados;
    private Integer puntosCompra;
    private Integer cantidadIngreso;
    private String clasificacion;
    private Float iva;
    private String proveedor;

    /*****MÉTODOS GETTER'S Y SETTER'S *****/

    public String getCodigoBarras ()
    {
        return codigoBarras;
    }
    public void setCodigoBarras(String codigoBarras)
    {
        this.codigoBarras = codigoBarras;
    }
    public Date getFechaRegistro ()
    {
        return fechaRegistro;
    }
    public void setFechaRegistro(Date fechaRegistro)
    {
        this.fechaRegistro = fechaRegistro;
    }
    public String getNombreProducto ()
    {
        return nombreProducto;
    }
    public void setNombreProducto(String nombreProducto)
    {
        this.nombreProducto = nombreProducto;
    }
    public Float getCostoCompra ()
    {
        return costoCompra;
    }
    public void setCostoCompra(Float costoCompra)
    {
        this.costoCompra = costoCompra;
    }
    public Float getCostoPublico ()
    {
        return costoPublico;
    }
    public void setCostoPublico(Float costoPublico)
    {
        this.costoPublico = costoPublico;
    }
    public Integer getPuntosAsignados ()
    {
        return puntosAsignados;
    }
}
```

```

public void setPuntosAsignados(Integer puntosAsignados)
{
    this.puntosAsignados = puntosAsignados;
}
public Integer getPuntosCompra()
{
    return puntosCompra;
}
public void setPuntosCompra(Integer puntosCompra)
{
    this.puntosCompra = puntosCompra;
}
public Integer getCantidadIngreso()
{
    return cantidadIngreso;
}
public void setCantidadIngreso(Integer cantidadIngreso)
{
    this.cantidadIngreso = cantidadIngreso;
}
public String getClasificacion()
{
    return clasificacion;
}
public void setClasificacion(String clasificacion)
{
    this.clasificacion = clasificacion;
}
public Float getIva()
{
    return iva;
}
public void setIva(Float iva)
{
    this.iva = iva;
}
public String getProveedor()
{
    return proveedor;
}
public void setProveedor(String proveedor)
{
    this.proveedor = proveedor;
}

@Override
public String toString()
{
    String mensaje = "";

    mensaje = codigoBarras + "\n" + fechaRegistro + "\n" + nombreProducto + "\n" + costoCompra + "\n" +
costoPublico + "\n" +
        puntosAsignados + "\n" + cantidadIngreso + "\n" + clasificacion + "\n" + iva + "\n" +
proveedor;

    return mensaje;
}
}

```

6.- ProductoVenta.java

```
package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto ProductoVenta con sus respectivos métodos
Getters y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras
Clases */

public class ProductoVenta
{
    private Integer idProductoVenta;
    private Integer cantidad;
    private Integer puntos;
    private Float importe;
    private String productoCodigoBarras;
    private Integer ventaNumeroTransaccion;

    /***** MÉTODOS GETTER'S Y SETTER'S *****/

    public Integer getIdProductoVenta()
    {
        return idProductoVenta;
    }
    public void setIdProductoVenta(Integer idProductoVenta)
    {
        this.idProductoVenta = idProductoVenta;
    }
    public Integer getCantidad()
    {
        return cantidad;
    }
    public void setCantidad(Integer cantidad)
    {
        this.cantidad = cantidad;
    }
    public Integer getPuntos()
    {
        return puntos;
    }
    public void setPuntos(Integer puntos)
    {
        this.puntos = puntos;
    }
    public Float getImporte()
    {
        return importe;
    }
    public void setImporte(Float importe)
    {
        this.importe = importe;
    }

    public String getProductoCodigoBarras()
    {
        return productoCodigoBarras;
    }
    public void setProductoCodigoBarras(String productoCodigoBarras)
    {
        this.productoCodigoBarras = productoCodigoBarras;
    }

    public Integer getVentaNumeroTransaccion()
    {
        return ventaNumeroTransaccion;
    }
    public void setVentaNumeroTransaccion(Integer ventaNumeroTransaccion)
    {
        this.ventaNumeroTransaccion = ventaNumeroTransaccion;
    }
}
```

```

@Override
public String toString()
{
    String mensaje = "";

    mensaje = idProductoVenta + "\n" + puntos + "\n" + importe + "\n" + productoCodigoBarras + "\n" +
ventaNumeroTransaccion;

    return mensaje;
}
}

```

7.- Proveedor.java

```

package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Proveedor con sus respectivos métodos
Getters y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras
Clases */

import java.sql.Date;

public class Proveedor
{
    private String rfcProveedor;
    private Date fechaRegistro;
    private String nombreProveedor;
    private String direccion;
    private String telefono;
    private String correo;
    private String nombreContacto;

    /***** MÉTODOS GETTER'S Y SETTER'S *****/

    public String getRfcProveedor()
    {
        return rfcProveedor;
    }
    public void setRfcProveedor(String rfcProveedor)
    {
        this.rfcProveedor = rfcProveedor;
    }
    public Date getFechaRegistro()
    {
        return fechaRegistro;
    }
    public void setFechaRegistro(Date fechaRegistro)
    {
        this.fechaRegistro = fechaRegistro;
    }

    public String getNombreProveedor()
    {
        return nombreProveedor;
    }
    public void setNombreProveedor(String nombreProveedor)
    {
        this.nombreProveedor = nombreProveedor;
    }
    public String getDireccion()
    {
        return direccion;
    }
    public void setDireccion(String direccion)
    {
        this.direccion = direccion;
    }
}

```

```

}
public String getTelefono()
{
    return telefono;
}
public void setTelefono(String telefono)
{
    this.telefono = telefono;
}
public String getCorreo()
{
    return correo;
}
public void setCorreo(String correo)
{
    this.correo = correo;
}
public String getNombreContacto()
{
    return nombreContacto;
}
public void setNombreContacto(String nombreContacto)
{
    this.nombreContacto = nombreContacto;
}

@Override
public String toString()
{
    String mensaje = "";

    mensaje = rfcProveedor + "\n" + fechaRegistro + "\n" + nombreProveedor + "\n" + direccion + "\n" +
        telefono + "\n" + correo + "\n" + nombreContacto;

    return mensaje;
}
}

```

8.- Venta.java

```

package hibernate.beans;

/* Esta Clase permite definir los atributos que conforman al objeto Venta con sus respectivos métodos Getters
y Setters para que dichos atributos puedan ser manipulados en otros métodos contenidos en otras Clases */

import java.sql.Date;

public class Venta
{
    private Integer numeroTransaccion;
    private Date fechaVenta;
    private String tipoPago;
    private Float subTotal;
    private Float iva;
    private Float total;
    private Integer puntosAcumulados;
    private Integer puntosPagados;
    private String clienteRFC;
    private String empleadoRFC;

    /***** MÉTODOS GETTER'S Y SETTER'S *****/

    public Integer getNumeroTransaccion()
    {
        return numeroTransaccion;
    }

    public void setNumeroTransaccion(Integer numeroTransaccion)

```



```

{
    this.numeroTransaccion = numeroTransaccion;
}
public Date getFechaVenta()
{
    return fechaVenta;
}
public void setFechaVenta(Date fechaVenta)
{
    this.fechaVenta = fechaVenta;
}
public String getTipoPago()
{
    return tipoPago;
}
public void setTipoPago(String tipoPago)
{
    this.tipoPago = tipoPago;
}
public Float getSubTotal()
{
    return subTotal;
}
public void setSubTotal(Float subTotal)
{
    this.subTotal = subTotal;
}
public Float getIva()
{
    return iva;
}
public void setIva(Float iva)
{
    this.iva = iva;
}
public Float getTotal()
{
    return total;
}
public void setTotal(Float total)
{
    this.total = total;
}
public Integer getPuntosAcumulados()
{
    return puntosAcumulados;
}
public void setPuntosAcumulados(Integer puntosAcumulados)
{
    this.puntosAcumulados = puntosAcumulados;
}
public String getClienteRFC()
{
    return clienteRFC;
}
public void setClienteRFC(String clienteRFC)
{
    this.clienteRFC = clienteRFC;
}
public String getEmpleadoRFC()
{
    return empleadoRFC;
}
public void setEmpleadoRFC(String empleadoRFC)
{
    this.empleadoRFC = empleadoRFC;
}
public Integer getPuntosPagados()

```

```

    {
        return puntosPagados;
    }
    public void setPuntosPagados(Integer puntosPagados)
    {
        this.puntosPagados = puntosPagados;
    }

    @Override
    public String toString()
    {
        String mensaje = "";

        mensaje = numeroTransaccion + "\n" + fechaVenta + "\n" + tipoPago + "\n" + subTotal + "\n" + iva +
"\n" +
                total + "\n" + puntosAcumulados + "\n" + clienteRFC + "\n" + empleadoRFC;

        return mensaje;
    }
}

```

ANEXO B. Código de clase que permite la conexión con la Base de Datos contenida en el paquete **hibernate.conexión** de la aplicación de escritorio y de la aplicación web:

1.- Conexión.java

```

package hibernate.conexion;

/** Esta Clase permite crear el objeto "SessionFactory" el cual ayuda a gestionar las sesiones de conexión a
la Base de Datos e inicializa Hibernate */

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Conexion
{
    private static final SessionFactory sessionFactory = buildSessionFactory();
    private static SessionFactory buildSessionFactory()
    {
        try
        {
            return new Configuration().configure().buildSessionFactory();
        }
        catch(Throwable ex)
        {
            System.err.println("ERROR AL CREAR sessionFactory " + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory()
    {
        return sessionFactory;
    }

    public static void closeSessionFactory()
    {
        sessionFactory.close();
    }
}

```

ANEXO C. Código de los archivos **hbm.xml** contenidos en el paquete **hibernate.mapeos** de la aplicación de escritorio y de la aplicación web:

1.- Cliente.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Cliente" con las columnas de la tabla
"Cliente" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Cliente" table="Cliente">
    <id name="rfcCliente" column="pk_rfc_cliente" type="string"/>
    <property name="fechaRegistro" column="fecha_registro" type="date"/>
    <property name="nombreCliente" column="nombre" type="string"/>
    <property name="direccion" column="direccion" type="string"/>
    <property name="telefono" column="telefono" type="string"/>
    <property name="puntosAcumulados" column="puntos" type="int"/>
    <property name="correo" column="correo" type="string"/>
    <property name="usuario" column="usuario" type="string"/>
    <property name="contrasenia" column="contraseña" type="string"/>
  </class>
</hibernate-mapping>
```

2.- Devolución.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Devolucion" con las columnas de la tabla
"Devolucion" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Devolucion" table="Devolucion">
    <id name="numeroDevolucion" column="pk_numero_devolucion" type="int"> <generator class="native"/>
  </id>
  <property name="fechaDevolucion" column="fecha_devolucion" type="date"/>
  <property name="clasificacion" column="clasificacion" type="string"/>
  <property name="motivo" column="motivo" type="string"/>
  <property name="cantidadProductos" column="cantidad_productos" type="int"/>
  <property name="costoDevolucion" column="costo_devolucion" type="float"/>
  <property name="statusDevolucion" column="status_devolucion" type="string"/>
  <property name="pedidoNumeroPedido" column="fk_pedido_numero_pedido" type="int"/>
  </class>
</hibernate-mapping>
```

3.- Empleado.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Empleado" con las columnas de la tabla
"Empleado" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Empleado" table="Empleado">
    <id name="rfcEmpleado" column="pk_rfc_empleado" type="string"/>
    <property name="fechaRegistro" column="fecha_registro" type="date"/>
    <property name="nombreEmpleado" column="nombre" type="string"/>
    <property name="direccion" column="direccion" type="string"/>
    <property name="telefono" column="telefono" type="string"/>
    <property name="correo" column="correo" type="string"/>
    <property name="usuario" column="usuario" type="string"/>
    <property name="contrasenia" column="contraseña" type="string"/>
  </class>
</hibernate-mapping>
```

4.- Pedido.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Pedido" con las columnas de la tabla
"Pedido" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Pedido" table="Pedido">
    <id name="numeroPedido" column="pk_numero_pedido" type="int"> <generator class="native"/> </id>
    <property name="fechaPedido" column="fecha_pedido" type="date"/>
    <property name="cantidadSolicitada" column="cantidad_solicitada" type="int"/>
    <property name="totalLiquidar" column="total_liquidar" type="float"/>
    <property name="statusPedido" column="status_pedido" type="string"/>
    <property name="productoCodigoBarras" column="fk_producto_codigo_barras" type="string"/>
  </class>
</hibernate-mapping>
```

5.- Producto.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Producto" con las columnas de la tabla
"Producto" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Producto" table="Producto">
    <id name="codigoBarras" column="pk_codigo_barras"/>
    <property name="fechaRegistro" column="fecha_registro" type="date"/>
    <property name="nombreProducto" column="nombre" type="string"/>
    <property name="costoCompra" column="costo_compra" type="float"/>
    <property name="costoPublico" column="costo_publico" type="float"/>
    <property name="puntosAsignados" column="puntos_asignados" type="int"/>
    <property name="puntosCompra" column="puntos_compra" type="int"/>
    <property name="cantidadIngreso" column="cantidad_ingreso" type="int"/>
    <property name="clasificacion" column="clasificacion" type="string"/>
    <property name="iva" column="iva" type="float"/>
    <property name="proveedor" column="fk_proveedor_rfc" type="string"/>
  </class>
</hibernate-mapping>
```

6.- ProductoVenta.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "ProductoVenta" con las columnas de la tabla
"Producto Venta" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.ProductoVenta" table="Producto_Venta">
    <id name="idProductoVenta" column="pk_id_producto_venta" type="int"> <generator class="native"/>
  </id>
  <property name="cantidad" column="cantidad" type="int"/>
  <property name="puntos" column="puntos" type="int"/>
  <property name="importe" column="importe" type="float"/>
  <property name="productoCodigoBarras" column="fk_producto_codigo_barras" type="string"/>
  <property name="ventaNumeroTransaccion" column="fk_venta_numero_transaccion" type="int"/>
  </class>
</hibernate-mapping>
```

7.- Proveedor.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Proveedor" con las columnas de la tabla
"Proveedor" de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Proveedor" table="Proveedor">
    <id name="rfcProveedor" column="pk_rfc_proveedor" type="string" />
    <property name="fechaRegistro" column="fecha_registro" type="date"/>
    <property name="nombreProveedor" column="nombre" type="string"/>
    <property name="direccion" column="direccion" type="string"/>
    <property name="telefono" column="telefono" type="string"/>
    <property name="correo" column="correo" type="string"/>
    <property name="nombreContacto" column="nombre_contacto" type="string"/>
  </class>
</hibernate-mapping>
```

8.- Venta.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite el mapeo de los atributos del objeto "Venta" con las columnas de la tabla "Venta"
de la Base de Datos Relacional implementada para esta Aplicación -->

<hibernate-mapping>
  <class name="hibernate.beans.Venta" table="Venta">
    <id name="numeroTransaccion" column="pk_numero_transaccion" type="int"> <generator
class="native"/> </id>
    <property name="fechaVenta" column="fecha_venta" type="date"/>
    <property name="tipoPago" column="tipo_pago" type="string"/>
    <property name="subTotal" column="sub_total" type="float"/>
    <property name="iva" column="iva" type="float"/>
    <property name="total" column="total" type="float"/>
    <property name="puntosAcumulados" column="puntos_acumulados" type="int"/>
    <property name="puntosPagados" column="puntos_pagados" type="int"/>
    <property name="clienteRFC" column="fk_cliente_rfc" type="string"/>
    <property name="empleadoRFC" column="fk_empleado_rfc" type="string"/>
  </class>
</hibernate-mapping>
```

ANEXO D. Código de las clases contenidas en el paquete **hibernate.operaciones** de la aplicación de escritorio y de la aplicación web:

1.- DatosEstaticos.java

```
package hibernate.operaciones;

/** Esta Clase permite obtener el RFC (clave) y Nombre del Empleado, después de autenticarse en el Sistema,
para ser utilizados en algunas interfaces de la Aplicación */

public class DatosEstaticos
{
  public static String claveEmpleado = "";
  public static String nombreEmpleado = "";
}
```

2.- OpCliente.java

```
package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con el Cliente
dentro del Sistema **/

import javax.persistence.PersistenceException;
import javax.swing.JOptionPane;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Cliente;
import hibernate.conexion.Conexion;

public class OpCliente
{
    private final static Logger elLogger = Logger.getLogger(OpProveedor.class);
    private Session sesion = null;
    private boolean resultado = false;

    /***** MÉTODO PARA INSERTAR CLIENTE *****/

    public boolean insertarCliente(Cliente elCliente)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(Throwable ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Insertar " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        try
        {
            elLogger.info("INSERTANDO AL CLIENTE");

            sesion.beginTransaction();
            sesion.save(elCliente);
            sesion.getTransaction().commit();

            resultado = true;
            elLogger.info("SE INSERTO CON EXITO AL CLIENTE");
        }
        catch(PersistenceException excep)
        {
            elLogger.error("ERROR AL INTENTAR INSERTAR AL CLIENTE");
        }
        finally
        {
            sesion.close();
        }

        if(elCliente.getNombreCliente() != null)
        {
            JOptionPane.showMessageDialog(null, "SE INSERTÓ CORRECTAMENTE EL CLIENTE");
        }

        return resultado;
    }

    /***** MÉTODO PARA ENVIAR CLIENTE SEGÚN LLAVE PRIMARIA (RFC) *****/
}
```

```

public Cliente buscarClienteRfc(String rfcCliente)
{
    PropertyConfigurator.configure("logger.properties");

    Cliente elCliente;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Enviar Cliente " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL NOMBRE DEL CLIENTE");

    elCliente = (Cliente) sesion.createQuery("SELECT CLIENTE FROM Cliente AS CLIENTE WHERE
CLIENTE.rfcCliente = '" + rfcCliente + "'").uniqueResult();

    sesion.close();

    return elCliente;
}

/***** MÉTODO PARA ELIMINAR CLIENTE POR RFC *****/

@SuppressWarnings("rawtypes")
public void eliminarCliente(String rfcCliente)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Eliminar Cliente " + ex);
        throw new ExceptionInInitializerError(ex);
    }
    elLogger.info("ELIMINANDO AL CLIENTE " + rfcCliente);

    sesion.beginTransaction();
    String sentencia = "DELETE FROM Cliente AS CLIENTE WHERE CLIENTE.rfcCliente = '" + rfcCliente
+ "'";
    Query elQuery = sesion.createQuery(sentencia);
    elQuery.executeUpdate();
    sesion.getTransaction().commit();

    elLogger.info("SE ELIMINO CORRECTAMENTE AL PROVEEDOR " + rfcCliente);

    JOptionPane.showMessageDialog(null, "SE ELIMINÓ CORRECTAMENTE AL CLIENTE");

    sesion.close();
}

/***** MÉTODO PARA ACTUALIZAR PUNTOS DEL CLIENTE *****/

public void actualizarPuntosCliente(String rfcCliente,Integer puntosDeCompra)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
}

```

```

        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Actualizar Puntos Cliente " +
ex);
            throw new ExceptionInInitializerError(ex);
        }

        Cliente clienteEncontrado = sesion.get(Cliente.class,rfcCliente);

        elLogger.info("ACTUALIZANDO LOS PUNTOS DEL CLIENTE " + rfcCliente);

        sesion.beginTransaction();
        clienteEncontrado.setPuntosAcumulados(puntosDeCompra);
        sesion.update(clienteEncontrado);
        sesion.getTransaction().commit();

        sesion.close();
    }

    /***** MÉTODO PARA ACTUALIZAR CLIENTE *****/

    public void actualizarCliente(Cliente elCliente)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Actualizar Cliente " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        Cliente clienteEncontrado = sesion.get(Cliente.class,elCliente.getRfcCliente());

        elLogger.info("ACTUALIZANDO AL CLIENTE " + elCliente.getRfcCliente());

        sesion.beginTransaction();
        clienteEncontrado.setFechaRegistro((java.sql.Date)elCliente.getFechaRegistro());
        clienteEncontrado.setRfcCliente(elCliente.getRfcCliente());
        clienteEncontrado.setNombreCliente(elCliente.getNombreCliente());
        clienteEncontrado.setDireccion(elCliente.getDireccion());
        clienteEncontrado.setTelefono(elCliente.getTelefono());
        clienteEncontrado.setCorreo(elCliente.getCorreo());
        sesion.update(clienteEncontrado);
        sesion.getTransaction().commit();

        elLogger.info("SE ACTUALIZÓ CORRECTAMENTE EL CLIENTE " + elCliente.getRfcCliente());

        JOptionPane.showMessageDialog(null, "SE ACTUALIZÓ CORRECTAMENTE EL CLIENTE");

        sesion.close();
    }
}

```

3.- OpDevolución.java

```

package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con la Devolución
dentro del Sistema */

import java.util.*;
import javax.persistence.PersistenceException;
import javax.swing.JOptionPane;

```



```

import org.apache.log4j.*;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Devolucion;
import hibernate.conexion.Conexion;

public class OpDevolucion
{
    private final static Logger elLogger = Logger.getLogger(OpDevolucion.class);
    Session sesion = null;

    /***** MÉTODO PARA REGISTRAR UNA DEVOLUCIÓN *****/

    public void registrarDevolucion(Devolucion laDevolucion)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(Throwable ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpDevolucion Insertar 1 " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        try
        {
            elLogger.info("REGISTRANDO LA DEVOLUCIÓN");

            sesion.beginTransaction();
            sesion.save(laDevolucion);
            sesion.getTransaction().commit();

            elLogger.info("SE INSERTO SE REGISTRÓ LA DEVOLUCIÓN");
        }
        catch(PersistenceException excep)
        {
            elLogger.error("ERROR AL INTENTAR REGISTRAR LA DEVOLUCIÓN");
        }
        finally
        {
            sesion.close();
        }

        JOptionPane.showMessageDialog(null, "SE INSERTÓ CORRECTAMENTE LA DEVOLUCIÓN");
    }

    /***** MÉTODO PARA MOSTRAR TODAS LAS DEVOLUCIONES DE UN PROVEEDOR *****/

    @SuppressWarnings({ "unchecked", "rawtypes" })
    public List<Devolucion> traerDevoluciones(String nombreProveedor)
    {
        PropertyConfigurator.configure("logger.properties");

        List<Devolucion> listaDevoluciones;
        String sentencia;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpDevolucion Datos de las Devoluciones " +
ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
}

```

```

    }

    elLogger.info("BUSCANDO LAS DEVOLUCIONES");

    sentencia = "SELECT DEVOLUCION FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor
PROVEEDOR,Devolucion DEVOLUCION WHERE "
        + "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras =
PEDIDO.productoCodigoBarras AND "
        + "DEVOLUCION.pedidoNumeroPedido = PEDIDO.numeroPedido AND PROVEEDOR.nombreProveedor = '"
+ nombreProveedor + "'";

    Query elQuery = sesion.createQuery(sentencia);

    listaDevoluciones = elQuery.list();

    elLogger.info("DEVOLUCIONES ENCONTRADAS");

    sesion.close();

    return listaDevoluciones;
}

/*****MÉTODO PARA MOSTRAR DEVOLUCIÓN SEGÚN NUMERO DE DEVOLUCIÓN*****/

public Devolucion buscarDevolucionPorNumero(Integer numDevolucion)
{
    PropertyConfigurator.configure("logger.properties");

    Devolucion laDevolucion;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpDevolucionr Devolución Llave Primaria " +
ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL NUMERO DE DEVOLUCIÓN");

    laDevolucion = (Devolucion) sesion.createQuery("SELECT DEVOLUCION FROM Devolucion AS DEVOLUCION
WHERE DEVOLUCION.numeroDevolucion = " + numDevolucion).uniqueResult();

    sesion.close();
    return laDevolucion;
}

/***** MÉTODO PARA MOSTRAR LA(S) DEVOLUCIÓN(ES) SEGÚN FILTROS DE CONSULTA *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Devolucion> buscarDevolucionesFiltros(String nombreProveedor,boolean
todasDevoluciones,String numeroDevolucion)
{
    PropertyConfigurator.configure("logger.properties");

    List<Devolucion> listaDevoluciones = null;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpDevolucion Consultar con Filtros " + ex);
    }
}

```

```

        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LAS DEVOLUCIONES");

    if(nombreProveedor.equals("Seleccione un Proveedor"))
    {
        JOptionPane.showMessageDialog(null,"Seleccione un Proveedor.");
    }
    else if(numeroDevolucion.equals("") && todasDevoluciones == false)
    {
        JOptionPane.showMessageDialog(null,"Ingrese un Número de Devolución.");
    }
    else if(todasDevoluciones == false)
    {
        elLogger.info("BUSCANDO LA DEVOLUCIÓN");

        sentencia = "SELECT DEVOLUCION FROM Devolucion DEVOLUCION, Pedido PEDIDO,Producto
PRODUCTO,Proveedor PROVEEDOR WHERE "
            + "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras =
PEDIDO.productoCodigoBarras AND DEVOLUCION.pedidoNumeroPedido = PEDIDO.numeroPedido AND "
            + "PROVEEDOR.nombreProveedor = '" + nombreProveedor + "' AND
DEVOLUCION.numeroDevolucion = '" + Integer.valueOf(numeroDevolucion) + "'";
        Query elQuery = sesion.createQuery(sentencia);
        listaDevoluciones = elQuery.list();

        elLogger.info("LA DEVOLUCIÓN FUE ENCONTRADA");
    }
    else if(todasDevoluciones == true)
    {
        elLogger.info("BUSCANDO LAS DEVOLUCIONES");

        sentencia = "SELECT DEVOLUCION FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor
PROVEEDOR,Devolucion DEVOLUCION WHERE "
            + "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras =
PEDIDO.productoCodigoBarras AND "
            + "DEVOLUCION.pedidoNumeroPedido = PEDIDO.numeroPedido AND PROVEEDOR.nombreProveedor
= '" + nombreProveedor + "'";
        Query elQuery = sesion.createQuery(sentencia);
        listaDevoluciones = elQuery.list();

        elLogger.info("LAS DEVOLUCIONES FUERON ENCONTRADAS");
    }

    sesion.close();

    return listaDevoluciones;
}

/*****MÉTODO PARA ACTUALIZAR DEVOLUCIÓN *****/

public void actualizarDevolucion(Devolucion laDevolucion)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESIÓN EN OpDevolucion Actualizar Devolucion " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    Devolucion devolucionEncontrada=sesion.get(Devolucion.class,laDevolucion.getNumeroDevolucion());

    elLogger.info("ACTUALIZANDO LA DEVOLUCIÓN " + laDevolucion.getNumeroDevolucion());
}

```

```

sesion.beginTransaction();
devolucionEncontrada.setFechaDevolucion((java.sql.Date)laDevolucion.getFechaDevolucion());
devolucionEncontrada.setClasificacion(laDevolucion.getClasificacion());
devolucionEncontrada.setMotivo(laDevolucion.getMotivo());
devolucionEncontrada.setCantidadProductos(laDevolucion.getCantidadProductos());;
devolucionEncontrada.setCostoDevolucion(laDevolucion.getCostoDevolucion());
devolucionEncontrada.setStatusDevolucion(laDevolucion.getStatusDevolucion());
devolucionEncontrada.setPedidoNumeroPedido(laDevolucion.getPedidoNumeroPedido());
sesion.update(devolucionEncontrada);

elLogger.info("SE ACTUALIZÓ CORRECTAMENTE LA DEVOLUCIÓN " + laDevolucion.getNumeroDevolucion());
sesion.getTransaction().commit();

JOptionPane.showMessageDialog(null, "SE ACTUALIZÓ CORRECTAMENTE LA DEVOLUCIÓN");

sesion.close();
}

/*****MÉTODO PARA ELIMINAR DEVOLUCION SEGÚN NÚMERO DE DEVOLUCIÓN *****/

public void eliminarDevolucion(Integer numeroDevolucion)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESIÓN EN OpDevolucion Eliminar Devolucion " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    try
    {
        elLogger.info("SE ELIMINARÁ LA DEVOLUCIÓN " + numeroDevolucion);
        sesion.beginTransaction();
        Devolucion devolucionBorrar = sesion.get(Devolucion.class, numeroDevolucion);
        sesion.delete(devolucionBorrar);
        sesion.getTransaction().commit();
        elLogger.info("SE HA ELIMINADO LA DEVOLUCIÓN " + numeroDevolucion);
    }
    catch (IllegalArgumentException arg)
    {
        elLogger.error("NO SE PUEDE ELIMINAR LA DEVOLUCIÓN");
    }
    finally
    {
        JOptionPane.showMessageDialog(null, "SE ELIMINÓ CORRECTAMENTE LA DEVOLUCIÓN");

        sesion.close();
    }
}
}

```

4.- OpEmpleado.java

```

package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con el Empleado
dentro del Sistema **/

import java.util.*;
import javax.persistence.PersistenceException;

```

```

import javax.swing.JOptionPane;
import org.apache.log4j.*;
import org.hibernate.Session;
import hibernate.beans.Empleado;
import hibernate.conexion.Conexion;

public class OpEmpleado
{
    private final static Logger elLogger = Logger.getLogger(OpEmpleado.class);
    Session sesion = null;
    boolean resultado = false;

    /*****MÉTODO PARA INSERTAR EMPLEADO *****/

    public boolean insertarEmpleado(Empleado elEmpleado)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(Throwable ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado insertar " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        try
        {
            elLogger.info("INSERTANDO EL EMPLEADO");

            sesion.beginTransaction();
            sesion.save(elEmpleado);
            sesion.getTransaction().commit();

            resultado = true;
            elLogger.info("SE INSERTO CON EXITO EL EMPLEADO");
        }
        catch(PersistenceException excep)
        {
            elLogger.error("ERROR AL INTENTAR INSERTAR EL EMPLEADO");
        }
        finally
        {
            sesion.close();
        }
        return resultado;
    }

    /***** MÉTODO PARA INSERTAR LISTA EMPLEADOS *****/

    public boolean insertarListaEmpleados(LinkedList<Empleado> listaEmpleados)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(Throwable ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado insertar lista " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        try
        {

```

```

        elLogger.info("INSERTANDO LA LISTA DE EMPLEADOS");

        sesion.beginTransaction();
        for(int i=0;i<listaEmpleados.size();i++)
        {
            sesion.save(listaEmpleados.get(i));
        }
        sesion.getTransaction().commit();

        resultado = true;
        elLogger.info("SE INSERTO CON EXITO LA LISTA DE EMPLEADOS");
    }
    catch(PersistenceException excep)
    {
        elLogger.error("ERROR AL INTENTAR INSERTAR LA LISTA DE EMPLEADOS");
    }
    finally
    {
        sesion.close();
    }
    return resultado;
}

/***** MÉTODO PARA ACTUALIZAR EMPLEADO SEGÚN RFC *****/

public boolean actualizarEmpleado(String numeroEmpleado)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado actualizar " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    try
    {
        elLogger.info("ACTUALIZANDO EL EMPLEADO " + numeroEmpleado);

        sesion.beginTransaction();
        Empleado empleadoEncontrado = sesion.get(Empleado.class,numeroEmpleado);
        sesion.update(empleadoEncontrado);
        sesion.getTransaction().commit();

        resultado = true;
        elLogger.info("SE ACTUALIZO CON EXITO EL EMPLEADO " + numeroEmpleado);
    }
    catch(NullPointerException e)
    {
        elLogger.error("ERROR AL INTENTAR ACTUALIZAR EL EMPLEADO " + numeroEmpleado);
    }
    finally
    {
        sesion.close();
    }
    return resultado;
}

/***** MÉTODO PARA ELIMINAR EMPLEADO SEGÚN RFC *****/

public boolean eliminarEmpleado(String numeroEmpleado)
{

```

```

PropertyConfigurator.configure("logger.properties");

try
{
    sesion = Conexion.getSessionFactory().openSession();
}
catch(Throwable ex)
{
    System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado eliminar " + ex);
    throw new ExceptionInInitializerError(ex);
}

try
{
    elLogger.info("ELIMINANDO EL EMPLEADO " + numeroEmpleado);

    sesion.beginTransaction();
    Empleado empleadoEncontrado = sesion.get(Empleado.class,numeroEmpleado);
    sesion.delete(empleadoEncontrado);
    System.out.println("EMPLEADO ELIMINADO.");
    sesion.getTransaction().commit();

    resultado = true;
    elLogger.info("SE ELIMINO CON EXITO EL EMPLEADO " + numeroEmpleado);
}
catch(IllegalArgumentException arg)
{
    elLogger.error("ERROR AL INTENTAR ELIMINAR EL EMPLEADO " + numeroEmpleado);
}
finally
{
    sesion.close();
}

return resultado;
}

/***** MÉTODO PARA MOSTRAR EMPLEADO SEGÚN RFC *****/

public Empleado consultarEmpleado(String numeroEmpleado)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado consultar " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO AL EMPLEADO " + numeroEmpleado);

    Empleado empleadoConsultado = sesion.get(Empleado.class,numeroEmpleado);

    elLogger.info("EMPLEADO SELECCIONANDO" + numeroEmpleado);

    return empleadoConsultado;
}

/***** MÉTODO PARA MOSTRAR EMPLEADO SEGÚN USUARIO Y CLAVE *****/

public Empleado buscarEmpleado(String usuario,String clave)
{
    PropertyConfigurator.configure("logger.properties");

```

```

Empleado elEmpleado = null;

try
{
    sesion = Conexion.getSessionFactory().openSession();
}
catch(ExceptionInInitializerError ex)
{
    System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado Enviar Empleado " + ex);
    throw new ExceptionInInitializerError(ex);
}

if(usuario.equals(""))
{
    JOptionPane.showMessageDialog(null,"INGRESE NOMBRE DE USUARIO");
}
else if(clave.equals(""))
{
    JOptionPane.showMessageDialog(null,"INGRESE PASSWORD");
}
else
{
    elLogger.info("BUSCANDO EL NOMBRE DEL EMPLEADO");
    elEmpleado = (Empleado) sesion.createQuery("SELECT EMPLEADO FROM Empleado AS EMPLEADO WHERE EMPLEADO.usuario='"+usuario+"' AND EMPLEADO.contrasenia='"+clave+"'").uniqueResult();
    elLogger.info("EMPLEADO ENCONTRADO");

    if(elEmpleado != null)
    {
        DatosEstaticos.claveEmpleado = elEmpleado.getRfcEmpleado();
        DatosEstaticos.nombreEmpleado = elEmpleado.getNombreEmpleado();
    }
    else
    {
        JOptionPane.showMessageDialog(null,"EMPLEADO NO EXISTE EN EL SISTEMA");
    }
}
sesion.close();
return elEmpleado;
}
}

```

5.- OpPedido.java

```

package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con el Pedido dentro del Sistema **/

import java.util.*;
import javax.persistence.PersistenceException;
import javax.swing.*;
import org.apache.log4j.*;
import org.hibernate.*;
import org.hibernate.query.Query;
import hibernate.beans.Pedido;
import hibernate.beans.Producto;
import hibernate.conexion.Conexion;

public class OpPedido
{
    private final static Logger elLogger = Logger.getLogger(OpPedido.class);
    private Session sesion = null;
    private boolean resultado = false;

    /******* MÉTODO PARA INSERTAR PEDIDO *****/

```



```

public boolean registrarPedido(Pedido elPedido)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(Throwable ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Insertar Un Pedido " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    try
    {
        elLogger.info("REGISTRANDO EL PEDIDO");

        sesion.beginTransaction();
        sesion.save(elPedido);
        sesion.getTransaction().commit();

        resultado = true;
        elLogger.info("SE REGISTRÓ CON EXITO EL PEDIDO");
    }
    catch(PersistenceException excep)
    {
        elLogger.error("ERROR AL INTENTAR REGISTRAR EL PEDIDO");
    }
    finally
    {
        sesion.close();
    }

    JOptionPane.showMessageDialog(null,"SE REGISTRÓ CORRECTAMENTE EL PEDIDO");
    return resultado;
}

/***** MÉTODO PARA CONSULTAR PEDIDO(S) SEGÚN FILTROS *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Pedido> buscarPedidosFiltros(String nombreProveedor,boolean todosPedidos,String
numeroPedido)
{
    PropertyConfigurator.configure("logger.properties");

    List<Pedido> listaPedidos = null;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Consultar con Filtros " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PEDIDOS");

    if(nombreProveedor.equals("Seleccione un Proveedor"))
    {
        JOptionPane.showMessageDialog(null,"Seleccione un Proveedor.");
    }
    else if(numeroPedido.equals("") && todosPedidos == false)
    {
        JOptionPane.showMessageDialog(null,"Ingrese un Número de Pedido.");
    }
}

```

```

    }
    else if(todosPedidos == false)
    {
        elLogger.info("BUSCANDO EL PEDIDO");

        sentencia = "SELECT PEDIDO FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor PROVEEDOR WHERE "
            + "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras = "
PEDIDO.productoCodigoBarras AND "
            + "PROVEEDOR.nombreProveedor = '" + nombreProveedor + "' AND PEDIDO.numeroPedido = "
'" + numeroPedido + "'";
        Query elQuery = sesion.createQuery(sentencia);
        listaPedidos = elQuery.list();

        elLogger.info("EL PEDIDO FUE ENCONTRADO");
    }
    else if(todosPedidos == true)
    {
        elLogger.info("BUSCANDO EL PEDIDO");

        sentencia = "SELECT PEDIDO FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor PROVEEDOR WHERE "
            + "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras = "
PEDIDO.productoCodigoBarras AND "
            + "PROVEEDOR.nombreProveedor = '" + nombreProveedor + "'";
        Query elQuery = sesion.createQuery(sentencia);
        listaPedidos = elQuery.list();

        elLogger.info("LOS PEDIDOS FUERON ENCONTRADOS");
    }

    sesion.close();

    return listaPedidos;
}

/***** MÉTODO PARA MOSTRAR TODOS LOS PEDIDOS DE UN PROVEEDOR *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Pedido> traerPedidos(String rfcProveedor)
{
    PropertyConfigurator.configure("logger.properties");

    List<Pedido> listaPedidos;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Datos de Pedidos " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS PEDIDOS");

    sentencia = "SELECT PEDIDO FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor PROVEEDOR WHERE "
        + "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras = "
PEDIDO.productoCodigoBarras AND "
        + "PROVEEDOR.rfcProveedor = :rfcProveedor";
    Query elQuery = sesion.createQuery(sentencia);
    elQuery.setParameter("rfcProveedor", rfcProveedor);

    listaPedidos = elQuery.list();

    sesion.close();

    return listaPedidos;
}

```

```

}

/***** MÉTODO PARA MOSTRAR PEDIDOS PENDIENTES *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Pedido> buscarPedidosFiltrosPendientes (String nombreProveedor,boolean
todosPedidos,String numeroPedido)
{
    PropertyConfigurator.configure("logger.properties");

    List<Pedido> listaPedidos = null;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Consultar con Filtros " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PEDIDOS");

    if(nombreProveedor.equals("Seleccione un Proveedor"))
    {
        JOptionPane.showMessageDialog(null,"Seleccione un Proveedor.");
    }
    else if(numeroPedido.equals("") && todosPedidos == false)
    {
        JOptionPane.showMessageDialog(null,"Ingrese un Número de Pedido.");
    }
    else if(todosPedidos == false)
    {
        elLogger.info("BUSCANDO EL PEDIDO");

        sentencia = "SELECT PEDIDO FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor PROVEEDOR WHERE "
+ "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras =
PEDIDO.productoCodigoBarras AND "
+ "PROVEEDOR.nombreProveedor = '" + nombreProveedor + "' AND PEDIDO.numeroPedido =
'" + numeroPedido
+ "' AND PEDIDO.statusPedido = 'Pendiente'";
        Query elQuery = sesion.createQuery(sentencia);
        listaPedidos = elQuery.list();

        elLogger.info("EL PEDIDO FUE ENCONTRADO");
    }
    else if(todosPedidos == true)
    {
        elLogger.info("BUSCANDO EL PEDIDO");

        sentencia = "SELECT PEDIDO FROM Pedido PEDIDO,Producto PRODUCTO,Proveedor PROVEEDOR WHERE "
+ "PROVEEDOR.rfcProveedor = PRODUCTO.proveedor AND PRODUCTO.codigoBarras =
PEDIDO.productoCodigoBarras AND "
+ "PROVEEDOR.nombreProveedor = '" + nombreProveedor + "' AND PEDIDO.statusPedido =
'Pendiente' " ;
        Query elQuery = sesion.createQuery(sentencia);
        listaPedidos = elQuery.list();

        elLogger.info("LOS PEDIDOS FUERON ENCONTRADOS");
    }
    sesion.close();
    return listaPedidos;
}

/***** MÉTODO PARA MOSTRAR PEDIDO SEGÚN SU NÚMERO *****/

```

```

public Pedido buscarPedidoPorNumero(Integer numPedido)
{
    PropertyConfigurator.configure("logger.properties");

    Pedido elPedido;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Pedido por Llave Primaria " +
ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL NUMERO DE PEDIDO");

    elPedido = (Pedido) sesion.createQuery("SELECT PEDIDO FROM Pedido AS PEDIDO WHERE
PEDIDO.numeroPedido = '" + numPedido + "'").uniqueResult();

    sesion.close();

    return elPedido;
}

/***** MÉTODO PARA MOSTRAR EL PRODUCTO DE UN PEDIDO *****/

public Producto traerProductoPedido(String codigoBarras)
{
    PropertyConfigurator.configure("logger.properties");

    Producto elProducto;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Datos de para obtener Proveedor "
+ ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO PRODUCTO");

    elProducto = (Producto) sesion.createQuery("SELECT PRODUCTO FROM Pedido PEDIDO,Producto PRODUCTO
WHERE PRODUCTO.codigoBarras = PEDIDO.productoCodigoBarras AND "
+ "PRODUCTO.codigoBarras = '" + codigoBarras + "'").uniqueResult();

    sesion.close();
    return elProducto;
}

/***** MÉTODO PARA ACTUALIZAR PEDIDO *****/

public void actualizarPedido(Pedido elPedido)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Actualizar Pedido " + ex);

```

```

        throw new ExceptionInInitializerError(ex);
    }

    Pedido pedidoEncontrado = sesion.get(Pedido.class, elPedido.getNumeroPedido());

    elLogger.info("ACTUALIZANDO EL PEDIDO " + elPedido.getNumeroPedido());

    sesion.beginTransaction();
    pedidoEncontrado.setFechaPedido((java.sql.Date)elPedido.getFechaPedido());
    pedidoEncontrado.setProductoCodigoBarras(elPedido.getProductoCodigoBarras());
    pedidoEncontrado.setCantidadSolicitada(elPedido.getCantidadSolicitada());
    pedidoEncontrado.setTotalLiquidar(elPedido.getTotalLiquidar());
    pedidoEncontrado.setStatusPedido(elPedido.getStatusPedido());
    sesion.update(pedidoEncontrado);
    sesion.getTransaction().commit();

    elLogger.info("SE ACTUALIZÓ CORRECTAMENTE EL PEDIDO " + elPedido.getNumeroPedido());

    JOptionPane.showMessageDialog(null, "SE ACTUALIZÓ CORRECTAMENTE EL PEDIDO");

    sesion.close();
}

/***** MÉTODO PARA ELIMINAR PEDIDO SEGÚN NÚMERO DE PEDIDO *****/

public void eliminarPedido(Integer numeroPedido)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpPedido Eliminar Pedido " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    try
    {
        elLogger.info("SE ELIMINARÁ EL PEDIDO " + numeroPedido);
        sesion.beginTransaction();
        Pedido pedidoBorrar = sesion.get(Pedido.class, numeroPedido);
        sesion.delete(pedidoBorrar);
        sesion.getTransaction().commit();
        elLogger.info("SE HA ELIMINADO EL PEDIDO " + numeroPedido);
    }
    catch (IllegalArgumentException arg)
    {
        elLogger.error("NO SE PUEDE ELIMINAR EL PEDIDO");
    }
    finally
    {
        sesion.close();
    }
}
}

```

6.- OpProducto.java

```
package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con el
Producto dentro del Sistema **/

import java.util.*;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Producto;
import hibernate.conexion.Conexion;
import org.apache.log4j.*;
import javax.persistence.PersistenceException;
import javax.swing.JOptionPane;

public class OpProducto
{
    private final static Logger elLogger = Logger.getLogger(OpProducto.class);
    private Session sesion = null;
    private boolean resultado = false;

    /***** MÉTODO PARA INSERTAR PRODUCTO *****/

    public boolean insertarProducto(Producto elProducto)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(Throwable ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN Producto " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        try
        {
            elLogger.info("INSERTANDO EL PRODUCTO");

            sesion.beginTransaction();
            sesion.save(elProducto);
            sesion.getTransaction().commit();

            resultado = true;
            elLogger.info("SE INSERTO CON EXITO EL PRODUCTO");
        }
        catch(PersistenceException excep)
        {
            elLogger.error("ERROR AL INTENTAR INSERTAR EL PRODUCTO");
        }
        finally
        {
            sesion.close();
        }

        JOptionPane.showMessageDialog(null, "SE INSERTÓ CORRECTAMENTE EL PRODUCTO");
        return resultado;
    }

    /***** MÉTODO PARA MOSTRAR LOS NOMBRES DE TODOS LOS PRODUCTOS *****/

    @SuppressWarnings({ "unchecked", "rawtypes" })
    public List<String> traerNombresProductos()
    {
        PropertyConfigurator.configure("logger.properties");
    }
}
```

```

List<String> listaProductos;
String sentencia;

try
{
    sesion = Conexion.getSessionFactory().openSession();
}
catch(ExceptionInInitializerError ex)
{
    System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Nombres Productos " + ex);
    throw new ExceptionInInitializerError(ex);
}

elLogger.info("SELECCIONANDO LOS NOMBRES DE LOS PRODUCTOS");

sentencia = "SELECT PRODUCTO.nombreProducto FROM Producto AS PRODUCTO ORDER BY
PRODUCTO.nombreProducto";
Query elQuery = sesion.createQuery(sentencia);
listaProductos = elQuery.list();

elLogger.info("SE ENCONTRARON LOS NOMBRES DE LOS PRODUCTOS");

sesion.close();

return listaProductos;
}

/***** MÉTODO PARA MOSTRAR PRODUCTO SEGÚN SU NOMBRE *****/

public Producto buscarProducto(String nombreProducto)
{
    PropertyConfigurator.configure("logger.properties");

    Producto elProducto;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Enviar Producto segun nombre
" + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL PRODUCTO");

    elProducto = (Producto) sesion.createQuery("SELECT PROD FROM Producto AS PROD WHERE
PROD.nombreProducto = '" + nombreProducto + "'").uniqueResult();

    elLogger.info("PRODUCTO ENCONTRADO");

    sesion.close();

    return elProducto;
}

/***** MÉTODO PARA ELIMINAR PRODUCTO SEGÚN SU NOMBRE *****/

@SuppressWarnings("rawtypes")
public void eliminarProducto(String nombreProducto)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
}

```

```

    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Eliminar Producto " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("ELIMINANDO AL PROVEEDOR " + nombreProducto);

    String sentencia = "DELETE FROM Producto AS PROD WHERE PROD.nombreProducto = '" +
nombreProducto + "'";

    sesion.beginTransaction();
    Query elQuery = sesion.createQuery(sentencia);
    elQuery.executeUpdate();
    sesion.getTransaction().commit();

    elLogger.info("SE ELIMINO CORRECTAMENTE AL PROVEEDOR " + nombreProducto);

    JOptionPane.showMessageDialog(null, "SE ELIMINÓ CORRECTAMENTE EL PRODUCTO");

    sesion.close();
}

/MÉTODO PARA ACTUALIZAR CANTIDAD DE PRODUCTOS SEGÚN CANTIDAD ENTREGADA POR PEDIDO/

@SuppressWarnings("rawtypes")
public void actualizarCantidadProductosAlmacenPedido(String codigo,Integer cantidadEntregada)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Actualizar Cantidad en
Almacen Pedido" + ex);
        throw new ExceptionInInitializerError(ex);
    }

    Producto productoEncontrado = sesion.get(Producto.class,codigo);
    productoEncontrado.setCantidadIngreso(productoEncontrado.getCantidadIngreso() +
cantidadEntregada);

    elLogger.info("ACTUALIZANDO EL PRODUCTO " + codigo);

    String sentencia = "UPDATE Producto SET cantidadIngreso='" +
productoEncontrado.getCantidadIngreso() + "' WHERE codigoBarras='" + codigo + "'";

    sesion.beginTransaction();
    Query elQuery = sesion.createQuery(sentencia);
    elQuery.executeUpdate();
    sesion.getTransaction().commit();

    elLogger.info("SE ACTUALIZO EL PRODUCTO " + codigo);

    sesion.close();
}

/**** MÉTODO PARA ACTUALIZAR CANTIDAD DE PRODUCTOS SEGÚN PRODUCTOS DEVUELTOS ****/

public void actualizarCantidadProductosAlmacenDevolucion(String codigo,Integer cantidadDevolta)
{
    PropertyConfigurator.configure("logger.properties");

```



```

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Actualizar Cantidad en
Almacen Devuelto" + ex);
        throw new ExceptionInInitializerError(ex);
    }

    Producto productoEncontrado = sesion.get(Producto.class,codigo);

    elLogger.info("ACTUALIZANDO LA CANTIDAD DE PRODUCTOS SEGÚN LA DEVOLUCION " + codigo);

    sesion.beginTransaction();
    productoEncontrado.setCantidadIngreso(productoEncontrado.getCantidadIngreso() -
cantidadDevuelta);
    sesion.update(productoEncontrado);
    sesion.getTransaction().commit();

    elLogger.info("SE ACTUALIZÓ LA CANTIDAD DE PRODUCTOS SEGÚN LA DEVOLUCION " + codigo);

    sesion.close();
}

/*****MÉTODO PARA ACTUALIZAR PRODUCTO GENERAL *****/

public void actualizarProducto(Producto elProducto)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Actualizar Producto " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    Producto productoEncontrado = sesion.get(Producto.class,elProducto.getCodigoBarras());

    elLogger.info("ACTUALIZANDO EL PRODUCTO " + elProducto.getCodigoBarras());

    sesion.beginTransaction();

    productoEncontrado.setFechaRegistro((java.sql.Date)elProducto.getFechaRegistro());
    productoEncontrado.setCodigoBarras(elProducto.getCodigoBarras());
    productoEncontrado.setNombreProducto(elProducto.getNombreProducto());
    productoEncontrado.setCostoCompra(elProducto.getCostoCompra());
    productoEncontrado.setCostoPublico(elProducto.getCostoPublico());
    productoEncontrado.setPuntosAsignados(elProducto.getPuntosAsignados());
    productoEncontrado.setPuntosCompra(elProducto.getPuntosCompra());
    productoEncontrado.setProveedor(elProducto.getProveedor());
    productoEncontrado.setCantidadIngreso(elProducto.getCantidadIngreso());
    productoEncontrado.setClasificacion(elProducto.getClasificacion());
    productoEncontrado.setIva(elProducto.getIva());
    sesion.update(productoEncontrado);

    sesion.getTransaction().commit();

    elLogger.info("SE ACTUALIZO EL PRODUCTO " + elProducto.getCodigoBarras());

    JOptionPane.showMessageDialog(null, "SE ACTUALIZÓ CORRECTAMENTE EL PRODUCTO");

    sesion.close();
}

```

```

}

/***** MÉTODO PARA MOSTRAR PRODUCTOS SEGÚN FILTROS DE CONSULTA *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Producto> buscarProductosFiltros(String desde,String hasta,String rfcProveedor,String
codigoBarras,String nombreProducto)
{
    PropertyConfigurator.configure("logger.properties");

    List<Producto> listaProductos = null;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Consultar con Filtros " +
ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PRODUCTOS");

    if(codigoBarras.equals("") && nombreProducto.equals("") && rfcProveedor.equals("Seleccione un
Proveedor"))
    {
        sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE PRODUCTO.fechaRegistro
BETWEEN '"+desde+"' AND '"+hasta+"' ORDER BY PRODUCTO.fechaRegistro ASC";
        Query elQuery = sesion.createQuery(sentencia);
        listaProductos = elQuery.list();

        elLogger.info("LOS PRODUCTOS FUERON SELECCIONADOS");
    }
    else if(codigoBarras.equals("") && nombreProducto.equals("") &&
!rfcProveedor.equals("Seleccione un Proveedor"))
    {
        sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE PRODUCTO.fechaRegistro
BETWEEN '"+desde+"' AND '"+hasta+"' AND PRODUCTO.proveedor='"+rfcProveedor+"' ORDER BY
PRODUCTO.fechaRegistro ASC";
        Query elQuery = sesion.createQuery(sentencia);
        listaProductos = elQuery.list();

        elLogger.info("LOS PRODUCTOS FUERON SELECCIONADOS");
    }
    else if(!codigoBarras.equals("") && nombreProducto.equals("") &&
rfcProveedor.equals("Seleccione un Proveedor"))
    {
        sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE PRODUCTO.fechaRegistro
BETWEEN '"+desde+"' AND '"+hasta+"' AND PRODUCTO.codigoBarras='"+codigoBarras+"' ORDER BY
PRODUCTO.fechaRegistro ASC";
        Query elQuery = sesion.createQuery(sentencia);
        listaProductos = elQuery.list();

        elLogger.info("LOS PRODUCTOS FUERON SELECCIONADOS");
    }
    else if(codigoBarras.equals("") && !nombreProducto.equals("") &&
rfcProveedor.equals("Seleccione un Proveedor"))
    {
        sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE PRODUCTO.fechaRegistro
BETWEEN '"+desde+"' AND '"+hasta+"' AND PRODUCTO.nombreProducto='"+nombreProducto+"' ORDER BY
PRODUCTO.fechaRegistro ASC";
        Query elQuery = sesion.createQuery(sentencia);
        listaProductos = elQuery.list();

        elLogger.info("LOS PRODUCTOS FUERON SELECCIONADOS");
    }
}

```

```

    }
    else if(codigoBarras.equals("") && !nombreProducto.equals("") &&
!rfcProveedor.equals("Seleccione un Proveedor"))
    {
        sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE PRODUCTO.fechaRegistro
BETWEEN '"+desde+"' AND '"+hasta+"' AND PRODUCTO.nombreProducto='"+nombreProducto+"' AND
PRODUCTO.proveedor='"+rfcProveedor+"' ORDER BY PRODUCTO.fechaRegistro ASC";
        Query elQuery = sesion.createQuery(sentencia);
        listaProductos = elQuery.list();

        elLogger.info("LOS PRODUCTOS FUERON SELECCIONADOS");
    }
    else if(!codigoBarras.equals("") && nombreProducto.equals("") &&
!rfcProveedor.equals("Seleccione un Proveedor"))
    {
        sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE PRODUCTO.fechaRegistro
BETWEEN '"+desde+"' AND '"+hasta+"' AND PRODUCTO.codigoBarras='"+codigoBarras+"' AND
PRODUCTO.proveedor='"+rfcProveedor+"' ORDER BY PRODUCTO.fechaRegistro ASC";
        Query elQuery = sesion.createQuery(sentencia);
        listaProductos = elQuery.list();

        elLogger.info("LOS PRODUCTOS FUERON SELECCIONADOS");
    }

    sesion.close();

    return listaProductos;
}
/***** MÉTODO PARA MOSTRAR PRODUCTO SEGÚN CÓDIGO DE BARRAS *****/
public Producto buscarProductoPorID(String codigoBarras)
{
    PropertyConfigurator.configure("logger.properties");

    Producto elProducto;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Enviar Producto " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL PRODUCTO");
    elProducto = (Producto) sesion.createQuery("SELECT PROD FROM Producto AS PROD WHERE
PROD.codigoBarras = '" + codigoBarras + "'").uniqueResult();
    elLogger.info("SE ENCONTRO EL PRODUCTO");
    sesion.close();
    return elProducto;
}
}

```

7.- OpProductoVenta.java

```

package hibernate.operaciones;

/** Esta Clase contiene el método que permite insertar una Linea de Venta en la cual se relaciona la
Tabla Producto y la Tabla Venta */

import javax.persistence.PersistenceException;
import org.apache.log4j.*;
import org.hibernate.Session;
import hibernate.beans.ProductoVenta;

```

```

import hibernate.conexion.Conexion;

public class OpProductoVenta
{
    private final static Logger elLogger = Logger.getLogger(OpProductoVenta.class);
    Session sesion = null;
    boolean resultado = false;

    /******* MÉTODO PARA REGISTRAR UNA LINEA DE VENTA *****/

    public boolean RegistrarLineaVenta(ProductoVenta lineaVenta)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(Throwable ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpProductoVenta Registrar " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        try
        {
            elLogger.info("INSERTANDO EL PRODUCTO-VENTA");

            sesion.beginTransaction();
            sesion.save(lineaVenta);
            sesion.getTransaction().commit();

            resultado = true;
            elLogger.info("SE REGISTRÓ CON EXITO EL PRODUCTO-VENTA");
        }
        catch(PersistenceException excep)
        {
            elLogger.error("ERROR AL REGISTRAR EL PRODUCTO-VENTA");
        }
        finally
        {
            sesion.close();
        }
        return resultado;
    }
}

```

8.- OpProveedor.java

```

package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con el
Proveedor dentro del Sistema */

import java.util.*;
import javax.persistence.PersistenceException;
import javax.swing.JOptionPane;
import org.apache.log4j.*;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Proveedor;
import hibernate.conexion.Conexion;

public class OpProveedor
{
    private final static Logger elLogger = Logger.getLogger(OpProveedor.class);

```

```

Session sesion = null;

/***** MÉTODO PARA INSERTAR PROVEEDOR *****/

public void insertarProveedor(Proveedor elProveedor)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(Throwable ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Insertar " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    try
    {
        elLogger.info("INSERTANDO AL PROVEEDOR");

        sesion.beginTransaction();
        sesion.save(elProveedor);
        sesion.getTransaction().commit();

        elLogger.info("SE INSERTO CON EXITO AL PROVEEDOR");
    }
    catch(PersistenceException excep)
    {
        elLogger.error("ERROR AL INTENTAR INSERTAR AL PROVEEDOR");
    }
    finally
    {
        sesion.close();
    }

    JOptionPane.showMessageDialog(null,"SE INSERTÓ CORRECTAMENTE EL PROVEEDOR");
}

/***** MÉTODO PARA MOSTRAR TODOS LOS NOMBRES DE LOS PROVEEDORES *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<String> traerNombresProveedores()
{
    PropertyConfigurator.configure("logger.properties");

    List<String> listaProveedores;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Nombres Proveedores " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS NOMBRES DE PROVEEDORES");

    sentencia = "SELECT PROVEEDOR.nombreProveedor FROM Proveedor AS PROVEEDOR ORDER BY
PROVEEDOR.nombreProveedor";
    Query elQuery = sesion.createQuery(sentencia);
    listaProveedores = elQuery.list();

    elLogger.info("NOMBRES DE PROVEEDORES SELECCIONADOS");
}

```

```

        sesion.close();

        return listaProveedores;
    }

/***** MÉTODO PARA MOSTRAR PROVEEDOR SEGÚN NOMBRE *****/

    public Proveedor buscarProveedor(String nombreProveedor)
    {
        PropertyConfigurator.configure("logger.properties");

        Proveedor elProveedor = null;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Enviar Proveedor " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("BUSCANDO EL NOMBRE DEL PROVEEDOR");

        elProveedor = (Proveedor) sesion.createQuery("SELECT PROV FROM Proveedor AS PROV WHERE
PROV.nombreProveedor = '" + nombreProveedor + "'").uniqueResult();

        elLogger.info("SE ENCONTRO EL NOMBRE DEL PROVEEDOR");

        sesion.close();

        return elProveedor;
    }

/***** MÉTODO PARA MOSTRAR PROVEEDOR SEGÚN RFC *****/

    public Proveedor buscarProveedorRfc(String rfcProveedor)
    {
        PropertyConfigurator.configure("logger.properties");

        Proveedor elProveedor;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Enviar Proveedor " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("BUSCANDO EL NOMBRE DEL PROVEEDOR");

        elProveedor = (Proveedor) sesion.createQuery("SELECT PROV FROM Proveedor AS PROV WHERE
PROV.rfcProveedor = '" + rfcProveedor + "'").uniqueResult();

        elLogger.info("PROVEEDOR FUE ENCONTRADO");

        sesion.close();

        return elProveedor;
    }

/***** MÉTODO PARA ELIMINAR PROVEEDOR SEGÚN NOMBRE *****/

```

```

@SuppressWarnings("rawtypes")
public void eliminarProveedor(String nombreProveedor)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Eliminar Proveedor " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("ELIMINANDO AL PROVEEDOR " + nombreProveedor);

    String sentencia = "DELETE FROM Proveedor AS PROV WHERE PROV.nombreProveedor = '" +
nombreProveedor + "'";

    sesion.beginTransaction();
    Query elQuery = sesion.createQuery(sentencia);
    elQuery.executeUpdate();
    sesion.getTransaction().commit();

    elLogger.info("SE ELIMINO CORRECTAMENTE EL PROVEEDOR " + nombreProveedor);

    JOptionPane.showMessageDialog(null, "SE ELIMINÓ CORRECTAMENTE EL PROVEEDOR");

    sesion.close();
}

/***** MÉTODO PARA ACTUALIZAR PROVEEDOR *****/

public void actualizarProveedor(Proveedor elProveedor)
{
    PropertyConfigurator.configure("logger.properties");

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Actualizar Proveedor " +
ex);
        throw new ExceptionInInitializerError(ex);
    }

    Proveedor proveedorEncontrado = sesion.get(Proveedor.class, elProveedor.getRfcProveedor());

    elLogger.info("ACTUALIZANDO AL PROVEEDOR " + elProveedor.getRfcProveedor());

    sesion.beginTransaction();

    proveedorEncontrado.setFechaRegistro((java.sql.Date)elProveedor.getFechaRegistro());
    proveedorEncontrado.setRfcProveedor(elProveedor.getRfcProveedor());
    proveedorEncontrado.setNombreProveedor(elProveedor.getNombreProveedor());
    proveedorEncontrado.setDireccion(elProveedor.getDireccion());
    proveedorEncontrado.setTelefono(elProveedor.getTelefono());
    proveedorEncontrado.setCorreo(elProveedor.getCorreo());
    proveedorEncontrado.setNombreContacto(elProveedor.getNombreContacto());
    sesion.update(proveedorEncontrado);

    sesion.getTransaction().commit();

    elLogger.info("SE ACTUALIZO CORRECTAMENTE EL PROVEEDOR " + elProveedor.getRfcProveedor());
}

```

```

        JOptionPane.showMessageDialog(null, "SE ACTUALIZÓ CORRECTAMENTE EL PROVEEDOR");
        sesion.close();
    }

    /*****MÉTODO PARA MOSTRAR LISTA DE PROVEEDORES *****/

    @SuppressWarnings({ "unchecked", "rawtypes" })
    public List<Proveedor> traerDatosProveedores()
    {
        PropertyConfigurator.configure("logger.properties");

        List<Proveedor> listaProveedores;
        String sentencia;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Datos Proveedores " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("SELECCIONANDO LOS DATOS DE PROVEEDORES");

        sentencia = "SELECT PROV FROM Proveedor AS PROV";
        Query elQuery = sesion.createQuery(sentencia);

        listaProveedores = elQuery.list();

        elLogger.info("LISTOS TODOS LOS DATOS DE PROVEEDORES");

        sesion.close();

        return listaProveedores;
    }
}

```

9.- OpVenta.java

```

package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio relacionados con la Venta
dentro del Sistema */

import java.util.List;
import javax.persistence.PersistenceException;
import javax.swing.*;
import org.apache.log4j.*;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Venta;
import hibernate.conexion.Conexion;

public class OpVenta
{
    private final static Logger elLogger = Logger.getLogger(OpVenta.class);
    Session sesion = null;
    boolean resultado = false;

    /***** MÉTODO PARA REGISTRAR VENTA *****/

    public boolean RegistrarVenta(Venta venta)
    {
        PropertyConfigurator.configure("logger.properties");
    }
}

```



```

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(Throwable ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpVenta Registrar " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    try
    {
        elLogger.info("INSERTANDO LA VENTA");

        sesion.beginTransaction();
        sesion.save(venta);
        sesion.getTransaction().commit();

        resultado = true;
        elLogger.info("SE REGISTRÓ CON EXITO LA VENTA");
    }
    catch(PersistenceException excep)
    {
        elLogger.error("ERROR AL INTENTAR REGISTRAR LA VENTA");
    }
    finally
    {
        sesion.close();
    }

    JOptionPane.showMessageDialog(null,"SE REGISTRÓ CORRECTAMENTE LA VENTA");
    return resultado;
}

/*****MÉTODO PARA OBTENER LA ULTIMA VENTA *****/

public int traerUltimaVenta()
{
    PropertyConfigurator.configure("logger.properties");

    int idVenta;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESIÓN EN OpVenta Ultima Venta " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO ÚLTIMA VENTA");

    idVenta = (int) sesion.createQuery("SELECT MAX(VENTA.numeroTransaccion) FROM Venta AS
VENTA").uniqueResult();

    sesion.close();

    return idVenta;
}

/***** MÉTODO PARA MOSTRAR LISTA DE VENTAS SEGÚN LOS FILTROS *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Venta> buscarProductosFiltros(String desde,String hasta,String rfcEmpleado,String
tipoPago)
{

```

```

PropertyConfigurator.configure("logger.properties");

List<Venta> listaVentas = null;
String sentencia;

try
{
    sesion = Conexion.getSessionFactory().openSession();
}
catch(ExceptionInInitializerError ex)
{
    System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Consultar con Filtros " + ex);
    throw new ExceptionInInitializerError(ex);
}

elLogger.info("SELECCIONANDO LAS VENTAS");

if(rfcEmpleado.equals("") && tipoPago.equals("Seleccione una Opción"))
{
    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN '"+desde+"' AND
"+hasta
                +" ORDER BY VENTA.fechaVenta ASC";
    Query elQuery = sesion.createQuery(sentencia);
    listaVentas = elQuery.list();

    elLogger.info("LAS VENTAS FUERON SELECCIONADAS");
}
else if(rfcEmpleado.equals("") && !tipoPago.equals("Seleccione una Opción"))
{
    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN '"+desde+"' AND
"+hasta
                +" AND VENTA.tipoPago='"+tipoPago+"' ORDER BY VENTA.fechaVenta ASC";
    Query elQuery = sesion.createQuery(sentencia);
    listaVentas = elQuery.list();

    elLogger.info("LAS VENTAS FUERON SELECCIONADAS");
}
else if(!rfcEmpleado.equals("") && tipoPago.equals("Seleccione una Opción"))
{
    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN '"+desde+"' AND
"+hasta
                +" AND VENTA.empleadoRFC='"+rfcEmpleado+"' ORDER BY VENTA.fechaVenta ASC";
    Query elQuery = sesion.createQuery(sentencia);
    listaVentas = elQuery.list();

    elLogger.info("LAS VENTAS FUERON SELECCIONADAS");
}
else if(!rfcEmpleado.equals("") && !tipoPago.equals("Seleccione una Opción"))
{
    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN '"+desde+"' AND
"+hasta
                +" AND VENTA.empleadoRFC='"+rfcEmpleado+"' AND
VENTA.tipoPago='"+tipoPago+"' ORDER BY VENTA.fechaVenta ASC";
    Query elQuery = sesion.createQuery(sentencia);
    listaVentas = elQuery.list();

    elLogger.info("LAS VENTAS FUERON SELECCIONADAS");
}

sesion.close();

return listaVentas;
}
}

```

10.- OpCliente.java (Web)

```
package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio
relacionados con el Cliente dentro del Sistema **/

import org.apache.log4j.*;
import org.hibernate.Session;
import hibernate.beans.Cliente;
import hibernate.conexion.Conexion;

public class OpCliente
{
    private final static Logger elLogger = Logger.getLogger(OpCliente.class);
    Session session = null;

    /******* MÉTODO PARA VALIDAR USUARIO Y CLAVE DEL CLIENTE *****/

    public Cliente validarCliente(String usuario,String clave)
    {
        PropertyConfigurator.configure("logger.properties");
        Cliente cliente;

        try
        {
            session = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Validar Cliente "
+ ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("BUSCANDO AL CLIENTE");

        cliente = (Cliente) session.createQuery("SELECT CLIENTE FROM Cliente AS CLIENTE
WHERE CLIENTE.usuario='"+usuario+"' AND CLIENTE.contrasenia='"+clave+"'").uniqueResult();

        elLogger.info("CLIENTE ENCONTRADO");

        session.close();

        return cliente;
    }

    /******* MÉTODO PARA MOSTRAR CLIENTE SEGÚN RFC *****/

    public Cliente buscarClienteRfc(String rfcCliente)
    {
        PropertyConfigurator.configure("logger.properties");

        Cliente elCliente;

        try
        {
            session = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Enviar Cliente "
+ ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("BUSCANDO NOMBRE DEL CLIENTE");
```

```

        elCliente = (Cliente) sesion.createQuery("SELECT CLIENTE FROM Cliente AS CLIENTE
WHERE CLIENTE.rfcCliente = '" + rfcCliente + "'").uniqueResult();

        elLogger.info("NOMBRE DEL CLIENTE ENCONTRADO");

        sesion.close();

        return elCliente;
    }

/***** MÉTODO PARA ACTUALIZAR CLIENTE *****/

    public void actualizarCliente(Cliente elCliente)
    {
        PropertyConfigurator.configure("logger.properties");

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpCliente Actualizar
Cliente " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        Cliente clienteEncontrado = sesion.get(Cliente.class,elCliente.getRfcCliente());

        sesion.beginTransaction();
        elLogger.info("ACTUALIZANDO AL CLIENTE " + elCliente.getRfcCliente());

        clienteEncontrado.setFechaRegistro((java.sql.Date)elCliente.getFechaRegistro());
        clienteEncontrado.setNombreCliente(elCliente.getNombreCliente());
        clienteEncontrado.setPuntosAcumulados(elCliente.getPuntosAcumulados());
        clienteEncontrado.setDireccion(elCliente.getDireccion());
        clienteEncontrado.setTelefono(elCliente.getTelefono());
        clienteEncontrado.setCorreo(elCliente.getCorreo());
        clienteEncontrado.setUsuario(elCliente.getUsuario());
        clienteEncontrado.setContrasenia(elCliente.getContrasenia());
        sesion.update(clienteEncontrado);

        elLogger.info("SE ACTUALIZÓ EL CLIENTE " + elCliente.getRfcCliente());
        sesion.getTransaction().commit();

        sesion.close();
    }
}

```

11.- OpVenta.java (Web)

```

package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio
relacionados con la Venta dentro del Sistema */

import java.util.List;
import org.apache.log4j.*;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Venta;
import hibernate.conexion.Conexion;

public class OpVenta
{
    private final static Logger elLogger = Logger.getLogger(OpVenta.class);
    Session sesion = null;
}

```

```

/***** MÉTODO PARA MOSTRAR LAS VENTAS DE UN CLIENTE *****/

@SuppressWarnings({ "rawtypes", "unchecked" })
public List<Venta> buscarVentasPorCliente(String rfcCliente)
{
    PropertyConfigurator.configure("logger.properties");

    List<Venta> listaVentas;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpVenta Consultar por RFC
Cliente " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LAS VENTAS");

    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.clienteRFC = '" +
rfcCliente + "'";
    Query elQuery = sesion.createQuery(sentencia);

    listaVentas = elQuery.list();

    sesion.close();

    return listaVentas;
}

/***** MÉTODO PARA BUSCAR LA VENTA *****/

public Venta buscarVenta(Integer numeroTransaccion)
{
    PropertyConfigurator.configure("logger.properties");

    Venta venta;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpVenta Buscar Venta " +
ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL PRODUCTO");

    venta = (Venta) sesion.createQuery("SELECT VENTA FROM Venta AS VENTA WHERE
VENTA.numeroTransaccion = '" + numeroTransaccion + "'").uniqueResult();

    sesion.close();

    return venta;
}

/***** MÉTODO PARA CONSULTAR VENTAS CON RANGO FECHAS *****/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Venta> buscarVentasPorFechas(String desde,String hasta)

```

```

    {
        PropertyConfigurator.configure("logger.properties");

        List<Venta> listaVentas;
        String sentencia;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpVenta Consultar por
Fechas " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("SELECCIONANDO LOS DATOS DE LAS VENTAS");

        sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN
"+"desde+" AND "+"hasta+" ORDER BY VENTA.numeroTransaccion ASC";
        Query elQuery = sesion.createQuery(sentencia);

        listaVentas = elQuery.list();

        sesion.close();

        return listaVentas;
    }

/***** MÉTODO PARA MOSTRAR VENTAS CON FECHAS Y RFC EMPLEADO *****/

    @SuppressWarnings({ "unchecked", "rawtypes" })
    public List<Venta> buscarVentasPorFechasEmpleado(String desde,String hasta,String
claveEmpleado)
    {
        PropertyConfigurator.configure("logger.properties");

        List<Venta> listaVentas;
        String sentencia;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpVenta Consultar por
Empleado " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("SELECCIONANDO LOS DATOS DE LAS VENTAS");

        sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN
"+"desde+" AND "+"hasta+" AND VENTA.empleadoRFC='"+claveEmpleado+"' ORDER BY
VENTA.numeroTransaccion ASC";
        Query elQuery = sesion.createQuery(sentencia);

        listaVentas = elQuery.list();

        sesion.close();

        return listaVentas;
    }

/***** MÉTODO PARA MOSTRAR VENTAS CON FECHAS Y TIPO DE PAGO *****/

```

```

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Venta> buscarVentasPorFechasTipoPago(String desde,String hasta,String
tipoPago)
{
    PropertyConfigurator.configure("logger.properties");

    List<Venta> listaVentas;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpVenta Consultar por Tipo
Pago " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LAS VENTAS");

    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN
"+"desde+" AND "+hasta+" AND VENTA.tipoPago='"+tipoPago+"' ORDER BY
VENTA.numeroTransaccion ASC";
    Query elQuery = sesion.createQuery(sentencia);

    listaVentas = elQuery.list();

    sesion.close();

    return listaVentas;
}

/* MÉTODO PARA MOSTRAR VENTAS CON FECHAS,RFC EMPLEADO Y TIPO DE PAGO */

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Venta> buscarventasPorFechasEmpleadoTipoPago(String desde,String
hasta,String claveEmpleado,String tipoPago)
{
    PropertyConfigurator.configure("logger.properties");

    List<Venta> listaVentas;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN Opventa Consultar por
Empleado y Tipo Pago " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LAS VENTAS");
    sentencia = "SELECT VENTA FROM Venta AS VENTA WHERE VENTA.fechaVenta BETWEEN
"+"desde+" AND "+hasta+" AND VENTA.empleadoRFC='"+claveEmpleado+"' AND
VENTA.tipoPago='"+tipoPago+"' ORDER BY VENTA.numeroTransaccion ASC";
    Query elQuery = sesion.createQuery(sentencia);

    listaVentas = elQuery.list();

    sesion.close();

    return listaVentas;
}
}

```

12.- OpEmpleado.java (Web)

```
package hibernate.operaciones;

/** Esta Clase contiene el método para validar el Usuario y Clave del Empleado dentro del
Sistema Web **/

import org.apache.log4j.*;
import org.hibernate.Session;
import hibernate.beans.Empleado;
import hibernate.conexion.Conexion;

public class OpEmpleado
{
    private final static Logger elLogger = Logger.getLogger(OpCliente.class);
    Session sesion = null;

    /******* MÉTODO PARA VALIDAR USUARIO Y CLAVE DEL EMPLEADO *****/

    public Empleado buscarEmpleado(String usuario,String clave)
    {
        PropertyConfigurator.configure("logger.properties");

        Empleado elEmpleado;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpEmpleado Enviar Empleado
" + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("BUSCANDO EL NOMBRE DEL EMPLEADO");

        elEmpleado = (Empleado) sesion.createQuery("SELECT EMPLEADO FROM Empleado AS
EMPLEADO WHERE EMPLEADO.usuario='"+usuario+"' AND
EMPLEADO.contrasenia='"+clave+"'").uniqueResult();

        sesion.close();
        return elEmpleado;
    }
}
```

13.- OpProducto.java (Web)

```
package hibernate.operaciones;

/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio
relacionados con el Producto dentro del Sistema **/

import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Producto;
import hibernate.conexion.Conexion;

public class OpProducto
{
    private final static Logger elLogger = Logger.getLogger(OpProducto.class);
    Session sesion = null;
```



```

/* MÉTODO PARA CONSULTAR LISTA DE PRODUCTOS CON SOLO RANGO FECHAS */

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Producto> buscarProductosPorFechas(String desde,String hasta)
{
    PropertyConfigurator.configure("logger.properties");

    List<Producto> listaProductos;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Consultar por
Fechas " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PRODUCTOS");

    sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE
PRODUCTO.fechaRegistro BETWEEN '"+desde+"' AND '"+hasta+"' ORDER BY
PRODUCTO.fechaRegistro ASC";
    Query elQuery = sesion.createQuery(sentencia);

    listaProductos = elQuery.list();

    sesion.close();

    return listaProductos;
}

/* MÉTODO MOSTRAR LISTA DE PRODUCTOS SEGÚN FECHAS Y RFC PROVEEDOR */

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Producto> buscarProductosPorFechasProveedor(String desde,String
hasta,String rfcProveedor)
{
    PropertyConfigurator.configure("logger.properties");

    List<Producto> listaProductos;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Consultar por
Proveedor " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PRODUCTOS");

    sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE
PRODUCTO.fechaRegistro BETWEEN '"+desde+"' AND '"+hasta+"' AND
PRODUCTO.proveedor='"+rfcProveedor+"' ORDER BY PRODUCTO.fechaRegistro ASC";
    Query elQuery = sesion.createQuery(sentencia);

    listaProductos = elQuery.list();

    sesion.close();
}

```

```

    }
    return listaProductos;
}

/* MÉTODO PARA MOSTRAR LISTA DE PRODUCTOS SEGÚN FECHAS Y CÓDIGO DE BARRAS */

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Producto> buscarProductosPorFechasCodigoBarras(String desde,String
hasta,String codigoBarras)
{
    PropertyConfigurator.configure("logger.properties");

    List<Producto> listaProductos;
    String sentencia;

    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Consultar por
Codigo de Barras " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PRODUCTOS");
    sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE
PRODUCTO.fechaRegistro BETWEEN '"+desde+"' AND '"+hasta+"' AND
PRODUCTO.codigoBarras='"+codigoBarras+"' ORDER BY PRODUCTO.fechaRegistro ASC";
    Query elQuery = sesion.createQuery(sentencia);
    listaProductos = elQuery.list();

    sesion.close();
    return listaProductos;
}

/* MÉTODO PARA MOSTRAR LISTA DE PRODUCTOS SEGÚN FECHAS, CÓDIGO DE BARRAS Y RFC PROVEEDOR
*/

@SuppressWarnings({ "unchecked", "rawtypes" })
public List<Producto> buscarProductosPorFechasCodigoBarrasProveedor(String
desde,String hasta,String codigoBarras,String rfcProveedor)
{
    PropertyConfigurator.configure("logger.properties");

    List<Producto> listaProductos;
    String sentencia;
    try
    {
        sesion = Conexion.getSessionFactory().openSession();
    }
    catch(ExceptionInInitializerError ex)
    {
        System.err.println("NO SE PUDO CREAR LA SESION EN OpProducto Consultar por
Codigo Barras y Proveedor " + ex);
        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("SELECCIONANDO LOS DATOS DE LOS PRODUCTOS");
    sentencia = "SELECT PRODUCTO FROM Producto AS PRODUCTO WHERE
PRODUCTO.fechaRegistro BETWEEN '"+desde+"' AND '"+hasta+"' AND
PRODUCTO.codigoBarras='"+codigoBarras+"' AND PRODUCTO.proveedor='"+rfcProveedor+"' ORDER
BY PRODUCTO.fechaRegistro ASC";
    Query elQuery = sesion.createQuery(sentencia);
    listaProductos = elQuery.list();
    sesion.close();
    return listaProductos;
}
}

```

14.- OpProveedor.java (Web)

```
package hibernate.operaciones;
/** Esta Clase contiene todos los métodos necesarios para la lógica de negocio
relacionados con el Proveedor dentro del Sistema */

import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;
import org.hibernate.query.Query;
import hibernate.beans.Proveedor;
import hibernate.conexion.Conexion;

public class OpProveedor
{
    private final static Logger elLogger = Logger.getLogger(OpProveedor.class);
    Session sesion = null;

    /* MÉTODO PARA MOSTRAR TODOS LOS PROVEEDORES REGISTRADOS */

    @SuppressWarnings({ "unchecked", "rawtypes" })
    public List<Proveedor> traerProveedores()
    {
        PropertyConfigurator.configure("logger.properties");

        List<Proveedor> listaProveedores;
        String sentencia;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Nombres
Proveedor " + ex);
            throw new ExceptionInInitializerError(ex);
        }

        elLogger.info("SELECCIONANDO LOS NOMBRES DE PROVEEDORES");

        sentencia = "SELECT PROVEEDOR FROM Proveedor AS PROVEEDOR ORDER BY
PROVEEDOR.nombreProveedor";
        Query elQuery = sesion.createQuery(sentencia);

        listaProveedores = elQuery.list();

        sesion.close();

        return listaProveedores;
    }

    /* MÉTODO PARA MOSTRAR PROVEEDOR SEGÚN RFC */

    public Proveedor buscarProveedorRfc(String rfcProveedor)
    {
        PropertyConfigurator.configure("logger.properties");

        Proveedor elProveedor;

        try
        {
            sesion = Conexion.getSessionFactory().openSession();
        }
        catch(ExceptionInInitializerError ex)
        {
            System.err.println("NO SE PUDO CREAR LA SESION EN OpProveedor Enviar
Proveedor " + ex);
        }
    }
}
```

```

        throw new ExceptionInInitializerError(ex);
    }

    elLogger.info("BUSCANDO EL NOMBRE DEL PROVEEDOR");
    elProveedor = (Proveedor) sesion.createQuery("SELECT PROV FROM Proveedor AS PROV
WHERE PROV.rfcProveedor = '" + rfcProveedor + "'").uniqueResult();
    sesion.close();
    return elProveedor;
}
}
}

```

ANEXO E. Código de las clases contenidas en el **paquete interfaces** de la aplicación de escritorio:

1.- ConsultarCuentasPorPagar.java

```

package Interfaces;

/** Esta Interfaz permite consultar los Pedidos que están Pendientes por Pagar para cada
Proveedor en el Sistema */

import java.awt.*;
import java.awt.event.*;
import java.util.List;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import hibernate.beans.Pedido;
import hibernate.beans.Producto;
import hibernate.operaciones.OpProveedor;
import hibernate.operaciones.OpPedido;

public class ConsultarCuentasPorPagar
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmCuentasPorPagar;
    private JPanel panelGeneral;

    /******* CAJAS Y ELEMENTOS DE LOS FILTROS *****/

    @SuppressWarnings("rawtypes")
    JComboBox cajaListaProveedores;
    private ButtonGroup losRadio;
    private JRadioButton radioUno,radioTodos;
    private JTextField cajaNumeroPedido;
    private TextPrompt leyenda;

    /******* TABLA CONTENEDOR DE LOS PEDIDOS *****/

    private JTable tablaPedidos;
    private DefaultTableModel modeloTabla;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
    private OpPedido operacionesPedido = new OpPedido();
    private Producto elProducto = new Producto();

    /******* MAIN *****/

    public static void main(String[] args)
    {

```

```

EventQueue.invokeLater(new Runnable()
{
    public void run()
    {
        try
        {
            ConsultarCuentasPorPagar window = new ConsultarCuentasPorPagar();
            window.frmCuentasPorPagar.setVisible(true);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
});
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmCuentasPorPagar.setVisible(visibilidad);
}

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

public ConsultarCuentasPorPagar()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DE CONSULTAR CUENTAS POR PAGAR *****/

@SuppressWarnings({ "rawtypes", "unchecked", "serial" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmCuentasPorPagar = new JFrame();
    frmCuentasPorPagar.setTitle("Cuentas por Pagar");
    frmCuentasPorPagar.getContentPane().setBackground(new Color(175, 238, 238));

    frmCuentasPorPagar.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProveedor.class.getResource("/Imagenes/Logotipo.png")));
    frmCuentasPorPagar.setBounds(100, 100, 450, 300);
    frmCuentasPorPagar.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmCuentasPorPagar.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmCuentasPorPagar.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1100));
    panelGeneral.setBackground(new Color(255,255,255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override

```

```

public void mouseClicked(MouseEvent arg0)
{
    MenuPrincipal menuPrincipal = new MenuPrincipal();

    menuPrincipal.setVisible(true);
    frmCuentasPorPagar.setVisible(false);
}

@Override
public void mouseEntered(MouseEvent e)
{
    etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
}
@Override
public void mouseExited(MouseEvent e)
{
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
}
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmCuentasPorPagar.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);

```

```

panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Cuentas por Pagar");
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(320, 31, 200, 41);
panelGeneral.add(etiquetaProductos);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox Filtros para Realizar las Consultas *****/

JPanel groupBoxConsultarCuentas = new JPanel();
groupBoxConsultarCuentas.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Consultar Cuentas por Pagar", TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(0, 0, 0)));
groupBoxConsultarCuentas.setBackground(new Color(255,255,255));
groupBoxConsultarCuentas.setBounds(423, 185, 525, 300);
groupBoxConsultarCuentas.setLayout(null);
panelGeneral.add(groupBoxConsultarCuentas);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.setBounds(219, 47, 221, 36);
groupBoxConsultarCuentas.add(cajaListaProveedores);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setBounds(79, 47, 131, 30);
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
groupBoxConsultarCuentas.add(etiquetaProveedor);

radioUno = new JRadioButton("1 Cuenta");
radioUno.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajaNumeroPedido.setEnabled(true);
    }
});
radioUno.setBackground(new Color(255, 255, 255));
radioUno.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioUno.setHorizontalAlignment(SwingConstants.CENTER);
radioUno.setBounds(50, 105, 184, 36);

```

```

groupBoxConsultarCuentas.add(radioUno);

radioTodos = new JRadioButton("Todas las Cuentas");
radioTodos.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajaNumeroPedido.setText("");
        cajaNumeroPedido.setEnabled(false);
    }
});
radioTodos.setBackground(new Color(255, 255, 255));
radioTodos.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioTodos.setHorizontalAlignment(SwingConstants.CENTER);
radioTodos.setBounds(250, 105, 190, 36);
radioTodos.setSelected(true);
groupBoxConsultarCuentas.add(radioTodos);

losRadio = new ButtonGroup();
losRadio.add(radioUno);
losRadio.add(radioTodos);

JLabel etiquetaNumeroPedido = new JLabel("Pedido:");
etiquetaNumeroPedido.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNumeroPedido.setBounds(141, 170, 93, 30);
groupBoxConsultarCuentas.add(etiquetaNumeroPedido);

cajaNumeroPedido = new JTextField();
cajaNumeroPedido.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaNumeroPedido.setEnabled(false);
cajaNumeroPedido.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Número de Pedido",cajaNumeroPedido);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNumeroPedido.setBounds(237, 170, 148, 36);
cajaNumeroPedido.setColumns(10);
cajaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 16));
groupBoxConsultarCuentas.add(cajaNumeroPedido);

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(50, 580, 1250, 330);
scrollTabla.setHorizontalScrollBar(null);
panelGeneral.add(scrollTabla);

tablaPedidos = new JTable();
modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
tablaPedidos = new JTable(modeloTabla);

```



```

scrollTabla.setViewportView(tablaPedidos);
tablaPedidos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
tablaPedidos.setRowHeight(30);
modeloTabla.addColumn("# PEDIDO");
modeloTabla.addColumn("FECHA");
modeloTabla.addColumn("CÓDIGO BARRAS");
modeloTabla.addColumn("PRODUCTO");
modeloTabla.addColumn("PROVEEDOR");
modeloTabla.addColumn("CANTIDAD");
modeloTabla.addColumn("PRECIO");
modeloTabla.addColumn("TOTAL");
modeloTabla.addColumn("ESTATUS");
tablaPedidos.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
tablaPedidos.getColumnModel().getColumn(0).setPreferredWidth(100);
tablaPedidos.getColumnModel().getColumn(1).setPreferredWidth(110);
tablaPedidos.getColumnModel().getColumn(2).setPreferredWidth(180);
tablaPedidos.getColumnModel().getColumn(3).setPreferredWidth(200);
tablaPedidos.getColumnModel().getColumn(4).setPreferredWidth(200);
tablaPedidos.getColumnModel().getColumn(5).setPreferredWidth(85);
tablaPedidos.getColumnModel().getColumn(6).setPreferredWidth(85);
tablaPedidos.getColumnModel().getColumn(7).setPreferredWidth(85);
tablaPedidos.getColumnModel().getColumn(8).setPreferredWidth(205);
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
tablaPedidos.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(2).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(3).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(4).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(5).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(6).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(7).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(8).setCellRenderer(justificacion);
scrollTabla.setViewportView(tablaPedidos);

/***** Formato y Funcionalidad del Botón "Ver" según Filtros *****/

JButton botonVer = new JButton("Ver");
botonVer.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        boolean radioTodosPedidos = true;

        while(modeloTabla.getRowCount()>0)
        {
            modeloTabla.removeRow(0);
        }

        if(radioUno.isSelected())
        {
            radioTodosPedidos = false;
        }

        List<Pedido> listaPedidos =
operacionesPedido.buscarPedidosFiltrosPendientes((String)cajaListaProveedores.getSelected
Item()),

radioTodosPedidos, cajaNumeroPedido.getText());
        if(listaPedidos != null)
        {
            if(listaPedidos.size() != 0)
            {
                for (Pedido pedido : listaPedidos)
                {
                    elProducto =
operacionesPedido.tracerProductoPedido(pedido.getProductoCodigoBarras());
                    Object[] renglon =
{pedido.getNumeroPedido(),pedido.getFechaPedido(),pedido.getProductoCodigoBarras(),

```

```

elProducto.getNombreProducto(), (String) cajaListaProveedores.getSelectedItem(),
pedido.getCantidadSolicitada(), elProducto.getCostoPublico(), pedido.getTotalLiquidar(),
                                pedido.getStatusPedido());
                                modeloTabla.addRow(renglon);
                                }
                                }
                                else
                                {
                                JOptionPane.showMessageDialog(null, "No se Encontraron Pedidos
para esta Consulta.");
                                }
                                }
                                });
                                botonVer.addMouseListener(new MouseAdapter()
                                {
                                @Override
                                public void mouseEntered(MouseEvent e)
                                {
                                botonVer.setForeground(new Color(12,2,235));
                                }
                                @Override
                                public void mouseExited(MouseEvent e)
                                {
                                botonVer.setForeground(new Color(0,0,0));
                                }
                                });
                                Border elBordeConsultar = new LineBorder(Color.black,1);
                                botonVer.setBorder(elBordeConsultar);
                                botonVer.setForeground(new Color(0,0,0));
                                botonVer.setBackground(new Color(216, 253, 241));
                                botonVer.setFont(new Font("Century", Font.PLAIN, 18));
                                botonVer.setBounds(230, 240, 80, 40);
                                groupBoxConsultarCuentas.add(botonVer);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonSalir = new JButton(" Salir");
botonSalir.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmCuentasPorPagar.setVisible(false);
    }
});
botonSalir.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonSalir.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonSalir.setForeground(new Color(0,0,0));
    }
});
botonSalir.setHorizontalAlignment(SwingConstants.CENTER);
Border elBordeSalir = new LineBorder(Color.black,1);
botonSalir.setBorder(elBordeSalir);
botonSalir.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));

```

```

        botonSalir.setForeground(new Color(0,0,0));
        botonSalir.setBackground(new Color(234,34,34));
        botonSalir.setFont(new Font("Century", Font.PLAIN, 20));
        botonSalir.setBounds(600, 980, 210, 50);
        panelGeneral.add(botonSalir);
    }
}

```

2.- ConsultarDevoluciones.java

```

package Interfaces;

/** Esta Interfaz permite consultar las Devoluciones que han sido registradas para cada
Proveedor en el Sistema */

import java.awt.*;
import java.awt.event.*;
import java.util.List;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import hibernate.beans.*;
import hibernate.operaciones.*;

public class ConsultarDevoluciones
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmConsultarDevoluciones;
    private JPanel panelGeneral;

    /******* CAJAS Y ELEMENTOS DE LOS FILTROS *****/

    @SuppressWarnings("rawtypes")
    private JComboBox cajaListaProveedores;
    private ButtonGroup losRadio;
    private JRadioButton radioUno,radioTodas;
    private JTextField cajaNumeroDevolucion;
    private TextPrompt leyenda;

    /******* TABLA CONTENEDOR DE LAS DEVOLUCIONES *****/

    private JTable tablaDevoluciones;
    private DefaultTableModel modeloTabla;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
    private OpPedido operacionesPedido = new OpPedido();
    private OpDevolucion operacionesDevolucion = new OpDevolucion();
    private OpProducto operacionesProducto = new OpProducto();
    private Pedido elPedido = new Pedido();
    private Producto elProducto = new Producto();

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {

```

```

        ConsultarDevoluciones window = new ConsultarDevoluciones();
        window.frmConsultarDevoluciones.setVisible(true);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    });
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmConsultarDevoluciones.setVisible(visibilidad);
}

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

public ConsultarDevoluciones()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DE CONSULTAR PRODUCTO *****/

@SuppressWarnings({ "rawtypes", "unchecked", "serial" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmConsultarDevoluciones = new JFrame();
    frmConsultarDevoluciones.setTitle("Consultar Devoluciones");
    frmConsultarDevoluciones.getContentPane().setBackground(new Color(175, 238,
238));

    frmConsultarDevoluciones.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProve
edor.class.getResource("/Imágenes/Logotipo.png")));
    frmConsultarDevoluciones.setBounds(100, 100, 450, 300);
    frmConsultarDevoluciones.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmConsultarDevoluciones.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmConsultarDevoluciones.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1100));
    panelGeneral.setBackground(new Color(255,255,255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);

```

```

        frmConsultarDevoluciones.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmConsultarDevoluciones.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaDevoluciones = new JLabel("Devoluciones");
etiquetaDevoluciones.addMouseListener(new MouseAdapter()
{

```

```

@Override
public void mouseClicked(MouseEvent e)
{
    MenuDevoluciones devoluciones = new MenuDevoluciones();

    devoluciones.setVisible(true);
    frmConsultarDevoluciones.setVisible(false);
}
@Override
public void mouseEntered(MouseEvent e)
{
    etiquetaDevoluciones.setForeground(new Color(232, 19, 245));
}
@Override
public void mouseExited(MouseEvent e)
{
    etiquetaDevoluciones.setForeground(new Color(0, 0, 255));
}
});
etiquetaDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDevoluciones.setForeground(Color.BLUE);
etiquetaDevoluciones.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaDevoluciones.setBounds(330, 31, 125, 41);
panelGeneral.add(etiquetaDevoluciones);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(455, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaConsultarDevoluciones = new JLabel("Consultar");
etiquetaConsultarDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaConsultarDevoluciones.setForeground(Color.BLUE);
etiquetaConsultarDevoluciones.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaConsultarDevoluciones.setBounds(465, 32, 119, 41);
panelGeneral.add(etiquetaConsultarDevoluciones);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox Filtros para Realizar las Consultas *****/

JPanel groupBoxConsultarDevoluciones = new JPanel();
groupBoxConsultarDevoluciones.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Consultar Devoluciones",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxConsultarDevoluciones.setBackground(new Color(255,255,255));
groupBoxConsultarDevoluciones.setBounds(423, 185, 525, 300);
groupBoxConsultarDevoluciones.setLayout(null);
panelGeneral.add(groupBoxConsultarDevoluciones);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));

```

```

cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.setBounds(219, 47, 221, 36);
groupBoxConsultarDevoluciones.add(cajaListaProveedores);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setBounds(79, 47, 131, 30);
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
groupBoxConsultarDevoluciones.add(etiquetaProveedor);

radioUno = new JRadioButton("1 Devolución");
radioUno.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajaNumeroDevolucion.setEnabled(true);
    }
});
radioUno.setBackground(new Color(255, 255, 255));
radioUno.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioUno.setHorizontalAlignment(SwingConstants.CENTER);
radioUno.setBounds(30, 105, 184, 36);
groupBoxConsultarDevoluciones.add(radioUno);

radioTodas = new JRadioButton("Todas las Devoluciones");
radioTodas.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajaNumeroDevolucion.setText("");
        cajaNumeroDevolucion.setEnabled(false);
    }
});
radioTodas.setBackground(new Color(255, 255, 255));
radioTodas.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioTodas.setHorizontalAlignment(SwingConstants.CENTER);
radioTodas.setBounds(220, 105, 230, 36);
radioTodas.setSelected(true);
groupBoxConsultarDevoluciones.add(radioTodas);

losRadio = new ButtonGroup();
losRadio.add(radioUno);
losRadio.add(radioTodas);

JLabel etiquetaNumeroDevolucion = new JLabel("Devolución:");
etiquetaNumeroDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNumeroDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNumeroDevolucion.setBounds(90, 170, 140, 30);
groupBoxConsultarDevoluciones.add(etiquetaNumeroDevolucion);

cajaNumeroDevolucion = new JTextField();
cajaNumeroDevolucion.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});

```

```

    }
  });
  cajaNumeroDevolucion.setEnabled(false);
  cajaNumeroDevolucion.setHorizontalAlignment(SwingConstants.CENTER);
  leyenda = new TextPrompt("Número de Devolución",cajaNumeroDevolucion);
  leyenda.setHorizontalAlignment(SwingConstants.CENTER);
  leyenda.changeAlpha(0.50f);
  leyenda.changeStyle(Font.ITALIC);
  cajaNumeroDevolucion.setBounds(237, 170, 148, 36);
  cajaNumeroDevolucion.setColumns(10);
  cajaNumeroDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
  groupBoxConsultarDevoluciones.add(cajaNumeroDevolucion);

```

/Formato y Funcionalidad de la Tabla para Visualizar las Devoluciones /

```

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(10, 580, 1330, 330);
scrollTabla.setHorizontalScrollBar(null);
panelGeneral.add(scrollTabla);

tablaDevoluciones = new JTable();
modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
tablaDevoluciones = new JTable(modeloTabla);
scrollTabla.setViewportView(scrollTabla);
tablaDevoluciones.setFont(new Font("Cambria Math", Font.PLAIN, 15));
tablaDevoluciones.setRowHeight(30);
modeloTabla.addColumn("# DEVOLUCIÓN");
modeloTabla.addColumn("FECHA");
modeloTabla.addColumn("CÓDIGO BARRAS");
modeloTabla.addColumn("PRODUCTO");
modeloTabla.addColumn("PROVEEDOR");
modeloTabla.addColumn("CANTIDAD");
modeloTabla.addColumn("PRECIO");
modeloTabla.addColumn("TOTAL");
modeloTabla.addColumn("MOTIVO");
modeloTabla.addColumn("CLASIFICACIÓN");
modeloTabla.addColumn("ESTATUS");
tablaDevoluciones.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
tablaDevoluciones.getColumnModel().getColumn(0).setPreferredWidth(100);
tablaDevoluciones.getColumnModel().getColumn(1).setPreferredWidth(90);
tablaDevoluciones.getColumnModel().getColumn(2).setPreferredWidth(150);
tablaDevoluciones.getColumnModel().getColumn(3).setPreferredWidth(150);
tablaDevoluciones.getColumnModel().getColumn(4).setPreferredWidth(150);
tablaDevoluciones.getColumnModel().getColumn(5).setPreferredWidth(85);
tablaDevoluciones.getColumnModel().getColumn(6).setPreferredWidth(85);
tablaDevoluciones.getColumnModel().getColumn(7).setPreferredWidth(85);
tablaDevoluciones.getColumnModel().getColumn(8).setPreferredWidth(200);
tablaDevoluciones.getColumnModel().getColumn(9).setPreferredWidth(135);
tablaDevoluciones.getColumnModel().getColumn(10).setPreferredWidth(100);
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
tablaDevoluciones.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(2).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(3).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(4).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(5).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(6).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(7).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(8).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(9).setCellRenderer(justificacion);

```



```

tablaDevoluciones.getColumnModel().getColumn(10).setCellRenderer(justificacion);
scrollTabla.setViewportView(tablaDevoluciones);

/***** Formato y Funcionalidad del Botón "Ver" según Filtros *****/

JButton botonVer = new JButton("Ver");
botonVer.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        boolean radioTodasDevoluciones = true;

        while(modeloTabla.getRowCount()>0)
        {
            modeloTabla.removeRow(0);
        }

        if(radioUno.isSelected())
        {
            radioTodasDevoluciones = false;
        }

        List<Devolucion> listaDevoluciones =
operacionesDevolucion.buscarDevolucionesFiltros((String)cajaListaProveedores.getSelectedI
tem(),

radioTodasDevoluciones, cajaNumeroDevolucion.getText());
        if(listaDevoluciones != null)
        {
            if(listaDevoluciones.size() != 0)
            {
                for (Devolucion devolucion : listaDevoluciones)
                {
                    elPedido =
operacionesPedido.buscarPedidoPorNumero(devolucion.getPedidoNumeroPedido());
                    elProducto =
operacionesProducto.buscarProductoPorID(elPedido.getProductoCodigoBarras());
                    Object[] renglon =
{devolucion.getNumeroDevolucion(),devolucion.getFechaDevolucion(),

elPedido.getProductoCodigoBarras(),elProducto.getNombreProducto(),(String)cajaListaProvee
dores.getSelectedItem(),

devolucion.getCantidadProductos(),elProducto.getCostoCompra(),devolucion.getCostoDevoluci
on(),

devolucion.getMotivo(),devolucion.getClasificacion(),devolucion.getStatusDevolucion()};
                    modeloTabla.addRow(renglon);
                }
            }
            else
            {
                JOptionPane.showMessageDialog(null,"No se Encontraron
Devoluciones para esta Busqueda.");
            }
        }
    }
});
botonVer.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonVer.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {

```

```

        botonVer.setForeground(new Color(0,0,0));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,1);
botonVer.setBorder(elBordeConsultar);
botonVer.setForeground(new Color(0,0,0));
botonVer.setBackground(new Color(216, 253, 241));
botonVer.setFont(new Font("Century", Font.PLAIN, 18));
botonVer.setBounds(230, 240, 80, 40);
groupBoxConsultarDevoluciones.add(botonVer);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonSalir = new JButton(" Salir");
botonSalir.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuDevoluciones menuDevoluciones = new MenuDevoluciones();

        menuDevoluciones.setVisible(true);
        frmConsultarDevoluciones.setVisible(false);
    }
});
botonSalir.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonSalir.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonSalir.setForeground(new Color(0,0,0));
    }
});
botonSalir.setHorizontalAlignment(SwingConstants.CENTER);
Border elBordeSalir = new LineBorder(Color.black,1);
botonSalir.setBorder(elBordeSalir);
botonSalir.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonSalir.setForeground(new Color(0,0,0));
botonSalir.setBackground(new Color(234,34,34));
botonSalir.setFont(new Font("Century", Font.PLAIN, 20));
botonSalir.setBounds(600, 980, 210, 50);
panelGeneral.add(botonSalir);

}
}

```

3.- ConsultarPedidos.java

```

package Interfaces;

/** Esta Interfaz permite consultar los Pedidos que han sido registrados para cada Proveedor
en el Sistema */

import java.awt.*;
import java.awt.event.*;
import java.util.List;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import hibernate.beans.Pedido;
import hibernate.beans.Producto;

```

```

import hibernate.operaciones.OpProveedor;
import hibernate.operaciones.OpPedido;

public class ConsultarPedidos
{
/*****CONTENEDOR JFrame BASE Y PANEL GENERAL *****/

private JFrame frmConsultarPedidos;
private JPanel panelGeneral;

/***** CAJAS Y ELEMENTOS DE LOS FILTROS *****/

@SuppressWarnings("rawtypes")
private JComboBox cajaListaProveedores;
private ButtonGroup losRadio;
private JRadioButton radioUno,radioTodos;
private JTextField cajaNumeroPedido;
private TextPrompt leyenda;

/***** TABLA CONTENEDOR DE LOS PEDIDOS *****/

private DefaultTableModel modeloTabla;
private JTable tablaPedidos;

/***** OPERACIONES PARA FUNCIONALIDAD *****/

private OpProveedor operacionesProveedor = new OpProveedor();
private OpPedido operacionesPedido = new OpPedido();
private Producto elProducto = new Producto();

/***** MAIN *****/

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                ConsultarPedidos window = new ConsultarPedidos();
                window.frmConsultarPedidos.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmConsultarPedidos.setVisible(visibilidad);
}

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

public ConsultarPedidos()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DE CONSULTAR PEDIDOS *****/

@SuppressWarnings({ "rawtypes", "unchecked", "serial" })

```

```

private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

frmConsultarPedidos = new JFrame();
frmConsultarPedidos.setTitle("Consultar Pedidos");
frmConsultarPedidos.getContentPane().setBackground(new Color(175, 238, 238));

frmConsultarPedidos.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProveedor.
class.getResource("/Imagenes/Logotipo.png")));
frmConsultarPedidos.setBounds(100, 100, 450, 300);
frmConsultarPedidos.setExtendedState(JFrame.MAXIMIZED_BOTH);
frmConsultarPedidos.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JScrollPane scrollGeneral = new JScrollPane();
scrollGeneral.setBackground(new Color(175, 238, 238));
scrollGeneral.setBounds(0, 0, 1365, 715);
scrollGeneral.setHorizontalScrollBar(null);
frmConsultarPedidos.getContentPane().add(scrollGeneral);

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1100));
panelGeneral.setBackground(new Color(255,255,255));
scrollGeneral.setViewportViewView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
@Override
public void mouseClicked(MouseEvent arg0)
{
MenuPrincipal menuPrincipal = new MenuPrincipal();

menuPrincipal.setVisible(true);
frmConsultarPedidos.setVisible(false);
}

@Override
public void mouseEntered(MouseEvent e)
{
etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
}
@Override
public void mouseExited(MouseEvent e)
{
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
}
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

```

```

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmConsultarPedidos.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Pedidos");
etiquetaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        MenuPedidos pedidos = new MenuPedidos();

        pedidos.setVisible(true);
        frmConsultarPedidos.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(0, 0, 255));
    }
});
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(312, 31, 115, 41);
panelGeneral.add(etiquetaProductos);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);

```

```

etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(405, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaConsultarProveedores = new JLabel("Consultar");
etiquetaConsultarProveedores.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaConsultarProveedores.setForeground(Color.BLUE);
etiquetaConsultarProveedores.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaConsultarProveedores.setBounds(365, 32, 219, 41);
panelGeneral.add(etiquetaConsultarProveedores);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox Filtros para Realizar las Consultas *****/

JPanel groupBoxConsultarPedido = new JPanel();
groupBoxConsultarPedido.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Consultar Pedidos",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxConsultarPedido.setBackground(new Color(255,255,255));
groupBoxConsultarPedido.setBounds(423, 185, 525, 300);
groupBoxConsultarPedido.setLayout(null);
panelGeneral.add(groupBoxConsultarPedido);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.setBounds(219, 47, 221, 36);
groupBoxConsultarPedido.add(cajaListaProveedores);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setBounds(79, 47, 131, 30);
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
groupBoxConsultarPedido.add(etiquetaProveedor);

radioUno = new JRadioButton("1 Pedido");
radioUno.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajaNumeroPedido.setEnabled(true);
    }
});
radioUno.setBackground(new Color(255, 255, 255));
radioUno.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioUno.setHorizontalAlignment(SwingConstants.CENTER);

```

```

radioUno.setBounds(50, 105, 184, 36);
groupBoxConsultarPedido.add(radioUno);

radioTodos = new JRadioButton("Todos los Pedidos");
radioTodos.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajaNumeroPedido.setText("");
        cajaNumeroPedido.setEnabled(false);
    }
});
radioTodos.setBackground(new Color(255, 255, 255));
radioTodos.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioTodos.setHorizontalAlignment(SwingConstants.CENTER);
radioTodos.setBounds(250, 105, 190, 36);
radioTodos.setSelected(true);
groupBoxConsultarPedido.add(radioTodos);

losRadio = new ButtonGroup();
losRadio.add(radioUno);
losRadio.add(radioTodos);

JLabel etiquetaNumeroPedido = new JLabel("Pedido:");
etiquetaNumeroPedido.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNumeroPedido.setBounds(141, 170, 93, 30);
groupBoxConsultarPedido.add(etiquetaNumeroPedido);

cajaNumeroPedido = new JTextField();
cajaNumeroPedido.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaNumeroPedido.setEnabled(false);
cajaNumeroPedido.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Número de Pedido",cajaNumeroPedido);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNumeroPedido.setBounds(237, 170, 148, 36);
cajaNumeroPedido.setColumns(10);
cajaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 16));
groupBoxConsultarPedido.add(cajaNumeroPedido);

/** Formato y Funcionalidad de la Tabla para Visualizar los Pedidos */

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(50, 580, 1250, 330);
scrollTabla.setHorizontalScrollBar(null);
panelGeneral.add(scrollTabla);

tablaPedidos = new JTable();
modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
}

```

```

    };
    DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
    tablaPedidos = new JTable(modeloTabla);
    scrollTabla.setViewportViewView(tablaPedidos);
    tablaPedidos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
    tablaPedidos.setRowHeight(30);
    modeloTabla.addColumn("# PEDIDO");
    modeloTabla.addColumn("FECHA");
    modeloTabla.addColumn("CÓDIGO BARRAS");
    modeloTabla.addColumn("PRODUCTO");
    modeloTabla.addColumn("PROVEEDOR");
    modeloTabla.addColumn("CANTIDAD");
    modeloTabla.addColumn("PRECIO");
    modeloTabla.addColumn("TOTAL");
    modeloTabla.addColumn("ESTATUS");
    tablaPedidos.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
    tablaPedidos.getColumnModel().getColumn(0).setPreferredWidth(100);
    tablaPedidos.getColumnModel().getColumn(1).setPreferredWidth(110);
    tablaPedidos.getColumnModel().getColumn(2).setPreferredWidth(180);
    tablaPedidos.getColumnModel().getColumn(3).setPreferredWidth(200);
    tablaPedidos.getColumnModel().getColumn(4).setPreferredWidth(200);
    tablaPedidos.getColumnModel().getColumn(5).setPreferredWidth(85);
    tablaPedidos.getColumnModel().getColumn(6).setPreferredWidth(85);
    tablaPedidos.getColumnModel().getColumn(7).setPreferredWidth(85);
    tablaPedidos.getColumnModel().getColumn(8).setPreferredWidth(205);
    justificacion.setHorizontalAlignment(SwingConstants.CENTER);
    tablaPedidos.getColumnModel().getColumn(0).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(1).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(2).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(3).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(4).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(5).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(6).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(7).setCellRenderer(justificacion);
    tablaPedidos.getColumnModel().getColumn(8).setCellRenderer(justificacion);
    scrollTabla.setViewportViewView(tablaPedidos);

    /***** Formato y Funcionalidad del Botón "Ver" según Filtros *****/

    JButton botonVer = new JButton("Ver");
    botonVer.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            boolean radioTodosPedidos = true;

            while(modeloTabla.getRowCount()>0)
            {
                modeloTabla.removeRow(0);
            }

            if(radioUno.isSelected())
            {
                radioTodosPedidos = false;
            }

            List<Pedido> listaPedidos =
operacionesPedido.buscarPedidosFiltros((String)cajaListaProveedores.getSelectedItem(),
radioTodosPedidos, cajaNumeroPedido.getText());

            if(listaPedidos != null)
            {
                if(listaPedidos.size() != 0)
                {
                    for (Pedido pedido : listaPedidos)
                    {

```



```

        elProducto =
operacionesPedido.tracerProductoPedido(pedido.getProductoCodigoBarras());
        Object[] renglon =
{pedido.getNumeroPedido(),pedido.getFechaPedido(),pedido.getProductoCodigoBarras(),
elProducto.getNombreProducto(),(String)cajaListaProveedores.getSelectedItem(),
pedido.getCantidadSolicitada(),elProducto.getCostoPublico(),pedido.getTotalLiquidar(),
        pedido.getStatusPedido()};
        modeloTabla.addRow(renglon);
    }
    }
    else
    {
        JOptionPane.showMessageDialog(null,"No se Encontraron Pedidos
para esta Consulta.");
    }
}
});
botonVer.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonVer.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonVer.setForeground(new Color(0,0,0));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,1);
botonVer.setBorder(elBordeConsultar);
botonVer.setForeground(new Color(0,0,0));
botonVer.setBackground(new Color(216, 253, 241));
botonVer.setFont(new Font("Century", Font.PLAIN, 18));
botonVer.setBounds(230, 240, 80, 40);
groupBoxConsultarPedido.add(botonVer);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonSalir = new JButton(" Salir");
botonSalir.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuPedidos menuPedidos = new MenuPedidos();

        menuPedidos.setVisible(true);
        frmConsultarPedidos.setVisible(false);
    }
});
botonSalir.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonSalir.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonSalir.setForeground(new Color(0,0,0));
    }
});
botonSalir.setHorizontalAlignment(SwingConstants.CENTER);

```

```

        Border elBordeSalir = new LineBorder(Color.black,1);
        botonSalir.setBorder(elBordeSalir);
        botonSalir.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
        botonSalir.setForeground(new Color(0,0,0));
        botonSalir.setBackground(new Color(234,34,34));
        botonSalir.setFont(new Font("Century", Font.PLAIN, 20));
        botonSalir.setBounds(600, 980, 210, 50);
        panelGeneral.add(botonSalir);
    }
}

```

4.- ConsultarProducto.java

```

package Interfaces;

/** Esta Interfaz permite consultar los Productos por medio de los filtros de rango de
fechas, código de barras, nombre de producto o por nombre de proveedor */

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import java.awt.event.*;
import java.text.SimpleDateFormat;
import java.util.List;
import com.toedter.calendar.JDateChooser;
import hibernate.operaciones.OpProveedor;
import hibernate.operaciones.OpProducto;
import hibernate.beans.Producto;
import hibernate.beans.Proveedor;

public class ConsultarProducto
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmConsultarProducto;
    private JPanel panelGeneral;

    /******* CAJAS DE LAS FECHAS *****/

    private JDateChooser fechaDesde;
    private JDateChooser fechaHasta;

    /******* CAJAS DE BUSQUEDA AVANZADA CON SU LEYENDA *****/

    private JTextField cajaCodigoBarras;
    private JTextField cajaNombreProducto;
    private TextPrompt leyenda;
    @SuppressWarnings("rawtypes")
    private JComboBox cajaListaProveedores;

    /******* TABLA CONTENEDOR DE PRODUCTOS *****/

    private JTable tablaProductos;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
    private OpProducto operacionesProducto = new OpProducto();
    private List<Producto> listaProductos;

    /******* ELEMENTOS UTILIZADOS EN EL BOTÓN "Consultar" *****/

    boolean cajasVerificadas;

    /******* MAIN *****/
}

```

```

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                ConsultarProducto window = new ConsultarProducto();
                window.frmConsultarProducto.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmConsultarProducto.setVisible(visibilidad);
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if (fechaDesde.getDate() == null || fechaHasta.getDate() == null)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

public ConsultarProducto()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DE CONSULTAR PRODUCTO *****/

@SuppressWarnings({ "rawtypes", "unchecked", "serial" })
private void initialize()
{
    /***** Formato del JFrame, Scroll y Página Principal *****/

    frmConsultarProducto = new JFrame();
    frmConsultarProducto.setTitle("Consultar Productos");

    frmConsultarProducto.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProducto.class.getResource("/Imagenes/Logotipo.png")));
    frmConsultarProducto.setBounds(100, 100, 450, 300);
    frmConsultarProducto.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmConsultarProducto.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmConsultarProducto.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();

```

```

scrollGeneral.setBackground(new Color(175, 238, 238));
scrollGeneral.setBounds(0, 0, 1365, 715);
scrollGeneral.setHorizontalScrollBar(null);
frmConsultarProducto.getContentPane().add(scrollGeneral);

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1300));
panelGeneral.setBackground(new Color(255, 255, 255));
scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmConsultarProducto.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmConsultarProducto.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {

```

```

        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Productos");
etiquetaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        MenuProductos productos = new MenuProductos();

        productos.setVisible(true);
        frmConsultarProducto.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(0, 0, 255));
    }
});
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(326, 31, 108, 41);
panelGeneral.add(etiquetaProductos);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(423, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaModificarProducto = new JLabel("Consultar");
etiquetaModificarProducto.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaModificarProducto.setForeground(Color.BLUE);
etiquetaModificarProducto.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaModificarProducto.setBounds(400, 31, 200, 41);
panelGeneral.add(etiquetaModificarProducto);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));

```

```

etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox de Rango de Fechas para la Consulta *****/

JPanel groupBoxRangoFechas = new JPanel();
groupBoxRangoFechas.setBackground(new Color(243, 243, 243));
groupBoxRangoFechas.setBorder(new TitledBorder(null, "Rango de Consulta",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
groupBoxRangoFechas.setBounds(184, 300, 350, 171);
groupBoxRangoFechas.setLayout(null);
panelGeneral.add(groupBoxRangoFechas);

JLabel etiquetaFechaDesde = new JLabel("Desde:");
etiquetaFechaDesde.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFechaDesde.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFechaDesde.setBounds(47, 50, 89, 30);
groupBoxRangoFechas.add(etiquetaFechaDesde);

fechaDesde = new JDateChooser();
fechaDesde.setDateFormatString("yyyy-MM-dd");
fechaDesde.setForeground(new Color(0, 0, 0));
fechaDesde.setBounds(146, 50, 143, 27);
groupBoxRangoFechas.add(fechaDesde);

JLabel etiquetaFechaHasta = new JLabel("Hasta:");
etiquetaFechaHasta.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFechaHasta.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFechaHasta.setBounds(47, 100, 89, 30);
groupBoxRangoFechas.add(etiquetaFechaHasta);

fechaHasta = new JDateChooser();
fechaHasta.setDateFormatString("yyyy-MM-dd");
fechaHasta.setForeground(new Color(0, 0, 0));
fechaHasta.setBounds(146, 100, 143, 27);
groupBoxRangoFechas.add(fechaHasta);

/***** GroupBox de Busqueda Avanzada para la Consulta *****/

JPanel groupBoxConsultaAvanzada = new JPanel();
groupBoxConsultaAvanzada.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Consulta Avanzada",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxConsultaAvanzada.setBackground(new Color(243,243,243));
groupBoxConsultaAvanzada.setBounds(660, 250, 550, 270);
panelGeneral.add(groupBoxConsultaAvanzada);
groupBoxConsultaAvanzada.setLayout(null);

JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setBounds(70, 40, 188, 30);
groupBoxConsultaAvanzada.add(etiquetaCodigoBarras);
etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));

cajaCodigoBarras = new JTextField();
cajaCodigoBarras.addKeyListener(new KeyAdapter()
{
@Override
public void keyTyped(KeyEvent e)
{

```

```

        char caracter = e.getKeyChar();

        if(!cajaCodigoBarras.getText().equals(""))
        {
            cajaNombreProducto.setEnabled(false);
        }
        else
        {
            cajaNombreProducto.setEnabled(true);
        }

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCodigoBarras.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Código",cajaCodigoBarras);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCodigoBarras.setBounds(270, 40, 198, 36);
cajaCodigoBarras.setColumns(10);
cajaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 16));
groupBoxConsultaAvanzada.add(cajaCodigoBarras);

JLabel etiquetaNombreProducto = new JLabel("Nombre del Producto:");
etiquetaNombreProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProducto.setBounds(30, 120, 236, 30);
groupBoxConsultaAvanzada.add(etiquetaNombreProducto);

cajaNombreProducto = new JTextField();
cajaNombreProducto.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyTyped(KeyEvent e)
    {
        if(!cajaNombreProducto.getText().equals(""))
        {
            cajaCodigoBarras.setEnabled(false);
        }
        else
        {
            cajaCodigoBarras.setEnabled(true);
        }
    }
});
cajaNombreProducto.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Nombre",cajaNombreProducto);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNombreProducto.setBounds(270, 120, 250, 36);
cajaNombreProducto.setColumns(10);
cajaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 16));
groupBoxConsultaAvanzada.add(cajaNombreProducto);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds(120, 200, 131, 30);
groupBoxConsultaAvanzada.add(etiquetaProveedor);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));

```

```

        cajaListaProveedores.addItem("Seleccione un Proveedor");
        List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
        for(int i=0;i<losNombresProveedores.size();i++)
        {
            cajaListaProveedores.addItem(losNombresProveedores.get(i));
        }
        cajaListaProveedores.setBounds(270, 200, 221, 36);
        groupBoxConsultaAvanzada.add(cajaListaProveedores);

/*Formato y Funcionalidad de la Tabla para Visualizar los Productos */

        JScrollPane scrollTabla = new JScrollPane();
        scrollTabla.setBounds(5, 780, 1335, 330);
        scrollTabla.setHorizontalScrollBar(null);
        panelGeneral.add(scrollTabla);

        tablaProductos = new JTable();
        DefaultTableModel modeloTabla = new DefaultTableModel()
        {
            @Override
            public boolean isCellEditable(int row, int column)
            {
                return false;
            }
        };
        DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
        tablaProductos = new JTable(modeloTabla);
        scrollTabla.setViewportViewView(tablaProductos);
        tablaProductos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
        tablaProductos.setRowHeight(30);
        modeloTabla.addColumn("CÓDIGO BARRAS");
        modeloTabla.addColumn("PROVEEDOR");
        modeloTabla.addColumn("FECHA");
        modeloTabla.addColumn("NOMBRE");
        modeloTabla.addColumn("$ COMPRA");
        modeloTabla.addColumn("$ PUBLICO");
        modeloTabla.addColumn("PTS. ASIG.");
        modeloTabla.addColumn("PTS. COMP.");
        modeloTabla.addColumn("INGRESO");
        modeloTabla.addColumn("CLASIFICACIÓN");
        modeloTabla.addColumn("IVA");
        tablaProductos.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        tablaProductos.getColumnModel().getColumn(0).setPreferredWidth(150);
        tablaProductos.getColumnModel().getColumn(1).setPreferredWidth(150);
        tablaProductos.getColumnModel().getColumn(2).setPreferredWidth(100);
        tablaProductos.getColumnModel().getColumn(3).setPreferredWidth(250);
        tablaProductos.getColumnModel().getColumn(4).setPreferredWidth(85);
        tablaProductos.getColumnModel().getColumn(5).setPreferredWidth(85);
        tablaProductos.getColumnModel().getColumn(6).setPreferredWidth(85);
        tablaProductos.getColumnModel().getColumn(7).setPreferredWidth(85);
        tablaProductos.getColumnModel().getColumn(8).setPreferredWidth(85);
        tablaProductos.getColumnModel().getColumn(9).setPreferredWidth(197);
        tablaProductos.getColumnModel().getColumn(10).setPreferredWidth(60);
        justificacion.setHorizontalAlignment(SwingConstants.CENTER);
        tablaProductos.getColumnModel().getColumn(0).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(1).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(2).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(3).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(4).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(5).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(6).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(7).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(8).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(9).setCellRenderer(justificacion);
        tablaProductos.getColumnModel().getColumn(10).setCellRenderer(justificacion);
        scrollTabla.setViewportViewView(tablaProductos);

```



```

/**** Formato y Funcionalidad del Botón "Consultar" según Filtros *****/

JButton botonConsultar = new JButton("Consultar");
botonConsultar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        cajasVerificadas = camposLlenos();

        if(cajasVerificadas == false)
        {
            JOptionPane.showMessageDialog(null,"Por Favor, seleccione el rango de
fechas a consultar.");
        }
        else
        {
            java.util.Date laFechaDesde = fechaDesde.getDate();
            SimpleDateFormat cambiarFechaDesde = new
SimpleDateFormat(fechaDesde.getDateFormatString());
            java.util.Date laFechaHasta = fechaHasta.getDate();
            SimpleDateFormat cambiarFechaHasta = new
SimpleDateFormat(fechaHasta.getDateFormatString());

            while(modeloTabla.getRowCount()>0)
            {
                modeloTabla.removeRow(0);
            }

            Proveedor proveedor =
operacionesProveedor.buscarProveedor((String)cajaListaProveedores.getSelectedItem());

            if(proveedor != null)
            {
                listaProductos =
operacionesProducto.buscarProductosFiltros(String.valueOf(cambiarFechaDesde.format(laFecha
aDesde)),
String.valueOf(cambiarFechaHasta.format(laFechaHasta)), proveedor.getRfcProveedor(),
cajaCodigoBarras.getText(),
cajaNombreProducto.getText());
            }
            else
            {
                listaProductos =
operacionesProducto.buscarProductosFiltros(String.valueOf(cambiarFechaDesde.format(laFecha
aDesde)),
String.valueOf(cambiarFechaHasta.format(laFechaHasta)),
(String)cajaListaProveedores.getSelectedItem(),
cajaCodigoBarras.getText(),
cajaNombreProducto.getText());
            }

            for(Producto producto:listaProductos)
            {
                Object[] renglon =
{producto.getCodigoBarras(),producto.getProveedor(),producto.getFechaRegistro(),
producto.getNombreProducto(),producto.getCostoCompra(),producto.getCostoPublico(),
producto.getPuntosAsignados(),producto.getPuntosCompra(),producto.getCantidadIngreso(),
producto.getClasificacion(),producto.getIva()};
                modeloTabla.addRow(renglon);
            }

            cajaCodigoBarras.setText("");
            cajaNombreProducto.setText("");

```

```

        cajaListaProveedores.setSelectedIndex(0);
    }
}
});
botonConsultar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(0,0,0));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,1);
botonConsultar.setBorder(elBordeConsultar);
botonConsultar.setForeground(Color.BLACK);
botonConsultar.setBackground(new Color(216, 253, 241));
botonConsultar.setFont(new Font("Century", Font.PLAIN, 20));
botonConsultar.setBounds(582, 586, 200, 50);
panelGeneral.add(botonConsultar);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuProductos menuProductos = new MenuProductos();

        menuProductos.setVisible(true);
        frmConsultarProducto.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0,0,0));
    }
});
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
Border elBordeTerminar = new LineBorder(Color.black,1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(new Color(0,0,0));
botonTerminar.setBackground(new Color(234,34,34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(600, 1180, 210, 50);
panelGeneral.add(botonTerminar);
}
}
}

```

5.- ConsultarProveedor.java

```
package Interfaces;

/** Esta Interfaz permite consultar la información de un proveedor en especial o de todos
los proveedores **/

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import hibernate.operaciones.OpProveedor;
import java.awt.event.*;
import java.util.List;
import hibernate.beans.Proveedor;

public class ConsultarProveedor
{
    /*******CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmConsultarProveedor;
    private JPanel panelGeneral;

    /******* CAJA DE PROVEEDORES *****/

    @SuppressWarnings("rawtypes")
    JComboBox cajaListaProveedores;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
    private Proveedor elProveedor;

    /******* TABLA CONTENEDORA DE DATOS DE LOS PROVEEDORES *****/

    private JTable tablaProveedores;
    private List<Proveedor> listaProveedores;
    private String nombreProveedor;

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    ConsultarProveedor window = new ConsultarProveedor();
                    window.frmConsultarProveedor.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmConsultarProveedor.setVisible(visibilidad);
    }

    /******* INICIALIZACIÓN GENERAL DEL JFRAME *****/
}
```

```

public ConsultarProveedor()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DE CONSULTAR PRODUCTO *****/

@SuppressWarnings({ "rawtypes", "unchecked", "serial"})
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmConsultarProveedor = new JFrame();
    frmConsultarProveedor.getContentPane().setBackground(new Color(175, 238, 238));

frmConsultarProveedor.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProveedor.class.getResource("/Imágenes/Logotipo.png")));
    frmConsultarProveedor.setTitle("Consultar Proveedor");
    frmConsultarProveedor.setBounds(100, 100, 450, 300);
    frmConsultarProveedor.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmConsultarProveedor.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmConsultarProveedor.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1000));
    panelGeneral.setBackground(new Color(255,255,255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmConsultarProveedor.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    panelGeneral.setLayout(null);

    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    panelGeneral.add(etiquetaPaginaPrincipal);

```

```

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmConsultarProveedor.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Proveedores");
etiquetaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        MenuProveedores proveedores = new MenuProveedores();

        proveedores.setVisible(true);
        frmConsultarProveedor.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(0, 0, 255));
    }
});
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);

```

```

etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(330, 31, 115, 41);
panelGeneral.add(etiquetaProductos);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(437, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaConsultarProveedores = new JLabel("Consultar");
etiquetaConsultarProveedores.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaConsultarProveedores.setForeground(Color.BLUE);
etiquetaConsultarProveedores.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaConsultarProveedores.setBounds(400, 32, 219, 41);
panelGeneral.add(etiquetaConsultarProveedores);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox con Lista Desplegable de los Proveedores *****/

JPanel groupBoxConsultarProveedor = new JPanel();
groupBoxConsultarProveedor.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Consultar",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxConsultarProveedor.setBackground(new Color(255,255,255));
groupBoxConsultarProveedor.setBounds(423, 185, 525, 133);
groupBoxConsultarProveedor.setLayout(null);
panelGeneral.add(groupBoxConsultarProveedor);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
cajaListaProveedores.addItem("Todos los Proveedores");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        nombreProveedor = (String)cajaListaProveedores.getSelectedItem();
    }
});
cajaListaProveedores.setBounds(219, 47, 221, 36);
groupBoxConsultarProveedor.add(cajaListaProveedores);

JLabel etiquetaProveedor = new JLabel("Proveedor:");

```

```

etiquetaProveedor.setBounds(79, 47, 131, 30);
groupBoxConsultarProveedor.add(etiquetaProveedor);
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(80, 530, 1188, 319);
scrollTabla.setHorizontalScrollBar(null);
panelGeneral.add(scrollTabla);

DefaultTableModel modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
tablaProveedores = new JTable(modeloTabla);
scrollTabla.setViewportViewView(tablaProveedores);
tablaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
tablaProveedores.setRowHeight(80);
modeloTabla.addColumn("RFC");
modeloTabla.addColumn("FECHA REGISTRO");
modeloTabla.addColumn("NOMBRE");
modeloTabla.addColumn("DIRECCIÓN");
modeloTabla.addColumn("TELÉFONO");
modeloTabla.addColumn("CORREO");
modeloTabla.addColumn("CONTACTO");
tablaProveedores.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
tablaProveedores.getColumnModel().getColumn(0).setPreferredWidth(110);
tablaProveedores.getColumnModel().getColumn(1).setPreferredWidth(120);
tablaProveedores.getColumnModel().getColumn(2).setPreferredWidth(250);
tablaProveedores.getColumnModel().getColumn(3).setPreferredWidth(215);
tablaProveedores.getColumnModel().getColumn(4).setPreferredWidth(130);
tablaProveedores.getColumnModel().getColumn(5).setPreferredWidth(200);
tablaProveedores.getColumnModel().getColumn(6).setPreferredWidth(160);
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
tablaProveedores.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaProveedores.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaProveedores.getColumnModel().getColumn(2).setCellRenderer(justificacion);
tablaProveedores.getColumnModel().getColumn(3).setCellRenderer(justificacion);
tablaProveedores.getColumnModel().getColumn(4).setCellRenderer(justificacion);
tablaProveedores.getColumnModel().getColumn(5).setCellRenderer(justificacion);
tablaProveedores.getColumnModel().getColumn(6).setCellRenderer(justificacion);

/**** Formato y Funcionalidad del Botón "Consultar" según Filtros ****/

JButton botonConsultar = new JButton("Consultar");
botonConsultar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        if(cajaListaProveedores.getSelectedItem().equals("Seleccione un
Proveedor"))
        {
            JOptionPane.showMessageDialog(null, "Seleccione un Proveedor");
        }
        else
        {
            while(modeloTabla.getRowCount()>0)
            {
                modeloTabla.removeRow(0);
            }
            if(nombreProveedor.equals("Todos los Proveedores"))
            {
                listaProveedores = operacionesProveedor.tracerDatosProveedores();
            }
        }
    }
});

```

```

        for(int i=0;i<listaProveedores.size();i++)
        {
            elProveedor = listaProveedores.get(i);
            Object[] renglon =
{elProveedor.getRfcProveedor(),elProveedor.getFechaRegistro(),elProveedor.getNombreProvee
dor(),elProveedor.getDireccion(),
elProveedor.getTelefono(),elProveedor.getCorreo(),elProveedor.getNombreContacto()};
            modeloTabla.addRow(renglon);
        }
    }
    else
    {
        elProveedor =
operacionesProveedor.buscarProveedor(nombreProveedor);
        Object[] renglon =
{elProveedor.getRfcProveedor(),elProveedor.getFechaRegistro(),elProveedor.getNombreProvee
dor(),elProveedor.getDireccion(),
elProveedor.getTelefono(),elProveedor.getCorreo(),elProveedor.getNombreContacto()};
        modeloTabla.addRow(renglon);
    }
}
});
botonConsultar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(0,0,0));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,1);
botonConsultar.setBorder(elBordeConsultar);
botonConsultar.setForeground(new Color(0,0,0));
botonConsultar.setBackground(new Color(216, 253, 241));
botonConsultar.setFont(new Font("Century", Font.PLAIN, 18));
botonConsultar.setBounds(600, 350, 150, 50);
panelGeneral.add(botonConsultar);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuProveedores menuProveedores = new MenuProveedores();

        menuProveedores.setVisible(true);
        frmConsultarProveedor.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)

```



```

        {
            botonTerminar.setForeground(new Color(0,0,0));
        }
    });
    Border elBordeTerminar = new LineBorder(Color.black,1);
    botonTerminar.setBorder(elBordeTerminar);
    botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
    botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imágenes/Tache.png")));
    botonTerminar.setForeground(new Color(0,0,0));
    botonTerminar.setBackground(new Color(234,34,34));
    botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
    botonTerminar.setBounds(600, 900, 210, 50);
    panelGeneral.add(botonTerminar);
}
}
}

```

6.- ConsultarVentas.java

```

package Interfaces;

/** Esta Interfaz permite consultar las ventas registradas en el Sistema utilizando filtros
para buscar por rango de fechas unicamente o también filtrar por vendedor y/o tipo de pago
**/

import java.awt.*;
import java.awt.event.*;
import java.text.SimpleDateFormat;
import java.util.List;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Venta;
import hibernate.operaciones.OpVenta;

public class ConsultarVentas
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmConsultarVentas;
    private JPanel panelGeneral;

    /******* CAJAS DE LAS FECHAS *****/

    private JDateChooser fechaDesde;
    private JDateChooser fechaHasta;

    /******* CAJAS DE BUSQUEDA AVANZADA CON SU LEYENDA *****/

    private JTextField cajaVendedor;
    private TextPrompt leyenda;
    @SuppressWarnings("rawtypes")
    private JComboBox cajaFormaPago;

    /******* TABLA CONTENEDOR DE VENTAS *****/

    private JTable tablaVentas;
    private DefaultTableModel modeloTabla;

    private OpVenta operacionesVenta = new OpVenta();

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    ConsultarVentas window = new ConsultarVentas();

```

```

        window.frmConsultarVentas.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
});
}

public ConsultarVentas()
{
    initialize();
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if (fechaDesde.getDate() == null || fechaHasta.getDate() == null)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmConsultarVentas.setVisible(visibilidad);
}

/***** DISEÑO DE LA PÁGINA DE CONSULTAR VENTAS *****/

@SuppressWarnings({ "rawtypes", "unchecked", "serial" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmConsultarVentas = new JFrame();
    frmConsultarVentas.setTitle("Consultar Ventas");

    frmConsultarVentas.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProducto.class.getResource("/Imagenes/Logotipo.png")));
    frmConsultarVentas.setBounds(100, 100, 450, 300);
    frmConsultarVentas.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmConsultarVentas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmConsultarVentas.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmConsultarVentas.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1300)); //PARA
    panelGeneral.setBackground(new Color(255, 255, 255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

```

```

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmConsultarVentas.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(220, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Ventas");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuVentas ventas = new MenuVentas();

        ventas.setVisible(true);
        frmConsultarVentas.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(240, 32, 65, 41);
panelGeneral.add(etiquetaAlmacen);

```

```

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(300, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Consultar");
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(270, 31, 200, 41);
panelGeneral.add(etiquetaProductos);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox de Rango de Fechas para la Consulta *****/

JPanel groupBoxRangoFechas = new JPanel();
groupBoxRangoFechas.setBackground(new Color(243, 243, 243));
groupBoxRangoFechas.setBorder(new TitledBorder(null, "Rango de Consulta",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
groupBoxRangoFechas.setBounds(184, 300, 350, 171);
groupBoxRangoFechas.setLayout(null);
panelGeneral.add(groupBoxRangoFechas);

JLabel etiquetaFechaDesde = new JLabel("Desde:");
etiquetaFechaDesde.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFechaDesde.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFechaDesde.setBounds(47, 50, 89, 30);
groupBoxRangoFechas.add(etiquetaFechaDesde);

fechaDesde = new JDateChooser();
fechaDesde.setDateFormatString("yyyy-MM-dd");
fechaDesde.setForeground(new Color(0, 0, 0));
fechaDesde.setBounds(146, 50, 143, 27);
groupBoxRangoFechas.add(fechaDesde);

JLabel etiquetaFechaHasta = new JLabel("Hasta:");
etiquetaFechaHasta.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFechaHasta.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFechaHasta.setBounds(47, 100, 89, 30);
groupBoxRangoFechas.add(etiquetaFechaHasta);

fechaHasta = new JDateChooser();
fechaHasta.setDateFormatString("yyyy-MM-dd");
fechaHasta.setForeground(new Color(0, 0, 0));
fechaHasta.setBounds(146, 100, 143, 27);
groupBoxRangoFechas.add(fechaHasta);

/***** GroupBox de Busqueda Avanzada para la Consulta *****/

JPanel groupBoxConsultaAvanzada = new JPanel();

```

```

        groupBoxConsultaAvanzada.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Consulta Avanzada",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
        groupBoxConsultaAvanzada.setBackground(new Color(243,243,243));
        groupBoxConsultaAvanzada.setBounds(660, 275, 550, 205);
        panelGeneral.add(groupBoxConsultaAvanzada);
        groupBoxConsultaAvanzada.setLayout(null);

        JLabel etiquetaFormaPago = new JLabel("Forma de Pago:");
        etiquetaFormaPago.setHorizontalAlignment(SwingConstants.RIGHT);
        etiquetaFormaPago.setFont(new Font("Cambria Math", Font.PLAIN, 25));
        etiquetaFormaPago.setBounds(52, 127, 183, 30);
        groupBoxConsultaAvanzada.add(etiquetaFormaPago);

        cajaFormaPago = new JComboBox();
        cajaFormaPago.setFont(new Font("Cambria Math", Font.PLAIN, 15));
        cajaFormaPago.setModel(new DefaultComboBoxModel(new String[] {}));
        cajaFormaPago.addItem("Seleccione una Opción");
        cajaFormaPago.addItem("Efectivo");
        cajaFormaPago.addItem("Puntos");
        cajaFormaPago.setBounds(243, 125, 221, 36);
        groupBoxConsultaAvanzada.add(cajaFormaPago);

        JLabel etiquetaVendedor = new JLabel("Vendedor:");
        etiquetaVendedor.setHorizontalAlignment(SwingConstants.RIGHT);
        etiquetaVendedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
        etiquetaVendedor.setBounds(68, 45, 137, 30);
        groupBoxConsultaAvanzada.add(etiquetaVendedor);

        cajaVendedor = new JTextField();
        cajaVendedor.setHorizontalAlignment(SwingConstants.CENTER);
        leyenda = new TextPrompt("Ingresar Clave Empleado",cajaVendedor);
        leyenda.setHorizontalAlignment(SwingConstants.CENTER);
        leyenda.changeAlpha(0.50f);
        leyenda.changeStyle(Font.ITALIC);
        cajaVendedor.setBounds(215, 42, 250, 36);
        cajaVendedor.setColumns(10);
        cajaVendedor.setFont(new Font("Cambria Math", Font.PLAIN, 16));
        groupBoxConsultaAvanzada.add(cajaVendedor);

/** Formato y Funcionalidad de la Tabla para Visualizar las Ventas **/

        JScrollPane scrollTabla = new JScrollPane();
        scrollTabla.setBounds(220, 780, 900, 330);
        scrollTabla.setHorizontalScrollBar(null);
        panelGeneral.add(scrollTabla);

        tablaVentas = new JTable();
        modeloTabla = new DefaultTableModel()
        {
            @Override
            public boolean isCellEditable(int row, int column)
            {
                return false;
            }
        };
        DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
        tablaVentas = new JTable(modeloTabla);
        scrollTabla.setViewportView(tablaVentas);
        tablaVentas.setFont(new Font("Cambria Math", Font.PLAIN, 15));
        tablaVentas.setRowHeight(30);
        modeloTabla.addColumn("TRANSACCIÓN");
        modeloTabla.addColumn("FECHA");
        modeloTabla.addColumn("VENDEDOR");
        modeloTabla.addColumn("TIPO DE PAGO");
        modeloTabla.addColumn("IMPORTE EFECTIVO");
        modeloTabla.addColumn("PUNTOS ACUMULADOS");
        tablaVentas.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

```

```

tablaVentas.getColumnModel().getColumn(0).setPreferredWidth(110);
tablaVentas.getColumnModel().getColumn(1).setPreferredWidth(100);
tablaVentas.getColumnModel().getColumn(2).setPreferredWidth(250);
tablaVentas.getColumnModel().getColumn(3).setPreferredWidth(140);
tablaVentas.getColumnModel().getColumn(4).setPreferredWidth(150);
tablaVentas.getColumnModel().getColumn(5).setPreferredWidth(150);
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
tablaVentas.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(2).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(3).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(4).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(5).setCellRenderer(justificacion);
scrollTabla.setViewportView(tablaVentas);

/**** Formato y Funcionalidad del Botón "Consultar" según Filtros ****/

JButton botonConsultar = new JButton("Consultar");
botonConsultar.addActionListener(new ActionListener()
{
    boolean cajasVerificadas;

    public void actionPerformed(ActionEvent e)
    {
        cajasVerificadas = camposLlenos();

        if(cajasVerificadas == false)
        {
            JOptionPane.showMessageDialog(null, "Por Favor, selecciona el rango de
fechas a consultar.");
        }
        else
        {
            java.util.Date laFechaDesde = fechaDesde.getDate();
            SimpleDateFormat cambiarFechaDesde = new
SimpleDateFormat(fechaDesde.getDateFormatString());
            java.util.Date laFechaHasta = fechaHasta.getDate();
            SimpleDateFormat cambiarFechaHasta = new
SimpleDateFormat(fechaHasta.getDateFormatString());

            String desde =
String.valueOf(cambiarFechaDesde.format(laFechaDesde));
            String hasta =
String.valueOf(cambiarFechaHasta.format(laFechaHasta));

            while(modeloTabla.getRowCount()>0)
            {
                modeloTabla.removeRow(0);
            }

            List<Venta> listaVentas =
operacionesVenta.buscarProductosFiltros(desde, hasta, cajaVendedor.getText(),
(String)cajaFormaPago.getSelectedItem());
            if(listaVentas.size() != 0)
            {
                for (Venta venta : listaVentas)
                {
                    Object[] renglon =
{venta.getNumeroTransaccion(), venta.getFechaVenta(), venta.getEmpleadoRFC(),
venta.getTipoPago(), venta.getTotal(), venta.getPuntosAcumulados()};
                    modeloTabla.addRow(renglon);
                }
            }
            else
            {

```

```

        JOptionPane.showMessageDialog(null, "No se Encontraron Ventas para
esta Busqueda.");
    }
}
});
botonConsultar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(0,0,0));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,1);
botonConsultar.setBorder(elBordeConsultar);
botonConsultar.setForeground(Color.BLACK);
botonConsultar.setBackground(new Color(216, 253, 241));
botonConsultar.setFont(new Font("Century", Font.PLAIN, 20));
botonConsultar.setBounds(582, 586, 200, 50);
panelGeneral.add(botonConsultar);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuVentas menuVentas = new MenuVentas();

        menuVentas.setVisible(true);
        frmConsultarVentas.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0,0,0));
    }
});
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
Border elBordeTerminar = new LineBorder(Color.black,1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(new Color(0,0,0));
botonTerminar.setBackground(new Color(234,34,34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(600, 1180, 210, 50);
panelGeneral.add(botonTerminar);
}
}
}

```

7.- Login.java

```
package Interfaces;

/** Esta Interfaz permite la autenticación de un Empleado para poder acceder a las
funcionalidades del Sistema **/

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import hibernate.beans.Empleado;
import hibernate.operaciones.OpEmpleado;
import java.awt.event.*;

public class Login
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmLogin;

    /******* CAJAS Y LEYENDA PARA USUARIO Y PASSWORD *****/

    private JTextField cajaUsuario;
    private JPasswordField cajaClave;
    private TextPrompt leyenda;

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    Login window = new Login();
                    window.frmLogin.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmLogin.setVisible(visibilidad);
    }

    public Login()
    {
        initialize();
        frmLogin.setLocationRelativeTo(null);
    }

    /******* DISEÑO DE LA PÁGINA LOGIN *****/

    private void initialize()
    {
        /*----- Formato del JFrame -----*/

        frmLogin = new JFrame();
        frmLogin.setFont(new Font("Constantia", Font.PLAIN, 20));
        frmLogin.setResizable(false);
    }
}
```



```

frmLogin.setIconImage(Toolkit.getDefaultToolkit().getImage(Login.class.getResource("/Imag
enes/Logotipo.png")));
    frmLogin.setTitle("Login");
    frmLogin.getContentPane().setBackground(new Color(255, 255, 255));
    frmLogin.setBounds(100, 100, 446, 550);
    frmLogin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmLogin.getContentPane().setLayout(null);

/*----- Nombre del Negocio y Logotipo -----*/

    JLabel etiquetaNegocio = new JLabel("Mi Negocio");
    etiquetaNegocio.setForeground(new Color(19, 60, 245));
    etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
    etiquetaNegocio.setBounds(194, 78, 202, 56);
    frmLogin.getContentPane().add(etiquetaNegocio);

    JLabel etiquetaLogotipo = new JLabel("");
    etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
    etiquetaLogotipo.setBackground(new Color(175, 238, 238));
    etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaLogotipo.setBounds(55, 55, 129, 104);
    frmLogin.getContentPane().add(etiquetaLogotipo);

/*----- Formato de Etiquetas y Cajas de Texto -----*/

    JLabel etiquetaUsuario = new JLabel("Usuario:");
    etiquetaUsuario.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaUsuario.setFont(new Font("Bookman Old Style", Font.PLAIN, 18));
    etiquetaUsuario.setBounds(86, 232, 77, 14);
    frmLogin.getContentPane().add(etiquetaUsuario);

    JLabel etiquetaClave = new JLabel("Clave:");
    etiquetaClave.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaClave.setFont(new Font("Bookman Old Style", Font.PLAIN, 18));
    etiquetaClave.setBounds(109, 290, 53, 14);
    frmLogin.getContentPane().add(etiquetaClave);

    cajaUsuario = new JTextField();
    leyenda = new TextPrompt("Escriba su Usuario",cajaUsuario);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaUsuario.setForeground(new Color(0, 0, 0));
    cajaUsuario.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    cajaUsuario.setBounds(171, 225, 159, 29);
    cajaUsuario.setColumns(10);
    frmLogin.getContentPane().add(cajaUsuario);

    cajaClave = new JPasswordField();
    leyenda = new TextPrompt("Escriba su Password",cajaClave);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaClave.setEchoChar('*');
    cajaClave.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    cajaClave.setBounds(172, 285, 158, 29);
    frmLogin.getContentPane().add(cajaClave);

/*----- Formato y Funcionalidad del Botón "Ingresar" -----*/

    JButton botonIngresar = new JButton("  Ingresar");
    botonIngresar.addMouseListener(new MouseAdapter ()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonIngresar.setForeground(new Color(12,2,235));
        }
    }

```

```

@Override
public void mouseExited(MouseEvent e)
{
    botonIngresar.setForeground(new Color(0, 0, 0));
}
@Override
public void mouseClicked(MouseEvent e)
{
    botonIngresar.setForeground(new Color(0,0,0));
    botonIngresar.setBackground(new Color(95, 255, 95));
}
});
botonIngresar.setForeground(new Color(0, 0, 0));
botonIngresar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        OpEmpleado operacionesEmpleado = new OpEmpleado();

        String usuario = cajaUsuario.getText();
        String clave = String.valueOf(cajaClave.getPassword());

        Empleado empleado = operacionesEmpleado.buscarEmpleado(usuario, clave);

        if(empleado != null)
        {
            MenuPrincipal principal = new MenuPrincipal();
            principal.setVisible(true);
            frmLogin.setVisible(false);
        }
    }
});
Border elBordeIngresar = new LineBorder(Color.black,2);
botonIngresar.setBorder(elBordeIngresar);
botonIngresar.setHorizontalAlignment(SwingConstants.CENTER);
botonIngresar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/IngresarUsuario.png")));
botonIngresar.setBackground(new Color(95, 255, 95));
botonIngresar.setFont(new Font("Century", Font.PLAIN, 18));
botonIngresar.setBounds(140, 390, 180, 55);
frmLogin.getContentPane().add(botonIngresar);
}
}

```

8.- MenuAlmacen.java

```

package Interfaces;

/** Esta Interfaz muestra las 5 secciones en las que se divide el Almacén: "Productos",
"Proveedores", "Pedidos", "Devoluciones" y "Cuentas por Pagar", a las cuales puede acceder
un Empleado que se encuentra autenticado en el Sistema */

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class MenuAlmacen
{
    /***** CONTENEDOR JFrame BASE *****/

    private JFrame frmAlmacen;

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()

```

```

    {
        public void run()
        {
            try
            {
                MenuAlmacen window = new MenuAlmacen();
                window.frmAlmacen.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmAlmacen.setVisible(visibilidad);
}

public MenuAlmacen()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DEL MENÚ ALMACÉN *****/

private void initialize()
{
    /*----- Formato del JFrame -----*/

    frmAlmacen = new JFrame();
    frmAlmacen.getContentPane().setBackground(new Color(255, 255, 255));

    frmAlmacen.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuAlmacen.class.getResource("/Imagenes/Logotipo.png")));
    frmAlmacen.setTitle("Almacen");
    frmAlmacen.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmAlmacen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmAlmacen.getContentPane().setLayout(null);

    /*----- Formato y Funcionalidad de las Migajas -----*/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmAlmacen.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
}

```

```

etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
frmAlmacen.getContentPane().add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal = new JLabel("/");
etiquetaDiagonal.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal.setForeground(new Color(0, 0, 255));
etiquetaDiagonal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal.setBounds(214, 32, 22, 39);
frmAlmacen.getContentPane().add(etiquetaDiagonal);

JLabel etiquetaAlmacen = new JLabel("Almacen");
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(237, 31, 85, 41);
frmAlmacen.getContentPane().add(etiquetaAlmacen);

/*----- Nombre del Negocio y Logotipo -----*/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
frmAlmacen.getContentPane().add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imágenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
frmAlmacen.getContentPane().add(etiquetaLogotipo);

/*----- Formato de las 5 Imágenes del Menú -----*/

JLabel imagenProductos = new JLabel("");
imagenProductos.setHorizontalAlignment(SwingConstants.CENTER);
imagenProductos.setIcon(new
ImageIcon(MenuAlmacen.class.getResource("/Imágenes/MA_Productos.png")));
imagenProductos.setBounds(141, 224, 181, 138);
frmAlmacen.getContentPane().add(imagenProductos);

JLabel imagenProveedores = new JLabel("");
imagenProveedores.setIcon(new
ImageIcon(MenuAlmacen.class.getResource("/Imágenes/MA_Proveedores.png")));
imagenProveedores.setHorizontalAlignment(SwingConstants.CENTER);
imagenProveedores.setBounds(612, 220, 157, 143);
frmAlmacen.getContentPane().add(imagenProveedores);

JLabel imagenDevoluciones = new JLabel("");
imagenDevoluciones.setIcon(new
ImageIcon(MenuAlmacen.class.getResource("/Imágenes/MA_Devoluciones.png")));
imagenDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
imagenDevoluciones.setBounds(1085, 203, 163, 159);
frmAlmacen.getContentPane().add(imagenDevoluciones);

JLabel imagenPedidos = new JLabel("");
imagenPedidos.setIcon(new
ImageIcon(MenuAlmacen.class.getResource("/Imágenes/MA_Pedidos.png")));
imagenPedidos.setHorizontalAlignment(SwingConstants.CENTER);
imagenPedidos.setBounds(374, 457, 199, 186);
frmAlmacen.getContentPane().add(imagenPedidos);

JLabel imagenCuentasPorPagar = new JLabel("");
imagenCuentasPorPagar.setIcon(new
ImageIcon(MenuAlmacen.class.getResource("/Imágenes/MA_CuentasPorPagar.png")));

```

```

imagenCuentasPorPagar.setHorizontalAlignment(SwingConstants.CENTER);
imagenCuentasPorPagar.setBounds(860, 468, 148, 174);
frmAlmacen.getContentPane().add(imagenCuentasPorPagar);

/*----- Formato y Funcionalidad de la Etiqueta de Cada Imágen -----*/

JLabel etiquetaProductos = new JLabel("Productos");
etiquetaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuProductos productos = new MenuProductos();

        productos.setVisible(true);
        frmAlmacen.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(204, 51, 51));
    }
});
etiquetaProductos.setForeground(new Color(204, 51, 51));
etiquetaProductos.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setBounds(155, 363, 147, 40);
frmAlmacen.getContentPane().add(etiquetaProductos);

JLabel etiquetaProveedores = new JLabel("Proveedores");
etiquetaProveedores.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuProveedores proveedores = new MenuProveedores();

        proveedores.setVisible(true);
        frmAlmacen.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProveedores.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProveedores.setForeground(new Color(204, 51, 51));
    }
});
etiquetaProveedores.setForeground(new Color(204, 51, 51));
etiquetaProveedores.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaProveedores.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProveedores.setBounds(611, 366, 163, 34);
frmAlmacen.getContentPane().add(etiquetaProveedores);

JLabel etiquetaDevoluciones = new JLabel("Devoluciones");
etiquetaDevoluciones.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {

```

```

        MenuDevoluciones devoluciones = new MenuDevoluciones();

        devoluciones.setVisible(true);
        frmAlmacen.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(204, 51, 51));
    }
});
etiquetaDevoluciones.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaDevoluciones.setForeground(new Color(204, 51, 51));
etiquetaDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDevoluciones.setBounds(1074, 363, 177, 40);
frmAlmacen.getContentPane().add(etiquetaDevoluciones);

JLabel etiquetaPedidos = new JLabel("Pedidos");
etiquetaPedidos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPedidos pedidos = new MenuPedidos();

        pedidos.setVisible(true);
        frmAlmacen.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPedidos.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPedidos.setForeground(new Color(204, 51, 51));
    }
});
etiquetaPedidos.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaPedidos.setForeground(new Color(204, 51, 51));
etiquetaPedidos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPedidos.setBounds(412, 642, 122, 32);
frmAlmacen.getContentPane().add(etiquetaPedidos);

JLabel etiquetaCuentasPorPagar = new JLabel("Cuentas Por Pagar");
etiquetaCuentasPorPagar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        ConsultarCuentasPorPagar cuentas = new ConsultarCuentasPorPagar();

        cuentas.setVisible(true);
        frmAlmacen.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaCuentasPorPagar.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)

```

```

        {
            etiquetaCuentasPorPagar.setForeground(new Color(204, 51, 51));
        }
    });
    etiquetaCuentasPorPagar.setFont(new Font("Times New Roman", Font.PLAIN, 30));
    etiquetaCuentasPorPagar.setForeground(new Color(204, 51, 51));
    etiquetaCuentasPorPagar.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaCuentasPorPagar.setBounds(811, 638, 247, 40);
    frmAlmacen.getContentPane().add(etiquetaCuentasPorPagar);
}
}

```

9.- MenuClientes.java

```

package Interfaces;

/** Esta Interfaz muestra las 2 secciones en las que se divide la parte de Clientes:
"Registrar" y "Modificar", a las cuales puede acceder un Empleado que se encuentra
autenticado en el Sistema */

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class MenuClientes
{
    /******* CONTENEDOR JFRAME BASE *****/

    private JFrame frmClientes;

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    MenuClientes window = new MenuClientes();
                    window.frmClientes.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmClientes.setVisible(visibilidad);
    }

    /******* INICIALIZACIÓN GENERAL DEL JFRAME *****/

    public MenuClientes()
    {
        initialize();
    }

    /******* DISEÑO DE LA PÁGINA DEL MENÚ PARA CLIENTES *****/

    private void initialize()
    {

```

```

/*----- Formato del JFrame -----*/

frmClientes = new JFrame();
frmClientes.getContentPane().setBackground(new Color(255, 255, 255));

frmClientes.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuClientes.class.getResource("/Imagenes/Logotipo.png")));
frmClientes.setTitle("Clientes");
frmClientes.setExtendedState(JFrame.MAXIMIZED_BOTH);
frmClientes.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frmClientes.getContentPane().setLayout(null);

/*----- Formato y Funcionalidad de las Migajas -----*/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9lgina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmClientes.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
frmClientes.getContentPane().add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal = new JLabel("/");
etiquetaDiagonal.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal.setForeground(new Color(0, 0, 255));
etiquetaDiagonal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal.setBounds(214, 32, 22, 39);
frmClientes.getContentPane().add(etiquetaDiagonal);

JLabel etiquetaClientes = new JLabel("Clientes");
etiquetaClientes.setForeground(new Color(0, 0, 255));
etiquetaClientes.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaClientes.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaClientes.setBounds(237, 31, 85, 41);
frmClientes.getContentPane().add(etiquetaClientes);

/*----- Nombre del Negocio y Logotipo -----*/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
frmClientes.getContentPane().add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));

```



```

etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
frmClientes.getContentPane().add(etiquetaLogotipo);

/*----- Formato de las 2 Imagenes del Menú -----*/

JLabel imagenRegistro = new JLabel("");
imagenRegistro.setHorizontalAlignment(SwingConstants.CENTER);
imagenRegistro.setIcon(new
ImageIcon(MenuClientes.class.getResource("/Imagenes/RegistroCliente.png")));
imagenRegistro.setBounds(300, 280, 210, 200);
frmClientes.getContentPane().add(imagenRegistro);

JLabel imagenModificar = new JLabel("");
imagenModificar.setIcon(new
ImageIcon(MenuClientes.class.getResource("/Imagenes/ModificarCliente.png")));
imagenModificar.setHorizontalAlignment(SwingConstants.CENTER);
imagenModificar.setBounds(850, 275, 210, 200);
frmClientes.getContentPane().add(imagenModificar);

/*----- Formato y Funcionalidad de la Etiqueta de Cada Imágen -----*/

JLabel etiquetaRegistrar = new JLabel("Registrar");
etiquetaRegistrar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        RegistrarCliente registrarCliente = new RegistrarCliente();

        registrarCliente.setVisible(true);
        frmClientes.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaRegistrar.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaRegistrar.setForeground(new Color(204, 51, 51));
    }
});
etiquetaRegistrar.setForeground(new Color(204, 51, 51));
etiquetaRegistrar.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaRegistrar.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaRegistrar.setBounds(330, 500, 147, 40);
frmClientes.getContentPane().add(etiquetaRegistrar);

JLabel etiquetaModificar = new JLabel("Modificar");
etiquetaModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        ModificarCliente modificarCliente = new ModificarCliente();

        modificarCliente.setVisible(true);
        frmClientes.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaModificar.setForeground(new Color(0, 153, 102));
    }
    @Override

```

```

        public void mouseExited(MouseEvent e)
        {
            etiquetaModificar.setForeground(new Color(204, 51, 51));
        }
    });
    etiquetaModificar.setForeground(new Color(204, 51, 51));
    etiquetaModificar.setFont(new Font("Times New Roman", Font.PLAIN, 30));
    etiquetaModificar.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaModificar.setBounds(875, 500, 163, 34);
    frmClientes.getContentPane().add(etiquetaModificar);
}
}

```

10.- MenuDevoluciones.java

```

package Interfaces;

/** Esta Interfaz muestra las 3 operaciones que se pueden realizar para una Devolución:
"Registrar", "Modificar" y "Consultar" */

import java.awt.*;
import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.LineBorder;
import java.awt.event.*;

public class MenuDevoluciones
{
    /******* CONTENEDOR JFRAME BASE *****/

    private JFrame frmMenuDevoluciones;

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    MenuDevoluciones window = new MenuDevoluciones();
                    window.frmMenuDevoluciones.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmMenuDevoluciones.setVisible(visibilidad);
    }

    /******* INICIALIZACIÓN GENERAL DEL JFRAME *****/

    public MenuDevoluciones()
    {
        initialize();
    }
}

```

```

/***** DISEÑO DE LA PÁGINA DEL MENÚ PARA DEVOLUCIONES *****/

private void initialize()
{
/*----- Formato del JFrame -----*/

    frmMenuDevoluciones = new JFrame();
    frmMenuDevoluciones.getContentPane().setBackground(new Color(255,255,255));

frmMenuDevoluciones.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuAlmacen.class.g
etResource("/Imagenes/Logotipo.png")));
    frmMenuDevoluciones.setTitle("Devoluciones");
    frmMenuDevoluciones.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmMenuDevoluciones.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmMenuDevoluciones.getContentPane().setLayout(null);

/*----- Formato y Funcionalidad de las Migajas -----*/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmMenuDevoluciones.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    frmMenuDevoluciones.getContentPane().add(etiquetaPaginaPrincipal);

    JLabel etiquetaDiagonal2 = new JLabel("/");
    etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal2.setForeground(new Color(0, 0, 255));
    etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal2.setBounds(313, 32, 22, 39);
    frmMenuDevoluciones.getContentPane().add(etiquetaDiagonal2);

    JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
    etiquetaAlmacen.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuAlmacen almacen = new MenuAlmacen();

            almacen.setVisible(true);
            frmMenuDevoluciones.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)

```

```

        {
            etiquetaAlmacen.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaAlmacen.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaAlmacen.setBounds(233, 32, 85, 41);
    frmMenuDevoluciones.getContentPane().add(etiquetaAlmacen);

    JLabel etiquetaDiagonal1 = new JLabel("/");
    etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal1.setForeground(Color.BLUE);
    etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal1.setBounds(214, 32, 22, 39);
    frmMenuDevoluciones.getContentPane().add(etiquetaDiagonal1);

    JLabel etiquetaProductos = new JLabel("Devoluciones");
    etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaProductos.setForeground(Color.BLUE);
    etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaProductos.setBounds(323, 32, 143, 41);
    frmMenuDevoluciones.getContentPane().add(etiquetaProductos);

    /*----- Nombre del Negocio y Logotipo -----*/

    JLabel etiquetaNegocio = new JLabel("Mi Negocio");
    etiquetaNegocio.setForeground(new Color(19, 60, 245));
    etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
    etiquetaNegocio.setBounds(1008, 55, 202, 56);
    frmMenuDevoluciones.getContentPane().add(etiquetaNegocio);

    JLabel etiquetaLogotipo = new JLabel("");
    etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
    etiquetaLogotipo.setBackground(new Color(175, 238, 238));
    etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaLogotipo.setBounds(1197, 35, 129, 104);
    frmMenuDevoluciones.getContentPane().add(etiquetaLogotipo);

    /*----- Imagen del Menú -----*/

    JLabel imagenProductos = new JLabel("");
    imagenProductos.setIcon(new
ImageIcon(MenuDevoluciones.class.getResource("/Imagenes/MDevoluciones.png")));
    imagenProductos.setHorizontalAlignment(SwingConstants.CENTER);
    imagenProductos.setBounds(194, 221, 500, 391);
    frmMenuDevoluciones.getContentPane().add(imagenProductos);

    /*----- Formato y Funcionalidad de los 3 Botones -----*/

    JButton botonRegistrar = new JButton("Registrar");
    botonRegistrar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            RegistrarDevolucion registrarDevol = new RegistrarDevolucion();

            registrarDevol.setVisible(true);
            frmMenuDevoluciones.setVisible(false);
        }
    });
    botonRegistrar.addMouseListener(new MouseAdapter()

```

```

{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeRegistrar = new LineBorder(Color.black,2);
botonRegistrar.setBorder(elBordeRegistrar);
botonRegistrar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonRegistrar.setForeground(new Color(165, 42, 42));
botonRegistrar.setBackground(new Color(195,255,250));
botonRegistrar.setBounds(836, 203, 264, 73);
frmMenuDevoluciones.getContentPane().add(botonRegistrar);

JButton botonModificar = new JButton("Modificar");
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ModificarDevolucion modificarDevol = new ModificarDevolucion();

        modificarDevol.setVisible(true);
        frmMenuDevoluciones.setVisible(false);
    }
});
botonModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonModificar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonModificar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeModificar = new LineBorder(Color.black,2);
botonModificar.setBorder(elBordeModificar);
botonModificar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonModificar.setForeground(new Color(165, 42, 42));
botonModificar.setBackground(new Color(195,255,250));
botonModificar.setBounds(836, 384, 264, 73);
frmMenuDevoluciones.getContentPane().add(botonModificar);

JButton botonConsultar = new JButton("Consultar");
botonConsultar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ConsultarDevoluciones devoluciones = new ConsultarDevoluciones();

        devoluciones.setVisible(true);
        frmMenuDevoluciones.setVisible(false);
    }
});
botonConsultar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {

```

```

        botonConsultar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,2);
botonConsultar.setBorder(elBordeConsultar);
botonConsultar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonConsultar.setForeground(new Color(165, 42, 42));
botonConsultar.setBackground(new Color(195,255,250));
botonConsultar.setBounds(836, 562, 264, 73);
frmMenuDevoluciones.getContentPane().add(botonConsultar);
}
}

```

11.- MenuPedidos.java

```

package Interfaces;

/** Esta Interfaz muestra las 3 operaciones que se pueden realizar para un Pedido:
"Registrar", "Modificar" y "Consultar" */

import java.awt.*;
import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.LineBorder;

import java.awt.event.*;

public class MenuPedidos
{
    /******* CONTENEDOR JFRAME BASE *****/

    private JFrame frmMenuPedidos;

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    MenuPedidos window = new MenuPedidos();
                    window.frmMenuPedidos.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmMenuPedidos.setVisible(visibilidad);
    }
}

```

```

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

public MenuPedidos()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DEL MENÚ PARA PEDIDOS *****/

private void initialize()
{
/*----- Formato del JFrame -----*/

    frmMenuPedidos = new JFrame();
    frmMenuPedidos.getContentPane().setBackground(new Color(255, 255, 255));

    frmMenuPedidos.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuAlmacen.class.getResource("/Imágenes/Logotipo.png")));
    frmMenuPedidos.setTitle("Pedidos");
    frmMenuPedidos.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmMenuPedidos.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmMenuPedidos.getContentPane().setLayout(null);

/*----- Formato y Funcionalidad de las Migajas -----*/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmMenuPedidos.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    frmMenuPedidos.getContentPane().add(etiquetaPaginaPrincipal);

    JLabel etiquetaDiagonal2 = new JLabel("/");
    etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal2.setForeground(new Color(0, 0, 255));
    etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal2.setBounds(313, 32, 22, 39);
    frmMenuPedidos.getContentPane().add(etiquetaDiagonal2);

    JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
    etiquetaAlmacen.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuAlmacen almacen = new MenuAlmacen();

```

```

        almacen.setVisible(true);
        frmMenuPedidos.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
frmMenuPedidos.getContentPane().add(etiquetaAlmacen);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setForeground(Color.BLUE);
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setBounds(214, 32, 22, 39);
frmMenuPedidos.getContentPane().add(etiquetaDiagonal1);

JLabel etiquetaProductos = new JLabel("Pedidos");
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(329, 32, 85, 41);
frmMenuPedidos.getContentPane().add(etiquetaProductos);

/*----- Nombre del Negocio y Logotipo -----*/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
frmMenuPedidos.getContentPane().add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imágenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
frmMenuPedidos.getContentPane().add(etiquetaLogotipo);

/*----- Imagen del Menú -----*/

JLabel imagenProductos = new JLabel("");
imagenProductos.setIcon(new
ImageIcon(MenuPedidos.class.getResource("/Imágenes/MPedidos.png")));
imagenProductos.setHorizontalAlignment(SwingConstants.CENTER);
imagenProductos.setBounds(193, 187, 419, 451);
frmMenuPedidos.getContentPane().add(imagenProductos);

/*----- Formato y Funcionalidad de los 3 Botones -----*/

JButton botonRegistrar = new JButton("Registrar");
botonRegistrar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {

```



```

RegistrarPedido registroPedido = new RegistrarPedido();

registroPedido.setVisible(true);
frmMenuPedidos.setVisible(false);
}
});
botonRegistrar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeRegistrar = new LineBorder(Color.black,2);
botonRegistrar.setBorder(elBordeRegistrar);
botonRegistrar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonRegistrar.setForeground(new Color(165, 42, 42));
botonRegistrar.setBackground(new Color(195,255,250));
botonRegistrar.setBounds(836, 203, 264, 73);
frmMenuPedidos.getContentPane().add(botonRegistrar);

JButton botonModificar = new JButton("Modificar");
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ModificarPedido modificarPedido = new ModificarPedido();

        modificarPedido.setVisible(true);
        frmMenuPedidos.setVisible(false);
    }
});
botonModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonModificar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonModificar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeModificar = new LineBorder(Color.black,2);
botonModificar.setBorder(elBordeModificar);
botonModificar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonModificar.setForeground(new Color(165, 42, 42));
botonModificar.setBackground(new Color(195,255,250));
botonModificar.setBounds(836, 384, 264, 73);
frmMenuPedidos.getContentPane().add(botonModificar);

JButton botonConsultar = new JButton("Consultar");
botonConsultar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ConsultarPedidos consultarPedidos = new ConsultarPedidos();

        consultarPedidos.setVisible(true);
        frmMenuPedidos.setVisible(false);
    }
});

```

```

    }
});
botonConsultar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,2);
botonConsultar.setBorder(elBordeConsultar);
botonConsultar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonConsultar.setForeground(new Color(165, 42, 42));
botonConsultar.setBackground(new Color(195,255,250));
botonConsultar.setBounds(836, 562, 264, 73);
frmMenuPedidos.getContentPane().add(botonConsultar);
}
}

```

12.- MenúPrincipal.java

```

package Interfaces;

/** Esta Interfaz muestra las 3 partes en las que se divide todo el Sistema y a las que se
puede acceder: "Almacen", "Ventas" y "Clientes" */

import java.awt.*;
import javax.swing.*;
import hibernate.operaciones.DatosEstaticos;
import java.awt.event.*;

public class MenuPrincipal
{
    /******* CONTENEDOR JFRAME BASE *****/

    private JFrame frmPginaPrincipal;

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    MenuPrincipal window = new MenuPrincipal();
                    window.frmPginaPrincipal.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {

```

```

        frmPginaPrincipal.setVisible(visibilidad);
    }

    /***** INICIALIZACIÓN GENERAL DEL JFrame *****/

    public MenuPrincipal()
    {
        initialize();
    }

    /***** DISEÑO DE LA PÁGINA DEL MENÚ PRINCIPAL *****/

    private void initialize()
    {
        /*----- Formato del JFrame -----*/

        frmPginaPrincipal = new JFrame();
        frmPginaPrincipal.getContentPane().setBackground(new Color(255, 255, 255));
        frmPginaPrincipal.setTitle("P\u00E1gina Principal");

        frmPginaPrincipal.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuPrincipal.class.g
        etResource("/Imagenes/Logotipo.png")));
        frmPginaPrincipal.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frmPginaPrincipal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmPginaPrincipal.getContentPane().setLayout(null);

        /*----- Formato de las 3 Imagenes del Menú -----*/

        JLabel imagenAlmacen = new JLabel("");
        imagenAlmacen.setIcon(new
        ImageIcon(MenuPrincipal.class.getResource("/Imagenes/MP_Almacen.png")));
        imagenAlmacen.setBounds(159, 58, 153, 155);
        frmPginaPrincipal.getContentPane().add(imagenAlmacen);

        JLabel imagenVentas = new JLabel("");
        imagenVentas.setIcon(new
        ImageIcon(MenuPrincipal.class.getResource("/Imagenes/MP_Ventas.png")));
        imagenVentas.setBounds(159, 293, 160, 126);
        frmPginaPrincipal.getContentPane().add(imagenVentas);

        JLabel imagenClientes = new JLabel("");
        imagenClientes.setIcon(new
        ImageIcon(MenuPrincipal.class.getResource("/Imagenes/MP_Clientes.png")));
        imagenClientes.setBounds(152, 525, 160, 126);
        frmPginaPrincipal.getContentPane().add(imagenClientes);

        /*----- Formato y Funcionalidad de la Etiqueta de Cada Imágen -----*/

        JLabel etiquetaImagenAlmacen = new JLabel("Almacen");
        etiquetaImagenAlmacen.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent e)
            {
                MenuAlmacen almacen = new MenuAlmacen();

                almacen.setVisible(true);
                frmPginaPrincipal.setVisible(false);
            }
            @Override
            public void mouseEntered(MouseEvent e)
            {
                etiquetaImagenAlmacen.setForeground(new Color(36,173,157));
            }
            @Override
            public void mouseExited(MouseEvent e)
            {

```

```

        etiquetaImagenAlmacen.setForeground(new Color(173, 36, 36));
    }
});
etiquetaImagenAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaImagenAlmacen.setForeground(new Color(173, 36, 36));
etiquetaImagenAlmacen.setFont(new Font("Lucida Calligraphy", Font.PLAIN, 40));
etiquetaImagenAlmacen.setBounds(345, 108, 212, 48);
frmPginaPrincipal.getContentPane().add(etiquetaImagenAlmacen);

JLabel etiquetaImagenVentas = new JLabel("Ventas");
etiquetaImagenVentas.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        MenuVentas menuVentas = new MenuVentas();

        menuVentas.setVisible(true);
        frmPginaPrincipal.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaImagenVentas.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaImagenVentas.setForeground(new Color(173, 36, 36));
    }
});
etiquetaImagenVentas.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaImagenVentas.setForeground(new Color(173, 36, 36));
etiquetaImagenVentas.setFont(new Font("Lucida Calligraphy", Font.PLAIN, 40));
etiquetaImagenVentas.setBounds(345, 333, 160, 43);
frmPginaPrincipal.getContentPane().add(etiquetaImagenVentas);

JLabel etiquetaImagenClientes = new JLabel("Clientes");
etiquetaImagenClientes.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        MenuClientes clientes = new MenuClientes();

        clientes.setVisible(true);
        frmPginaPrincipal.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaImagenClientes.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaImagenClientes.setForeground(new Color(173, 36, 36));
    }
});
etiquetaImagenClientes.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaImagenClientes.setForeground(new Color(173, 36, 36));
etiquetaImagenClientes.setFont(new Font("Lucida Calligraphy", Font.PLAIN, 40));
etiquetaImagenClientes.setBounds(345, 564, 174, 48);
frmPginaPrincipal.getContentPane().add(etiquetaImagenClientes);

/*----- Nombre del Negocio y Logotipo -----*/

JLabel imagenLogotipo = new JLabel("");

```

```

        imagenLogotipo.setIcon(new
ImageIcon(MenuPrincipal.class.getResource("/Imagenes/MP_Logotipo.png")));
        imagenLogotipo.setBounds(886, 243, 232, 181);
        frmPginaPrincipal.getContentPane().add(imagenLogotipo);

        JLabel etiquetaLogotipo = new JLabel("Mi Negocio");
        etiquetaLogotipo.setForeground(new Color(19, 60, 245));
        etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
        etiquetaLogotipo.setFont(new Font("Vivaldi", Font.PLAIN, 80));
        etiquetaLogotipo.setBounds(803, 435, 403, 89);
        frmPginaPrincipal.getContentPane().add(etiquetaLogotipo);

/*----- Etiqueta de Nombre del Empleado y de "Cerrar Sesión" -----*/

        JLabel etiquetaNombreVendedor = new JLabel();
        etiquetaNombreVendedor.setText(DatosEstaticos.nombreEmpleado);
        etiquetaNombreVendedor.setHorizontalAlignment(SwingConstants.RIGHT);
        etiquetaNombreVendedor.setFont(new Font("Cambria Math", Font.ITALIC, 20));
        etiquetaNombreVendedor.setBounds(886, 34, 406, 30);
        frmPginaPrincipal.getContentPane().add(etiquetaNombreVendedor);

        JLabel etiquetaCerrarSesion = new JLabel("Cerrar Sesi\u00F3n");
        etiquetaCerrarSesion.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent arg0)
            {
                Login login = new Login();

                login.setVisible(true);
                frmPginaPrincipal.setVisible(false);
            }
            @Override
            public void mouseEntered(MouseEvent arg0)
            {
                etiquetaCerrarSesion.setForeground(new Color(232,19,245));
            }
            @Override
            public void mouseExited(MouseEvent e)
            {
                etiquetaCerrarSesion.setForeground(new Color(0, 0, 255));
            }
        });
        etiquetaCerrarSesion.setForeground(new Color(0, 0, 255));
        etiquetaCerrarSesion.setHorizontalAlignment(SwingConstants.CENTER);
        etiquetaCerrarSesion.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC,
20));
        etiquetaCerrarSesion.setBounds(1162, 62, 130, 34);
        frmPginaPrincipal.getContentPane().add(etiquetaCerrarSesion);
    }
}

```

13.- MenuProductos.java

```

package Interfaces;

/** Esta Interfaz muestra las 3 operaciones que se pueden realizar para un Producto:
"Nuevo", "Modificar" y "Consultar" */

import java.awt.*;
import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.LineBorder;
import java.awt.event.*;

public class MenuProductos

```

```

{
/***** CONTENEDOR JFrame BASE *****/

private JFrame frmMenuProductos;

/***** MAIN *****/

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                MenuProductos window = new MenuProductos();
                window.frmMenuProductos.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmMenuProductos.setVisible(visibilidad);
}

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

public MenuProductos()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DEL MENÚ PARA PRODUCTOS *****/

private void initialize()
{
/*----- Formato del JFrame -----*/

    frmMenuProductos = new JFrame();
    frmMenuProductos.getContentPane().setBackground(new Color(255,255,255));

    frmMenuProductos.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuAlmacen.class.getResource("/Imágenes/Logotipo.png")));
    frmMenuProductos.setTitle("Productos");
    frmMenuProductos.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmMenuProductos.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmMenuProductos.getContentPane().setLayout(null);

/*----- Formato y Funcionalidad de las Migajas -----*/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmMenuProductos.setVisible(false);
        }
    });
}
}

```

```

    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
frmMenuProductos.getContentPane().add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setForeground(new Color(0, 0, 255));
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setBounds(313, 32, 22, 39);
frmMenuProductos.getContentPane().add(etiquetaDiagonal2);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmMenuProductos.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
frmMenuProductos.getContentPane().add(etiquetaAlmacen);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setForeground(Color.BLUE);
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setBounds(214, 32, 22, 39);
frmMenuProductos.getContentPane().add(etiquetaDiagonal1);

JLabel etiquetaProductos = new JLabel("Productos");
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(326, 31, 108, 41);
frmMenuProductos.getContentPane().add(etiquetaProductos);

```

```

/*----- Nombre del Negocio y Logotipo -----*/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
frmMenuProductos.getContentPane().add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
frmMenuProductos.getContentPane().add(etiquetaLogotipo);

/*----- Imagen del Menú -----*/

JLabel imagenProductos = new JLabel("");
imagenProductos.setHorizontalAlignment(SwingConstants.CENTER);
imagenProductos.setIcon(new
ImageIcon(MenuProductos.class.getResource("/Imagenes/MProductos.png")));
imagenProductos.setBounds(129, 209, 530, 425);
frmMenuProductos.getContentPane().add(imagenProductos);

/*----- Formato y Funcionalidad de los 3 Botones -----*/

JButton botonNuevo = new JButton("Nuevo");
botonNuevo.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        NuevoProducto nuevoProducto = new NuevoProducto();

        nuevoProducto.setVisible(true);
        frmMenuProductos.setVisible(false);
    }
});
botonNuevo.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonNuevo.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonNuevo.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeNuevo = new LineBorder(Color.black,2);
botonNuevo.setBorder(elBordeNuevo);
botonNuevo.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonNuevo.setForeground(new Color(165, 42, 42));
botonNuevo.setBackground(new Color(195,255,250));
botonNuevo.setBounds(836, 203, 264, 73);
frmMenuProductos.getContentPane().add(botonNuevo);

JButton botonModificar = new JButton("Modificar");
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ModificarProducto modificarProducto = new ModificarProducto();

        modificarProducto.setVisible(true);
    }
});

```



```

        frmMenuProductos.setVisible(false);
    }
});
 botonModificar.addMouseListener(new MouseAdapter()
 {
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonModificar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonModificar.setForeground(new Color(165, 42, 42));
    }
 });
Border elBordeModificar = new LineBorder(Color.black,2);
 botonModificar.setBorder(elBordeModificar);
 botonModificar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
 botonModificar.setForeground(new Color(165, 42, 42));
 botonModificar.setBackground(new Color(195,255,250));
 botonModificar.setBounds(836, 384, 264, 73);
 frmMenuProductos.getContentPane().add(botonModificar);

 JButton botonConsultar = new JButton("Consultar");
 botonConsultar.addActionListener(new ActionListener()
 {
    public void actionPerformed(ActionEvent e)
    {
        ConsultarProducto consultaProducto = new ConsultarProducto();

        consultaProducto.setVisible(true);
        frmMenuProductos.setVisible(false);
    }
 });
 botonConsultar.addMouseListener(new MouseAdapter()
 {
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(165, 42, 42));
    }
 });
Border elBordeConsultar = new LineBorder(Color.black,2);
 botonConsultar.setBorder(elBordeConsultar);
 botonConsultar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
 botonConsultar.setForeground(new Color(165, 42, 42));
 botonConsultar.setBackground(new Color(195,255,250));
 botonConsultar.setBounds(836, 562, 264, 73);
 frmMenuProductos.getContentPane().add(botonConsultar);
}
}
}

```

14.- MenúProveedores.java

```

package Interfaces;

/** Esta Interfaz muestra las 3 operaciones que se pueden realizar para un Proveedor:
"Nuevo", "Modificar" y "Consultar" */

import java.awt.*;
import javax.swing.*;

```

```

import javax.swing.border.*;
import java.awt.event.*;

public class MenuProveedores
{
/***** CONTENEDOR JFRAME BASE *****/

    private JFrame frmMenuProveedores;

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    MenuProveedores window = new MenuProveedores();
                    window.frmMenuProveedores.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

/***** VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmMenuProveedores.setVisible(visibilidad);
    }

    public MenuProveedores()
    {
        initialize();
    }

/***** DISEÑO DE LA PÁGINA DEL MENÚ PARA PROVEEDORES *****/

    private void initialize()
    {
/*----- Formato del JFrame -----*/

        frmMenuProveedores = new JFrame();
        frmMenuProveedores.getContentPane().setBackground(new Color(255, 255, 255));

        frmMenuProveedores.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuAlmacen.class.getResource("/Imágenes/Logotipo.png")));
        frmMenuProveedores.setTitle("Proveedores");
        frmMenuProveedores.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frmMenuProveedores.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmMenuProveedores.getContentPane().setLayout(null);

/*----- Formato y Funcionalidad de las Migajas -----*/

        JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
        etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent arg0)
            {
                MenuPrincipal menuPrincipal = new MenuPrincipal();

                menuPrincipal.setVisible(true);
                frmMenuProveedores.setVisible(false);
            }
        });
    }
}

```

```

    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
frmMenuProveedores.getContentPane().add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setForeground(new Color(0, 0, 255));
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setBounds(313, 32, 22, 39);
frmMenuProveedores.getContentPane().add(etiquetaDiagonal2);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmMenuProveedores.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
frmMenuProveedores.getContentPane().add(etiquetaAlmacen);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setForeground(Color.BLUE);
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setBounds(214, 32, 22, 39);
frmMenuProveedores.getContentPane().add(etiquetaDiagonal1);

JLabel etiquetaProveedores = new JLabel("Proveedores");
etiquetaProveedores.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProveedores.setForeground(Color.BLUE);
etiquetaProveedores.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProveedores.setBounds(326, 31, 135, 41);
frmMenuProveedores.getContentPane().add(etiquetaProveedores);

```

```

/*----- Nombre del Negocio y Logotipo -----*/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
frmMenuProveedores.getContentPane().add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
frmMenuProveedores.getContentPane().add(etiquetaLogotipo);

/*----- Imagen del Menú -----*/

JLabel imagenProveedores = new JLabel("");
imagenProveedores.setIcon(new
ImageIcon(MenuProveedores.class.getResource("/Imagenes/MProveedores.png")));
imagenProveedores.setHorizontalAlignment(SwingConstants.CENTER);
imagenProveedores.setBounds(178, 180, 481, 480);
frmMenuProveedores.getContentPane().add(imagenProveedores);

/*----- Formato y Funcionalidad de los 3 Botones -----*/

JButton botonNuevo = new JButton("Nuevo");
botonNuevo.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        NuevoProveedor nuevoProveedor = new NuevoProveedor();

        nuevoProveedor.setVisible(true);
        frmMenuProveedores.setVisible(false);
    }
});
botonNuevo.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonNuevo.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonNuevo.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeNuevo = new LineBorder(Color.black,2);
botonNuevo.setBorder(elBordeNuevo);
botonNuevo.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonNuevo.setForeground(new Color(165, 42, 42));
botonNuevo.setBackground(new Color(195,255,250));
botonNuevo.setBounds(836, 203, 264, 73);
frmMenuProveedores.getContentPane().add(botonNuevo);

JButton botonModificar = new JButton("Modificar");
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ModificarProveedor modificarProveedor = new ModificarProveedor();

        modificarProveedor.setVisible(true);
    }
});

```

```

        frmMenuProveedores.setVisible(false);
    }
});
botonModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonModificar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonModificar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeModificar = new LineBorder(Color.black,2);
botonModificar.setBorder(elBordeModificar);
botonModificar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonModificar.setForeground(new Color(165, 42, 42));
botonModificar.setBackground(new Color(195,255,250));
botonModificar.setBounds(836, 384, 264, 73);
frmMenuProveedores.getContentPane().add(botonModificar);

JButton botonConsultar = new JButton("Consultar");
botonConsultar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ConsultarProveedor consultarProveedor = new ConsultarProveedor();

        consultarProveedor.setVisible(true);
        frmMenuProveedores.setVisible(false);
    }
});
botonConsultar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(36,173,157));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonConsultar.setForeground(new Color(165, 42, 42));
    }
});
Border elBordeConsultar = new LineBorder(Color.black,2);
botonConsultar.setBorder(elBordeConsultar);
botonConsultar.setFont(new Font("Cambria Math", Font.PLAIN, 40));
botonConsultar.setForeground(new Color(165, 42, 42));
botonConsultar.setBackground(new Color(195,255,250));
botonConsultar.setBounds(836, 562, 264, 73);
frmMenuProveedores.getContentPane().add(botonConsultar);
}
}
}

```

15.- MenuVentas.java

```

package Interfaces;

/** Esta Interfaz muestra las 2 secciones: "Caja" y "Consultar", que conforman la parte de
"Ventas" */

import java.awt.Color;

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MenuVentas
{
    /******* CONTENEDOR JFRAME BASE *****/

    private JFrame frmMenuventas;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    MenuVentas window = new MenuVentas();
                    window.frmMenuventas.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public MenuVentas()
    {
        initialize();
    }

    /******* VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmMenuventas.setVisible(visibilidad);
    }

    /******* DISEÑO DE LA PÁGINA DEL MENÚ PARA VENTAS *****/

    private void initialize()
    {
        /*----- Formato del JFrame -----*/

        frmMenuventas = new JFrame();
        frmMenuventas.setTitle("MenuVentas");
        frmMenuventas.getContentPane().setBackground(new Color(255, 255, 255));

        frmMenuventas.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuClientes.class.getResource("/Imagenes/Logotipo.png")));
        frmMenuventas.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frmMenuventas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmMenuventas.getContentPane().setLayout(null);

        /*----- Formato y Funcionalidad de las Migajas -----*/

        JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
        etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent arg0)
            {
                MenuPrincipal menuPrincipal = new MenuPrincipal();

                menuPrincipal.setVisible(true);
                frmMenuventas.setVisible(false);
            }

            @Override
            public void mouseEntered(MouseEvent e)
            {

```

```

        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
frmMenuventas.getContentPane().add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal = new JLabel("/");
etiquetaDiagonal.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal.setForeground(new Color(0, 0, 255));
etiquetaDiagonal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal.setBounds(214, 32, 22, 39);
frmMenuventas.getContentPane().add(etiquetaDiagonal);

JLabel etiquetaClientes = new JLabel("Ventas");
etiquetaClientes.setForeground(new Color(0, 0, 255));
etiquetaClientes.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaClientes.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaClientes.setBounds(225, 31, 85, 41);
frmMenuventas.getContentPane().add(etiquetaClientes);

/*----- Nombre del Negocio y Logotipo -----*/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
frmMenuventas.getContentPane().add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imágenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
frmMenuventas.getContentPane().add(etiquetaLogotipo);

/*----- Formato de las 2 Imágenes del Menú -----*/

JLabel imagenRegistro = new JLabel("");
imagenRegistro.setHorizontalAlignment(SwingConstants.CENTER);
imagenRegistro.setIcon(new
ImageIcon(MenuClientes.class.getResource("/Imágenes/MV_Caja.png")));
imagenRegistro.setBounds(300, 280, 210, 200);
frmMenuventas.getContentPane().add(imagenRegistro);

JLabel imagenModificar = new JLabel("");
imagenModificar.setIcon(new
ImageIcon(MenuClientes.class.getResource("/Imágenes/MV_Consultar.png")));
imagenModificar.setHorizontalAlignment(SwingConstants.CENTER);
imagenModificar.setBounds(850, 275, 210, 200);
frmMenuventas.getContentPane().add(imagenModificar);

/*----- Formato y Funcionalidad de la Etiqueta de Cada Imagen -----*/

JLabel etiquetaCaja = new JLabel("Caja");
etiquetaCaja.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {

```

```

RegistrarVentas registrarVentas = new RegistrarVentas();

registrarVentas.setVisible(true);
frmMenuventas.setVisible(false);
}
@Override
public void mouseEntered(MouseEvent e)
{
    etiquetaCaja.setForeground(new Color(0, 153, 102));
}
@Override
public void mouseExited(MouseEvent e)
{
    etiquetaCaja.setForeground(new Color(204, 51, 51));
}
});
etiquetaCaja.setForeground(new Color(204, 51, 51));
etiquetaCaja.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaCaja.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaCaja.setBounds(330, 500, 147, 40);
frmMenuventas.getContentPane().add(etiquetaCaja);

JLabel etiquetaModificar = new JLabel("Consultar");
etiquetaModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        ConsultarVentas ventas = new ConsultarVentas();

        ventas.setVisible(true);
        frmMenuventas.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaModificar.setForeground(new Color(0, 153, 102));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaModificar.setForeground(new Color(204, 51, 51));
    }
});
etiquetaModificar.setForeground(new Color(204, 51, 51));
etiquetaModificar.setFont(new Font("Times New Roman", Font.PLAIN, 30));
etiquetaModificar.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaModificar.setBounds(875, 500, 163, 34);
frmMenuventas.getContentPane().add(etiquetaModificar);
}
}

```

16.- ModificarCliente.java

```

package Interfaces;

/** Esta Interfaz permite modificar la información de un Cliente o eliminarlo en caso de
que no se encuentre ligado con una venta */

import java.awt.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import javax.swing.*;
import javax.swing.border.*;
import com.toedter.calendar.JDateChooser;

```



```

import hibernate.beans.Cliente;
import hibernate.operaciones.OpCliente;

public class ModificarCliente
{
/***** CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

private JFrame frmModificarCliente;
private JPanel panelGeneral;

/***** CAJA PARA BUSCAR CLIENTE EN GROUPBOX *****/

private JTextField cajaBusquedaRfc;

/***** ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

private TextPrompt leyenda;
private JTextPane panelTextoDireccion;
private JTextField cajaRfcCliente, cajaTelefono;
private JTextField cajaCorreo, cajaNombreCliente;
private JDateChooser fecha;

/***** OPERACIONES PARA FUNCIONALIDAD *****/

private OpCliente operacionesCliente = new OpCliente();
private Cliente elCliente = new Cliente();

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                ModificarCliente window = new ModificarCliente();
                window.frmModificarCliente.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public ModificarCliente()
{
    initialize();
}

/***** VISIBILIDAD DEL JFRAME *****/

public void setVisible(boolean visibilidad)
{
    frmModificarCliente.setVisible(visibilidad);
}

/***** DISEÑO DE LA PÁGINA DE MODIFICAR CLIENTE *****/

private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmModificarCliente = new JFrame();
    frmModificarCliente.setTitle("Modificar Cliente");

frmModificarCliente.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProducto.class.getResource("/Imágenes/Logotipo.png")));
    frmModificarCliente.setBounds(100, 100, 450, 300);
    frmModificarCliente.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmModificarCliente.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmModificarCliente.getContentPane().setLayout(null);
}
}

```

```

JScrollPane scrollGeneral = new JScrollPane();
scrollGeneral.setBackground(new Color(175, 238, 238));
scrollGeneral.setBounds(0, 0, 1365, 715);
scrollGeneral.setHorizontalScrollBar(null);
frmModificarCliente.getContentPane().add(scrollGeneral);

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1450));
panelGeneral.setBackground(new Color(255, 255, 255));
scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmModificarCliente.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Clientes");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuClientes clientes = new MenuClientes();

        clientes.setVisible(true);
        frmModificarCliente.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)

```

```

        {
            etiquetaAlmacen.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaAlmacen.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaAlmacen.setBounds(233, 32, 85, 41);
    panelGeneral.add(etiquetaAlmacen);

    JLabel etiquetaDiagonal2 = new JLabel("/");
    etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal2.setForeground(Color.BLUE);
    etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal2.setBounds(214, 32, 22, 39);
    panelGeneral.add(etiquetaDiagonal2);

    JLabel etiquetaClientes = new JLabel("Modificar");
    etiquetaClientes.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaClientes.setForeground(Color.BLUE);
    etiquetaClientes.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaClientes.setBounds(326, 31, 108, 41);
    panelGeneral.add(etiquetaClientes);

    /***** Nombre del Negocio y Logotipo *****/

    JLabel etiquetaNegocio = new JLabel("Mi Negocio");
    etiquetaNegocio.setForeground(new Color(19, 60, 245));
    etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
    etiquetaNegocio.setBounds(1008, 55, 202, 56);
    panelGeneral.add(etiquetaNegocio);

    JLabel etiquetaLogotipo = new JLabel("");
    etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
    etiquetaLogotipo.setBackground(new Color(175, 238, 238));
    etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaLogotipo.setBounds(1197, 35, 129, 104);
    panelGeneral.add(etiquetaLogotipo);

    /***** GroupBox para Buscar Clientes Registrados *****/

    JPanel groupBoxBuscarCliente = new JPanel();
    groupBoxBuscarCliente.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Buscar Cliente",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
    groupBoxBuscarCliente.setBackground(new Color(243,243,243));
    groupBoxBuscarCliente.setBounds(356, 250, 624, 270);
    panelGeneral.add(groupBoxBuscarCliente);
    groupBoxBuscarCliente.setLayout(null);

    JLabel etiquetaNumeroCliente = new JLabel("Número de Cliente:");
    etiquetaNumeroCliente.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaNumeroCliente.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaNumeroCliente.setBounds(113, 60, 223, 30);
    groupBoxBuscarCliente.add(etiquetaNumeroCliente);

    cajaBusquedaRfc = new JTextField();
    cajaBusquedaRfc.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar RFC",cajaBusquedaRfc);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);

```

```

cajaBusquedaRfc.setBounds(344, 56, 170, 36);
cajaBusquedaRfc.setColumns(10);
cajaBusquedaRfc.setFont(new Font("Cambria Math", Font.PLAIN, 16));
groupBoxBuscarCliente.add(cajaBusquedaRfc);

/***** GroupBox con Tabla para ver los Clientes Registrados *****/

JPanel groupBoxDatosCliente = new JPanel();
groupBoxDatosCliente.setBackground(new Color(255,255,255));
groupBoxDatosCliente.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Datos del Cliente",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxDatosCliente.setBounds(100, 608, 1150, 650);
groupBoxDatosCliente.setLayout(null);
panelGeneral.add(groupBoxDatosCliente);

/***** Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
panelFormulario.setBounds(30, 30, 1088, 600);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 750));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
groupBoxDatosCliente.add(panelFormulario);

/***** Elementos del Formulario *****/

JLabel etiquetaRFC = new JLabel("R.F.C.");
etiquetaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaRFC.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaRFC.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaRFC);

cajaRfcCliente = new JTextField();
cajaRfcCliente.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar RFC",cajaRfcCliente);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaRfcCliente.setBounds(536, 45, 169, 36);
cajaRfcCliente.setColumns(10);
cajaRfcCliente.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaRfcCliente);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 130, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 130, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaNombreProveedor = new JLabel("Nombre del Cliente:");
etiquetaNombreProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProveedor.setBounds(272, 218, 257, 30);
panelFormulario.add(etiquetaNombreProveedor);

cajaNombreCliente = new JTextField();
cajaNombreCliente.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Nombre",cajaNombreCliente);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);

```

```

leyenda.changeStyle(Font.ITALIC);
cajaNombreCliente.setBounds(535, 215, 250, 36);
cajaNombreCliente.setColumns(10);
cajaNombreCliente.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaNombreCliente);

JLabel etiquetaDireccin = new JLabel("Direcci\u00F3n:");
etiquetaDireccin.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaDireccin.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaDireccin.setBounds(421, 330, 109, 30);
panelFormulario.add(etiquetaDireccin);

panelTextoDireccion = new JTextPane();
leyenda = new TextPrompt("Ingresar Dirección",panelTextoDireccion);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
panelTextoDireccion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelTextoDireccion.setBounds(536, 300, 229, 88);
panelFormulario.add(panelTextoDireccion);

JLabel etiquetaTelefono = new JLabel("Tel\u00E9fono:");
etiquetaTelefono.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTelefono.setBounds(421, 430, 109, 30);
panelFormulario.add(etiquetaTelefono);

cajaTelefono = new JTextField();
leyenda = new TextPrompt("(55) 55-55-55-55",cajaTelefono);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaTelefono.setHorizontalAlignment(SwingConstants.CENTER);
cajaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaTelefono.setColumns(10);
cajaTelefono.setBounds(536, 430, 155, 36);
panelFormulario.add(cajaTelefono);

JLabel etiquetaCorreo = new JLabel("Correo:");
etiquetaCorreo.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCorreo.setBounds(445, 515, 85, 30);
panelFormulario.add(etiquetaCorreo);

cajaCorreo = new JTextField();
leyenda = new TextPrompt("correo@correo.com",cajaCorreo);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCorreo.setHorizontalAlignment(SwingConstants.CENTER);
cajaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCorreo.setColumns(10);
cajaCorreo.setBounds(536, 515, 273, 36);
panelFormulario.add(cajaCorreo);

/***** Formato y Funcionalidad de Botón "Modificar" *****/

JButton botonModificar = new JButton("  Modificar");
Border elBordeModificar = new LineBorder(Color.black,1);
botonModificar.setBorder(elBordeModificar);
botonModificar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imágenes/Modificar.png")));
botonModificar.setBackground(new Color(216,253,241));
botonModificar.setFont(new Font("Century", Font.PLAIN, 18));
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {

```

```

        elCliente =
operacionesCliente.buscarClienteRfc(cajaBusquedaRfc.getText());

        if(elCliente != null)
        {
            cajaRfcCliente.setText(elCliente.getRfcCliente());
            fecha.setDate(elCliente.getFechaRegistro());
            cajaNombreCliente.setText(elCliente.getNombreCliente());
            panelTextoDireccion.setText(elCliente.getDireccion());
            cajaTelefono.setText(elCliente.getTelefono());
            cajaCorreo.setText(elCliente.getCorreo());
        }
        else
        {
            JOptionPane.showMessageDialog(null,"El Cliente no esta Registrado en
el Sistema");
        }
    }
});
botonModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonModificar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonModificar.setForeground(new Color(0,0,0));
    }
});
botonModificar.setBounds(366, 175, 183, 49);
groupBoxBuscarCliente.add(botonModificar);

/***** Formato y Funcionalidad de Botón "Eliminar" *****/

JButton botonEliminar = new JButton("  Eliminar");
Border elBordeEliminar = new LineBorder(Color.black,1);
botonEliminar.setBorder(elBordeEliminar);
botonEliminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        elCliente =
operacionesCliente.buscarClienteRfc(cajaBusquedaRfc.getText());

        if(elCliente != null)
        {
            operacionesCliente.eliminarCliente(cajaBusquedaRfc.getText());

            fecha.setDate(null);
            cajaRfcCliente.setText("");
            cajaNombreCliente.setText("");
            panelTextoDireccion.setText("");
            cajaTelefono.setText("");
            cajaCorreo.setText("");
            cajaBusquedaRfc.setText("");
        }
        else
        {
            JOptionPane.showMessageDialog(null,"El Cliente no esta Registrado en
el Sistema");
        }
    }
});
botonEliminar.addMouseListener(new MouseAdapter()
{

```

```

        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonEliminar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonEliminar.setForeground(new Color(0,0,0));
        }
    });
    botonEliminar.setBackground(new Color(216,253,241));
    botonEliminar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Eliminar.png")));
    botonEliminar.setFont(new Font("Century", Font.PLAIN, 18));
    botonEliminar.setBounds(80, 175, 183, 49);
    groupBoxBuscarCliente.add(botonEliminar);

/***** Formato y Funcionalidad de Botón "Salir" *****/

    JButton botonSalir = new JButton(" Salir");
    botonSalir.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            MenuClientes menuClientes = new MenuClientes();

            menuClientes.setVisible(true);
            frmModificarCliente.setVisible(false);
        }
    });
    botonSalir.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonSalir.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonSalir.setForeground(new Color(0,0,0));
        }
    });
    Border elBordeTerminar = new LineBorder(Color.black,1);
    botonSalir.setBorder(elBordeTerminar);
    botonSalir.setHorizontalAlignment(SwingConstants.CENTER);
    botonSalir.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
    botonSalir.setForeground(new Color(0,0,0));
    botonSalir.setBackground(new Color(234,34,34));
    botonSalir.setFont(new Font("Century", Font.PLAIN, 20));
    botonSalir.setBounds(300, 1300, 197, 50);
    panelGeneral.add(botonSalir);

/***** Formato y Funcionalidad de Botón "Guardar" *****/

    JButton botonRegistrar = new JButton(" Guardar");
    botonRegistrar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            java.util.Date laFecha = fecha.getDate();
            SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

            elCliente.setRfcCliente(cajaRfcCliente.getText());

```

```

elCliente.setFechaRegistro(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))));
    elCliente.setNombreCliente(cajaNombreCliente.getText());
    elCliente.setDireccion(panelTextoDireccion.getText());
    elCliente.setTelefono(cajaTelefono.getText());
    elCliente.setCorreo(cajaCorreo.getText());

    operacionesCliente.actualizarCliente(elCliente);

    fecha.setDate(null);
    cajaRfcCliente.setText("");
    cajaNombreCliente.setText("");
    panelTextoDireccion.setText("");
    cajaTelefono.setText("");
    cajaCorreo.setText("");
    cajaBusquedaRfc.setText("");
}
});
botonRegistrar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(0,0,0));
    }
});
Border elBordeRegistrar = new LineBorder(Color.black,1);
botonRegistrar.setBorder(elBordeRegistrar);
botonRegistrar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imágenes/Palomita.png")));
botonRegistrar.setForeground(new Color(0,0,0));
botonRegistrar.setBackground(new Color(95,255,95));
botonRegistrar.setFont(new Font("Century", Font.PLAIN, 20));
botonRegistrar.setBounds(850, 1300, 208, 50);
panelGeneral.add(botonRegistrar);
}
}
}

```

17.- ModificarDevolución.java

```

package Interfaces;

/** Esta Interfaz permite modificar la información de una Devolución, siempre y cuando su
estado sea distinto de "Devuelto", o eliminar la Devolución */

import java.awt.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.List;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.*;
import hibernate.operaciones.*;

public class ModificarDevolucion
{
    /******* CONTENEDOR JFrame BASE Y PANEL GENERAL *****/

    private JFrame frmModificarDevolucin;

```



```

private JPanel panelGeneral;

/***** ELEMENTOS DEL GROUPBOX "LISTA DEVOLUCIONES" *****/

@SuppressWarnings("rawtypes")
private JComboBox cajaListaProveedores;
private JTable tablaDevoluciones;
private DefaultTableModel modeloTabla;
private int fila;

/***** ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

private JDateChooser fecha;
private JTextField cajaCantidadProductos, cajaNumeroPedido;
private JTextPane panelTextoMotivo;
private TextPrompt leyenda;
private JLabel etiquetaVerProveedor = new JLabel();
private JLabel etiquetaVerNombreProducto = new JLabel();
private JLabel etiquetaVerCostoUnidad = new JLabel();
private JLabel etiquetaVerCostoDevolucion = new JLabel();
private JLabel etiquetaVerCodigoBarras = new JLabel();
@SuppressWarnings("rawtypes")
private JComboBox cajaTipoProductos, cajaStatusDevolucion;

/***** OPERACIONES PARA FUNCIONALIDAD *****/

private OpProveedor operacionesProveedor = new OpProveedor();
private OpDevolucion operacionesDevolucion = new OpDevolucion();
private OpProducto operacionesProducto = new OpProducto();
private OpPedido operacionesPedido = new OpPedido();
private Devolucion laDevolucion = new Devolucion();
private Producto elProducto = new Producto();
private List<Devolucion> lasDevoluciones;

/***** ELEMENTOS UTILIZADOS EN EL BOTÓN "Guardar" *****/

boolean cajasVerificadas;

/***** MAIN *****/

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                ModificarDevolucion window = new ModificarDevolucion();
                window.frmModificarDevolucin.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFRAME *****/

public void setVisible(boolean visibilidad)
{
    frmModificarDevolucin.setVisible(visibilidad);
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

```

```

public boolean camposLlenos()
{
    boolean verificacion;
    if (fecha.getDate() == null || cajaNumeroPedido.getText().isEmpty() ||
    etiquetaVerCodigoBarras.getText().isEmpty() ||
        etiquetaVerNombreProducto.getText().isEmpty() ||
    etiquetaVerProveedor.getText().isEmpty() || cajaCantidadProductos.getText().isEmpty() ||
        etiquetaVerCostoUnidad.getText().isEmpty() ||
    etiquetaVerCostoDevolucion.getText().isEmpty() || panelTextoMotivo.getText().isEmpty() ||
        cajaTipoProductos.getSelectedIndex() == 0 ||
    cajaStatusDevolucion.getSelectedIndex() == 0)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** INICIALIZACIÓN GENERAL DEL JFRAME *****/

public ModificarDevolucion()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA DE MODIFICAR DEVOLUCIÓN *****/

@SuppressWarnings({ "unchecked", "rawtypes", "serial" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmModificarDevolucin = new JFrame();

frmModificarDevolucin.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProducto
.class.getResource("/Imágenes/Logotipo.png")));
    frmModificarDevolucin.setTitle("Modificar Devoluciones");
    frmModificarDevolucin.setBounds(100, 100, 450, 300);
    frmModificarDevolucin.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmModificarDevolucin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmModificarDevolucin.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmModificarDevolucin.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1950));
    panelGeneral.setBackground(new Color(255, 255, 255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("Página Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

```

```

        menuPrincipal.setVisible(true);
        frmModificarDevolucin.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
    });
panelGeneral.setLayout(null);
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmModificarDevolucin.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaDevoluciones = new JLabel("Devoluciones");
etiquetaDevoluciones.addMouseListener(new MouseAdapter()

```

```

{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuDevoluciones devoluciones = new MenuDevoluciones();

        devoluciones.setVisible(true);
        frmModificarDevolucin.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(0, 0, 255));
    }
});
etiquetaDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDevoluciones.setForeground(Color.BLUE);
etiquetaDevoluciones.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaDevoluciones.setBounds(332, 31, 121, 41);
panelGeneral.add(etiquetaDevoluciones);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(450, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaModificarDevolucion = new JLabel("Modificar");
etiquetaModificarDevolucion.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaModificarDevolucion.setForeground(Color.BLUE);
etiquetaModificarDevolucion.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaModificarDevolucion.setBounds(430, 32, 170, 41);
panelGeneral.add(etiquetaModificarDevolucion);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox para ver la Lista de Devoluciones *****/

JPanel groupBoxListaDevoluciones = new JPanel();
groupBoxListaDevoluciones.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Lista de Devoluciones",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxListaDevoluciones.setBackground(new Color(243, 243, 243));
groupBoxListaDevoluciones.setBounds(250, 250, 850, 400);
groupBoxListaDevoluciones.setLayout(null);
panelGeneral.add(groupBoxListaDevoluciones);

cajaListaProveedores = new JComboBox();

```

```

cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        while(modeloTabla.getRowCount()>0)
        {
            modeloTabla.removeRow(0);
        }

        if(!((String)cajaListaProveedores.getSelectedItem()).equals("ninguno"))
        {
            lasDevoluciones =
operacionesDevolucion.tracerDevoluciones((String)cajaListaProveedores.getSelectedItem());
            for(int i=0;i<lasDevoluciones.size();i++)
            {
                Object[] renglon =
{lasDevoluciones.get(i).getFechaDevolucion(),lasDevoluciones.get(i).getNumeroDevolucion()
};
                modeloTabla.addRow(renglon);
            }
        }
    }
});
cajaListaProveedores.setBounds(320, 50, 221, 36);
groupBoxListaDevoluciones.add(cajaListaProveedores);

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(80, 150, 433, 180);
scrollTabla.setHorizontalScrollBar(null);
groupBoxListaDevoluciones.add(scrollTabla);

modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
tablaDevoluciones = new JTable(modeloTabla);
tablaDevoluciones.setFont(new Font("Cambria Math", Font.PLAIN, 15));
DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
modeloTabla.addColumn("FECHA");
modeloTabla.addColumn("NÚMERO DE DEVOLUCIÓN");
tablaDevoluciones.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaDevoluciones.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaDevoluciones.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        fila = tablaDevoluciones.getSelectedRow();
        laDevolucion =
operacionesDevolucion.buscarDevolucionPorNumero(Integer.valueOf(tablaDevoluciones.getValueAt(fila,1).toString()));
    }
});

```

```

scrollTabla.setViewportView(tablaDevoluciones);

/***** GroupBox para ver los Datos de las Devoluciones *****/

JPanel groupBoxDatosDevoluciones = new JPanel();
groupBoxDatosDevoluciones.setBackground(new Color(255,255,255));
groupBoxDatosDevoluciones.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Datos de las Devoluciones",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxDatosDevoluciones.setBounds(100, 760, 1150, 1040);
groupBoxDatosDevoluciones.setLayout(null);
panelGeneral.add(groupBoxDatosDevoluciones);

/***** Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
panelFormulario.setBounds(30, 30, 1088, 980);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 750));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
groupBoxDatosDevoluciones.add(panelFormulario);

/***** Elementos del Formulario *****/

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 45, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaNumeroPedido = new JLabel("Pedido:");
etiquetaNumeroPedido.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNumeroPedido.setBounds(342, 120, 188, 30);
panelFormulario.add(etiquetaNumeroPedido);

cajaNumeroPedido = new JTextField();
cajaNumeroPedido.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Número de Pedido",cajaNumeroPedido);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNumeroPedido.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            Pedido elPedido =
operacionesPedido.buscarPedidoPorNumero(Integer.valueOf(cajaNumeroPedido.getText()));

            if(cajaNumeroPedido.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingrese un Número de
Pedido.");
            }
            else if(elPedido == null)
            {

```

```

        JOptionPane.showMessageDialog(null, "Este Pedido no existe en el
Sistema.");
    }
    else
    {
etiquetaVerCodigoBarras.setText(elPedido.getProductoCodigoBarras());

etiquetaVerNombreProducto.setText(operacionesProducto.buscarProductoPorID(elPedido.getPro
ductoCodigoBarras()).getNombreProducto());

etiquetaVerProveedor.setText(operacionesProducto.buscarProductoPorID(elPedido.getProducto
CodigoBarras()).getProveedor());

etiquetaVerCostoUnidad.setText(String.valueOf(operacionesProducto.buscarProductoPorID(elP
edido.getProductoCodigoBarras()).getCostoCompra()));
    }
}

public void keyTyped(KeyEvent e)
{
    char caracter = e.getKeyChar();

    if((caracter < '0') || (caracter > '9')) && (caracter != '\b')
    {
        e.consume();
    }
}
});
cajaNumeroPedido.setBounds(536, 120, 124, 36);
cajaNumeroPedido.setColumns(10);
cajaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaNumeroPedido);

JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCodigoBarras.setBounds(342, 205, 188, 30);
panelFormulario.add(etiquetaCodigoBarras);

etiquetaVerCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCodigoBarras.setBounds(536, 205, 250, 36);
panelFormulario.add(etiquetaVerCodigoBarras);

JLabel etiquetaNombreProducto = new JLabel("Producto:");
etiquetaNombreProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProducto.setBounds(296, 280, 236, 30);
panelFormulario.add(etiquetaNombreProducto);

etiquetaVerNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerNombreProducto.setBounds(536, 280, 250, 36);
panelFormulario.add(etiquetaVerNombreProducto);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds(399, 350, 131, 30);
panelFormulario.add(etiquetaProveedor);

etiquetaVerProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerProveedor.setBounds(540, 350, 320, 36);
panelFormulario.add(etiquetaVerProveedor);

JLabel etiquetaCantidadProductos = new JLabel("Cantidad de Productos:");
etiquetaCantidadProductos.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCantidadProductos.setFont(new Font("Cambria Math", Font.PLAIN, 25));

```

```

etiquetaCantidadProductos.setBounds(250, 435, 280, 36);
panelFormulario.add(etiquetaCantidadProductos);

cajaCantidadProductos = new JTextField();
cajaCantidadProductos.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Cantidad", cajaCantidadProductos);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCantidadProductos.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            elProducto =
operacionesProducto.buscarProductoPorID(etiquetaVerCodigoBarras.getText());

            if(cajaCantidadProductos.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null, "Ingrese una Cantidad.");
            }
            else
            {
                etiquetaVerCostoDevolucion.setText(String.valueOf(Float.parseFloat(cajaCantidadProductos.
getText()) * elProducto.getCostoCompra()));
            }
        }
    }

    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCantidadProductos.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCantidadProductos.setColumns(10);
cajaCantidadProductos.setBounds(536, 435, 69, 36);
panelFormulario.add(cajaCantidadProductos);

JLabel etiquetaCostoUnidad = new JLabel("Costo por Unidad:");
etiquetaCostoUnidad.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoUnidad.setBounds(300, 520, 230, 30);
panelFormulario.add(etiquetaCostoUnidad);

etiquetaVerCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCostoUnidad.setBounds(540, 520, 95, 36);
panelFormulario.add(etiquetaVerCostoUnidad);

JLabel etiquetaCostoDevolucion = new JLabel("Costo de la Devolución:");
etiquetaCostoDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoDevolucion.setBounds(250, 625, 280, 30);
panelFormulario.add(etiquetaCostoDevolucion);

etiquetaVerCostoDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCostoDevolucion.setBounds(540, 625, 95, 36);
panelFormulario.add(etiquetaVerCostoDevolucion);

```



```

JLabel etiquetaMotivo = new JLabel("Motivo:");
etiquetaMotivo.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaMotivo.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaMotivo.setBounds(421, 720, 109, 30);
panelFormulario.add(etiquetaMotivo);

panelTextoMotivo = new JTextPane();
leyenda = new TextPrompt("Motivo de la Devolución",panelTextoMotivo);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
panelTextoMotivo.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelTextoMotivo.setBounds(536, 694, 229, 88);
panelFormulario.add(panelTextoMotivo);

JLabel etiquetaTipoProducto = new JLabel("Clasificación:");
etiquetaTipoProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTipoProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTipoProducto.setBounds(280, 820, 250, 30);
panelFormulario.add(etiquetaTipoProducto);

cajaTipoProductos = new JComboBox();
cajaTipoProductos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaTipoProductos.setModel(new DefaultComboBoxModel(new String[] {}));
cajaTipoProductos.addItem("Seleccione una Opción");
cajaTipoProductos.addItem("Abarrotes");
cajaTipoProductos.addItem("Bebidas");
cajaTipoProductos.addItem("Cuidado e Higiene Personal");
cajaTipoProductos.addItem("Dulceria");
cajaTipoProductos.addItem("Lacteos");
cajaTipoProductos.setBounds(536, 820, 270, 36);
panelFormulario.add(cajaTipoProductos);

JLabel etiquetaStatusDevolucion = new JLabel("Estatus:");
etiquetaStatusDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaStatusDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaStatusDevolucion.setBounds(440, 900, 90, 30);
panelFormulario.add(etiquetaStatusDevolucion);

cajaStatusDevolucion = new JComboBox();
cajaStatusDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaStatusDevolucion.setModel(new DefaultComboBoxModel(new String[] {}));
cajaStatusDevolucion.addItem("Seleccione una Opción");
cajaStatusDevolucion.addItem("Cancelado");
cajaStatusDevolucion.addItem("Devuelto");
cajaStatusDevolucion.addItem("Pendiente");
cajaStatusDevolucion.addItem("Rechazado");
cajaStatusDevolucion.setBounds(536, 900, 200, 36);
panelFormulario.add(cajaStatusDevolucion);

/***** Formato y Funcionalidad de Botón "Modificar" *****/

JButton botonModificar = new JButton("  Modificar");
Border elBordeModificar = new LineBorder(Color.black,1);
botonModificar.setBorder(elBordeModificar);
botonModificar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Modificar.png")));
botonModificar.setBackground(new Color(216,253,241));
botonModificar.setFont(new Font("Century", Font.PLAIN, 18));
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        Pedido elPedido =
operacionesPedido.buscarPedidoPorNumero(laDevolucion.getPedidoNumeroPedido());
        Producto elProducto =
operacionesProducto.buscarProductoPorID(elPedido.getProductoCodigoBarras());

```

```

        fecha.setDate(laDevolucion.getFechaDevolucion());

cajaNumeroPedido.setText(String.valueOf(laDevolucion.getPedidoNumeroPedido()));
etiquetaVerCodigoBarras.setText(elPedido.getProductoCodigoBarras());
etiquetaVerNombreProducto.setText(elProducto.getNombreProducto());
etiquetaVerProveedor.setText(elProducto.getProveedor());

cajaCantidadProductos.setText(String.valueOf(laDevolucion.getCantidadProductos()));

etiquetaVerCostoUnidad.setText(String.valueOf(elProducto.getCostoCompra()));

etiquetaVerCostoDevolucion.setText(String.valueOf(laDevolucion.getCostoDevolucion()));
panelTextoMotivo.setText(laDevolucion.getMotivo());
for(int i=0;i<=cajaTipoProductos.getItemCount()+1;i++)
{
if((laDevolucion.getClasificacion()).equals(cajaTipoProductos.getItemAt(i)))
    {
        cajaTipoProductos.setSelectedIndex(i);
        break;
    }
}
for(int i=0;i<=cajaStatusDevolucion.getItemCount()+1;i++)
{
if((laDevolucion.getStatusDevolucion()).equals(cajaStatusDevolucion.getItemAt(i)))
    {
        if(laDevolucion.getStatusDevolucion().equals("Devuelto"))
        {
            fecha.setEnabled(false);
            cajaNumeroPedido.setEnabled(false);
            cajaCantidadProductos.setEnabled(false);
            panelTextoMotivo.setEnabled(false);
            cajaTipoProductos.setEnabled(false);
            cajaStatusDevolucion.setEnabled(false);
        }
        cajaStatusDevolucion.setSelectedIndex(i);
        break;
    }
}
});
botonModificar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonModificar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonModificar.setForeground(new Color(0,0,0));
    }
});
botonModificar.setBounds(600, 150, 183, 61);
groupBoxListaDevoluciones.add(botonModificar);

/***** Formato y Funcionalidad de Botón "Eliminar" *****/

JButton botonEliminar = new JButton(" Eliminar");
Border elBordeEliminar = new LineBorder(Color.black,1);
botonEliminar.setBorder(elBordeEliminar);
botonEliminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {

```

```

operacionesDevolucion.eliminarDevolucion(Integer.parseInt(tablaDevoluciones.getValueAt(fi
la,1).toString()));

        fecha.setDate(null);
        cajaNumeroPedido.setText("");
        etiquetaVerCodigoBarras.setText("");
        etiquetaVerNombreProducto.setText("");
        etiquetaVerProveedor.setText("");
        cajaCantidadProductos.setText("");
        etiquetaVerCostoUnidad.setText("");
        etiquetaVerCostoDevolucion.setText("");
        panelTextoMotivo.setText("");
        cajaTipoProductos.setSelectedIndex(0);
        cajaStatusDevolucion.setSelectedIndex(0);

        while(modeloTabla.getRowCount()>0)
        {
            modeloTabla.removeRow(0);
        }

        if(!((String)cajaListaProveedores.getSelectedItem()).equals(""))
        {
            lasDevoluciones =
operacionesDevolucion.tracerDevoluciones((String)cajaListaProveedores.getSelectedItem());
            for(int i=0;i<lasDevoluciones.size();i++)
            {
                Object[] renglon =
{lasDevoluciones.get(i).getFechaDevolucion(),lasDevoluciones.get(i).getNumeroDevolucion()
};
                modeloTabla.addRow(renglon);
            }
        }
    });
    botonEliminar.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonEliminar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonEliminar.setForeground(new Color(0,0,0));
        }
    });
    botonEliminar.setBackground(new Color(216,253,241));
    botonEliminar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Eliminar.png")));
    botonEliminar.setFont(new Font("Century", Font.PLAIN, 18));
    botonEliminar.setBounds(600, 260, 183, 61);
    groupBoxListaDevoluciones.add(botonEliminar);

/***** Formato y Funcionalidad de Botón "Salir" *****/

    JButton botonSalir = new JButton(" Salir");
    botonSalir.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            MenuDevoluciones menuDevoluciones = new MenuDevoluciones();

            menuDevoluciones.setVisible(true);
            frmModificarDevolucin.setVisible(false);
        }
    });

```

```

        botonSalir.setHorizontalAlignment(SwingConstants.CENTER);
        botonSalir.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imágenes/Tache.png")));
        botonSalir.setForeground(Color.BLACK);
        botonSalir.setBackground(new Color(234,34,34));
        botonSalir.setFont(new Font("Century", Font.PLAIN, 20));
        botonSalir.setBounds(310, 1850, 210, 48);
        panelGeneral.add(botonSalir);

/***** Formato y Funcionalidad de Botón "Guardar" *****/

        JButton botonGuardar = new JButton("  Guardar");
        botonGuardar.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                cajasVerificadas = camposLlenos();

                if(cajasVerificadas == false)
                {
                    JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
                }
                else
                {
                    java.util.Date laFecha = fecha.getDate();
                    SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

                    laDevolucion.setFechaDevolucion(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha)
)));

                    laDevolucion.setClasificacion((String)cajaTipoProductos.getSelectedItem());
                    laDevolucion.setMotivo(panelTextoMotivo.getText());

                    laDevolucion.setCantidadProductos(Integer.parseInt(cajaCantidadProductos.getText()));

                    laDevolucion.setCostoDevolucion(Float.parseFloat(etiquetaVerCostoDevolucion.getText()));

                    laDevolucion.setStatusDevolucion((String)cajaStatusDevolucion.getSelectedItem());

                    laDevolucion.setPedidoNumeroPedido(Integer.parseInt(cajaNumeroPedido.getText()));

                    if(((String)cajaStatusDevolucion.getSelectedItem()).compareTo("Devuelto") == 0)
                    {

                        operacionesProducto.actualizarCantidadProductosAlmacenDevolucion(operacionesPedido.buscar
PedidoPorNumero(

Integer.parseInt(cajaNumeroPedido.getText()).getProductoCodigoBarras(),Integer.valueOf(c
ajaCantidadProductos.getText()));

                            }

                            operacionesDevolucion.actualizarDevolucion(laDevolucion);

                            fecha.setDate(null);
                            cajaNumeroPedido.setText("");
                            etiquetaVerCodigoBarras.setText("");
                            etiquetaVerNombreProducto.setText("");
                            etiquetaVerProveedor.setText("");
                            cajaCantidadProductos.setText("");
                            etiquetaVerCostoUnidad.setText("");
                            etiquetaVerCostoDevolucion.setText("");
                            panelTextoMotivo.setText("");
                            cajaTipoProductos.setSelectedIndex(0);
                            cajaStatusDevolucion.setSelectedIndex(0);

```

```

    }
    });
    botonGuardar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
    botonGuardar.setForeground(new Color(0, 0, 0));
    botonGuardar.setBackground(new Color(95,255,95));
    botonGuardar.setFont(new Font("Century", Font.PLAIN, 20));
    botonGuardar.setBounds(840, 1850, 210, 48);
    panelGeneral.add(botonGuardar);
}
}
}

```

18.- ModificarPedido.java

```

package Interfaces;

/** Esta Interfaz permite modificar la información de un Pedido, siempre y cuando su estado
sea distinto de "Realizado", o eliminar el Pedido */

import java.awt.*;
import java.util.List;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Pedido;
import hibernate.beans.Producto;
import hibernate.beans.Proveedor;
import hibernate.operaciones.OpProveedor;
import hibernate.operaciones.OpPedido;
import hibernate.operaciones.OpProducto;

public class ModificarPedido
{
/***** CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmModificarPedido;
    private JPanel panelGeneral;

/***** ELEMENTOS DEL GROUPBOX "LISTA DE PEDIDOS" *****/

    @SuppressWarnings("rawtypes")
    private JComboBox cajaListaProveedores;
    private JTable tablaPedidos;
    private DefaultTableModel modeloTabla;

/***** ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaCodigoBarras,cajaCantidadSolicitar;
    private TextPrompt leyenda;
    private JLabel etiquetaVerNombreProducto = new JLabel();
    private JLabel etiquetaVerRfcProveedor = new JLabel();
    private JLabel etiquetaVerCostoUnidad = new JLabel();
    private JLabel etiquetaVerTotalLiquidar= new JLabel();
    @SuppressWarnings("rawtypes")
    private JComboBox cajaStatusPedido;

/***** OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
    private OpPedido operacionesPedido = new OpPedido();

```

```

private OpProducto operacionesProducto = new OpProducto();
private Pedido elPedido = new Pedido();
private Proveedor elProveedor = new Proveedor();
private Producto elProducto = new Producto();

private List<Pedido> losPedidos = new ArrayList<>();
private int fila;

/***** MAIN *****/

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                ModificarPedido window = new ModificarPedido();
                window.frmModificarPedido.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if (fecha.getDate() == null || cajaCodigoBarras.getText().isEmpty() ||
        cajaCantidadSolicitar.getText().isEmpty() ||
        etiquetaVerNombreProducto.getText().isEmpty() ||
        etiquetaVerRfcProveedor.getText().isEmpty() || etiquetaVerCostoUnidad.getText().isEmpty()
        ||
        etiquetaVerTotalLiquidar.getText().isEmpty() ||
        cajaStatusPedido.getSelectedIndex() == 0 || tablaPedidos.getSelectedRow() == -1 ||
        cajaListaProveedores.getSelectedIndex() == 0)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** INICIALIZACIÓN GENERAL DEL JFRAME *****/

public ModificarPedido()
{
    initialize();
}

/***** VISIBILIDAD DEL JFRAME *****/

public void setVisible(boolean visibilidad)
{
    frmModificarPedido.setVisible(visibilidad);
}

/***** DISEÑO DE LA PÁGINA DE MODIFICAR PEDIDO *****/

```

```

@SuppressWarnings({ "rawtypes", "unchecked", "serial" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmModificarPedido = new JFrame();
    frmModificarPedido.setTitle("Modificar Pedido");

frmModificarPedido.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProducto.cl
ass.getResource("/Imágenes/Logotipo.png")));
    frmModificarPedido.setBounds(100, 100, 450, 300);
    frmModificarPedido.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmModificarPedido.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmModificarPedido.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmModificarPedido.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1600));
    panelGeneral.setBackground(new Color(255, 255, 255));
    scrollGeneral.setViewportViewView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmModificarPedido.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    panelGeneral.setLayout(null);
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    panelGeneral.add(etiquetaPaginaPrincipal);

    JLabel etiquetaDiagonal1 = new JLabel("/");
    etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
    etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal1.setBounds(313, 32, 22, 39);
    panelGeneral.add(etiquetaDiagonal1);

```

```

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmModificarPedido.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaPedidos = new JLabel("Pedidos");
etiquetaPedidos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPedidos pedidos = new MenuPedidos();

        pedidos.setVisible(true);
        frmModificarPedido.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPedidos.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPedidos.setForeground(new Color(0, 0, 255));
    }
});
etiquetaPedidos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPedidos.setForeground(Color.BLUE);
etiquetaPedidos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPedidos.setBounds(312, 31, 121, 41);
panelGeneral.add(etiquetaPedidos);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));

```



```

etiquetaDiagonal3.setBounds(409, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaModificarPedido = new JLabel("Modificar");
etiquetaModificarPedido.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaModificarPedido.setForeground(Color.BLUE);
etiquetaModificarPedido.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaModificarPedido.setBounds(390, 32, 170, 41);
panelGeneral.add(etiquetaModificarPedido);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox para ver la Lista de Pedidos *****/

JPanel groupBoxListaPedidos = new JPanel();
groupBoxListaPedidos.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Lista de Pedidos",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxListaPedidos.setBackground(new Color(243, 243, 243));
groupBoxListaPedidos.setBounds(250, 230, 850, 400);
groupBoxListaPedidos.setLayout(null);
panelGeneral.add(groupBoxListaPedidos);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds(180, 50, 131, 30);
groupBoxListaPedidos.add(etiquetaProveedor);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        while(modeloTabla.getRowCount()>0)
        {
            modeloTabla.removeRow(0);
        }

        if(!((String)cajaListaProveedores.getSelectedItem()).equals("ninguno"))
        {
            elProveedor =
operacionesProveedor.buscarProveedor(((String)cajaListaProveedores.getSelectedItem()));

```

```

        losPedidos =
operacionesPedido.traerPedidos(elProveedor.getRfcProveedor());
        for(int i=0;i<losPedidos.size();i++)
        {
            Object[] renglon =
{losPedidos.get(i).getFechaPedido(),losPedidos.get(i).getNumeroPedido()};
            modeloTabla.addRow(renglon);
        }
    }
});
cajaListaProveedores.setBounds(320, 50, 221, 36);
groupBoxListaPedidos.add(cajaListaProveedores);

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(106, 150, 377, 180);
scrollTabla.setHorizontalScrollBar(null);
groupBoxListaPedidos.add(scrollTabla);

modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
tablaPedidos = new JTable(modeloTabla);
tablaPedidos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
modeloTabla.addColumn("FECHA");
modeloTabla.addColumn("NÚMERO DE PEDIDO");
tablaPedidos.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaPedidos.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaPedidos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        fila = tablaPedidos.getSelectedRow();
        elPedido =
operacionesPedido.buscarPedidoPorNumero(Integer.valueOf(tablaPedidos.getValueAt(fila,1)).t
oString());
    }
});
scrollTabla.setViewportViewView(tablaPedidos);

/***** Formato y Funcionalidad de Botón "Modificar" *****/

JButton botonModificar = new JButton("  Modificar");
Border elBordeModificar = new LineBorder(Color.black,1);
botonModificar.setBorder(elBordeModificar);
botonModificar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imágenes/Modificar.png")));
botonModificar.setBackground(new Color(216,253,241));
botonModificar.setFont(new Font("Century", Font.PLAIN, 18));
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        fecha.setDate(elPedido.getFechaPedido());
        cajaCodigoBarras.setText(elPedido.getProductoCodigoBarras());
        cajaCodigoBarras.setEnabled(false);

etiquetaVerNombreProducto.setText(operacionesProducto.buscarProductoPorID(elPedido.getPro
ductoCodigoBarras()).getNombreProducto());

```

```

etiquetaVerRfcProveedor.setText(operacionesPedido.tracerProductoPedido(elPedido.getProduct
oCodigoBarras()).getProveedor());

cajaCantidadSolicitar.setText(Integer.toString(elPedido.getCantidadSolicitada()));

etiquetaVerCostoUnidad.setText(String.valueOf(operacionesProducto.buscarProductoPorID(elP
edido.getProductCodigoBarras()).getCostoCompra()));

etiquetaVerTotalLiquidar.setText(String.valueOf(elPedido.getTotalLiquidar()));

        for(int i=0;i<=cajaStatusPedido.getComponentCount()+1;i++)
        {
if((elPedido.getStatusPedido()).equals(cajaStatusPedido.getItemAt(i)))
        {
            if(elPedido.getStatusPedido().equals("Realizado"))
            {
                cajaCantidadSolicitar.setEnabled(false);
                cajaStatusPedido.setEnabled(false);
            }
            cajaStatusPedido.setSelectedIndex(i);
            break;
        }
        }
    });
    botonModificar.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonModificar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonModificar.setForeground(new Color(0,0,0));
        }
    });
    botonModificar.setBounds(576, 150, 183, 61);
    groupBoxListaPedidos.add(botonModificar);

/***** Formato y Funcionalidad de Botón "Eliminar" *****/

    JButton botonEliminar = new JButton("  Eliminar");
    Border elBordeEliminar = new LineBorder(Color.black,1);
    botonEliminar.setBorder(elBordeEliminar);
    botonEliminar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {

operacionesPedido.eliminarPedido(Integer.valueOf(tablaPedidos.getValueAt(fila,1).toString
()));

        fecha.setDate(null);
        cajaCodigoBarras.setText("");
        etiquetaVerNombreProducto.setText("");
        etiquetaVerRfcProveedor.setText("");
        cajaCantidadSolicitar.setText("");
        etiquetaVerCostoUnidad.setText("");
        etiquetaVerTotalLiquidar.setText("");
        cajaStatusPedido.setSelectedIndex(0);

        while(modeloTabla.getRowCount()>0)
        {
            modeloTabla.removeRow(0);

```

```

    }

    if(!((String)cajaListaProveedores.getSelectedItem()).equals(""))
    {
        elProveedor =
operacionesProveedor.buscarProveedor((String)cajaListaProveedores.getSelectedItem());
        losPedidos =
operacionesPedido.trazerPedidos(elProveedor.getRfcProveedor());

        for(int i=0;i<losPedidos.size();i++)
        {
            Object[] renglon =
{losPedidos.get(i).getFechaPedido(),losPedidos.get(i).getNumeroPedido()};
            modeloTabla.addRow(renglon);
        }
    }
    tablaPedidos.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            fila = tablaPedidos.getSelectedRow();
            elPedido =
operacionesPedido.buscarPedidoPorNumero(Integer.valueOf(tablaPedidos.getValueAt(fila,1).t
oString()));
        }
    });
}
});
botonEliminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonEliminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonEliminar.setForeground(new Color(0,0,0));
    }
});
botonEliminar.setBackground(new Color(216,253,241));
botonEliminar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Eliminar.png")));
botonEliminar.setFont(new Font("Century", Font.PLAIN, 18));
botonEliminar.setBounds(578, 260, 183, 61);
groupBoxListaPedidos.add(botonEliminar);

/***** GroupBox para ver los Datos de los Pedidos *****/

JPanel groupBoxDatosPedidos = new JPanel();
groupBoxDatosPedidos.setBackground(new Color(255,255,255));
groupBoxDatosPedidos.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Datos de los Pedidos",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxDatosPedidos.setBounds(100, 700, 1150, 720);
groupBoxDatosPedidos.setLayout(null);
panelGeneral.add(groupBoxDatosPedidos);

/***** Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
panelFormulario.setBounds(30, 30, 1088, 670);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 750));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);

```

```

groupBoxDatosPedidos.add(panelFormulario);

/***** Elementos del Formulario *****/

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 45, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCodigoBarras.setBounds(342, 120, 188, 30);
panelFormulario.add(etiquetaCodigoBarras);

cajaCodigoBarras = new JTextField();
cajaCodigoBarras.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            elProducto =
operacionesProducto.buscarProductoPorID(cajaCodigoBarras.getText());

            if(cajaCodigoBarras.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingrese un C\u00f3digo.");
            }
            else if(elProducto == null)
            {
                JOptionPane.showMessageDialog(null,"Este C\u00f3digo no existe en el
Sistema.");
            }
            else
            {
                etiquetaVerNombreProducto.setText(elProducto.getNombreProducto());
                etiquetaVerRfcProveedor.setText(elProducto.getProveedor());
                etiquetaVerCostoUnidad.setText(String.valueOf(elProducto.getCostoCompra()));
            }
        }
    }

    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCodigoBarras.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar C\u00f3digo",cajaCodigoBarras);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);

```

```

leyenda.changeStyle(Font.ITALIC);
cajaCodigoBarras.setBounds(536, 120, 184, 36);
cajaCodigoBarras.setColumns(10);
cajaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaCodigoBarras);

JLabel etiquetaNombreProducto = new JLabel("Producto:");
etiquetaNombreProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProducto.setBounds(296, 205, 236, 30);
panelFormulario.add(etiquetaNombreProducto);

etiquetaVerNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerNombreProducto.setBounds(536, 205, 250, 36);
panelFormulario.add(etiquetaVerNombreProducto);

JLabel etiquetaProveedor2 = new JLabel("Proveedor:");
etiquetaProveedor2.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor2.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor2.setBounds(399, 280, 131, 30);
panelFormulario.add(etiquetaProveedor2);

etiquetaVerRfcProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerRfcProveedor.setBounds(540, 280, 320, 36);
panelFormulario.add(etiquetaVerRfcProveedor);

JLabel etiquetaCantidadSolicitar = new JLabel("Cantidad a Solicitar:");
etiquetaCantidadSolicitar.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCantidadSolicitar.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCantidadSolicitar.setBounds(250, 350, 280, 36);
panelFormulario.add(etiquetaCantidadSolicitar);

cajaCantidadSolicitar = new JTextField();
cajaCantidadSolicitar.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Cantidad", cajaCantidadSolicitar);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCantidadSolicitar.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            if(cajaCantidadSolicitar.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null, "Ingrese una Cantidad.");
            }
            else
            {
                etiquetaVerTotalLiquidar.setText(String.valueOf(Float.parseFloat(cajaCantidadSolicitar.ge
tText()) * Float.valueOf(etiquetaVerCostoUnidad.getText())));
            }
        }
    }

    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
}

```

```

});
cajaCantidadSolicitar.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCantidadSolicitar.setColumns(10);
cajaCantidadSolicitar.setBounds(536, 350, 69, 36);
panelFormulario.add(cajaCantidadSolicitar);

JLabel etiquetaCostoUnidad = new JLabel("Costo por Unidad:");
etiquetaCostoUnidad.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoUnidad.setBounds(300, 435, 230, 30);
panelFormulario.add(etiquetaCostoUnidad);

etiquetaVerCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCostoUnidad.setBounds(540, 435, 95, 36);
panelFormulario.add(etiquetaVerCostoUnidad);

JLabel etiquetaTotalLiquidar = new JLabel("Costo Total:");
etiquetaTotalLiquidar.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTotalLiquidar.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTotalLiquidar.setBounds(325, 520, 200, 30);
panelFormulario.add(etiquetaTotalLiquidar);

etiquetaVerTotalLiquidar.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerTotalLiquidar.setBounds(540, 520, 95, 36);
panelFormulario.add(etiquetaVerTotalLiquidar);

JLabel etiquetaStatusDevolucion = new JLabel("Estatus:");
etiquetaStatusDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaStatusDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaStatusDevolucion.setBounds(440, 600, 90, 30);
panelFormulario.add(etiquetaStatusDevolucion);

cajaStatusPedido = new JComboBox();
cajaStatusPedido.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaStatusPedido.setModel(new DefaultComboBoxModel(new String[] {}));
cajaStatusPedido.addItem("Seleccione un Opción");
cajaStatusPedido.addItem("Cancelado");
cajaStatusPedido.addItem("Pendiente");
cajaStatusPedido.addItem("Realizado");
cajaStatusPedido.addItem("Rechazado");
cajaStatusPedido.setBounds(536, 600, 200, 36);
panelFormulario.add(cajaStatusPedido);

/***** Formato y Funcionalidad de Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuPedidos menuPedidos = new MenuPedidos();

        menuPedidos.setVisible(true);
        frmModificarPedido.setVisible(false);
    }
});
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(Color.BLACK);
botonTerminar.setBackground(new Color(234, 34, 34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(310, 1490, 210, 48);
panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad de Botón "Guardar" *****/

JButton botonGuardar = new JButton(" Guardar");

```

```

        botonGuardar.addActionListener(new ActionListener ()
        {
            boolean cajasVerificadas;

            public void actionPerformed(ActionEvent e)
            {
                cajasVerificadas = camposLlenos();

                if(cajasVerificadas == false)
                {
                    JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
                }
                else
                {
                    java.util.Date laFecha = fecha.getDate();
                    SimpleDateFormat cambiarFecha = new
SimpleDateFormat (fecha.getDateFormatString());

                    elPedido.setFechaPedido(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))));
                    elPedido.setProductoCodigoBarras(cajaCodigoBarras.getText());

                    elPedido.setCantidadSolicitada(Integer.valueOf(cajaCantidadSolicitar.getText()));

                    elPedido.setTotalLiquidar(Float.valueOf(etiquetaVerTotalLiquidar.getText()));
                    elPedido.setStatusPedido((String)cajaStatusPedido.getSelectedItem());

                    if(((String)cajaStatusPedido.getSelectedItem()).compareTo("Realizado")==0)
                    {
                        operacionesProducto.actualizarCantidadProductosAlmacenPedido(cajaCodigoBarras.getText(),I
nteger.valueOf(cajaCantidadSolicitar.getText()));
                    }

                    operacionesPedido.actualizarPedido(elPedido);

                    fecha.setDate(null);
                    cajaCodigoBarras.setText("");
                    etiquetaVerNombreProducto.setText("");
                    etiquetaVerRfcProveedor.setText("");
                    cajaCantidadSolicitar.setText("");
                    etiquetaVerCostoUnidad.setText("");
                    etiquetaVerTotalLiquidar.setText("");
                    cajaStatusPedido.setSelectedIndex(0);
                }
            }
        });
        botonGuardar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
        botonGuardar.setForeground(new Color(0, 0, 0));
        botonGuardar.setBackground(new Color(95,255,95));
        botonGuardar.setFont(new Font("Century", Font.PLAIN, 20));
        botonGuardar.setBounds(840, 1490, 210, 48);
        panelGeneral.add(botonGuardar);
    }
}

```


19.- ModificarProducto.java

```
package Interfaces;

/** Esta Interfaz permite modificar la información de un Producto o eliminarlo **/

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Producto;
import hibernate.operaciones.OpProducto;
import hibernate.operaciones.OpProveedor;

public class ModificarProducto
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmModificarProducto;
    private JPanel panelGeneral;

    /******* ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaNombreProducto,cajaCostoCompra,cajaCostoPublico;
    private JTextField cajaPuntos,cajaPuntosCompra,cajaCantidadIngreso;
    private JTextField cajaClasificacion,cajaCodigoBarras;
    private TextPrompt leyenda;
    @SuppressWarnings("rawtypes")
    private JComboBox cajaListaProveedores;
    private ButtonGroup losRadio;
    private JRadioButton radioIval,radioIva2;

    /******* ELEMENTOS DE LA TABLA DE PRODUCTOS *****/

    private JTable tablaProductos;
    private DefaultTableModel modeloTabla;
    private List<String> nombresProductos;
    private int fila;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProducto operacionesProducto = new OpProducto();
    private OpProveedor operacionesProveedor = new OpProveedor();
    private Producto elProducto = new Producto();

    /******* ELEMENTOS UTILIZADOS EN EL BOTÓN "Guardar" *****/

    private boolean cajasVerificadas;

    /******* MAIN *****/

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    ModificarProducto window = new ModificarProducto();
                    window.frmModificarProducto.setVisible(true);
                }
            }
        }
    }
}
```

```

            catch (Exception e)
            {
                e.printStackTrace();
            }
        });
    }

    /***** VISIBILIDAD DEL JFrame *****/

    public void setVisible(boolean visibilidad)
    {
        frmModificarProducto.setVisible(visibilidad);
    }

    /***** VALIDACIÓN DE CAMPOS VACÍOS *****/

    public boolean camposLlenos()
    {
        boolean verificacion;
        if (fecha.getDate() == null || cajaCodigoBarras.getText().isEmpty() ||
            cajaNombreProducto.getText().isEmpty() ||
                cajaCostoCompra.getText().isEmpty() || cajaCostoPublico.getText().isEmpty()
            || cajaPuntos.getText().isEmpty() ||
                cajaPuntosCompra.getText().isEmpty() ||
            cajaCantidadIngreso.getText().isEmpty() || cajaClasificacion.getText().isEmpty() ||
                cajaListaProveedores.getSelectedIndex() == 0 ||
            tablaProductos.getSelectedRow() == -1)
        {
            verificacion = false;
        }
        else
        {
            verificacion = true;
        }
        return verificacion;
    }

    /***** INICIALIZACIÓN GENERAL DEL JFrame *****/

    public ModificarProducto()
    {
        initialize();
    }

    /***** DISEÑO DE LA PÁGINA DE MODIFICAR PRODUCTO *****/

    @SuppressWarnings({ "rawtypes", "unchecked", "serial" })
    private void initialize()
    {
    /***** Formato del JFrame, Scroll y Página Principal *****/

        frmModificarProducto = new JFrame();

        frmModificarProducto.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmModificarProducto.setIconImage(Toolkit.getDefaultToolkit().getImage(ModificarProducto.class.getResource("/imagenes/Logotipo.png")));
        frmModificarProducto.setTitle("Modificar Producto");
        frmModificarProducto.setBounds(100, 100, 450, 300);
        frmModificarProducto.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frmModificarProducto.getContentPane().setLayout(null);

        JScrollPane scrollGeneral = new JScrollPane();
        scrollGeneral.setBackground(new Color(175, 238, 238));
        scrollGeneral.setBounds(0, 0, 1365, 715);
        scrollGeneral.setHorizontalScrollBar(null);
        frmModificarProducto.getContentPane().add(scrollGeneral);
    }
}

```

```

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1650));
panelGeneral.setBackground(new Color(255,255,255));
scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmModificarProducto.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmModificarProducto.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {

```

```

        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Productos");
etiquetaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        MenuProductos productos = new MenuProductos();

        productos.setVisible(true);
        frmModificarProducto.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(0, 0, 255));
    }
});
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(326, 31, 108, 41);
panelGeneral.add(etiquetaProductos);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(423, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaModificarProducto = new JLabel("Modificar");
etiquetaModificarProducto.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaModificarProducto.setForeground(Color.BLUE);
etiquetaModificarProducto.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaModificarProducto.setBounds(400, 31, 190, 41);
panelGeneral.add(etiquetaModificarProducto);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");

```

```

    etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
    etiquetaLogotipo.setBackground(new Color(175, 238, 238));
    etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaLogotipo.setBounds(1197, 35, 129, 104);
    panelGeneral.add(etiquetaLogotipo);

/***** GroupBox para ver los Productos Registrados *****/

    JPanel groupBoxTabla = new JPanel();
    groupBoxTabla.setForeground(Color.BLACK);
    groupBoxTabla.setBackground(new Color(255, 255, 255));
    groupBoxTabla.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Lista de Productos",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
    groupBoxTabla.setBounds(282, 201, 744, 297);
    groupBoxTabla.setLayout(null);
    panelGeneral.add(groupBoxTabla);

    JScrollPane scrollTabla = new JScrollPane();
    scrollTabla.setBounds(50, 55, 433, 180);
    scrollTabla.setHorizontalScrollBar(null);
    groupBoxTabla.add(scrollTabla);

    modeloTabla = new DefaultTableModel()
    {
        @Override
        public boolean isCellEditable(int row, int column)
        {
            return false;
        }
    };
    tablaProductos = new JTable(modeloTabla);
    tablaProductos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
    modeloTabla.addColumn("Nombre de Productos");
    nombresProductos = operacionesProducto.traerNombresProductos();
    for(int i=0;i<nombresProductos.size();i++)
    {
        Object[] renglon = {nombresProductos.get(i)};
        modeloTabla.addRow(renglon);
    }
    tablaProductos.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            fila = tablaProductos.getSelectedRow();
            elProducto =
operacionesProducto.buscarProducto(tablaProductos.getValueAt(fila,0).toString());
        }
    });
    scrollTabla.setViewportViewView(tablaProductos);

/***** Formato y Funcionalidad de Botón "Modificar" *****/

    JButton botonModificar = new JButton("  Modificar");
    Border elBordeModificar = new LineBorder(Color.black,1);
    botonModificar.setBorder(elBordeModificar);
    botonModificar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Modificar.png")));
    botonModificar.setBackground(new Color(216,253,241));
    botonModificar.setFont(new Font("Century", Font.PLAIN, 18));
    botonModificar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            fecha.setDate(elProducto.getFechaRegistro());
            cajaCodigoBarras.setText(elProducto.getCodigoBarras());
        }
    });

```

```

        cajaNombreProducto.setText (elProducto.getNombreProducto());
        cajaCostoCompra.setText (Float.toString (elProducto.getCostoCompra()));
        cajaCostoPublico.setText (Float.toString (elProducto.getCostoPublico()));
        cajaPuntos.setText (Integer.toString (elProducto.getPuntosAsignados()));
        cajaPuntosCompra.setText (Integer.toString (elProducto.getPuntosCompra()));
        String nombreProveedor = "";
        for (int i=0;i<=cajaListaProveedores.getItemCount()+1;i++)
        {
            nombreProveedor =
operacionesProveedor.buscarProveedorRfc (elProducto.getProveedor()) .getNombreProveedor ()
;
            if (nombreProveedor.equals (cajaListaProveedores.getItemAt (i)))
            {
                cajaListaProveedores.setSelectedIndex (i);
                break;
            }
        }
cajaCantidadIngreso.setText (Integer.toString (elProducto.getCantidadIngreso()));
cajaClasificacion.setText (elProducto.getClasificacion());
if (elProducto.getIva () != 0.00f)
{
    radioIva1.setSelected (true);
}
else
{
    radioIva2.setSelected (true);
}
});
botonModificar.addMouseListener (new MouseAdapter ()
{
    @Override
    public void mouseEntered (MouseEvent e)
    {
        botonModificar.setForeground (new Color (12,2,235));
    }
    @Override
    public void mouseExited (MouseEvent e)
    {
        botonModificar.setForeground (new Color (0,0,0));
    }
});
botonModificar.setBounds (521, 66, 183, 61);
groupBoxTabla.add (botonModificar);

/***** Formato y Funcionalidad de Botón "Eliminar" *****/

JButton botonEliminar = new JButton (" Eliminar");
Border elBordeEliminar = new LineBorder (Color.black,1);
botonEliminar.setBorder (elBordeEliminar);
botonEliminar.addActionListener (new ActionListener ()
{
    public void actionPerformed (ActionEvent arg0)
    {
operacionesProducto.eliminarProducto (tablaProductos.getValueAt (fila,0).toString());

        fecha.setDate (null);
        cajaCodigoBarras.setText ("");
        cajaNombreProducto.setText ("");
        cajaCostoCompra.setText ("");
        cajaCostoPublico.setText ("");
        cajaPuntos.setText ("");
        cajaPuntosCompra.setText ("");
        cajaListaProveedores.setSelectedIndex (0);
        cajaCantidadIngreso.setText ("");
        cajaClasificacion.setText ("");

```

```

radioIva1.setSelected(false);
radioIva2.setSelected(false);

while(modeloTabla.getRowCount()>0)
{
    modeloTabla.removeRow(0);
}
nombresProductos = operacionesProducto.tracerNombresProductos();
for(int i=0;i<nombresProductos.size();i++)
{
    Object[] renglon = {nombresProductos.get(i)};
    modeloTabla.addRow(renglon);
}
tablaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        fila = tablaProductos.getSelectedRow();
        elProducto =
operacionesProducto.buscarProducto(tablaProductos.getValueAt(fila,0).toString());
    }
});
});
botonEliminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonEliminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonEliminar.setForeground(new Color(0,0,0));
    }
});
botonEliminar.setBackground(new Color(216,253,241));
botonEliminar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imágenes/Eliminar.png")));
botonEliminar.setFont(new Font("Century", Font.PLAIN, 18));
botonEliminar.setBounds(521, 162, 183, 61);
groupBoxTabla.add(botonEliminar);

/***** GroupBox del Formulario *****/

JPanel groupBoxDatos = new JPanel();
groupBoxDatos.setBackground(new Color(255,255,255));
groupBoxDatos.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Datos de los Productos",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxDatos.setBounds(100, 549, 1150, 950);
groupBoxDatos.setLayout(null);
panelGeneral.add(groupBoxDatos);

/***** Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
panelFormulario.setBounds(30, 30, 1088, 900);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 750));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
groupBoxDatos.add(panelFormulario);

/***** Elementos del Formulario *****/

```

```

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 50, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 50, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCodigoBarras.setBounds(342, 130, 188, 30);
panelFormulario.add(etiquetaCodigoBarras);

cajaCodigoBarras = new JTextField();
cajaCodigoBarras.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCodigoBarras.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar C\u00f3digo", cajaCodigoBarras);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCodigoBarras.setBounds(536, 130, 184, 36);
cajaCodigoBarras.setColumns(10);
cajaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaCodigoBarras);

JLabel etiquetaNombreProducto = new JLabel("Nombre del Producto:");
etiquetaNombreProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProducto.setBounds(296, 210, 236, 30);
panelFormulario.add(etiquetaNombreProducto);

cajaNombreProducto = new JTextField();
cajaNombreProducto.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Nombre", cajaNombreProducto);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNombreProducto.setBounds(536, 210, 250, 36);
cajaNombreProducto.setColumns(10);
cajaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaNombreProducto);

JLabel etiquetaCostoCompra = new JLabel("Costo de Compra:");
etiquetaCostoCompra.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoCompra.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoCompra.setBounds(345, 290, 188, 30);
panelFormulario.add(etiquetaCostoCompra);

cajaCostoCompra = new JTextField();
cajaCostoCompra.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)

```



```

        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '.'))
            {
                e.consume();
            }
        }
    });
    cajaCostoCompra.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar Costo",cajaCostoCompra);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaCostoCompra.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    cajaCostoCompra.setColumns(10);
    cajaCostoCompra.setBounds(536, 290, 95, 36);
    panelFormulario.add(cajaCostoCompra);

    JLabel etiquetaCostoPublico = new JLabel("Costo al P\u00F0Fablico:");
    etiquetaCostoPublico.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaCostoPublico.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaCostoPublico.setBounds(356, 369, 177, 30);
    panelFormulario.add(etiquetaCostoPublico);

    cajaCostoPublico = new JTextField();
    cajaCostoPublico.addKeyListener(new KeyAdapter()
    {
        public void keyTyped(KeyEvent e)
        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '.'))
            {
                e.consume();
            }
        }
    });
    cajaCostoPublico.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar Costo",cajaCostoPublico);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaCostoPublico.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    cajaCostoPublico.setColumns(10);
    cajaCostoPublico.setBounds(536, 369, 95, 36);
    panelFormulario.add(cajaCostoPublico);

    JLabel etiquetaPuntos = new JLabel("Puntos Asignados:");
    etiquetaPuntos.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaPuntos.setBounds(334, 449, 197, 36);
    panelFormulario.add(etiquetaPuntos);

    cajaPuntos = new JTextField();
    cajaPuntos.addKeyListener(new KeyAdapter()
    {
        public void keyTyped(KeyEvent e)
        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
            {
                e.consume();
            }
        }
    });
    cajaPuntos.setHorizontalAlignment(SwingConstants.CENTER);

```

```

leyenda = new TextPrompt("Puntos",cajaPuntos);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaPuntos.setColumns(10);
cajaPuntos.setBounds(536, 449, 69, 36);
panelFormulario.add(cajaPuntos);

JLabel etiquetaPuntosCompra = new JLabel("Puntos Compra:");
etiquetaPuntosCompra.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaPuntosCompra.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaPuntosCompra.setBounds(334, 525, 197, 36);
panelFormulario.add(etiquetaPuntosCompra);

cajaPuntosCompra = new JTextField();
cajaPuntosCompra.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaPuntosCompra.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Puntos",cajaPuntosCompra);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaPuntosCompra.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaPuntosCompra.setColumns(10);
cajaPuntosCompra.setBounds(536, 525, 69, 36);
panelFormulario.add(cajaPuntosCompra);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds(399, 605, 131, 30);
panelFormulario.add(etiquetaProveedor);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione una Opción");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.setBounds(536, 605, 320, 36);
panelFormulario.add(cajaListaProveedores);

JLabel etiquetaCantidadIngreso = new JLabel("Cantidad de Ingreso:");
etiquetaCantidadIngreso.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCantidadIngreso.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCantidadIngreso.setBounds(310, 685, 221, 36);
panelFormulario.add(etiquetaCantidadIngreso);

cajaCantidadIngreso = new JTextField();
cajaCantidadIngreso.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {

```

```

        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCantidadIngreso.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Cantidad",cajaCantidadIngreso);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCantidadIngreso.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCantidadIngreso.setColumns(10);
cajaCantidadIngreso.setBounds(536, 685, 69, 36);
panelFormulario.add(cajaCantidadIngreso);

JLabel etiquetaClasificacion = new JLabel("Clasificación:");
etiquetaClasificacion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaClasificacion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaClasificacion.setBounds(345, 765, 188, 36);
panelFormulario.add(etiquetaClasificacion);

cajaClasificacion = new JTextField();
cajaClasificacion.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Tipo de Producto",cajaClasificacion);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaClasificacion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaClasificacion.setColumns(10);
cajaClasificacion.setBounds(536, 765, 197, 36);
panelFormulario.add(cajaClasificacion);

JLabel etiquetaIVA = new JLabel("I.V.A.:");
etiquetaIVA.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaIVA.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaIVA.setBounds(445, 845, 87, 36);
panelFormulario.add(etiquetaIVA);

radioIva1 = new JRadioButton("Agregar IVA 16%");
radioIva1.setBackground(new Color(243, 243, 243));
radioIva1.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioIva1.setHorizontalAlignment(SwingConstants.CENTER);
radioIva1.setBounds(536, 845, 184, 36);
panelFormulario.add(radioIva1);

radioIva2 = new JRadioButton("No Agregar IVA");
radioIva2.setBackground(new Color(243, 243, 243));
radioIva2.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioIva2.setHorizontalAlignment(SwingConstants.CENTER);
radioIva2.setBounds(730, 845, 177, 36);
panelFormulario.add(radioIva2);

losRadio = new ButtonGroup();
losRadio.add(radioIva1);
losRadio.add(radioIva2);

/***** Formato y Funcionalidad de Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuProductos menuProductos = new MenuProductos();
    }
});

```

```

        menuProductos.setVisible(true);
        frmModificarProducto.setVisible(false);
    }
});
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(Color.BLACK);
botonTerminar.setBackground(new Color(234,34,34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(310, 1550, 210, 48);
panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad de Botón "Guardar" *****/

JButton botonGuardar = new JButton("  Guardar");
botonGuardar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        cajasVerificadas = camposLlenos();

        if(cajasVerificadas == false)
        {
            JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
        }
        else
        {
            DecimalFormat limitarDecimales = new DecimalFormat("#.00");
            java.util.Date laFecha = fecha.getDate();
            SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

elProducto.setFechaRegistro(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))););
elProducto.setCodigoBarras(cajaCodigoBarras.getText());
elProducto.setNombreProducto(cajaNombreProducto.getText());
elProducto.setCostoCompra(Float.valueOf(cajaCostoCompra.getText()));

elProducto.setCostoPublico(Float.valueOf(cajaCostoPublico.getText()));
elProducto.setPuntosAsignados(Integer.valueOf(cajaPuntos.getText()));

elProducto.setPuntosCompra(Integer.valueOf(cajaPuntosCompra.getText()));

elProducto.setProveedor(operacionesProveedor.buscarProveedor((String)cajaListaProveedores
.getItemAt(cajaListaProveedores.getSelectedIndex()).getRfcProveedor());

elProducto.setCantidadIngreso(Integer.valueOf(cajaCantidadIngreso.getText()));
elProducto.setClasificacion(cajaClasificacion.getText());
if (radioIva1.isSelected())
{
elProducto.setIva(Float.valueOf(limitarDecimales.format(Float.valueOf(cajaCostoPublico.ge
tText()) * 0.16f)));
}
else if (radioIva2.isSelected())
{
    elProducto.setIva(Float.valueOf(limitarDecimales.format(0f)));
}

operacionesProducto.actualizarProducto(elProducto);

fecha.setDate(null);
cajaCodigoBarras.setText("");
cajaNombreProducto.setText("");
cajaCostoCompra.setText("");
cajaCostoPublico.setText("");

```

```

        cajaPuntos.setText("");
        cajaPuntosCompra.setText("");
        cajaListaProveedores.setSelectedIndex(0);
        cajaCantidadIngreso.setText("");
        cajaClasificacion.setText("");
        losRadio.clearSelection();
    }
}
});
botonGuardar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
botonGuardar.setForeground(new Color(0, 0, 0));
botonGuardar.setBackground(new Color(95,255,95));
botonGuardar.setFont(new Font("Century", Font.PLAIN, 20));
botonGuardar.setBounds(840, 1550, 210, 48);
panelGeneral.add(botonGuardar);
}
}
}

```

20.- ModificarProveedor.java

```

package Interfaces;

/** Esta Interfaz permite modificar la información de un Proveedor o eliminarlo */

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.List;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Proveedor;
import hibernate.operaciones.OpProveedor;

public class ModificarProveedor
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmModificarProveedor;
    private JPanel panelGeneral;

    /******* ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaNombreProveedor,cajaTelefono;
    private JTextField cajaCorreo,cajaContacto,cajaRFC;
    private JTextPane panelTextoDireccion;
    private TextPrompt leyenda;

    /******* ELEMENTOS DE LA TABLA DE PRODUCTOS *****/

    private JTable tablaProveedores;
    private DefaultTableModel modeloTabla;
    private int fila;
    private List<String> losNombresProveedores;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
    private Proveedor elProveedor;

    /******* MAIN *****/

    public static void main(String[] args)

```

```

    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    ModificarProveedor window = new ModificarProveedor();
                    window.frmModificarProveedor.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

/***** VISIBILIDAD DEL JFrame *****/

    public void setVisible(boolean visibilidad)
    {
        frmModificarProveedor.setVisible(visibilidad);
    }

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

    public boolean camposLlenos()
    {
        boolean verificacion;
        if (fecha.getDate() == null || cajaRFC.getText().isEmpty() ||
            cajaNombreProveedor.getText().isEmpty() ||
            panelTextoDireccion.getText().isEmpty() || cajaCorreo.getText().isEmpty() ||
            cajaTelefono.getText().isEmpty() ||
            cajaContacto.getText().isEmpty())
        {
            verificacion = false;
        }
        else
        {
            verificacion = true;
        }
        return verificacion;
    }

/***** INICIALIZACIÓN GENERAL DEL JFrame *****/

    public ModificarProveedor()
    {
        initialize();
    }

/***** DISEÑO DE LA PÁGINA DE MODIFICAR PRODUCTO *****/

    @SuppressWarnings("serial")
    private void initialize()
    {
/***** Formato del JFrame, Scroll y Página Principal *****/

        frmModificarProveedor = new JFrame();
        frmModificarProveedor.getContentPane().setBackground(new Color(175, 238, 238));

        frmModificarProveedor.setIconImage(Toolkit.getDefaultToolkit().getImage(ModificarProveedor.class.getResource("/Imágenes/Logotipo.png")));
        frmModificarProveedor.setTitle("Modificar Proveedor");
        frmModificarProveedor.setBounds(100, 100, 450, 300);
        frmModificarProveedor.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frmModificarProveedor.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

```

```

frmModificarProveedor.getContentPane().setLayout(null);

JScrollPane scrollGeneral = new JScrollPane();
scrollGeneral.setBackground(new Color(175, 238, 238));
scrollGeneral.setBounds(0, 0, 1365, 715);
scrollGeneral.setHorizontalScrollBar(null);
frmModificarProveedor.getContentPane().add(scrollGeneral);

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new Dimension(1366, 1540));
panelGeneral.setBackground(new Color(255, 255, 255));
scrollGeneral.setViewportViewView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmModificarProveedor.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmModificarProveedor.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)

```

```

        {
            etiquetaAlmacen.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaAlmacen.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaAlmacen.setBounds(233, 32, 85, 41);
    panelGeneral.add(etiquetaAlmacen);

    JLabel etiquetaDiagonal2 = new JLabel("/");
    etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal2.setForeground(Color.BLUE);
    etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal2.setBounds(214, 32, 22, 39);
    panelGeneral.add(etiquetaDiagonal2);

    JLabel etiquetaProveedores = new JLabel("Proveedores");
    etiquetaProveedores.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent e)
        {
            MenuProveedores proveedores = new MenuProveedores();

            proveedores.setVisible(true);
            frmModificarProveedor.setVisible(false);
        }
        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaProveedores.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaProveedores.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaProveedores.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaProveedores.setForeground(Color.BLUE);
    etiquetaProveedores.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaProveedores.setBounds(326, 31, 123, 41);
    panelGeneral.add(etiquetaProveedores);

    JLabel etiquetaDiagonal3 = new JLabel("/");
    etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal3.setForeground(Color.BLUE);
    etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal3.setBounds(438, 31, 22, 39);
    panelGeneral.add(etiquetaDiagonal3);

    JLabel etiquetaModificarProveedor = new JLabel("Modificar");
    etiquetaModificarProveedor.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaModificarProveedor.setForeground(Color.BLUE);
    etiquetaModificarProveedor.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaModificarProveedor.setBounds(400, 32, 210, 41);
    panelGeneral.add(etiquetaModificarProveedor);

    /***** Nombre del Negocio y Logotipo *****/

    JLabel etiquetaNegocio = new JLabel("Mi Negocio");
    etiquetaNegocio.setForeground(new Color(19, 60, 245));

```



```

etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** GroupBox con Tabla para ver los Proveedores Registrados *****/

JPanel groupBoxTablaProveedores = new JPanel();
groupBoxTablaProveedores.setForeground(Color.BLACK);
groupBoxTablaProveedores.setBackground(new Color(255,255,255));
groupBoxTablaProveedores.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Lista de Proveedores",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxTablaProveedores.setBounds(282, 201, 744, 280);
groupBoxTablaProveedores.setLayout(null);
panelGeneral.add(groupBoxTablaProveedores);

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(50, 55, 433, 180);
scrollTabla.setHorizontalScrollBar(null);
groupBoxTablaProveedores.add(scrollTabla);

modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
tablaProveedores = new JTable(modeloTabla);
tablaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
modeloTabla.addColumn("Nombre de Proveedores");
losNombresProveedores = operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    Object[] renglon = {losNombresProveedores.get(i)};
    modeloTabla.addRow(renglon);
}
tablaProveedores.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        fila = tablaProveedores.getSelectedRow();
        elProveedor =
operacionesProveedor.buscarProveedor(tablaProveedores.getValueAt(fila,0).toString());
    }
});
scrollTabla.setViewportViewView(tablaProveedores);

/**** GroupBox para ver los Datos de los Proveedores Registrados *****/

JPanel groupBoxDatosProveedores = new JPanel();
groupBoxDatosProveedores.setBackground(new Color(255,255,255));
groupBoxDatosProveedores.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Datos de los Proveedores",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
groupBoxDatosProveedores.setBounds(100, 549, 1150, 800);
groupBoxDatosProveedores.setLayout(null);
panelGeneral.add(groupBoxDatosProveedores);

```

```

/***** Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
panelFormulario.setBounds(30, 30, 1088, 745);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 750));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
groupBoxDatosProveedores.add(panelFormulario);

/***** Elementos del Formulario *****/

JLabel etiquetaRFC = new JLabel("R.F.C.");
etiquetaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaRFC.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaRFC.setBounds(461, 55, 69, 30);
panelFormulario.add(etiquetaRFC);

cajaRFC = new JTextField();
cajaRFC.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar RFC",cajaRFC);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaRFC.setBounds(536, 55, 169, 36);
cajaRFC.setColumns(10);
cajaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaRFC);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 140, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 140, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaNombreProveedor = new JLabel("Nombre del Proveedor:");
etiquetaNombreProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProveedor.setBounds(273, 225, 257, 30);
panelFormulario.add(etiquetaNombreProveedor);

cajaNombreProveedor = new JTextField();
leyenda = new TextPrompt("Ingresar Nombre",cajaNombreProveedor);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNombreProveedor.setBounds(536, 225, 230, 36);
cajaNombreProveedor.setColumns(10);
cajaNombreProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaNombreProveedor);

JLabel etiquetaDireccin = new JLabel("Direcci\u00F3n:");
etiquetaDireccin.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaDireccin.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaDireccin.setBounds(421, 365, 109, 30);
panelFormulario.add(etiquetaDireccin);

panelTextoDireccion = new JTextPane();
leyenda = new TextPrompt("Ingresar Direcci\u00F3n",panelTextoDireccion);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);

```

```

leyenda.changeStyle(Font.ITALIC);
panelTextoDireccion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelTextoDireccion.setBounds(536, 320, 229, 88);
panelFormulario.add(panelTextoDireccion);

JLabel etiquetaTelefono = new JLabel("Tel\u00E9fono:");
etiquetaTelefono.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTelefono.setBounds(421, 477, 109, 30);
panelFormulario.add(etiquetaTelefono);

cajaTelefono = new JTextField();
leyenda = new TextPrompt("(55) 55-55-55-55",cajaTelefono);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaTelefono.setHorizontalAlignment(SwingConstants.CENTER);
cajaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaTelefono.setColumns(10);
cajaTelefono.setBounds(536, 477, 155, 36);
panelFormulario.add(cajaTelefono);

JLabel etiquetaCorreo = new JLabel("Correo:");
etiquetaCorreo.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCorreo.setBounds(445, 568, 85, 30);
panelFormulario.add(etiquetaCorreo);

cajaCorreo = new JTextField();
leyenda = new TextPrompt("correo@correo.com",cajaCorreo);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCorreo.setHorizontalAlignment(SwingConstants.CENTER);
cajaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCorreo.setColumns(10);
cajaCorreo.setBounds(536, 568, 273, 36);
panelFormulario.add(cajaCorreo);

JLabel etiquetaContacto = new JLabel("Contacto:");
etiquetaContacto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaContacto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaContacto.setBounds(416, 660, 119, 30);
panelFormulario.add(etiquetaContacto);

cajaContacto = new JTextField();
leyenda = new TextPrompt("Ingresar Nombre",cajaContacto);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaContacto.setHorizontalAlignment(SwingConstants.CENTER);
cajaContacto.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaContacto.setColumns(10);
cajaContacto.setBounds(537, 660, 272, 36);
panelFormulario.add(cajaContacto);

/***** Formato y Funcionalidad de Bot\u00f3n "Modificar" *****/

JButton botonModificar = new JButton("  Modificar");
Border elBordeModificar = new LineBorder(Color.black,1);
botonModificar.setBorder(elBordeModificar);
botonModificar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Modificar.png")));
botonModificar.setBackground(new Color(216,253,241));
botonModificar.setFont(new Font("Century", Font.PLAIN, 18));
botonModificar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)

```

```

        {
            if(tablaProveedores.getSelectedRow() == -1)
            {
                JOptionPane.showMessageDialog(null,"Seleccione un Proveedor.");
            }
            else
            {
                cajaRFC.setText(elProveedor.getRfcProveedor());
                fecha.setDate(elProveedor.getFechaRegistro());
                cajaNombreProveedor.setText(elProveedor.getNombreProveedor());
                panelTextoDireccion.setText(elProveedor.getDireccion());
                cajaTelefono.setText(elProveedor.getTelefono());
                cajaCorreo.setText(elProveedor.getCorreo());
                cajaContacto.setText(elProveedor.getNombreContacto());

                tablaProveedores.clearSelection();
            }
        }
    });
    botonModificar.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonModificar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonModificar.setForeground(new Color(0,0,0));
        }
    });
    botonModificar.setBounds(521, 66, 183, 61);
    groupBoxTablaProveedores.add(botonModificar);

/***** Formato y Funcionalidad de Botón "Eliminar" *****/

    JButton botonEliminar = new JButton(" Eliminar");
    Border elBordeEliminar = new LineBorder(Color.black,1);
    botonEliminar.setBorder(elBordeEliminar);
    botonEliminar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            if(tablaProveedores.getSelectedRow() == -1)
            {
                JOptionPane.showMessageDialog(null,"Seleccione un Proveedor.");
            }
            else
            {
operacionesProveedor.eliminarProveedor(tablaProveedores.getValueAt(fila,0).toString());

                fecha.setDate(null);
                cajaRFC.setText("");
                cajaNombreProveedor.setText("");
                panelTextoDireccion.setText("");
                cajaTelefono.setText("");
                cajaCorreo.setText("");
                cajaContacto.setText("");

                while(modeloTabla.getRowCount()>0)
                {
                    modeloTabla.removeRow(0);
                }
                losNombresProveedores =
operacionesProveedor.trainerNombresProveedores();
                for(int i=0;i<losNombresProveedores.size();i++)

```

```

        {
            Object[] renglon = {losNombresProveedores.get(i)};
            modeloTabla.addRow(renglon);
        }
        tablaProveedores.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent arg0)
            {
                fila = tablaProveedores.getSelectedRow();
                elProveedor =
operacionesProveedor.buscarProveedor(tablaProveedores.getValueAt(fila,0).toString());
            }
        });
    }
});
botonEliminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonEliminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonEliminar.setForeground(new Color(0,0,0));
    }
});
botonEliminar.setBackground(new Color(216,253,241));
botonEliminar.setIcon(new
ImageIcon(ModificarProducto.class.getResource("/Imagenes/Eliminar.png")));
botonEliminar.setFont(new Font("Century", Font.PLAIN, 18));
botonEliminar.setBounds(521, 162, 183, 61);
groupBoxTablaProveedores.add(botonEliminar);

/***** Formato y Funcionalidad de Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuProveedores menuProveedores = new MenuProveedores();

        menuProveedores.setVisible(true);
        frmModificarProveedor.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0,0,0));
    }
});
Border elBordeTerminar = new LineBorder(Color.black,1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));

```

```

        botonTerminar.setForeground(Color.BLACK);
        botonTerminar.setBackground(new Color(234,34,34));
        botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
        botonTerminar.setBounds(310, 1400, 210, 50);
        panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad de Botón "Guardar" *****/

        JButton botonGuardar = new JButton("  Guardar");
        botonGuardar.addActionListener(new ActionListener ()
        {
            boolean cajasVerificadas;

            public void actionPerformed(ActionEvent e)
            {
                cajasVerificadas = camposLlenos();
                java.util.Date laFecha = fecha.getDate();
                SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

                if(cajasVerificadas == false)
                {
                    JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
                }
                else
                {
                    elProveedor.setRfcProveedor(cajaRFC.getText());

elProveedor.setFechaRegistro(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))));
                    elProveedor.setNombreProveedor(cajaNombreProveedor.getText());
                    elProveedor.setDireccion(panelTextoDireccion.getText());
                    elProveedor.setTelefono(cajaTelefono.getText());
                    elProveedor.setCorreo(cajaCorreo.getText());
                    elProveedor.setNombreContacto(cajaContacto.getText());

                    operacionesProveedor.actualizarProveedor(elProveedor);

                    fecha.setDate(null);
                    cajaRFC.setText("");
                    cajaNombreProveedor.setText("");
                    panelTextoDireccion.setText("");
                    cajaTelefono.setText("");
                    cajaCorreo.setText("");
                    cajaContacto.setText("");

                    while(modeloTabla.getRowCount()>0)
                    {
                        modeloTabla.removeRow(0);
                    }
                    losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
                    for(int i=0;i<losNombresProveedores.size();i++)
                    {
                        Object[] renglon = {losNombresProveedores.get(i)};
                        modeloTabla.addRow(renglon);
                    }
                    tablaProveedores.addMouseListener(new MouseAdapter ()
                    {
                        @Override
                        public void mouseClicked(MouseEvent arg0)
                        {
                            {
                                fila = tablaProveedores.getSelectedRow();
                                elProveedor =
operacionesProveedor.buscarProveedor(tablaProveedores.getValueAt(fila,0).toString());
                            }
                        }
                    });
                }
            }
        }

```

```

    }
    });
    botonGuardar.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonGuardar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonGuardar.setForeground(new Color(0,0,0));
        }
    });
    Border elBordeGuardar = new LineBorder(Color.black,1);
    botonGuardar.setBorder(elBordeGuardar);
    botonGuardar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imágenes/Palomita.png")));
    botonGuardar.setForeground(new Color(0, 0, 0));
    botonGuardar.setBackground(new Color(95,255,95));
    botonGuardar.setFont(new Font("Century", Font.PLAIN, 20));
    botonGuardar.setBounds(840, 1400, 210, 50);
    panelGeneral.add(botonGuardar);
}
}
}

```

21.- NuevoProducto.java

```

package Interfaces;

/** Esta Interfaz permite registrar la información de un Nuevo Producto **/

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.*;
import java.util.List;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Producto;
import hibernate.operaciones.OpProducto;
import hibernate.operaciones.OpProveedor;

public class NuevoProducto
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmNuevoProducto;
    private JPanel panelGeneral;

    /******* ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaCodigoBarras,cajaNombreProducto;
    private JTextField cajaCostoCompra,cajaCostoPublico,cajaPuntos;
    private JTextField cajaPuntosCompra,cajaCantidadIngreso,cajaClasificacion;
    private TextPrompt leyenda;
    @SuppressWarnings("rawtypes")
    private JComboBox cajaListaProveedores;
    private ButtonGroup losRadio;
    private JRadioButton radioIva1,radioIva2;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();

```

```

private OpProducto operacionesProducto = new OpProducto();

/***** ELEMENTOS UTILIZADOS EN EL BOTÓN "Registrar" *****/

private Producto elProducto = new Producto();
private boolean cajasVerificadas;

/***** MAIN *****/

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                NuevoProducto window = new NuevoProducto();
                window.frmNuevoProducto.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFRAME *****/

public void setVisible(boolean visibilidad)
{
    frmNuevoProducto.setVisible(visibilidad);
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if (fecha.getDate() == null || cajaCodigoBarras.getText().isEmpty() ||
    cajaNombreProducto.getText().isEmpty() ||
        cajaCostoCompra.getText().isEmpty() || cajaCostoPublico.getText().isEmpty()
    || cajaPuntos.getText().isEmpty() ||
        cajaPuntosCompra.getText().isEmpty() ||
    cajaCantidadIngreso.getText().isEmpty() || cajaClasificacion.getText().isEmpty() ||
        cajaListaProveedores.getSelectedIndex() == 0)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** INICIALIZACIÓN GENERAL DEL JFRAME *****/

public NuevoProducto()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA PARA NUEVO PRODUCTO *****/

@SuppressWarnings({ "rawtypes", "unchecked" })
private void initialize()

```



```

    {
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmNuevoProducto = new JFrame();
    frmNuevoProducto.getContentPane().setBackground(new Color(175, 238, 238));

frmNuevoProducto.setIconImage(Toolkit.getDefaultToolkit().getImage(MenuAlmacen.class.getResource("/Imágenes/Logotipo.png")));
    frmNuevoProducto.setTitle("Nuevo Producto");
    frmNuevoProducto.setSize(Toolkit.getDefaultToolkit().getScreenSize().width,
4000);
    frmNuevoProducto.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmNuevoProducto.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmNuevoProducto.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmNuevoProducto.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new Dimension(1366, 1300));
    panelGeneral.setBackground(new Color(255, 255, 255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmNuevoProducto.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    panelGeneral.setLayout(null);
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    panelGeneral.add(etiquetaPaginaPrincipal);

    JLabel etiquetaDiagonal1 = new JLabel("/");
    etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
    etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal1.setBounds(313, 32, 22, 39);
    panelGeneral.add(etiquetaDiagonal1);

    JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");

```

```

etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmNuevoProducto.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Productos");
etiquetaProductos.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuProductos productos = new MenuProductos();

        productos.setVisible(true);
        frmNuevoProducto.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaProductos.setForeground(new Color(0, 0, 255));
    }
});
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(326, 31, 108, 41);
panelGeneral.add(etiquetaProductos);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));

```

```

etiquetaDiagonal3.setBounds(423, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaNuevoProducto = new JLabel("Nuevo");
etiquetaNuevoProducto.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaNuevoProducto.setForeground(Color.BLUE);
etiquetaNuevoProducto.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaNuevoProducto.setBounds(390, 31, 163, 41);
panelGeneral.add(etiquetaNuevoProducto);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** Formato del Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
Border elBordeFormulario = new LineBorder(Color.black,1);
panelFormulario.setBorder(elBordeFormulario);
panelFormulario.setBounds(125, 250, 1088, 900);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 980));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
panelGeneral.add(panelFormulario);

/***** Funcionalidad y Elementos del Formulario *****/

JLabel textoIndicacion = new JLabel("Capture la informaci\u00F3n que se solicita
a continuaci\u00F3n:");
textoIndicacion.setFont(new Font("Book Antiqua", Font.PLAIN, 25));
textoIndicacion.setHorizontalAlignment(SwingConstants.CENTER);
textoIndicacion.setBounds(100, 200, 595, 32);
panelGeneral.add(textoIndicacion);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 45, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCodigoBarras.setBounds(342, 120, 188, 30);
panelFormulario.add(etiquetaCodigoBarras);

cajaCodigoBarras = new JTextField();
cajaCodigoBarras.addKeyListener(new KeyAdapter()
{

```

```

        public void keyTyped(KeyEvent e)
        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
            {
                e.consume();
            }
        }
    });
    cajaCodigoBarras.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar C\u00f3digo",cajaCodigoBarras);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaCodigoBarras.setBounds(536, 120, 184, 36);
    cajaCodigoBarras.setColumns(10);
    cajaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    panelFormulario.add(cajaCodigoBarras);

    JLabel etiquetaNombreProducto = new JLabel("Nombre del Producto:");
    etiquetaNombreProducto.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaNombreProducto.setBounds(296, 205, 236, 30);
    panelFormulario.add(etiquetaNombreProducto);

    cajaNombreProducto = new JTextField();
    cajaNombreProducto.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar Nombre",cajaNombreProducto);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaNombreProducto.setBounds(536, 205, 250, 36);
    cajaNombreProducto.setColumns(10);
    cajaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    panelFormulario.add(cajaNombreProducto);

    JLabel etiquetaCostoCompra = new JLabel("Costo de Compra:");
    etiquetaCostoCompra.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaCostoCompra.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaCostoCompra.setBounds(345, 285, 188, 30);
    panelFormulario.add(etiquetaCostoCompra);

    cajaCostoCompra = new JTextField();
    cajaCostoCompra.addKeyListener(new KeyAdapter()
    {
        public void keyTyped(KeyEvent e)
        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '.'))
            {
                e.consume();
            }
        }
    });
    cajaCostoCompra.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar Costo",cajaCostoCompra);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaCostoCompra.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    cajaCostoCompra.setColumns(10);
    cajaCostoCompra.setBounds(536, 285, 95, 36);
    panelFormulario.add(cajaCostoCompra);

    JLabel etiquetaCostoPublico = new JLabel("Costo al P\u00fablico:");
    etiquetaCostoPublico.setHorizontalAlignment(SwingConstants.RIGHT);

```

```

etiquetaCostoPublico.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoPublico.setBounds(356, 364, 177, 30);
panelFormulario.add(etiquetaCostoPublico);

cajaCostoPublico = new JTextField();
cajaCostoPublico.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '.'))
        {
            e.consume();
        }
    }
});
cajaCostoPublico.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Costo",cajaCostoPublico);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCostoPublico.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCostoPublico.setColumns(10);
cajaCostoPublico.setBounds(536, 364, 95, 36);
panelFormulario.add(cajaCostoPublico);

JLabel etiquetaPuntos = new JLabel("Puntos Asignados:");
etiquetaPuntos.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaPuntos.setBounds(334, 444, 197, 36);
panelFormulario.add(etiquetaPuntos);

cajaPuntos = new JTextField();
cajaPuntos.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaPuntos.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Puntos",cajaPuntos);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaPuntos.setColumns(10);
cajaPuntos.setBounds(536, 444, 69, 36);
panelFormulario.add(cajaPuntos);

JLabel etiquetaPuntosCompra = new JLabel("Puntos Compra:");
etiquetaPuntosCompra.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaPuntosCompra.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaPuntosCompra.setBounds(334, 525, 197, 36);
panelFormulario.add(etiquetaPuntosCompra);

cajaPuntosCompra = new JTextField();
cajaPuntosCompra.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

```

```

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaPuntosCompra.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Puntos",cajaPuntosCompra);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaPuntosCompra.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaPuntosCompra.setColumns(10);
cajaPuntosCompra.setBounds(536, 525, 69, 36);
panelFormulario.add(cajaPuntosCompra);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds(399, 605, 131, 30);
panelFormulario.add(etiquetaProveedor);

cajaListaProveedores = new JComboBox();
cajaListaProveedores.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaListaProveedores.setModel(new DefaultComboBoxModel(new String[] {}));
cajaListaProveedores.addItem("Seleccione un Proveedor");
List<String> losNombresProveedores =
operacionesProveedor.tracerNombresProveedores();
for(int i=0;i<losNombresProveedores.size();i++)
{
    cajaListaProveedores.addItem(losNombresProveedores.get(i));
}
cajaListaProveedores.setBounds(536, 605, 320, 36);
panelFormulario.add(cajaListaProveedores);

JLabel etiquetaCantidadIngreso = new JLabel("Cantidad de Ingreso:");
etiquetaCantidadIngreso.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCantidadIngreso.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCantidadIngreso.setBounds(310, 685, 221, 36);
panelFormulario.add(etiquetaCantidadIngreso);

cajaCantidadIngreso = new JTextField();
cajaCantidadIngreso.addKeyListener(new KeyAdapter()
{
    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
}));
cajaCantidadIngreso.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Cantidad",cajaCantidadIngreso);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCantidadIngreso.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCantidadIngreso.setColumns(10);
cajaCantidadIngreso.setBounds(536, 685, 69, 36);
panelFormulario.add(cajaCantidadIngreso);

JLabel etiquetaClasificacion = new JLabel("Clasificación:");
etiquetaClasificacion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaClasificacion.setFont(new Font("Cambria Math", Font.PLAIN, 25));

```

```

etiquetaClasificacion.setBounds(345, 765, 188, 36);
panelFormulario.add(etiquetaClasificacion);

cajaClasificacion = new JTextField();
cajaClasificacion.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Tipo de Producto", cajaClasificacion);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaClasificacion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaClasificacion.setColumns(10);
cajaClasificacion.setBounds(536, 765, 197, 36);
panelFormulario.add(cajaClasificacion);

JLabel etiquetaIVA = new JLabel("I.V.A.");
etiquetaIVA.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaIVA.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaIVA.setBounds(445, 845, 87, 36);
panelFormulario.add(etiquetaIVA);

radioIva1 = new JRadioButton("Agregar IVA 16%");
radioIva1.setBackground(new Color(243, 243, 243));
radioIva1.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioIva1.setHorizontalAlignment(SwingConstants.CENTER);
radioIva1.setBounds(536, 845, 184, 36);
radioIva1.setSelected(true);
panelFormulario.add(radioIva1);

radioIva2 = new JRadioButton("No Agregar IVA");
radioIva2.setBackground(new Color(243, 243, 243));
radioIva2.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioIva2.setHorizontalAlignment(SwingConstants.CENTER);
radioIva2.setBounds(730, 845, 177, 36);
panelFormulario.add(radioIva2);

losRadio = new ButtonGroup();
losRadio.add(radioIva1);
losRadio.add(radioIva2);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuProductos menuProductos = new MenuProductos();

        menuProductos.setVisible(true);
        frmNuevoProducto.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12, 2, 235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0, 0, 0));
    }
});
Border elBordeTerminar = new LineBorder(Color.black, 1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);

```

```

        botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
        botonTerminar.setForeground(new Color(0,0,0));
        botonTerminar.setBackground(new Color(234,34,34));
        botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
        botonTerminar.setBounds(300, 1200, 197, 50);
        panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad del Botón "Registrar" *****/

        JButton botonRegistrar = new JButton(" Registrar");
        botonRegistrar.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                cajasVerificadas = camposLlenos();

                if(cajasVerificadas == false)
                {
                    JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
                }
                else
                {
                    DecimalFormat limitarDecimales = new DecimalFormat("#.00");
                    java.util.Date laFecha = fecha.getDate();
                    SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

                    elProducto.setFechaRegistro(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))););
                    elProducto.setCodigoBarras(cajaCodigoBarras.getText());
                    elProducto.setNombreProducto(cajaNombreProducto.getText());
                    elProducto.setCostoCompra(Float.valueOf(cajaCostoCompra.getText()));

                    elProducto.setCostoPublico(Float.valueOf(cajaCostoPublico.getText()));
                    elProducto.setPuntosAsignados(Integer.valueOf(cajaPuntos.getText()));

                    elProducto.setPuntosCompra(Integer.valueOf(cajaPuntosCompra.getText()));

                    elProducto.setProveedor(operacionesProveedor.buscarProveedor((String)cajaListaProveedores
.getSelectedItemAt()).getRfcProveedor());

                    elProducto.setCantidadIngreso(Integer.valueOf(cajaCantidadIngreso.getText()));
                    elProducto.setClasificacion(cajaClasificacion.getText());
                    if(radioIva1.isSelected())
                    {
                        elProducto.setIva(Float.valueOf(limitarDecimales.format(Float.valueOf(cajaCostoPublico.ge
tText()) * 0.16f)));
                    }
                    else if(radioIva2.isSelected())
                    {
                        elProducto.setIva(Float.valueOf(limitarDecimales.format(0f)));
                    }

                    operacionesProducto.insertarProducto(elProducto);

                    fecha.setDate(null);
                    cajaCodigoBarras.setText("");
                    cajaNombreProducto.setText("");
                    cajaCostoCompra.setText("");
                    cajaCostoPublico.setText("");
                    cajaPuntos.setText("");
                    cajaPuntosCompra.setText("");
                    cajaListaProveedores.setSelectedIndex(0);
                    cajaCantidadIngreso.setText("");
                    cajaClasificacion.setText("");
                }
            }
        });

```



```

        radioIval.setSelected(true);
    }
}
});
botonRegistrar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(0,0,0));
    }
});
Border elBordeRegistrar = new LineBorder(Color.black,1);
botonRegistrar.setBorder(elBordeRegistrar);
botonRegistrar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
botonRegistrar.setForeground(new Color(0,0,0));
botonRegistrar.setBackground(new Color(95,255,95));
botonRegistrar.setFont(new Font("Century", Font.PLAIN, 20));
botonRegistrar.setBounds(850, 1200, 208, 48);
panelGeneral.add(botonRegistrar);
}
}
}

```

22.- NuevoProveedor.java

```

package Interfaces;

/** Esta Interfaz permite registrar la información de un Nuevo Proveedor */

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Proveedor;
import hibernate.operaciones.OpProveedor;

public class NuevoProveedor
{
    /******* CONTENEDOR JFrame BASE Y PANEL GENERAL *****/

    private JFrame frmNuevoProveedor;
    private JPanel panelGeneral;

    /******* ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaTelefono;
    private JTextField cajaCorreo;
    private JTextField cajaContacto;
    private JTextField cajaNombreProveedor;
    private JTextField cajaRFC;
    private JTextPane panelTextoDireccion;
    private TextPrompt leyenda;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProveedor operacionesProveedor = new OpProveedor();
}

```

```

/***** ELEMENTOS UTILIZADOS EN EL BOTÓN "Registrar" *****/

private Proveedor elProveedor = new Proveedor();
private boolean cajasVerificadas;

/***** MAIN *****/

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                NuevoProveedor window = new NuevoProveedor();
                window.frmNuevoProveedor.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFRAME *****/

public void setVisible(boolean visibilidad)
{
    frmNuevoProveedor.setVisible(visibilidad);
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if (fecha.getDate() == null || cajaRFC.getText().isEmpty() ||
        cajaNombreProveedor.getText().isEmpty() ||
        panelTextoDireccion.getText().isEmpty() || cajaCorreo.getText().isEmpty() ||
        cajaTelefono.getText().isEmpty() ||
        cajaContacto.getText().isEmpty())
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** INICIALIZACIÓN GENERAL DEL JFRAME *****/

public NuevoProveedor()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA PARA NUEVO PROVEEDOR *****/

private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmNuevoProveedor = new JFrame();

```

```

frmNuevoProveedor.getContentPane().setBackground(new Color(255, 255, 255));

frmNuevoProveedor.setIconImage(Toolkit.getDefaultToolkit().getImage(NuevoProveedor.class.
getResource("/Imagenes/Logotipo.png")));
frmNuevoProveedor.setTitle("Nuevo Proveedor");
frmNuevoProveedor.setSize(Toolkit.getDefaultToolkit().getScreenSize().width,
1000);
frmNuevoProveedor.setExtendedState(JFrame.MAXIMIZED_BOTH);
frmNuevoProveedor.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frmNuevoProveedor.getContentPane().setLayout(null);

JScrollPane scrollGeneral = new JScrollPane();
scrollGeneral.setBackground(new Color(175, 238, 238));
scrollGeneral.setBounds(0, 0, 1365, 715);
scrollGeneral.setHorizontalScrollBar(null);
frmNuevoProveedor.getContentPane().add(scrollGeneral);

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new Dimension(1366, 1150));
panelGeneral.setBackground(new Color(255, 255, 255));
scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmNuevoProveedor.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)

```

```

        {
            MenuAlmacen almacen = new MenuAlmacen();

            almacen.setVisible(true);
            frmNuevoProveedor.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaAlmacen.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaAlmacen.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaAlmacen.setBounds(233, 32, 85, 41);
    panelGeneral.add(etiquetaAlmacen);

    JLabel etiquetaDiagonal2 = new JLabel("/");
    etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal2.setForeground(Color.BLUE);
    etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal2.setBounds(214, 32, 22, 39);
    panelGeneral.add(etiquetaDiagonal2);

    JLabel etiquetaProveedores = new JLabel("Proveedores");
    etiquetaProveedores.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuProveedores proveedores = new MenuProveedores();

            proveedores.setVisible(true);
            frmNuevoProveedor.setVisible(false);
        }
        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaProveedores.setForeground(new Color(232, 19, 245));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaProveedores.setForeground(new Color(0, 0, 255));
        }
    });
    etiquetaProveedores.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaProveedores.setForeground(Color.BLUE);
    etiquetaProveedores.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaProveedores.setBounds(326, 31, 121, 41);
    panelGeneral.add(etiquetaProveedores);

    JLabel etiquetaDiagonal3 = new JLabel("/");
    etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaDiagonal3.setForeground(Color.BLUE);
    etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
    etiquetaDiagonal3.setBounds(438, 31, 22, 39);
    panelGeneral.add(etiquetaDiagonal3);

    JLabel etiquetaNuevoProveedor = new JLabel("Nuevo");
    etiquetaNuevoProveedor.setHorizontalAlignment(SwingConstants.CENTER);

```

```

etiquetaNuevoProveedor.setForeground(Color.BLUE);
etiquetaNuevoProveedor.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaNuevoProveedor.setBounds(400, 32, 170, 41);
panelGeneral.add(etiquetaNuevoProveedor);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** Formato del Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
Border elBordeFormulario = new LineBorder(Color.black,1);
panelFormulario.setBorder(elBordeFormulario);
panelFormulario.setBounds(125, 250, 1088, 700);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 500));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
panelGeneral.add(panelFormulario);

/***** Funcionalidad y Elementos del Formulario *****/

JLabel textoIndicacion = new JLabel("Capture la informaci\u00F3n que se solicita
a continuaci\u00F3n:");
textoIndicacion.setFont(new Font("Book Antiqua", Font.PLAIN, 25));
textoIndicacion.setHorizontalAlignment(SwingConstants.CENTER);
textoIndicacion.setBounds(100, 200, 595, 32);
panelGeneral.add(textoIndicacion);

JLabel etiquetaRFC = new JLabel("R.F.C.");
etiquetaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaRFC.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaRFC.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaRFC);

cajaRFC = new JTextField();
cajaRFC.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar RFC",cajaRFC);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaRFC.setBounds(536, 45, 169, 36);
cajaRFC.setColumns(10);
cajaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaRFC);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 130, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));

```

```

fecha.setBounds(536, 130, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaNombreProveedor = new JLabel("Nombre del Proveedor:");
etiquetaNombreProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProveedor.setBounds(272, 218, 257, 30);
panelFormulario.add(etiquetaNombreProveedor);

cajaNombreProveedor = new JTextField();
cajaNombreProveedor.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Nombre", cajaNombreProveedor);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNombreProveedor.setBounds(535, 215, 250, 36);
cajaNombreProveedor.setColumns(10);
cajaNombreProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaNombreProveedor);

JLabel etiquetaDireccin = new JLabel("Direcci\u00F3n:");
etiquetaDireccin.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaDireccin.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaDireccin.setBounds(421, 330, 109, 30);
panelFormulario.add(etiquetaDireccin);

panelTextoDireccion = new JTextPane();
leyenda = new TextPrompt("Ingresar Dirección", panelTextoDireccion);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
panelTextoDireccion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelTextoDireccion.setBounds(536, 300, 229, 88);
panelFormulario.add(panelTextoDireccion);

JLabel etiquetaTelefono = new JLabel("Tel\u00E9fono:");
etiquetaTelefono.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTelefono.setBounds(421, 430, 109, 30);
panelFormulario.add(etiquetaTelefono);

cajaTelefono = new JTextField();
leyenda = new TextPrompt("(55) 55-55-55-55", cajaTelefono);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaTelefono.setHorizontalAlignment(SwingConstants.CENTER);
cajaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaTelefono.setColumns(10);
cajaTelefono.setBounds(536, 430, 155, 36);
panelFormulario.add(cajaTelefono);

JLabel etiquetaCorreo = new JLabel("Correo:");
etiquetaCorreo.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCorreo.setBounds(445, 515, 85, 30);
panelFormulario.add(etiquetaCorreo);

cajaCorreo = new JTextField();
leyenda = new TextPrompt("correo@correo.com", cajaCorreo);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCorreo.setHorizontalAlignment(SwingConstants.CENTER);
cajaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCorreo.setColumns(10);
cajaCorreo.setBounds(536, 515, 273, 36);
panelFormulario.add(cajaCorreo);

```

```

JLabel etiquetaContacto = new JLabel("Contacto:");
etiquetaContacto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaContacto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaContacto.setBounds(416, 600, 119, 30);
panelFormulario.add(etiquetaContacto);

cajaContacto = new JTextField();
leyenda = new TextPrompt("Ingresar Nombre",cajaContacto);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaContacto.setHorizontalAlignment(SwingConstants.CENTER);
cajaContacto.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaContacto.setColumns(10);
cajaContacto.setBounds(537, 600, 272, 36);
panelFormulario.add(cajaContacto);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuProveedores menuProveedores = new MenuProveedores();

        menuProveedores.setVisible(true);
        frmNuevoProveedor.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0,0,0));
    }
});
Border elBordeTerminar = new LineBorder(Color.black,1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(new Color(0,0,0));
botonTerminar.setBackground(new Color(234,34,34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(300, 1000, 197, 50);
panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad del Botón "Registrar" *****/

JButton botonRegistrar = new JButton(" Registrar");
botonRegistrar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        cajasVerificadas = camposLlenos();

        if(cajasVerificadas == false)
        {
            JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");

```

```

    }
    else
    {
        java.util.Date laFecha = fecha.getDate();
        SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

        elProveedor.setRfcProveedor(cajaRFC.getText());

elProveedor.setFechaRegistro(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))));
elProveedor.setNombreProveedor(cajaNombreProveedor.getText());
elProveedor.setDireccion(panelTextoDireccion.getText());
elProveedor.setTelefono(cajaTelefono.getText());
elProveedor.setCorreo(cajaCorreo.getText());
elProveedor.setNombreContacto(cajaContacto.getText());

        operacionesProveedor.insertarProveedor(elProveedor);

        fecha.setDate(null);
        cajaRFC.setText("");
        cajaNombreProveedor.setText("");
        panelTextoDireccion.setText("");
        cajaTelefono.setText("");
        cajaCorreo.setText("");
        cajaContacto.setText("");
    }
}
});
botonRegistrar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(0,0,0));
    }
});
Border elBordeRegistrar = new LineBorder(Color.black,1);
botonRegistrar.setBorder(elBordeRegistrar);
botonRegistrar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
botonRegistrar.setForeground(new Color(0,0,0));
botonRegistrar.setBackground(new Color(95,255,95));
botonRegistrar.setFont(new Font("Century", Font.PLAIN, 20));
botonRegistrar.setBounds(850, 1000, 208, 50);
panelGeneral.add(botonRegistrar);
}
}
}

```

23.- RegistrarCliente.java

```

package Interfaces;

/** Esta Interfaz permite registrar la información de un Nuevo Cliente */

import java.awt.Color;
import java.awt.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import javax.swing.*;
import javax.swing.border.*;
import com.toedter.calendar.JDateChooser;

```



```

import hibernate.beans.Cliente;
import hibernate.operaciones.*;

public class RegistrarCliente
{
/***** CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmRegistrarCliente;
    private JPanel panelGeneral;

/***** ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaRFC;
    private JTextField cajaNombreCliente;
    private JTextField cajaTelefono;
    private JTextField cajaCorreo;
    private JTextPane panelTextoDireccion;
    private TextPrompt leyenda;

/***** ELEMENTOS UTILIZADOS EN EL BOTÓN "Registrar" *****/

    private OpCliente operacionesCliente = new OpCliente();
    private Cliente elCliente = new Cliente();
    private boolean cajasVerificadas;

/***** MAIN *****/

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    RegistrarCliente window = new RegistrarCliente();
                    window.frmRegistrarCliente.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

/***** INICIALIZACIÓN GENERAL DEL JFRAME *****/

    public RegistrarCliente()
    {
        initialize();
    }

/***** VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmRegistrarCliente.setVisible(visibilidad);
    }

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

    public boolean camposLlenos()
    {
        boolean verificacion;
        if(fecha.getDate() == null || cajaRFC.getText().isEmpty() ||
cajaNombreCliente.getText().isEmpty() ||
        panelTextoDireccion.getText().isEmpty() || cajaCorreo.getText().isEmpty() ||
cajaTelefono.getText().isEmpty())
        {
            verificacion = false;
        }
        else

```

```

    {
        verificacion = true;
    }
    return verificacion;
}

/***** DISEÑO DE LA PÁGINA PARA NUEVO PROVEEDOR *****/

private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmRegistrarCliente = new JFrame();
    frmRegistrarCliente.setTitle("Registrar Cliente");
    frmRegistrarCliente.getContentPane().setBackground(new Color(255, 255, 255));

frmRegistrarCliente.setIconImage(Toolkit.getDefaultToolkit().getImage(NuevoProveedor.class.getResource("/Imágenes/Logotipo.png")));
    frmRegistrarCliente.setSize(Toolkit.getDefaultToolkit().getScreenSize().width,
1000);
    frmRegistrarCliente.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmRegistrarCliente.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmRegistrarCliente.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(175, 238, 238));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmRegistrarCliente.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new Dimension(1366, 1050));
    panelGeneral.setBackground(new Color(255, 255, 255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("Página Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmRegistrarCliente.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    panelGeneral.setLayout(null);
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    panelGeneral.add(etiquetaPaginaPrincipal);
}

```

```

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Clientes");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuClientes clientes = new MenuClientes();

        clientes.setVisible(true);
        frmRegistrarCliente.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProveedores = new JLabel("Registrar");
etiquetaProveedores.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProveedores.setForeground(Color.BLUE);
etiquetaProveedores.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProveedores.setBounds(326, 31, 121, 41);
panelGeneral.add(etiquetaProveedores);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** Formato del Panel del Formulario *****/

```

```

JPanel panelFormulario = new JPanel();
Border elBordeFormulario = new LineBorder(Color.black,1);
panelFormulario.setBorder(elBordeFormulario);
panelFormulario.setBounds(125, 250, 1088, 600);
panelFormulario.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 500));
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
panelGeneral.add(panelFormulario);

/***** Funcionalidad y Elementos del Formulario *****/

JLabel textoIndicacion = new JLabel("Capture la informaci\u00F3n que se solicita
a continuaci\u00F3n:");
textoIndicacion.setFont(new Font("Book Antiqua", Font.PLAIN, 25));
textoIndicacion.setHorizontalAlignment(SwingConstants.CENTER);
textoIndicacion.setBounds(100, 200, 595, 32);
panelGeneral.add(textoIndicacion);

JLabel etiquetaRFC = new JLabel("R.F.C.");
etiquetaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaRFC.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaRFC.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaRFC);

cajaRFC = new JTextField();
cajaRFC.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar RFC",cajaRFC);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaRFC.setBounds(536, 45, 169, 36);
cajaRFC.setColumns(10);
cajaRFC.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaRFC);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 130, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 130, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaNombreProveedor = new JLabel("Nombre del Cliente:");
etiquetaNombreProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProveedor.setBounds(272, 218, 257, 30);
panelFormulario.add(etiquetaNombreProveedor);

cajaNombreCliente = new JTextField();
cajaNombreCliente.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Nombre",cajaNombreCliente);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNombreCliente.setBounds(535, 215, 230, 36);
cajaNombreCliente.setColumns(10);
cajaNombreCliente.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaNombreCliente);

JLabel etiquetaDireccin = new JLabel("Direcci\u00F3n:");
etiquetaDireccin.setHorizontalAlignment(SwingConstants.RIGHT);

```

```

etiquetaDireccin.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaDireccin.setBounds(421, 330, 109, 30);
panelFormulario.add(etiquetaDireccin);

panelTextoDireccion = new JTextPane();
leyenda = new TextPrompt("Ingresar Dirección",panelTextoDireccion);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
panelTextoDireccion.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelTextoDireccion.setBounds(536, 300, 229, 88);
panelFormulario.add(panelTextoDireccion);

JLabel etiquetaTelefono = new JLabel("Tel\u00E9fono:");
etiquetaTelefono.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTelefono.setBounds(421, 430, 109, 30);
panelFormulario.add(etiquetaTelefono);

cajaTelefono = new JTextField();
leyenda = new TextPrompt("(55) 55-55-55-55",cajaTelefono);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaTelefono.setHorizontalAlignment(SwingConstants.CENTER);
cajaTelefono.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaTelefono.setColumns(10);
cajaTelefono.setBounds(536, 430, 155, 36);
panelFormulario.add(cajaTelefono);

JLabel etiquetaCorreo = new JLabel("Correo:");
etiquetaCorreo.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCorreo.setBounds(445, 515, 85, 30);
panelFormulario.add(etiquetaCorreo);

cajaCorreo = new JTextField();
leyenda = new TextPrompt("correo@correo.com",cajaCorreo);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCorreo.setHorizontalAlignment(SwingConstants.CENTER);
cajaCorreo.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCorreo.setColumns(10);
cajaCorreo.setBounds(536, 515, 273, 36);
panelFormulario.add(cajaCorreo);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonSalir = new JButton(" Salir");
botonSalir.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuClientes menuClientes = new MenuClientes();

        menuClientes.setVisible(true);
        frmRegistrarCliente.setVisible(false);
    }
});
botonSalir.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonSalir.setForeground(new Color(12,2,235));
    }
    @Override

```

```

        public void mouseExited(MouseEvent e)
        {
            botonSalir.setForeground(new Color(0,0,0));
        }
    });
    Border elBordeTerminar = new LineBorder(Color.black,1);
    botonSalir.setBorder(elBordeTerminar);
    botonSalir.setHorizontalAlignment(SwingConstants.CENTER);
    botonSalir.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
    botonSalir.setForeground(new Color(0,0,0));
    botonSalir.setBackground(new Color(234,34,34));
    botonSalir.setFont(new Font("Century", Font.PLAIN, 20));
    botonSalir.setBounds(300, 900, 197, 50);
    panelGeneral.add(botonSalir);

/***** Formato y Funcionalidad del Botón "Registrar" *****/

    JButton botonRegistrar = new JButton(" Registrar");
    botonRegistrar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            cajasVerificadas = camposLlenos();

            if(cajasVerificadas == false)
            {
                JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
            }
            else
            {
                java.util.Date laFecha = fecha.getDate();
                SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

                elCliente.setRfcCliente(cajaRFC.getText());

elCliente.setFechaRegistro(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha))));
                elCliente.setNombreCliente(cajaNombreCliente.getText());
                elCliente.setDireccion(panelTextoDireccion.getText());
                elCliente.setTelefono(cajaTelefono.getText());
                elCliente.setCorreo(cajaCorreo.getText());

                operacionesCliente.insertarCliente(elCliente);

                fecha.setDate(null);
                cajaRFC.setText("");
                cajaNombreCliente.setText("");
                panelTextoDireccion.setText("");
                cajaTelefono.setText("");
                cajaCorreo.setText("");
            }
        }
    });
    botonRegistrar.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonRegistrar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonRegistrar.setForeground(new Color(0,0,0));
        }
    });

```

```

        Border elBordeRegistrar = new LineBorder(Color.black,1);
        botonRegistrar.setBorder(elBordeRegistrar);
        botonRegistrar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
        botonRegistrar.setForeground(new Color(0,0,0));
        botonRegistrar.setBackground(new Color(95,255,95));
        botonRegistrar.setFont(new Font("Century", Font.PLAIN, 20));
        botonRegistrar.setBounds(850, 900, 208, 50);
        panelGeneral.add(botonRegistrar);
    }
}

```

24.- RegistrarDevolución.java

```

package Interfaces;

/** Esta Interfaz permite registrar la información de una Nueva Devolución */

import java.awt.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import javax.swing.*;
import javax.swing.border.*;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Devolucion;
import hibernate.beans.Producto;
import hibernate.beans.Pedido;
import hibernate.operaciones.*;

public class RegistrarDevolucion
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmRegistrarDevolucin;
    private JPanel panelGeneral;

    /*******ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private JDateChooser fecha;
    private JTextField cajaCantidadProductos,cajaNumeroPedido;
    private TextPrompt leyenda;
    private JLabel etiquetaVerProveedor = new JLabel();
    private JLabel etiquetaVerNombreProducto = new JLabel();
    private JLabel etiquetaVerCostoUnidad = new JLabel();
    private JLabel etiquetaVerCostoDevolucion = new JLabel();
    private JLabel etiquetaVerCodigoBarras = new JLabel();
    @SuppressWarnings("rawtypes")
    private JComboBox cajaTipoProductos,cajaStatusDevolucion;
    private JTextPane panelTextoMotivo;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProducto operacionesProducto = new OpProducto();
    private OpPedido operacionesPedido = new OpPedido();
    private OpDevolucion operacionesDevolucion = new OpDevolucion();
    private Producto elProducto = new Producto();

    /******* ELEMENTOS UTILIZADOS EN EL BOTÓN "Registrar" *****/

    private Devolucion laDevolucion = new Devolucion();
    boolean cajasVerificadas;

    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {

```

```

        public void run()
        {
            try
            {
                RegistrarDevolucion window = new RegistrarDevolucion();
                window.frmRegistrarDevolucin.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFRAME *****/

public void setVisible(boolean visibilidad)
{
    frmRegistrarDevolucin.setVisible(visibilidad);
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if (fecha.getDate() == null || cajaNumeroPedido.getText().isEmpty() ||
    etiquetaVerCodigoBarras.getText().isEmpty() ||
    etiquetaVerNombreProducto.getText().isEmpty() ||
    etiquetaVerProveedor.getText().isEmpty() || cajaCantidadProductos.getText().isEmpty() ||
    etiquetaVerCostoUnidad.getText().isEmpty() ||
    etiquetaVerCostoDevolucion.getText().isEmpty() || panelTextoMotivo.getText().isEmpty() ||
    cajaTipoProductos.getSelectedIndex() == 0 ||
    cajaStatusDevolucion.getSelectedIndex() == 0)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

/***** INICIALIZACIÓN GENERAL DEL JFRAME *****/

public RegistrarDevolucion()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA PARA NUEVO PRODUCTO *****/

@SuppressWarnings({ "rawtypes", "unchecked" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmRegistrarDevolucin = new JFrame();
    frmRegistrarDevolucin.setTitle("Registrar Devoluci\u00F3n");
    frmRegistrarDevolucin.getContentPane().setBackground(new Color(255,255,255));

    frmRegistrarDevolucin.setIconImage(Toolkit.getDefaultToolkit().getImage(RegistrarDevoluci
on.class.getResource("/Imagenes/Logotipo.png")));
    frmRegistrarDevolucin.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmRegistrarDevolucin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmRegistrarDevolucin.getContentPane().setLayout(null);
}
}

```



```

JScrollPane scrollGeneral = new JScrollPane();
scrollGeneral.setBackground(new Color(255,255,255));
scrollGeneral.setBounds(0, 0, 1365, 715);
scrollGeneral.setHorizontalScrollBar(null);
frmRegistrarDevolucin.getContentPane().add(scrollGeneral);

panelGeneral = new JPanel();
panelGeneral.setAlignmentX(0.0f);
panelGeneral.setAlignmentY(0.0f);
panelGeneral.setPreferredSize(new Dimension(1366, 1400));
panelGeneral.setBackground(new Color(255, 255, 255));
scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPrincipal menuPrincipal = new MenuPrincipal();

        menuPrincipal.setVisible(true);
        frmRegistrarDevolucin.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    }
});
panelGeneral.setLayout(null);
etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmRegistrarDevolucin.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {

```

```

        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaDevoluciones = new JLabel("Devoluciones");
etiquetaDevoluciones.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuDevoluciones devoluciones = new MenuDevoluciones();

        devoluciones.setVisible(true);
        frmRegistrarDevolucion.setVisible(false);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(232, 19, 245));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(0, 0, 255));
    }
});
etiquetaDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDevoluciones.setForeground(Color.BLUE);
etiquetaDevoluciones.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaDevoluciones.setBounds(332, 31, 121, 41);
panelGeneral.add(etiquetaDevoluciones);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(450, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaRegistrarDevolucion = new JLabel("Registrar");
etiquetaRegistrarDevolucion.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaRegistrarDevolucion.setForeground(Color.BLUE);
etiquetaRegistrarDevolucion.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaRegistrarDevolucion.setBounds(430, 32, 170, 41);
panelGeneral.add(etiquetaRegistrarDevolucion);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));

```

```

etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** Formato del Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
Border elBordeFormulario = new LineBorder(Color.black,1);
panelFormulario.setBorder(elBordeFormulario);
panelFormulario.setBounds(125, 250, 1088, 1000);
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
panelGeneral.add(panelFormulario);

/***** Funcionalidad y Elementos del Formulario *****/

JLabel textoIndicacion = new JLabel("Capture la informaci\u00F3n que se solicita
a continuaci\u00F3n:");
textoIndicacion.setFont(new Font("Book Antiqua", Font.PLAIN, 25));
textoIndicacion.setHorizontalAlignment(SwingConstants.CENTER);
textoIndicacion.setBounds(100, 200, 595, 32);
panelGeneral.add(textoIndicacion);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 45, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaNumeroPedido = new JLabel("Pedido:");
etiquetaNumeroPedido.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNumeroPedido.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNumeroPedido.setBounds(342, 120, 188, 30);
panelFormulario.add(etiquetaNumeroPedido);

cajaNumeroPedido = new JTextField();
cajaNumeroPedido.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Número de Pedido",cajaNumeroPedido);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaNumeroPedido.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            Pedido elPedido =
operacionesPedido.buscarPedidoPorNumero(Integer.valueOf(cajaNumeroPedido.getText()));

            if(cajaNumeroPedido.getText().equals(""))
            {

```

```

        JOptionPane.showMessageDialog(null, "Ingrese un Número de
Pedido.");
    }
    else if(elPedido == null)
    {
        JOptionPane.showMessageDialog(null, "Este Pedido no existe en el
Sistema.");
    }
    else
    {

etiquetaVerCodigoBarras.setText (elPedido.getProductoCodigoBarras ());

etiquetaVerNombreProducto.setText (operacionesProducto.buscarProductoPorID (elPedido.getPro
ductoCodigoBarras ()).getNombreProducto ());

etiquetaVerProveedor.setText (operacionesProducto.buscarProductoPorID (elPedido.getProducto
CodigoBarras ()).getProveedor ());

etiquetaVerCostoUnidad.setText (String.valueOf (operacionesProducto.buscarProductoPorID (elP
edido.getProductoCodigoBarras ()).getCostoCompra ()));
    }
}

public void keyTyped(KeyEvent e)
{
    char caracter = e.getKeyChar ();

    if((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
    {
        e.consume ();
    }
}
});
cajaNumeroPedido.setBounds (536, 120, 124, 36);
cajaNumeroPedido.setColumns (10);
cajaNumeroPedido.setFont (new Font ("Cambria Math", Font.PLAIN, 16));
panelFormulario.add (cajaNumeroPedido);

JLabel etiquetaCodigoBarras = new JLabel ("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setHorizontalAlignment (SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont (new Font ("Cambria Math", Font.PLAIN, 25));
etiquetaCodigoBarras.setBounds (342, 205, 188, 30);
panelFormulario.add (etiquetaCodigoBarras);

etiquetaVerCodigoBarras.setFont (new Font ("Cambria Math", Font.PLAIN, 18));
etiquetaVerCodigoBarras.setBounds (536, 205, 250, 36);
panelFormulario.add (etiquetaVerCodigoBarras);

JLabel etiquetaNombreProducto = new JLabel ("Producto:");
etiquetaNombreProducto.setHorizontalAlignment (SwingConstants.RIGHT);
etiquetaNombreProducto.setFont (new Font ("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProducto.setBounds (296, 280, 236, 30);
panelFormulario.add (etiquetaNombreProducto);

etiquetaVerNombreProducto.setFont (new Font ("Cambria Math", Font.PLAIN, 18));
etiquetaVerNombreProducto.setBounds (536, 280, 250, 36);
panelFormulario.add (etiquetaVerNombreProducto);

JLabel etiquetaProveedor = new JLabel ("Proveedor:");
etiquetaProveedor.setHorizontalAlignment (SwingConstants.RIGHT);
etiquetaProveedor.setFont (new Font ("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds (399, 350, 131, 30);
panelFormulario.add (etiquetaProveedor);

etiquetaVerProveedor.setFont (new Font ("Cambria Math", Font.PLAIN, 18));
etiquetaVerProveedor.setBounds (540, 350, 320, 36);

```

```

panelFormulario.add(etiquetaVerProveedor);

JLabel etiquetaCantidadProductos = new JLabel("Cantidad de Productos:");
etiquetaCantidadProductos.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCantidadProductos.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCantidadProductos.setBounds(250, 435, 280, 36);
panelFormulario.add(etiquetaCantidadProductos);

cajaCantidadProductos = new JTextField();
cajaCantidadProductos.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Cantidad", cajaCantidadProductos);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCantidadProductos.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            elProducto =
operacionesProducto.buscarProductoPorID(etiquetaVerCodigoBarras.getText());

            if(cajaCantidadProductos.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null, "Ingrese una Cantidad.");
            }
            else
            {
                etiquetaVerCostoDevolucion.setText(String.valueOf(Float.parseFloat(cajaCantidadProductos.
getText()) * elProducto.getCostoCompra()));
            }
        }
    }

    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCantidadProductos.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCantidadProductos.setColumns(10);
cajaCantidadProductos.setBounds(536, 435, 69, 36);
panelFormulario.add(cajaCantidadProductos);

JLabel etiquetaCostoUnidad = new JLabel("Costo por Unidad:");
etiquetaCostoUnidad.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoUnidad.setBounds(300, 520, 230, 30);
panelFormulario.add(etiquetaCostoUnidad);

etiquetaVerCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCostoUnidad.setBounds(540, 520, 95, 36);
panelFormulario.add(etiquetaVerCostoUnidad);

JLabel etiquetaCostoDevolucion = new JLabel("Costo de la Devolución:");
etiquetaCostoDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoDevolucion.setBounds(250, 625, 280, 30);
panelFormulario.add(etiquetaCostoDevolucion);

```

```

etiquetaVerCostoDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCostoDevolucion.setBounds(540, 625, 95, 36);
panelFormulario.add(etiquetaVerCostoDevolucion);

JLabel etiquetaMotivo = new JLabel("Motivo:");
etiquetaMotivo.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaMotivo.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaMotivo.setBounds(421, 720, 109, 30);
panelFormulario.add(etiquetaMotivo);

panelTextoMotivo = new JTextPane();
leyenda = new TextPrompt("Motivo de la Devolución",panelTextoMotivo);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
panelTextoMotivo.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelTextoMotivo.setBounds(536, 694, 229, 88);
panelFormulario.add(panelTextoMotivo);

JLabel etiquetaTipoProducto = new JLabel("Clasificación:");
etiquetaTipoProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTipoProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTipoProducto.setBounds(280, 820, 250, 30);
panelFormulario.add(etiquetaTipoProducto);

cajaTipoProductos = new JComboBox();
cajaTipoProductos.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaTipoProductos.setModel(new DefaultComboBoxModel(new String[] {}));
cajaTipoProductos.addItem("Seleccione una Opción");
cajaTipoProductos.addItem("Abarrotes");
cajaTipoProductos.addItem("Bebidas");
cajaTipoProductos.addItem("Cuidado e Higiene Personal");
cajaTipoProductos.addItem("Dulceria");
cajaTipoProductos.addItem("Lacteos");
cajaTipoProductos.setBounds(536, 820, 270, 36);
panelFormulario.add(cajaTipoProductos);

JLabel etiquetaStatusDevolucion = new JLabel("Estatus:");
etiquetaStatusDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaStatusDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaStatusDevolucion.setBounds(440, 900, 90, 30);
panelFormulario.add(etiquetaStatusDevolucion);

cajaStatusDevolucion = new JComboBox();
cajaStatusDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaStatusDevolucion.setModel(new DefaultComboBoxModel(new String[] {}));
cajaStatusDevolucion.addItem("Seleccione una Opción");
cajaStatusDevolucion.addItem("Cancelado");
cajaStatusDevolucion.addItem("Devuelto");
cajaStatusDevolucion.addItem("Pendiente");
cajaStatusDevolucion.addItem("Rechazado");
cajaStatusDevolucion.setBounds(536, 900, 200, 36);
panelFormulario.add(cajaStatusDevolucion);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuDevoluciones menuDevoluciones = new MenuDevoluciones();

        menuDevoluciones.setVisible(true);
        frmRegistrarDevolucin.setVisible(false);
    }
});

```

```

        botonTerminar.addMouseListener(new MouseAdapter ()
        {
            @Override
            public void mouseEntered(MouseEvent e)
            {
                botonTerminar.setForeground(new Color(12,2,235));
            }
            @Override
            public void mouseExited(MouseEvent e)
            {
                botonTerminar.setForeground(new Color(0,0,0));
            }
        });
        Border elBordeTerminar = new LineBorder(Color.black,1);
        botonTerminar.setBorder(elBordeTerminar);
        botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
        botonTerminar.setIcon(new
        ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
        botonTerminar.setForeground(new Color(0,0,0));
        botonTerminar.setBackground(new Color(234,34,34));
        botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
        botonTerminar.setBounds(300, 1330, 197, 50);
        panelGeneral.add(botonTerminar);

        /***** Formato y Funcionalidad del Botón "Registrar" *****/

        JButton botonRegistrar = new JButton(" Registrar");
        botonRegistrar.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                cajasVerificadas = camposLlenos();

                if(cajasVerificadas == false)
                {
                    JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
                }
                else
                {
                    java.util.Date laFecha = fecha.getDate();
                    SimpleDateFormat cambiarFecha = new
SimpleDateFormat(fecha.getDateFormatString());

                    laDevolucion.setFechaDevolucion(Date.valueOf(String.valueOf(cambiarFecha.format(laFecha)
)));

                    laDevolucion.setClasificacion((String)cajaTipoProductos.getSelectedItem());
                    laDevolucion.setMotivo(panelTextoMotivo.getText());

                    laDevolucion.setCantidadProductos(Integer.parseInt(cajaCantidadProductos.getText()));

                    laDevolucion.setCostoDevolucion(Float.parseFloat(etiquetaVerCostoDevolucion.getText()));

                    laDevolucion.setStatusDevolucion((String)cajaStatusDevolucion.getSelectedItem());

                    laDevolucion.setPedidoNumeroPedido(Integer.parseInt(cajaNumeroPedido.getText()));

                    if(((String)cajaStatusDevolucion.getSelectedItem()).compareTo("Devuelto") == 0)
                    {

                        operacionesProducto.actualizarCantidadProductosAlmacenDevolucion(operacionesPedido.buscar
PedidoPorNumero(
                            Integer.parseInt(cajaNumeroPedido.getText()).getProductoCodigoBarras(),Integer.valueOf(c
ajaCantidadProductos.getText()));
                    }
                }
            }
        });

```

```

    }

    operacionesDevolucion.registrarDevolucion(laDevolucion);

    fecha.setDate(null);
    cajaNumeroPedido.setText("");
    etiquetaVerCodigoBarras.setText("");
    etiquetaVerNombreProducto.setText("");
    etiquetaVerProveedor.setText("");
    cajaCantidadProductos.setText("");
    etiquetaVerCostoUnidad.setText("");
    etiquetaVerCostoDevolucion.setText("");
    panelTextoMotivo.setText("");
    cajaTipoProductos.setSelectedIndex(0);
    cajaStatusDevolucion.setSelectedIndex(0);
}
}
});
botonRegistrar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonRegistrar.setForeground(new Color(0,0,0));
    }
});
Border elBordeRegistrar = new LineBorder(Color.black,1);
botonRegistrar.setBorder(elBordeRegistrar);
botonRegistrar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imágenes/Palomita.png")));
botonRegistrar.setForeground(new Color(0,0,0));
botonRegistrar.setBackground(new Color(95,255,95));
botonRegistrar.setFont(new Font("Century", Font.PLAIN, 20));
botonRegistrar.setBounds(850, 1330, 208, 50);
panelGeneral.add(botonRegistrar);
}
}
}

```

25.- RegistrarPedido.java

```

package Interfaces;

/** Esta Interfaz permite registrar la información de un Nuevo Pedido */

import java.awt.*;
import java.awt.event.*;
import java.sql.Date;
import java.text.SimpleDateFormat;
import javax.swing.*;
import javax.swing.border.*;
import com.toedter.calendar.JDateChooser;
import hibernate.beans.Pedido;
import hibernate.beans.Producto;
import hibernate.operaciones.OpPedido;
import hibernate.operaciones.OpProducto;

public class RegistrarPedido
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmRegistrarPedido;
    private JPanel panelGeneral;
}

```



```

/***** ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

private JDateChooser fecha;
private JTextField cajaCodigoBarras, cajaCantidadSolicitar;
private TextPrompt leyenda;
private JLabel etiquetaVerNombreProducto = new JLabel();
private JLabel etiquetaVerRfcProveedor = new JLabel();
private JLabel etiquetaVerCostoUnidad = new JLabel();
private JLabel etiquetaVerTotalLiquidar = new JLabel();
@SuppressWarnings("rawtypes")
private JComboBox cajaStatusPedido;

/***** OPERACIONES PARA FUNCIONALIDAD *****/

private OpProducto operacionesProducto = new OpProducto();
private OpPedido operacionesPedido = new OpPedido();
private Producto elProducto = new Producto();
private Pedido elPedido = new Pedido();
private boolean cajasVerificadas;

public static void main(String[] args)
{
    EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            try
            {
                RegistrarPedido window = new RegistrarPedido();
                window.frmRegistrarPedido.setVisible(true);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    });
}

/***** VISIBILIDAD DEL JFrame *****/

public void setVisible(boolean visibilidad)
{
    frmRegistrarPedido.setVisible(visibilidad);
}

/***** VALIDACIÓN DE CAMPOS VACÍOS *****/

public boolean camposLlenos()
{
    boolean verificacion;
    if(fecha.getDate() == null || cajaCodigoBarras.getText().isEmpty() ||
cajaCantidadSolicitar.getText().isEmpty() ||
    etiquetaVerNombreProducto.getText().isEmpty() ||
etiquetaVerRfcProveedor.getText().isEmpty() || etiquetaVerCostoUnidad.getText().isEmpty()
||
    etiquetaVerTotalLiquidar.getText().isEmpty() ||
cajaStatusPedido.getSelectedIndex() == 0)
    {
        verificacion = false;
    }
    else
    {
        verificacion = true;
    }
    return verificacion;
}

```

```

public RegistrarPedido()
{
    initialize();
}

/***** DISEÑO DE LA PÁGINA "REGISTRAR PEDIDO" *****/

@SuppressWarnings({ "rawtypes", "unchecked" })
private void initialize()
{
/***** Formato del JFrame, Scroll y Página Principal *****/

    frmRegistrarPedido = new JFrame();
    frmRegistrarPedido.setTitle("Registrar Pedido");
    frmRegistrarPedido.getContentPane().setBackground(new Color(255,255,255));

frmRegistrarPedido.setIconImage(Toolkit.getDefaultToolkit().getImage(RegistrarDevolucion.
class.getResource("/Imagenes/Logotipo.png")));
    frmRegistrarPedido.setExtendedState(JFrame.MAXIMIZED_BOTH);
    frmRegistrarPedido.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmRegistrarPedido.getContentPane().setLayout(null);

    JScrollPane scrollGeneral = new JScrollPane();
    scrollGeneral.setBackground(new Color(255,255,255));
    scrollGeneral.setBounds(0, 0, 1365, 715);
    scrollGeneral.setHorizontalScrollBar(null);
    frmRegistrarPedido.getContentPane().add(scrollGeneral);

    panelGeneral = new JPanel();
    panelGeneral.setAlignmentX(0.0f);
    panelGeneral.setAlignmentY(0.0f);
    panelGeneral.setPreferredSize(new Dimension(1366, 1150));
    panelGeneral.setBackground(new Color(255, 255, 255));
    scrollGeneral.setViewportView(panelGeneral);

/***** Formato y Funcionalidad de las Migajas *****/

    JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E9gina Principal");
    etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent arg0)
        {
            MenuPrincipal menuPrincipal = new MenuPrincipal();

            menuPrincipal.setVisible(true);
            frmRegistrarPedido.setVisible(false);
        }

        @Override
        public void mouseEntered(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
        }

        @Override
        public void mouseExited(MouseEvent e)
        {
            etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
        }
    });
    panelGeneral.setLayout(null);
    etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
    etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
    etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
    etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
    panelGeneral.add(etiquetaPaginaPrincipal);

```

```

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(313, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Almac\u00E9n");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuAlmacen almacen = new MenuAlmacen();

        almacen.setVisible(true);
        frmRegistrarPedido.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(233, 32, 85, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(214, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaDevoluciones = new JLabel("Pedidos");
etiquetaDevoluciones.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuPedidos pedidos = new MenuPedidos();

        pedidos.setVisible(true);
        frmRegistrarPedido.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaDevoluciones.setForeground(new Color(0, 0, 255));
    }
});
etiquetaDevoluciones.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDevoluciones.setForeground(Color.BLUE);
etiquetaDevoluciones.setFont(new Font("Century", Font.ITALIC, 20));

```

```

etiquetaDevoluciones.setBounds(310, 31, 121, 41);
panelGeneral.add(etiquetaDevoluciones);

JLabel etiquetaDiagonal3 = new JLabel("/");
etiquetaDiagonal3.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal3.setForeground(Color.BLUE);
etiquetaDiagonal3.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal3.setBounds(405, 31, 22, 39);
panelGeneral.add(etiquetaDiagonal3);

JLabel etiquetaRegistrarDevolucion = new JLabel("Registrar");
etiquetaRegistrarDevolucion.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaRegistrarDevolucion.setForeground(Color.BLUE);
etiquetaRegistrarDevolucion.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaRegistrarDevolucion.setBounds(385, 32, 170, 41);
panelGeneral.add(etiquetaRegistrarDevolucion);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** Formato del Panel del Formulario *****/

JPanel panelFormulario = new JPanel();
Border elBordeFormulario = new BorderLayout(Color.black,1);
panelFormulario.setBorder(elBordeFormulario);
panelFormulario.setBounds(125, 250, 1088, 700);
panelFormulario.setBackground(new Color(243, 243, 243));
panelFormulario.setLayout(null);
panelGeneral.add(panelFormulario);

/***** Funcionalidad y Elementos del Formulario *****/

JLabel textoIndicacion = new JLabel("Capture la informaci\u00F3n que se solicita
a continuaci\u00F3n:");
textoIndicacion.setFont(new Font("Book Antiqua", Font.PLAIN, 25));
textoIndicacion.setHorizontalAlignment(SwingConstants.CENTER);
textoIndicacion.setBounds(100, 200, 595, 32);
panelGeneral.add(textoIndicacion);

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(461, 45, 69, 30);
panelFormulario.add(etiquetaFecha);

fecha = new JDateChooser();
fecha.setDateFormatString("yyyy-MM-dd");
fecha.setForeground(new Color(0, 0, 0));
fecha.setBounds(536, 45, 119, 27);
panelFormulario.add(fecha);

JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCodigoBarras.setBounds(342, 120, 188, 30);

```

```

panelFormulario.add(etiquetaCodigoBarras);

cajaCodigoBarras = new JTextField();
cajaCodigoBarras.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Ingresar Código",cajaCodigoBarras);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCodigoBarras.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            elProducto =
operacionesProducto.buscarProductoPorID(cajaCodigoBarras.getText());

            if(cajaCodigoBarras.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingrese un Código.");
            }
            else if(elProducto == null)
            {
                JOptionPane.showMessageDialog(null,"Este Código no existe en el
Sistema.");
            }
            else
            {
                etiquetaVerNombreProducto.setText(elProducto.getNombreProducto());
                etiquetaVerRfcProveedor.setText(elProducto.getProveedor());

                etiquetaVerCostoUnidad.setText(String.valueOf(elProducto.getCostoCompra()));
            }
        }
    }

    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCodigoBarras.setBounds(536, 120, 184, 36);
cajaCodigoBarras.setColumns(10);
cajaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelFormulario.add(cajaCodigoBarras);

JLabel etiquetaNombreProducto = new JLabel("Producto:");
etiquetaNombreProducto.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreProducto.setBounds(296, 205, 236, 30);
panelFormulario.add(etiquetaNombreProducto);

etiquetaVerNombreProducto.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerNombreProducto.setBounds(536, 205, 250, 36);
panelFormulario.add(etiquetaVerNombreProducto);

JLabel etiquetaProveedor = new JLabel("Proveedor:");
etiquetaProveedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaProveedor.setBounds(399, 280, 131, 30);

```

```

panelFormulario.add(etiquetaProveedor);

etiquetaVerRfcProveedor.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerRfcProveedor.setBounds(540, 280, 320, 36);
panelFormulario.add(etiquetaVerRfcProveedor);

JLabel etiquetaCantidadSolicitar = new JLabel("Cantidad a Solicitar:");
etiquetaCantidadSolicitar.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCantidadSolicitar.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCantidadSolicitar.setBounds(250, 350, 280, 36);
panelFormulario.add(etiquetaCantidadSolicitar);

cajaCantidadSolicitar = new JTextField();
cajaCantidadSolicitar.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Cantidad",cajaCantidadSolicitar);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaCantidadSolicitar.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyPressed(KeyEvent arg0)
    {
        if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
        {
            if(cajaCantidadSolicitar.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingrese una Cantidad.");
            }
            else
            {
                etiquetaVerTotalLiquidar.setText(String.valueOf(Float.parseFloat(cajaCantidadSolicitar.ge
tText()) * elProducto.getCostoCompra()));
            }
        }
    }

    public void keyTyped(KeyEvent e)
    {
        char caracter = e.getKeyChar();

        if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
        {
            e.consume();
        }
    }
});
cajaCantidadSolicitar.setFont(new Font("Cambria Math", Font.PLAIN, 16));
cajaCantidadSolicitar.setColumns(10);
cajaCantidadSolicitar.setBounds(536, 350, 69, 36);
panelFormulario.add(cajaCantidadSolicitar);

JLabel etiquetaCostoUnidad = new JLabel("Costo por Unidad:");
etiquetaCostoUnidad.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaCostoUnidad.setBounds(300, 435, 230, 30);
panelFormulario.add(etiquetaCostoUnidad);

etiquetaVerCostoUnidad.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerCostoUnidad.setBounds(540, 435, 95, 36);
panelFormulario.add(etiquetaVerCostoUnidad);

JLabel etiquetaTotalLiquidar = new JLabel("Costo Total:");
etiquetaTotalLiquidar.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTotalLiquidar.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaTotalLiquidar.setBounds(325, 520, 200, 30);

```

```

panelFormulario.add(etiquetaTotalLiquidar);

etiquetaVerTotalLiquidar.setFont(new Font("Cambria Math", Font.PLAIN, 18));
etiquetaVerTotalLiquidar.setBounds(540, 520, 95, 36);
panelFormulario.add(etiquetaVerTotalLiquidar);

JLabel etiquetaStatusDevolucion = new JLabel("Estatus:");
etiquetaStatusDevolucion.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaStatusDevolucion.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaStatusDevolucion.setBounds(440, 600, 90, 30);
panelFormulario.add(etiquetaStatusDevolucion);

cajaStatusPedido = new JComboBox();
cajaStatusPedido.setFont(new Font("Cambria Math", Font.PLAIN, 15));
cajaStatusPedido.setModel(new DefaultComboBoxModel(new String[] {}));
cajaStatusPedido.addItem("Seleccione un Opción");
cajaStatusPedido.addItem("Cancelado");
cajaStatusPedido.addItem("Pendiente");
cajaStatusPedido.addItem("Realizado");
cajaStatusPedido.addItem("Rechazado");
cajaStatusPedido.setBounds(536, 600, 200, 36);
panelFormulario.add(cajaStatusPedido);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuPedidos menuPedidos = new MenuPedidos();

        menuPedidos.setVisible(true);
        frmRegistrarPedido.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0,0,0));
    }
});
Border elBordeTerminar = new LineBorder(Color.black,1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(new Color(0,0,0));
botonTerminar.setBackground(new Color(234,34,34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(300, 1020, 197, 50);
panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad del Botón "Registrar" *****/

JButton botonRegistrar = new JButton(" Registrar");
botonRegistrar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        cajasVerificadas = camposLlenos();
    }
});

```

```

        if(cajasVerificadas == false)
        {
            JOptionPane.showMessageDialog(null,"Por Favor, llenar todos los
campos.");
        }
        else
        {
            java.util.Date laFecha = fecha.getDate();
            SimpleDateFormat cambiarFecha = new
SimpleDateFormat (fecha.getDateFormatString());

elPedido.setFechaPedido (Date.valueOf (String.valueOf (cambiarFecha.format (laFecha))));
            elPedido.setProductoCodigoBarras (cajaCodigoBarras.getText ());

elPedido.setCantidadSolicitada (Integer.valueOf (cajaCantidadSolicitar.getText ());

elPedido.setTotalLiquidar (Float.valueOf (etiquetaVerTotalLiquidar.getText ());
            elPedido.setStatusPedido ((String) cajaStatusPedido.getSelectedItem ());

if (((String) cajaStatusPedido.getSelectedItem ()) .compareTo ("Realizado") == 0)
        {

operacionesProducto.actualizarCantidadProductosAlmacenPedido (cajaCodigoBarras.getText (), I
nTEGER.valueOf (cajaCantidadSolicitar.getText ());
        }

operacionesPedido.registrarPedido (elPedido);

        fecha.setDate (null);
        cajaCodigoBarras.setText ("");
        etiquetaVerNombreProducto.setText ("");
        etiquetaVerRfcProveedor.setText ("");
        cajaCantidadSolicitar.setText ("");
        etiquetaVerCostoUnidad.setText ("");
        etiquetaVerTotalLiquidar.setText ("");
        cajaStatusPedido.setSelectedIndex (0);
    }
}
});
botonRegistrar.addMouseListener (new MouseAdapter ()
{
    @Override
    public void mouseEntered (MouseEvent e)
    {
        botonRegistrar.setForeground (new Color (12,2,235));
    }
    @Override
    public void mouseExited (MouseEvent e)
    {
        botonRegistrar.setForeground (new Color (0,0,0));
    }
});
Border elBordeRegistrar = new LineBorder (Color.black,1);
botonRegistrar.setBorder (elBordeRegistrar);
botonRegistrar.setIcon (new
ImageIcon (NuevoProducto.class.getResource ("/Imagenes/Palomita.png")));
botonRegistrar.setForeground (new Color (0,0,0));
botonRegistrar.setBackground (new Color (95,255,95));
botonRegistrar.setFont (new Font ("Century", Font.PLAIN, 20));
botonRegistrar.setBounds (850, 1020, 208, 50);
panelGeneral.add (botonRegistrar);
}
}

```


26.- RegistrarVentas.java

```
package Interfaces;

/** Esta Interfaz permite registrar las Ventas **/

import java.awt.*;
import java.awt.event.*;
import java.text.SimpleDateFormat;
import java.util.*;
import java.sql.Date;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.*;
import hibernate.beans.*;
import hibernate.operaciones.*;

public class RegistrarVentas
{
    /******* CONTENEDOR JFRAME BASE Y PANEL GENERAL *****/

    private JFrame frmCaja;
    private JPanel panelGeneral;

    /******* ELEMENTOS DEL FORMULARIO Y LEYENDA DE LAS MISMAS *****/

    private Calendar calendario;
    private JLabel etiquetaVerFecha = new JLabel();
    private JLabel etiquetaVerClaveVendedor = new JLabel();
    private JLabel etiquetaVerNombreVendedor = new JLabel();
    private JLabel etiquetaVerNombreCliente = new JLabel();
    private JLabel etiquetaVerPuntosAcumulados = new JLabel();
    private ButtonGroup losRadio;
    private JRadioButton radioEfectivo,radioPuntos;
    private JTextField cajaNumeroCliente,cajaSubTotal,cajaIva;
    private JTextField cajaPagarPuntos,cajaTotal,cajaCodigoBarras,cajaPuntosAcum;
    private TextPrompt leyenda;

    /******* OPERACIONES PARA FUNCIONALIDAD *****/

    private OpProducto operacionesProducto = new OpProducto();
    private OpVenta operacionesVenta = new OpVenta();
    private OpProductoVenta operacionesProductoVenta = new OpProductoVenta();
    private OpCliente operacionesCliente = new OpCliente();
    private Venta laVenta = new Venta();
    private Cliente elCliente = new Cliente();
    private Producto unProducto = new Producto();
    private ProductoVenta elProductoVenta = new ProductoVenta();

    /******* TABLA CONTENEDOR DE LA VENTA *****/

    private JTable tablaVentas;
    private DefaultTableModel modeloTabla;

    private double total = 0.0, iva = 0.0;
    private int puntos = 0,puntosCobrar = 0;
    private int cantidad;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    RegistrarVentas window = new RegistrarVentas();
                    window.frmCaja.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

    }

    public RegistrarVentas()
    {
        initialize();
    }

    /***** VISIBILIDAD DEL JFRAME *****/

    public void setVisible(boolean visibilidad)
    {
        frmCaja.setVisible(visibilidad);
    }

    /***** DISEÑO DE LA PÁGINA PARA REGISTRAR VENTA *****/

    @SuppressWarnings("serial")
    private void initialize()
    {
        /***** Formato del JFrame, Scroll y Página Principal *****/

        frmCaja = new JFrame();
        frmCaja.setTitle("RegistrarVentas");

        frmCaja.setIconImage(Toolkit.getDefaultToolkit().getImage(ConsultarProducto.class.getResource("/Imagenes/Logotipo.png")));
        frmCaja.setBounds(100, 100, 450, 300);
        frmCaja.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frmCaja.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmCaja.getContentPane().setLayout(null);

        JScrollPane scrollGeneral = new JScrollPane();
        scrollGeneral.setBackground(new Color(175, 238, 238));
        scrollGeneral.setBounds(0, 0, 1365, 715);
        scrollGeneral.setHorizontalScrollBar(null);
        frmCaja.getContentPane().add(scrollGeneral);

        panelGeneral = new JPanel();
        panelGeneral.setAlignmentX(0.0f);
        panelGeneral.setAlignmentY(0.0f);
        panelGeneral.setPreferredSize(new
Dimension(Toolkit.getDefaultToolkit().getScreenSize().width, 1400));
        panelGeneral.setBackground(new Color(255, 255, 255));
        scrollGeneral.setViewportView(panelGeneral);

        /***** Formato y Funcionalidad de las Migajas *****/

        JLabel etiquetaPaginaPrincipal = new JLabel("P\u00E1gina Principal");
        etiquetaPaginaPrincipal.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent arg0)
            {
                MenuPrincipal menuPrincipal = new MenuPrincipal();

                menuPrincipal.setVisible(true);
                frmCaja.setVisible(false);
            }

            @Override
            public void mouseEntered(MouseEvent e)
            {
                etiquetaPaginaPrincipal.setForeground(new Color(232, 19, 245));
            }

            @Override
            public void mouseExited(MouseEvent e)
            {
                etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
            }
        });
    }
}

```

```

    }
});
panelGeneral.setLayout (null);

etiquetaPaginaPrincipal.setForeground(new Color(0, 0, 255));
etiquetaPaginaPrincipal.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaPaginaPrincipal.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaPaginaPrincipal.setBounds(55, 32, 163, 40);
panelGeneral.add(etiquetaPaginaPrincipal);

JLabel etiquetaDiagonal1 = new JLabel("/");
etiquetaDiagonal1.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal1.setForeground(new Color(0, 0, 255));
etiquetaDiagonal1.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal1.setBounds(210, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal1);

JLabel etiquetaAlmacen = new JLabel("Ventas");
etiquetaAlmacen.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent arg0)
    {
        MenuVentas ventas = new MenuVentas();

        ventas.setVisible(true);
        frmCaja.setVisible(false);
    }

    @Override
    public void mouseEntered(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(232, 19, 245));
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        etiquetaAlmacen.setForeground(new Color(0, 0, 255));
    }
});
etiquetaAlmacen.setForeground(new Color(0, 0, 255));
etiquetaAlmacen.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaAlmacen.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaAlmacen.setBounds(230, 32, 65, 41);
panelGeneral.add(etiquetaAlmacen);

JLabel etiquetaDiagonal2 = new JLabel("/");
etiquetaDiagonal2.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaDiagonal2.setForeground(Color.BLUE);
etiquetaDiagonal2.setFont(new Font("Century", Font.ITALIC, 25));
etiquetaDiagonal2.setBounds(290, 32, 22, 39);
panelGeneral.add(etiquetaDiagonal2);

JLabel etiquetaProductos = new JLabel("Caja");
etiquetaProductos.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaProductos.setForeground(Color.BLUE);
etiquetaProductos.setFont(new Font("Century", Font.ITALIC, 20));
etiquetaProductos.setBounds(285, 31, 90, 41);
panelGeneral.add(etiquetaProductos);

/***** Nombre del Negocio y Logotipo *****/

JLabel etiquetaNegocio = new JLabel("Mi Negocio");
etiquetaNegocio.setForeground(new Color(19, 60, 245));
etiquetaNegocio.setFont(new Font("Vivaldi", Font.PLAIN, 40));
etiquetaNegocio.setBounds(1008, 55, 202, 56);
panelGeneral.add(etiquetaNegocio);

```

```

JLabel etiquetaLogotipo = new JLabel("");
etiquetaLogotipo.setIcon(new
ImageIcon(Login.class.getResource("/Imagenes/Logotipo.png")));
etiquetaLogotipo.setBackground(new Color(175, 238, 238));
etiquetaLogotipo.setHorizontalAlignment(SwingConstants.CENTER);
etiquetaLogotipo.setBounds(1197, 35, 129, 104);
panelGeneral.add(etiquetaLogotipo);

/***** Formato y Funcionalidad de las Etiquetas para ver Datos *****/

JLabel etiquetaFecha = new JLabel("Fecha:");
etiquetaFecha.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaFecha.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaFecha.setBounds(100, 190, 69, 30);
panelGeneral.add(etiquetaFecha);

SimpleDateFormat cambiarFecha = new SimpleDateFormat("yyyy-MM-dd");
calendario = Calendar.getInstance();
etiquetaVerFecha.setText(cambiarFecha.format(calendario.getTime()));
etiquetaVerFecha.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaVerFecha.setBounds(180, 193, 120, 27);
panelGeneral.add(etiquetaVerFecha);

JLabel etiquetaClaveVendedor = new JLabel("Clave Vendedor:");
etiquetaClaveVendedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaClaveVendedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaClaveVendedor.setBounds(400, 185, 175, 30);
panelGeneral.add(etiquetaClaveVendedor);

etiquetaVerClaveVendedor.setText(DatosEstaticos.claveEmpleado);
etiquetaVerClaveVendedor.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaVerClaveVendedor.setBounds(575, 184, 1350, 36);
panelGeneral.add(etiquetaVerClaveVendedor);

JLabel etiquetaNombreVendedor = new JLabel("Nombre Vendedor:");
etiquetaNombreVendedor.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaNombreVendedor.setFont(new Font("Cambria Math", Font.PLAIN, 25));
etiquetaNombreVendedor.setBounds(795, 186, 205, 30);
panelGeneral.add(etiquetaNombreVendedor);

etiquetaVerNombreVendedor.setText(DatosEstaticos.nombreEmpleado);
etiquetaVerNombreVendedor.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaVerNombreVendedor.setBounds(1005, 184, 280, 36);
panelGeneral.add(etiquetaVerNombreVendedor);

/***** GroupBox del Tipo de Pago *****/

JPanel groupBoxTipoPago = new JPanel();
groupBoxTipoPago.setBackground(new Color(243, 243, 243));
groupBoxTipoPago.setBorder(new TitledBorder(null, "Tipo de Pago",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
groupBoxTipoPago.setBounds(149, 345, 350, 130);
groupBoxTipoPago.setLayout(null);
panelGeneral.add(groupBoxTipoPago);

radioEfectivo = new JRadioButton("Efectivo");
radioEfectivo.setBackground(new Color(243, 243, 243));
radioEfectivo.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioEfectivo.setHorizontalAlignment(SwingConstants.CENTER);
radioEfectivo.setBounds(84, 27, 184, 36);
radioEfectivo.setSelected(true);
groupBoxTipoPago.add(radioEfectivo);

radioPuntos = new JRadioButton("Puntos");
radioPuntos.setBackground(new Color(243, 243, 243));
radioPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 20));
radioPuntos.setHorizontalAlignment(SwingConstants.CENTER);
radioPuntos.setBounds(82, 69, 177, 36);

```

```

        groupBoxTipoPago.add(radioPuntos);

        losRadio = new ButtonGroup();
        losRadio.add(radioEfectivo);
        losRadio.add(radioPuntos);

/***** GroupBox para los Datos del Cliente *****/

        JPanel groupBoxCliente = new JPanel();
        groupBoxCliente.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Cliente", TitledBorder.LEADING,
TitledBorder.TOP, null, new Color(0, 0, 0)));
        groupBoxCliente.setBackground(new Color(243,243,243));
        groupBoxCliente.setBounds(660, 270, 550, 270);
        groupBoxCliente.setLayout(null);
        panelGeneral.add(groupBoxCliente);

        JLabel etiquetaNumeroCliente = new JLabel("Número de Cliente:");
        etiquetaNumeroCliente.setHorizontalAlignment(SwingConstants.RIGHT);
        etiquetaNumeroCliente.setFont(new Font("Cambria Math", Font.PLAIN, 25));
        etiquetaNumeroCliente.setBounds(35, 45, 231, 30);
        groupBoxCliente.add(etiquetaNumeroCliente);

        cajaNumeroCliente = new JTextField();
        cajaNumeroCliente.setHorizontalAlignment(SwingConstants.CENTER);
        leyenda = new TextPrompt("Número de Cliente",cajaNumeroCliente);
        leyenda.setHorizontalAlignment(SwingConstants.CENTER);
        leyenda.changeAlpha(0.50f);
        leyenda.changeStyle(Font.ITALIC);
        cajaNumeroCliente.addKeyListener(new KeyAdapter()
        {
            @Override
            public void keyPressed(KeyEvent arg0)
            {
                if((arg0.getKeyCode() == KeyEvent.VK_TAB) || (arg0.getKeyCode() ==
KeyEvent.VK_ENTER))
                {
                    if(cajaNumeroCliente.getText().equals(""))
                    {
                        JOptionPane.showMessageDialog(null,"INGRESE LA CLAVE DEL
CLIENTE");
                    }
                    else
                    {
                        elCliente =
operacionesCliente.buscarClienteRfc(cajaNumeroCliente.getText());

                        if(elCliente == null)
                        {
                            Cliente cliente = new Cliente();
                            cliente.setRfcCliente(cajaNumeroCliente.getText());
                            operacionesCliente.insertarCliente(cliente);

                            etiquetaVerNombreCliente.setText("");
                            etiquetaVerPuntosAcumulados.setText("");
                            radioPuntos.setEnabled(false);
                        }
                        else
                        {
                            etiquetaVerNombreCliente.setText(elCliente.getNombreCliente());
                            etiquetaVerPuntosAcumulados.setText(String.valueOf(elCliente.getPuntosAcumulados()));
                        }
                    }
                }
            }
        }
    }

```

```

        public void keyTyped(KeyEvent e)
        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
            {
                e.consume();
            }
        }
    });
    cajaNumeroCliente.setBounds(273, 45, 150, 36);
    cajaNumeroCliente.setColumns(10);
    cajaNumeroCliente.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    groupBoxCliente.add(cajaNumeroCliente);

    JLabel etiquetaNombreCliente = new JLabel("Nombre Cliente:");
    etiquetaNombreCliente.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaNombreCliente.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaNombreCliente.setBounds(15, 119, 205, 30);
    groupBoxCliente.add(etiquetaNombreCliente);

    etiquetaVerNombreCliente.setFont(new Font("Cambria Math", Font.PLAIN, 20));
    etiquetaVerNombreCliente.setBounds(230, 118, 280, 36);
    groupBoxCliente.add(etiquetaVerNombreCliente);

    JLabel etiquetaPuntosAcumulados = new JLabel("Puntos Acumulados:");
    etiquetaPuntosAcumulados.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaPuntosAcumulados.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    etiquetaPuntosAcumulados.setBounds(93, 197, 231, 30);
    groupBoxCliente.add(etiquetaPuntosAcumulados);

    etiquetaVerPuntosAcumulados.setFont(new Font("Cambria Math", Font.PLAIN, 20));
    etiquetaVerPuntosAcumulados.setBounds(333, 196, 78, 36);
    groupBoxCliente.add(etiquetaVerPuntosAcumulados);

    /***** Elementos para Capturar Código de Barras *****/

    JLabel etiquetaCodigoBarras = new JLabel("C\u00F3digo de Barras:");
    etiquetaCodigoBarras.setBounds(370, 610, 188, 30);
    etiquetaCodigoBarras.setHorizontalAlignment(SwingConstants.RIGHT);
    etiquetaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 25));
    panelGeneral.add(etiquetaCodigoBarras);

    cajaCodigoBarras = new JTextField();
    cajaCodigoBarras.addKeyListener(new KeyAdapter()
    {
        public void keyTyped(KeyEvent e)
        {
            char caracter = e.getKeyChar();

            if(((caracter < '0') || (caracter > '9')) && (caracter != '\b'))
            {
                e.consume();
            }
        }
    });
    cajaCodigoBarras.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda = new TextPrompt("Ingresar Código",cajaCodigoBarras);
    leyenda.setHorizontalAlignment(SwingConstants.CENTER);
    leyenda.changeAlpha(0.50f);
    leyenda.changeStyle(Font.ITALIC);
    cajaCodigoBarras.setBounds(570, 610, 198, 36);
    cajaCodigoBarras.setColumns(10);
    cajaCodigoBarras.setFont(new Font("Cambria Math", Font.PLAIN, 16));
    panelGeneral.add(cajaCodigoBarras);

    / Formato y Funcionalidad de la Tabla para Visualizar los Productos de la Venta /

```

```

JScrollPane scrollTabla = new JScrollPane();
scrollTabla.setBounds(220, 709, 900, 330);
scrollTabla.setHorizontalScrollBar(null);
panelGeneral.add(scrollTabla);

tablaVentas = new JTable();
modeloTabla = new DefaultTableModel()
{
    @Override
    public boolean isCellEditable(int row, int column)
    {
        return false;
    }
};
DefaultTableCellRenderer justificacion = new DefaultTableCellRenderer();
tablaVentas = new JTable(modeloTabla);
scrollTabla.setViewportViewView(tablaVentas);
tablaVentas.setFont(new Font("Cambria Math", Font.PLAIN, 15));
tablaVentas.setRowHeight(30);
modeloTabla.addColumn("CANTIDAD");
modeloTabla.addColumn("PRODUCTO");
modeloTabla.addColumn("PRECIO");
modeloTabla.addColumn("PUNTOS");
modeloTabla.addColumn("IMPORTE EFECTIVO");
tablaVentas.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
tablaVentas.getColumnModel().getColumn(0).setPreferredWidth(150);
tablaVentas.getColumnModel().getColumn(1).setPreferredWidth(300);
tablaVentas.getColumnModel().getColumn(2).setPreferredWidth(150);
tablaVentas.getColumnModel().getColumn(3).setPreferredWidth(150);
tablaVentas.getColumnModel().getColumn(4).setPreferredWidth(148);
justificacion.setHorizontalAlignment(SwingConstants.CENTER);
tablaVentas.getColumnModel().getColumn(0).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(1).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(2).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(3).setCellRenderer(justificacion);
tablaVentas.getColumnModel().getColumn(4).setCellRenderer(justificacion);
scrollTabla.setViewportViewView(tablaVentas);

```

/Formato y Funcionalidad de las Cajas y Etiquetas (subtotal,iva,total,etc.)/

```

JLabel etiquetaSubTotal = new JLabel("Sub-total:");
etiquetaSubTotal.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaSubTotal.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaSubTotal.setBounds(895, 1050, 90, 30);
panelGeneral.add(etiquetaSubTotal);

```

```

cajaSubTotal = new JTextField();
cajaSubTotal.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Sub-total",cajaSubTotal);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaSubTotal.setBounds(990, 1050, 130, 30);
cajaSubTotal.setColumns(10);
cajaSubTotal.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelGeneral.add(cajaSubTotal);

```

```

JLabel etiquetaIva = new JLabel("I.V.A.");
etiquetaIva.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaIva.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaIva.setBounds(905, 1080, 80, 30);
panelGeneral.add(etiquetaIva);

```

```

cajaIva = new JTextField();
cajaIva.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("I.V.A.",cajaIva);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);

```

```

leyenda.changeStyle(Font.ITALIC);
cajaIva.setBounds(990, 1080, 130, 30);
cajaIva.setColumns(10);
cajaIva.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelGeneral.add(cajaIva);

JLabel etiquetaPagarPuntos = new JLabel("Pagar con Puntos:");
etiquetaPagarPuntos.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaPagarPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaPagarPuntos.setBounds(815, 1110, 170, 30);
panelGeneral.add(etiquetaPagarPuntos);

cajaPagarPuntos = new JTextField();
cajaPagarPuntos.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Puntos", cajaPagarPuntos);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaPagarPuntos.setBounds(990, 1110, 80, 30);
cajaPagarPuntos.setColumns(10);
cajaPagarPuntos.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelGeneral.add(cajaPagarPuntos);

JLabel etiquetaPuntosAcum = new JLabel("Puntos Acumulados:");
etiquetaPuntosAcum.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaPuntosAcum.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaPuntosAcum.setBounds(810, 1140, 175, 30);
panelGeneral.add(etiquetaPuntosAcum);

cajaPuntosAcum = new JTextField();
cajaPuntosAcum.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("Puntos", cajaPuntosAcum);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaPuntosAcum.setBounds(990, 1140, 80, 30);
cajaPuntosAcum.setColumns(10);
cajaPuntosAcum.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelGeneral.add(cajaPuntosAcum);

JLabel etiquetaTotal = new JLabel("Total:");
etiquetaTotal.setHorizontalAlignment(SwingConstants.RIGHT);
etiquetaTotal.setFont(new Font("Cambria Math", Font.PLAIN, 20));
etiquetaTotal.setBounds(905, 1170, 80, 30);
panelGeneral.add(etiquetaTotal);

cajaTotal = new JTextField();
cajaTotal.setHorizontalAlignment(SwingConstants.CENTER);
leyenda = new TextPrompt("total", cajaTotal);
leyenda.setHorizontalAlignment(SwingConstants.CENTER);
leyenda.changeAlpha(0.50f);
leyenda.changeStyle(Font.ITALIC);
cajaTotal.setBounds(990, 1170, 130, 30);
cajaTotal.setColumns(10);
cajaTotal.setFont(new Font("Cambria Math", Font.PLAIN, 16));
panelGeneral.add(cajaTotal);

/***** Formato y Funcionalidad del Botón "Agregar" *****/

JButton botonAgregar = new JButton("Agregar");
botonAgregar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        unProducto =
operacionesProducto.buscarProductoPorID(cajaCodigoBarras.getText());
        if(cajaCodigoBarras.getText().equals(""))
        {

```



```

        JOptionPane.showMessageDialog(null, "INGRESE UN CÓDIGO DE BARRAS");
    }
    else if(unProducto == null)
    {
        cajaCodigoBarras.setText("");
        JOptionPane.showMessageDialog(null, "NO ESTA REGISTRADO EL PRODUCTO");
    }
    else
    {
        cantidad = Integer.valueOf(JOptionPane.showInputDialog("Cantidad de "
+ unProducto.getNombreProducto()));
        Object[] renglon =
{cantidad,unProducto.getNombreProducto(),unProducto.getCostoPublico(),cantidad*unProducto
.getPuntosAsignados(),cantidad*unProducto.getCostoPublico()};
        modeloTabla.addRow(renglon);

        cajaPuntosAcum.setText(String.valueOf(puntos +=
cantidad*unProducto.getPuntosAsignados()));
        cajaPagarPuntos.setText(String.valueOf(puntosCobrar +=
cantidad*unProducto.getPuntosCompra()));
        cajaTotal.setText(String.valueOf(String.format("%.2f",total +=
cantidad*unProducto.getCostoPublico())));
        cajaIva.setText(String.valueOf(String.format("%.2f",iva = total *
0.16)));
        cajaSubTotal.setText(String.valueOf(String.format("%.2f",total -
iva)));

        cajaCodigoBarras.setText("");

        if(etiquetaVerPuntosAcumulados.getText().equals("") ||
Integer.valueOf(cajaPagarPuntos.getText()) >
Integer.valueOf(etiquetaVerPuntosAcumulados.getText()))
        {
            radioPuntos.setEnabled(false);
        }
        else if((Integer.valueOf(cajaPagarPuntos.getText()) <=
Integer.valueOf(etiquetaVerPuntosAcumulados.getText()))
        {
            radioPuntos.setEnabled(true);
        }
    }
});
botonAgregar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonAgregar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonAgregar.setForeground(new Color(0,0,0));
    }
});
Border elBordeAgregar = new LineBorder(Color.black,1);
botonAgregar.setBorder(elBordeAgregar);
botonAgregar.setForeground(Color.BLACK);
botonAgregar.setBackground(new Color(216, 253, 241));
botonAgregar.setFont(new Font("Century", Font.PLAIN, 18));
botonAgregar.setBounds(790, 610, 100, 30);
panelGeneral.add(botonAgregar);

/***** Formato y Funcionalidad del Botón "Quitar de la Lista" *****/

JButton botonQuitar = new JButton("Quitar de la Lista");
botonQuitar.addActionListener(new ActionListener()

```

```

    {
        public void actionPerformed(ActionEvent e)
        {
            DefaultTableModel modeloTabla =
(DefaultTableModel) tablaVentas.getModel();
            modeloTabla.removeRow(tablaVentas.getSelectedRow());
            total = 0.0;
            puntos = 0;
            puntosCobrar = 0;
            if(tablaVentas.getRowCount() == 0)
            {
                cajaSubTotal.setText("");
                cajaIva.setText("");
                cajaPagarPuntos.setText("");
                cajaPuntosAcum.setText("");
                cajaTotal.setText("");
            }
            else
            {
                for (int i=0;i<tablaVentas.getRowCount();i++)
                {
                    unProducto =
operacionesProducto.buscarProducto((String) tablaVentas.getValueAt(i,1));
                    Integer cantidadLinea = (int) tablaVentas.getValueAt(i,0);

                    cajaPuntosAcum.setText(String.valueOf(puntos +=
(int)tablaVentas.getValueAt(i,3)));
                    cajaTotal.setText(String.valueOf(String.format("%.2f",total +=
Double.valueOf(String.valueOf(tablaVentas.getValueAt(i,4))))));
                    cajaPagarPuntos.setText(String.valueOf(puntosCobrar +=
cantidadLinea*unProducto.getPuntosCompra()));
                }
                cajaIva.setText(String.valueOf(String.format("%.2f",iva = total *
0.16)));
                cajaSubTotal.setText(String.valueOf(String.format("%.2f",total -
iva)));
            }

            if(cajaPagarPuntos.getText().equals(""))
            {
                radioPuntos.setEnabled(true);
            }
            else if(Integer.valueOf(cajaPagarPuntos.getText()) <=
Integer.valueOf(etiquetaVerPuntosAcumulados.getText()))
            {
                radioPuntos.setEnabled(true);
            }
            else if(Integer.valueOf(cajaPagarPuntos.getText()) >
Integer.valueOf(etiquetaVerPuntosAcumulados.getText()))
            {
                radioPuntos.setEnabled(false);
                radioPuntos.setSelected(false);
                radioEfectivo.setSelected(true);
            }
        }
    });
    botonQuitar.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseEntered(MouseEvent e)
        {
            botonQuitar.setForeground(new Color(12,2,235));
        }
        @Override
        public void mouseExited(MouseEvent e)
        {
            botonQuitar.setForeground(new Color(0,0,0));
        }
    });
}

```

```

});
Border elBordeQuitar = new LineBorder(Color.black,1);
botonQuitar.setBorder(elBordeQuitar);
botonQuitar.setForeground(Color.BLACK);
botonQuitar.setBackground(new Color(216, 253, 241));
botonQuitar.setFont(new Font("Century", Font.PLAIN, 18));
botonQuitar.setBounds(910, 610, 170, 30);
panelGeneral.add(botonQuitar);

/***** Formato y Funcionalidad del Botón "Salir" *****/

JButton botonTerminar = new JButton(" Salir");
botonTerminar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        MenuVentas menuVentas = new MenuVentas();

        menuVentas.setVisible(true);
        frmCaja.setVisible(false);
    }
});
botonTerminar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonTerminar.setForeground(new Color(0,0,0));
    }
});
Border elBordeTerminar = new LineBorder(Color.black,1);
botonTerminar.setBorder(elBordeTerminar);
botonTerminar.setHorizontalAlignment(SwingConstants.CENTER);
botonTerminar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Tache.png")));
botonTerminar.setForeground(new Color(0,0,0));
botonTerminar.setBackground(new Color(234,34,34));
botonTerminar.setFont(new Font("Century", Font.PLAIN, 20));
botonTerminar.setBounds(300, 1250, 197, 50);
panelGeneral.add(botonTerminar);

/***** Formato y Funcionalidad del Botón "Pagar" *****/

JButton botonPagar = new JButton(" Pagar");
botonPagar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(cajaNumeroCliente.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingresa el RFC del Cliente");
        }
        else
        {
            if(radioEfectivo.isSelected())
            {
                laVenta.setTipoPago("Efectivo");

                if(elCliente != null && elCliente.getPuntosAcumulados() != null)
                {
                    elCliente.setPuntosAcumulados(elCliente.getPuntosAcumulados()
+ Integer.valueOf(cajaPuntosAcum.getText()));
                }
            }
        }
    }
});

```

```

operacionesCliente.actualizarPuntosCliente(cajaNumeroCliente.getText(),elCliente.getPuntosAcumulados());
    }
    else if(elCliente != null && elCliente.getPuntosAcumulados() ==
null)
    {
elCliente.setPuntosAcumulados(Integer.valueOf(cajaPuntosAcum.getText()));
operacionesCliente.actualizarPuntosCliente(cajaNumeroCliente.getText(),elCliente.getPuntosAcumulados());
    }
    }
    else if(radioPuntos.isSelected())
    {
        laVenta.setTipoPago("Puntos");

        if(elCliente != null)
        {
            System.out.println(elCliente.getPuntosAcumulados() -
Integer.valueOf(cajaPagarPuntos.getText()));
            elCliente.setPuntosAcumulados(elCliente.getPuntosAcumulados()
- Integer.valueOf(cajaPagarPuntos.getText()));

operacionesCliente.actualizarPuntosCliente(cajaNumeroCliente.getText(),Integer.valueOf(cajaPuntosAcum.getText()) + elCliente.getPuntosAcumulados());
        }
    }

    laVenta.setEmpleadoRFC(etiquetaVerClaveVendedor.getText());
    laVenta.setFechaVenta(Date.valueOf(etiquetaVerFecha.getText()));
    laVenta.setClienteRFC(cajaNumeroCliente.getText());

laVenta.setPuntosAcumulados(Integer.valueOf(cajaPuntosAcum.getText()));
laVenta.setSubTotal(Float.valueOf(cajaSubTotal.getText()));
laVenta.setIva(Float.valueOf(cajaIva.getText()));
laVenta.setTotal(Float.valueOf(cajaTotal.getText()));
laVenta.setPuntosPagados(Integer.valueOf(cajaPagarPuntos.getText()));
operacionesVenta.RegistrarVenta(laVenta);

int ultimaVenta = operacionesVenta.tracerUltimaVenta();
for (int i=0;i<tablaVentas.getRowCount();i++)
{
elProductoVenta.setCantidad(Integer.valueOf(String.valueOf(tablaVentas.getValueAt(i,0))))
;
elProductoVenta.setPuntos(Integer.valueOf(String.valueOf(tablaVentas.getValueAt(i,3))))
;
elProductoVenta.setImporte(Float.valueOf(String.valueOf(tablaVentas.getValueAt(i,4))))
;
elProductoVenta.setProductoCodigoBarras(String.valueOf(operacionesProducto.buscarProducto
(String.valueOf(tablaVentas.getValueAt(i,1))).getCodigoBarras()));
    elProductoVenta.setVentaNumeroTransaccion(ultimaVenta);
    operacionesProductoVenta.RegistrarLineaVenta(elProductoVenta);

operacionesProducto.actualizarCantidadProductosAlmacenDevolucion(String.valueOf(operacion
esProducto.buscarProducto(
String.valueOf(tablaVentas.getValueAt(i,1))).getCodigoBarras()),
Integer.valueOf(String.valueOf(tablaVentas.getValueAt(i,0))))
;
    }

    while(modeloTabla.getRowCount()>0)
    {

```

```

        modeloTabla.removeRow(0);
    }
    cajaNumeroCliente.setText("");
    etiquetaVerNombreCliente.setText("");
    etiquetaVerPuntosAcumulados.setText("");
    cajaSubTotal.setText("");
    cajaIva.setText("");
    cajaTotal.setText("");
    cajaPagarPuntos.setText("");
    cajaPuntosAcum.setText("");
    radioEfectivo.setSelected(true);
    radioPuntos.setSelected(false);
    }
    }
});
botonPagar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botonPagar.setForeground(new Color(12,2,235));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        botonPagar.setForeground(new Color(0,0,0));
    }
});
Border elBordePagar = new LineBorder(Color.black,1);
botonPagar.setBorder(elBordePagar);
botonPagar.setIcon(new
ImageIcon(NuevoProducto.class.getResource("/Imagenes/Palomita.png")));
botonPagar.setForeground(new Color(0,0,0));
botonPagar.setBackground(new Color(95,255,95));
botonPagar.setFont(new Font("Century", Font.PLAIN, 20));
botonPagar.setBounds(850, 1250, 208, 48);
panelGeneral.add(botonPagar);
}
}

```

ANEXO F. Código de la clase que permite colocar una leyenda dentro de las cajas de texto en la aplicación de escritorio:

1.- TextPrompt.java

```

package Interfaces;

/** Esta Clase permite colocar una leyenda de ayuda dentro de las cajas de texto */

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.event.*;
import javax.swing.text.*;

@SuppressWarnings("serial")
public class TextPrompt extends JLabel
    implements FocusListener, DocumentListener
{
    public enum Show
    {
        ALWAYS,
        FOCUS_GAINED,
    }
}

```

```

        FOCUS_LOST;
    }

    private JTextComponent component;
    private Document document;
    private Show show;
    private boolean showPromptOnce;
    private int focusLost;

    public TextPrompt(String text, JTextComponent component)
    {
        this(text, component, Show.ALWAYS);
    }

    public TextPrompt(String text, JTextComponent component, Show show)
    {
        this.component = component;
        setShow( show );
        document = component.getDocument();

        setText( text );
        setFont( component.getFont() );
        setForeground( component.getForeground() );
        setBorder( new EmptyBorder( component.getInsets() ) );
        setHorizontalAlignment( JLabel.LEADING );

        component.addFocusListener( this );
        document.addDocumentListener( this );

        component.setLayout( new BorderLayout() );
        component.add( this );
        checkForPrompt();
    }

    public void changeAlpha(float alpha)
    {
        changeAlpha( (int)(alpha * 255) );
    }

    public void changeAlpha(int alpha)
    {
        alpha = alpha > 255 ? 255 : alpha < 0 ? 0 : alpha;

        Color foreground = getForeground();
        int red = foreground.getRed();
        int green = foreground.getGreen();
        int blue = foreground.getBlue();

        Color withAlpha = new Color( red, green, blue, alpha );
        super.setForeground( withAlpha );
    }

    public void changeStyle(int style)
    {
        setFont( getFont().deriveFont( style ) );
    }

    public Show getShow()
    {
        return show;
    }

    public void setShow(Show show)
    {
        this.show = show;
    }

    public boolean getShowPromptOnce()

```

```

    {
        return showPromptOnce;
    }

    public void setShowPromptOnce(boolean showPromptOnce)
    {
        this.showPromptOnce = showPromptOnce;
    }

    private void checkForPrompt()
    {
        if (document.getLength() > 0)
        {
            setVisible( false );
            return;
        }

        if (showPromptOnce && focusLost > 0)
        {
            setVisible(false);
            return;
        }

        if (component.hasFocus())
        {
            if (show == Show.ALWAYS || show == Show.FOCUS_GAINED)
                setVisible( true );
            else
                setVisible( false );
        }
        else
        {
            if (show == Show.ALWAYS
                || show == Show.FOCUS_LOST)
                setVisible( true );
            else
                setVisible( false );
        }
    }

    public void focusGained(FocusEvent e)
    {
        checkForPrompt();
    }

    public void focusLost(FocusEvent e)
    {
        focusLost++;
        checkForPrompt();
    }

    public void insertUpdate(DocumentEvent e)
    {
        checkForPrompt();
    }

    public void removeUpdate(DocumentEvent e)
    {
        checkForPrompt();
    }

    public void changedUpdate(DocumentEvent e) {}
}

```

ANEXO G. Código del archivo de configuración de Hibernate de la aplicación de escritorio:

1.-Hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>

<!-- En este Archivo se realiza la configuración general de Hibernate en donde se capturan los parámetros necesarios para la conexión y manejo de la Base de Datos utilizada en la Aplicación (MySQL), también se captura la ubicación de los archivos que contienen el mapeo de los atributos de los objetos con las columnas de las tablas de la Base de Datos -->

<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class"> com.mysql.jdbc.Driver </property>
    <property name="hibernate.connection.url"> jdbc:mysql://localhost:3306/business_db </property>
    <property name="hibernate.connection.username"> root </property>
    <property name="hibernate.connection.password"> root </property>
    <property name="hibernate.dialect"> org.hibernate.dialect.MySQLDialect </property>

    <mapping resource="hibernate/mapeos/Proveedor.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Empleado.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Cliente.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Producto.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Venta.hbm.xml"/>
    <mapping resource="hibernate/mapeos/ProductoVenta.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Pedido.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Devolucion.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

ANEXO H. Código de las clases controladoras contenidas en el paquete **uam.pt.controladores** de la aplicación web:

1.- AlmacenController.java

```
package uam.pt.controladores;

/** Esta Clase es el Controlador que permite la interacción de la Vista de Consultar Almacén con los Datos y Operaciones del Sistema */

import java.util.List;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import hibernate.operaciones.*;
import uam.pt.formas.*;
import hibernate.beans.*;

@Controller
public class AlmacenController
{
  /******* MÉTODO PARA MOSTRAR PÁGINA CONSULTAR ALMACÉN *****/

  @RequestMapping("ConsultarAlmacen")
  public ModelAndView irConsultarAlmacen()
  {
    AlmacenFechasFrm filtrosBusqueda = new AlmacenFechasFrm();
    OpProveedor operacionesProveedor = new OpProveedor();
    List<Proveedor> listaProveedores = operacionesProveedor.traerProveedores();
    filtrosBusqueda.setListaProveedores(listaProveedores);
    ModelAndView modeloConsultarAlmacen = new ModelAndView("consultarAlmacen");

    modeloConsultarAlmacen.addObject("filtrosBusquedaAlmacen",filtrosBusqueda);
  }
}
```



```

        modeloConsultarAlmacen.addObject("DatosRecuperados", filtrosBusqueda);

        return modeloConsultarAlmacen;
    }

    /***** MÉTODO PARA MOSTRAR PRODUCTOS SEGÚN FILTROS *****/

    @RequestMapping(value="filtrosAlmacen", method=RequestMethod.POST)
    public ModelAndView buscarProductos(@ModelAttribute("filtrosBusquedaAlmacen")
    AlmacenFechasFrm filtrosBusqueda)
    {
        ModelAndView modeloFiltrarBusqueda = null;
        OpProveedor operacionesProveedor = new OpProveedor();
        OpProducto operacionesProducto = new OpProducto();
        AlmacenFechasFrm productosEncontrados = new AlmacenFechasFrm();
        List<Proveedor> listaProveedores = operacionesProveedor.tracerProveedores();
        List<Producto> listaProductos;

        String desde = String.valueOf(filtrosBusqueda.getDesde());
        String hasta = String.valueOf(filtrosBusqueda.getHasta());

        productosEncontrados.setListaProveedores(listaProveedores);

        if(filtrosBusqueda.getCodigoBarras().equals("") &&
        filtrosBusqueda.getProveedorRfc().equals("Ninguno"))
        {
            listaProductos = operacionesProducto.buscarProductosPorFechas(desde, hasta);

            productosEncontrados.setListaProductos(listaProductos);

            modeloFiltrarBusqueda = new
ModelAndView("consultarAlmacen", "DatosRecuperados", productosEncontrados);
        }
        else if(!filtrosBusqueda.getCodigoBarras().equals("") &&
        filtrosBusqueda.getProveedorRfc().equals("Ninguno"))
        {
            listaProductos =
operacionesProducto.buscarProductosPorFechasCodigoBarras(desde,
hasta, filtrosBusqueda.getCodigoBarras());
            productosEncontrados.setListaProductos(listaProductos);

            modeloFiltrarBusqueda = new
ModelAndView("consultarAlmacen", "DatosRecuperados", productosEncontrados);
        }
        else if(filtrosBusqueda.getCodigoBarras().equals("") &&
!filtrosBusqueda.getProveedorRfc().equals("Ninguno"))
        {
            listaProductos = operacionesProducto.buscarProductosPorFechasProveedor(desde,
hasta, filtrosBusqueda.getProveedorRfc());
            productosEncontrados.setListaProductos(listaProductos);

            modeloFiltrarBusqueda = new
ModelAndView("consultarAlmacen", "DatosRecuperados", productosEncontrados);
        }
        else if(!filtrosBusqueda.getCodigoBarras().equals("") &&
!filtrosBusqueda.getProveedorRfc().equals("Ninguno"))
        {
            listaProductos =
operacionesProducto.buscarProductosPorFechasCodigoBarrasProveedor(desde,
hasta, filtrosBusqueda.getCodigoBarras(), filtrosBusqueda.getProveedorRfc());
            productosEncontrados.setListaProductos(listaProductos);

            modeloFiltrarBusqueda = new
ModelAndView("consultarAlmacen", "DatosRecuperados", productosEncontrados);
        }
        return modeloFiltrarBusqueda;
    }
}

```

2.- ClienteController.java

```
package uam.pt.controladores;

/** Esta Clase es el Controlador que permite la interacción de las Vistas con los Datos y
Operaciones del Sistema que involucran
 * al Cliente **/

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import java.util.*;
import hibernate.beans.*;
import hibernate.operaciones.*;
import uam.pt.formas.ClienteVentaFrm;

@Controller
public class ClienteController
{

/***** MÉTODO PARA MOSTRAR PÁGINA PRINCIPAL *****/

    @RequestMapping("Principal")
    public ModelAndView irPrincipal()
    {
        return new ModelAndView("index");
    }

/***** MÉTODO PARA MOSTRAR MENÚ CLIENTE *****/

    @RequestMapping("MenuCliente")
    public ModelAndView irMenuCliente()
    {
        return new ModelAndView("menuCliente");
    }

/***** MÉTODO PARA MOSTRAR PÁGINA NUEVA CUENTA *****/

    @RequestMapping("NuevaCuenta")
    public ModelAndView irNuevaCuenta()
    {
        Cliente cliente = new Cliente();
        ModelAndView modeloNuevaCuenta = new ModelAndView("nuevaCuenta");

        modeloNuevaCuenta.addObject("formularioCliente",cliente);

        return modeloNuevaCuenta;
    }

/***** MÉTODO PARA BUSCAR CLIENTE POR RFC *****/

    @RequestMapping(value="pasarRfcCliente", method=RequestMethod.POST)
    public ModelAndView buscarCliente(@ModelAttribute("formularioCliente") Cliente
cliente)
    {
        OpCliente operacionesCliente = new OpCliente();
        ModelAndView modeloRegistrar;

        Cliente clienteEncontrado =
operacionesCliente.buscarClienteRfc(cliente.getRfcCliente());

        if(clienteEncontrado != null)
        {
            modeloRegistrar = new
ModelAndView("nuevaCuenta","DatoRecuperado",clienteEncontrado);
        }
        else
        {

```

```

        modeloRegistrar = new ModelAndView("nuevaCuenta");
    }

    return modeloRegistrar;
}

/* MÉTODO REGISTRAR CLIENTE Y ABRIR PÁGINA PARA CONSULTAR PUNTOS */

@RequestMapping(value="guardarDatosCliente", method=RequestMethod.POST)
public ModelAndView registrarCliente(@ModelAttribute("formularioCliente") Cliente
cliente)
{
    ModelAndView modeloRegistrar;
    OpCliente operacionesCliente = new OpCliente();
    OpVenta operacionesVenta = new OpVenta();
    ClienteVentaFrm clienteVenta = new ClienteVentaFrm();
    Integer sumaPuntosAcumulados = 0;

    List<Venta> ventas =
operacionesVenta.buscarVentasPorCliente(cliente.getRfcCliente());

    for(Venta venta:ventas)
    {
        sumaPuntosAcumulados += venta.getPuntosAcumulados();
        if(venta.getTipoPago().equals("Efectivo"))
        {
            venta.setPuntosPagados(0);
        }
    }

cliente.setPuntosAcumulados(operacionesCliente.buscarClienteRfc(cliente.getRfcCliente()).
getPuntosAcumulados());
    clienteVenta.setCliente(cliente);
    clienteVenta.setVentas(ventas);

    operacionesCliente.actualizarCliente(cliente);

    modeloRegistrar = new
ModelAndView("consultarPuntos","DatosRecuperados",clienteVenta);

    return modeloRegistrar;
}

/***** MÉTODO PARA MOSTRAR PÁGINA LOGIN DEL CLIENTE *****/

@RequestMapping("LoginCliente")
public ModelAndView irLoginCliente()
{
    Cliente cliente = new Cliente();
    ModelAndView modeloLoginCliente = new ModelAndView("loginCliente");

    modeloLoginCliente.addObject("formularioCliente",cliente);

    return modeloLoginCliente;
}

/* MÉTODO PARA MOSTRAR PÁGINA CONSULTAR PUNTOS DESPUES DE VALIDAR AL CLIENTE */

@RequestMapping(value="validarCliente", method=RequestMethod.POST)
public ModelAndView validarCliente(@ModelAttribute("formularioCliente") Cliente
cliente)
{
    OpCliente operacionesCliente = new OpCliente();
    OpVenta operacionesVenta = new OpVenta();
    ModelAndView modeloValidacion;
    ClienteVentaFrm clienteVenta = new ClienteVentaFrm();
    Integer sumaPuntosAcumulados = 0;

```

```

        Cliente clienteEncontrado =
operacionesCliente.validarCliente(cliente.getUsuario(), cliente.getContrasenia());

        if(clienteEncontrado != null)
        {
            List<Venta> ventas =
operacionesVenta.buscarVentasPorCliente(clienteEncontrado.getRfcCliente());

            for(Venta venta:ventas)
            {
                if(venta.getTipoPago().equals("Efectivo"))
                {
                    venta.setPuntosPagados(0);
                    sumaPuntosAcumulados += venta.getPuntosAcumulados();
                }
                else
                {
                    sumaPuntosAcumulados -= venta.getPuntosPagados();
                }
            }

            clienteVenta.setTotalPuntos(clienteEncontrado.getPuntosAcumulados());
            clienteVenta.setCliente(clienteEncontrado);
            clienteVenta.setVentas(ventas);

            modeloValidacion = new
ModelAndView("consultarPuntos","DatosRecuperados",clienteVenta);
        }
        else
        {
            modeloValidacion = new ModelAndView("loginCliente");
        }

        return modeloValidacion;
    }
}

```

3.- ProveedorController.java

```

package uam.pt.controladores;

/** Esta Clase es el Controlador que permite la interacción de la Vista de Consultar
Proveedores con los Datos y Operaciones del Sistema */

import java.util.*;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import hibernate.beans.Proveedor;
import hibernate.operaciones.OpProveedor;
import uam.pt.formas.AlmacenFechasFrm;

@Controller
public class ProveedorController
{
    /******* MÉTODO MOSTRAR PÁGINA CONSULTAR ALMACÉN *****/

    @RequestMapping("ConsultarProveedores")
    public ModelAndView irConsultarProveedores()
    {
        AlmacenFechasFrm filtrosBusqueda = new AlmacenFechasFrm();
        Proveedor proveedor = new Proveedor();
        OpProveedor operacionesProveedor = new OpProveedor();
        List<Proveedor> listaProveedores = operacionesProveedor.traerProveedores();
    }
}

```

```

        filtrosBusqueda.setListaProveedores (listaProveedores);

        ModelAndView modeloConsultarProveedores = new
ModelAndView("consultarProveedores");

        modeloConsultarProveedores.addObject ("filtroProveedor",proveedor);
        modeloConsultarProveedores.addObject ("proveedores",filtrosBusqueda);

        return modeloConsultarProveedores;
    }

/** MÉTODO PARA REALIZAR BUSQUEDA DEL O LOS PROVEEDORES **/

    @RequestMapping(value="buscarProveedores", method=RequestMethod.POST)
    public ModelAndView procesarBusquedaProveedores (@ModelAttribute("filtroProveedor")
Proveedor proveedor)
    {
        ModelAndView modeloVerProveedores;
        OpProveedor operacionesProveedor = new OpProveedor();
        AlmacenFechasFrm proveedores = new AlmacenFechasFrm();
        List<Proveedor> listaProveedores = operacionesProveedor.tracerProveedores();

        if(proveedor.getRfcProveedor().equals("Ninguno"))
        {
            modeloVerProveedores = new ModelAndView("consultarProveedores");
        }
        else if(proveedor.getRfcProveedor().equals("Todos"))
        {
            proveedores.setListaProveedores (listaProveedores);
            modeloVerProveedores = new
ModelAndView("consultarProveedores","ProveedoresRecuperados",proveedores);
            modeloVerProveedores.addObject ("proveedores",proveedores);
        }
        else
        {
            Proveedor proveedorEncontrado =
operacionesProveedor.buscarProveedorRfc (proveedor.getRfcProveedor());

            if(proveedorEncontrado == null)
            {
                proveedores.setListaProveedores (listaProveedores);
                modeloVerProveedores = new ModelAndView("consultarProveedores");
                modeloVerProveedores.addObject ("proveedores",proveedores);
            }
            else
            {
                proveedores.setListaProveedores (listaProveedores);
                modeloVerProveedores = new
ModelAndView("consultarProveedores","ProveedorRecuperado",proveedorEncontrado);
                modeloVerProveedores.addObject ("proveedores",proveedores);
            }
        }

        return modeloVerProveedores;
    }
}

```

4.- VentaController.java

```
package uam.pt.controladores;

/** Esta Clase es el Controlador que permite la interacción de la Vista de Consultar
Ventas con los Datos y Operaciones del Sistema **/

import java.util.List;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import hibernate.beans.*;
import hibernate.operaciones.*;
import uam.pt.formas.VentasFechasFrm;

@Controller
public class VentaController
{
    /******* MÉTODO MOSTRAR LOGIN ADMINISTRADOR *****/

    @RequestMapping("LoginAdministrador")
    public ModelAndView irLoginAdministrador()
    {
        Empleado empleado = new Empleado();
        ModelAndView modeloLoginAdministrador = new ModelAndView("loginAdministrador");

        modeloLoginAdministrador.addObject("usuarioClaveEmpleado", empleado);

        return modeloLoginAdministrador;
    }

    /******* MÉTODO MOSTRAR MENÚ ENCARGADO *****/

    @RequestMapping("MenuEncargado")
    public ModelAndView irMenuEncargado()
    {
        return new ModelAndView("menuEncargado");
    }

    /******* MÉTODO VALIDAR CLAVE Y USUARIO DEL ADMINISTRADOR *****/

    @RequestMapping(value="validarEmpleado", method=RequestMethod.POST)
    public ModelAndView validarEncargado(@ModelAttribute("usuarioClaveEmpleado") Empleado
empleado)
    {
        OpEmpleado operacionesEmpleado = new OpEmpleado();
        ModelAndView modeloValidarEmpleado;
        Empleado empleadoEncontrado =
operacionesEmpleado.buscarEmpleado(empleado.getUsuario(), empleado.getContrasenia());

        if(empleadoEncontrado != null)
        {
            modeloValidarEmpleado = new ModelAndView("menuEncargado");
        }
        else
        {
            modeloValidarEmpleado = new ModelAndView("LoginAdministrador");
        }

        return modeloValidarEmpleado;
    }

    /******* MÉTODO MOSTRAR PÁGINA CONSULTAR VENTAS *****/

    @RequestMapping("ConsultarVentas")
    public ModelAndView irConsultarVentas()
    {
        VentasFechasFrm filtrosBusqueda = new VentasFechasFrm();
    }
}
```

```

        ModelAndView modeloConsultarVentas = new ModelAndView("consultarVentas");

        modeloConsultarVentas.addObject("filtrosBusquedaVentas", filtrosBusqueda);

        return modeloConsultarVentas;
    }

    /***** MÉTODO MOSTRAR VENTAS SEGÚN FILTROS *****/

    @RequestMapping(value="filtros", method=RequestMethod.POST)
    public ModelAndView buscarVentas(@ModelAttribute("filtrosBusquedaVentas")
    VentasFechasFrm filtrosBusqueda)
    {
        OpVenta operacionesVenta = new OpVenta();
        VentasFechasFrm ventasEncontradas = new VentasFechasFrm();
        List<Venta> listaVentas;
        ModelAndView modeloBuscar = null;
        String desde = String.valueOf(filtrosBusqueda.getDesde());
        String hasta = String.valueOf(filtrosBusqueda.getHasta());

        if(filtrosBusqueda.getEmpleadoRFC().equals("Ninguno") &&
        filtrosBusqueda.getTipoPago().equals("Ninguno"))
        {
            listaVentas = operacionesVenta.buscarVentasPorFechas(desde, hasta);
            ventasEncontradas.setListaVentas(listaVentas);

            modeloBuscar = new
            ModelAndView("consultarVentas", "VentasRecuperadas", ventasEncontradas);
        }
        else if(!filtrosBusqueda.getEmpleadoRFC().equals("Ninguno") &&
        filtrosBusqueda.getTipoPago().equals("Ninguno"))
        {
            listaVentas = operacionesVenta.buscarVentasPorFechasEmpleado(desde,
            hasta, filtrosBusqueda.getEmpleadoRFC());
            ventasEncontradas.setListaVentas(listaVentas);

            modeloBuscar = new
            ModelAndView("consultarVentas", "VentasRecuperadas", ventasEncontradas);
        }
        else if(filtrosBusqueda.getEmpleadoRFC().equals("Ninguno") &&
        !filtrosBusqueda.getTipoPago().equals("Ninguno"))
        {
            listaVentas = operacionesVenta.buscarVentasPorFechasTipoPago(desde,
            hasta, filtrosBusqueda.getTipoPago());
            ventasEncontradas.setListaVentas(listaVentas);

            modeloBuscar = new
            ModelAndView("consultarVentas", "VentasRecuperadas", ventasEncontradas);
        }
        else if(!filtrosBusqueda.getEmpleadoRFC().equals("Ninguno") &&
        !filtrosBusqueda.getTipoPago().equals("Ninguno"))
        {
            listaVentas = operacionesVenta.buscarVentasPorFechasEmpleadoTipoPago(desde,
            hasta, filtrosBusqueda.getEmpleadoRFC(), filtrosBusqueda.getTipoPago());
            ventasEncontradas.setListaVentas(listaVentas);

            modeloBuscar = new
            ModelAndView("consultarVentas", "VentasRecuperadas", ventasEncontradas);
        }

        return modeloBuscar;
    }
}

```

ANEXO I. Código de las clases contenidas en el paquete **uam.pt.formas** de la aplicación web:

1.-AlmacenFechasFrm.java

```
package uam.pt.formas;

/** Esta Clase Especial permite el manejo de datos que pertenecen a distintas Clases que
son necesarios para ser mostrados en la Vista
 * de Consultar Almacén **/

import java.sql.Date;
import java.util.List;
import hibernate.beans.*;

public class AlmacenFechasFrm
{
    private Date desde;
    private Date hasta;
    private String codigoBarras;
    private String proveedorRfc;
    private List<Producto> listaProductos;
    private List<Proveedor> listaProveedores;

    /***** MÉTODOS GETTER'S Y SETTER'S *****/

    public Date getDesde()
    {
        return desde;
    }
    public void setDesde(Date desde)
    {
        this.desde = desde;
    }

    public Date getHasta()
    {
        return hasta;
    }
    public void setHasta(Date hasta)
    {
        this.hasta = hasta;
    }

    public String getCodigoBarras()
    {
        return codigoBarras;
    }
    public void setCodigoBarras(String codigoBarras)
    {
        this.codigoBarras = codigoBarras;
    }

    public String getProveedorRfc()
    {
        return proveedorRfc;
    }
    public void setProveedorRfc(String proveedorRfc)
    {
        this.proveedorRfc = proveedorRfc;
    }

    public List<Producto> getListaProductos()
    {
        return listaProductos;
    }
    public void setListaProductos(List<Producto> listaProductos)

```



```

    {
        this.listaProductos = listaProductos;
    }

    public List<Proveedor> getListaProveedores()
    {
        return listaProveedores;
    }
    public void setListaProveedores(List<Proveedor> listaProveedores)
    {
        this.listaProveedores = listaProveedores;
    }
}

```

2.- ClienteVentaFrm.java

```

package uam.pt.formas;

/** Esta Clase Especial permite el manejo de datos que pertenecen a distintas Clases que
son necesarios para ser mostrados en las Vistas
 * de Nueva Cuenta y Consultar Puntos **/

import java.util.*;
import hibernate.beans.*;

public class ClienteVentaFrm
{
    private List<Venta> ventas;
    private Cliente cliente;
    private Integer totalPuntos;

    /******* MÉTODOS GETTTER's Y SETTER's *****/

    public List<Venta> getVentas()
    {
        return ventas;
    }
    public void setVentas(List<Venta> ventas)
    {
        this.ventas = ventas;
    }
    public Cliente getCliente()
    {
        return cliente;
    }
    public void setCliente(Cliente cliente)
    {
        this.cliente = cliente;
    }
    public Integer getTotalPuntos()
    {
        return totalPuntos;
    }
    public void setTotalPuntos(Integer totalPuntos)
    {
        this.totalPuntos = totalPuntos;
    }
}

```

3.- VentasFechasFrm.java

```
package uam.pt.formas;

/** Esta Clase Especial permite el manejo de datos que pertenecen a distintas Clases que
son necesarios para ser mostrados en la Vista
 * de Consultar Ventas **/

import java.sql.Date;
import java.util.List;
import hibernate.beans.*;

public class VentasFechasFrm
{
    private Date desde;
    private Date hasta;
    private String tipoPago;
    private String empleadoRFC;
    private List<Venta> listaVentas;

    /******* MÉTODOS GETTTER's Y SETTER's *****/

    public Date getDesde()
    {
        return desde;
    }
    public void setDesde(Date desde)
    {
        this.desde = desde;
    }

    public Date getHasta()
    {
        return hasta;
    }
    public void setHasta(Date hasta)
    {
        this.hasta = hasta;
    }

    public String getTipoPago()
    {
        return tipoPago;
    }
    public void setTipoPago(String tipoPago)
    {
        this.tipoPago = tipoPago;
    }

    public String getEmpleadoRFC()
    {
        return empleadoRFC;
    }
    public void setEmpleadoRFC(String empleadoRFC)
    {
        this.empleadoRFC = empleadoRFC;
    }

    public List<Venta> getListaVentas()
    {
        return listaVentas;
    }
    public void setListaVentas(List<Venta> listaVentas)
    {
        this.listaVentas = listaVentas;
    }
}
}
```

ANEXO J. Código del archivo `servlet.xml` de la aplicación web:

1.- controlador-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite indicar la ubicación de los controladores que serán utilizados
de manera automática por la aplicación y
también se esta configurando la forma en que se van a redireccionar las vistas
después de ser invocadas -->

<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="uam.pt.controladores"/>

    <bean id="viewResolver"
class="org.springframework.web.servlet.view.UrlBasedViewResolver">
        <property name="viewClass"
value="org.springframework.web.servlet.view.JstlView" />
        <property name="prefix" value="/jsp"/>
        <property name="suffix" value=".jsp"/>
    </bean>

    <mvc:resources mapping="/recursos/**" location="/WEB-INF"/>
    <mvc:annotation-driven/>
</beans>
```

2.- web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Este Archivo permite dar de alta el Servlet "DispatcherServlet" para poder atender
las tareas y capturar las peticiones -->
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">

    <display-name>SistemaWebAdministracionInformacion</display-name>

    <welcome-file-list>
        <welcome-file>jsp/index.jsp</welcome-file>
    </welcome-file-list>

    <servlet>
        <servlet-name>controlador</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>controlador</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
</web-app>
```

ANEXO K. Código de las vistas .jsp que conforman la Aplicación Web:

1.- index.jsp

```
<!-- Esta Vista muestra un Carrusel de ofertas que serán definidas por el Negocio y
también se muestran 2 enlaces para acceder
al Sistema Web como "Cliente" o como "Administrador" -->

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Página de Inicio</title>
    <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" />
    <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" />
    <link rel="stylesheet" href="<c:url
value="/recursos/css_propios/principal.css"/>" />
    <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
    <script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
  </head>
  <body>
    <div class="container">
      <header class="row encabezado">
        <div class="col-lg-4 col-md-4 hidden-sm hidden-xs col-lg-offset-3
col-md-offset-3 nombreNegocio">
          <font face="Vivaldi">Mi Negocio</font>
        </div>
        <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
          <figure>
            " />
          </figure>
        </div>
      </header>
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
        <span class="glyphicon glyphicon-menu-hamburger"></span>
      </button>
      <div class="row cuerpo">
        <div class="row migajas">
          <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
            <ol class="breadcrumb">
              <li><a href="MenuCliente.html">Cliente</a></li>
              <li><a href="LoginAdministrador.html">Administrador</a></li>
            </ol>
          </div>
        </div>
        <div class="titulo">
          <font face="Cambria"><h1>Consulta Nuestras Promociones</h1></font>
        </div>
        <div id="carrusel" class="carousel slide" data-interval="3000" data-
ride="carousel">
          <ol class="carousel-indicators">
            <li data-target="#carrusel" data-slide-to="0"></li>
            <li data-target="#carrusel" data-slide-to="1"></li>
            <li data-target="#carrusel" data-slide-to="2"></li>
            <li data-target="#carrusel" data-slide-to="3"></li>
          </ol>
          <div class="carousel-inner">
            <div class="active item">
              " />
              <div class="carousel-caption">
```

```

                <h2>2 Lts de Blanqueador con 20% de Descuento de $30 a
$24</h2>
            </div>
        </div>
        <div class="item">
            " />
            <div class="carousel-caption">
                <h2>Llévate 2x1 <br/> Solo por Hoy</h2>
            </div>
        </div>
        <div class="item">
            " />
            <div class="carousel-caption">
                <h2>Pan Integral Chico con 25% de Descuento <br/> De $28
a $21 solo por Hoy</h2>
            </div>
        </div>
        <div class="item">
            " />
            <div class="carousel-caption">
                <h2>En la Compra de unas Papas Grandes Llévate otras
chicas Gratis <br/>
                Costo de Papas Grandes $25</h2>
            </div>
        </div>
    </div>
    <div>
        <a class="carousel-control left" href="#carrusel" data-slide="prev">
            <span class="glyphicon glyphicon-chevron-left"></span>
        </a>
        <a class="carousel-control right" href="#carrusel" data-slide="next">
            <span class="glyphicon glyphicon-chevron-right"></span>
        </a>
    </div>
</div>
</div>
</body>
</html>

```

2.- menuCliente.jsp

```

<!-- Esta Vista muestra las 2 secciones a las que puede acceder un Cliente que son:
"Nueva Cuenta" y "Puntos" -->

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Menú Cliente</title>
        <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" />
        <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" />
        <link rel="stylesheet" href="<c:url
value="/recursos/css_propios/menuCliente.css"/>" />
        <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
        <script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
    </head>
    <body>

        <div class="container">
            <header class="row encabezado">

```

```

        <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
            <font face="Vivaldi">Mi Negocio</font>
        </div>
        <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
            <figure>
                "/>
            </figure>
        </div>
    </header>
    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
        <span class="glyphicon glyphicon-menu-hamburger"></span>
    </button>
    <div class="row cuerpo">
        <div class="row migajas">
            <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
                <ol class="breadcrumb">
                    <li><a href="Principal.html">Inicio</a></li>
                    <li class="active">Menú Cliente</li>
                </ol>
            </div>
            <div class="row iconosMenu">
                <div class="col-lg-offset-2 col-md-offset-2 col-sm-offset-2 col-lg-3
col-md-3 col-sm-3">
                    <figure>
                        "/>
                        <label>
                            <font face="Cambria"><a href="NuevaCuenta.html">Cuenta
Nueva</a></font>
                        </label>
                    </figure>
                </div>
                <div class="col-lg-offset-2 col-md-offset-2 col-sm-offset-2 col-lg-3
col-md-3 col-sm-3">
                    <figure>
                        "/>
                        <label class="puntos">
                            <font face="Cambria"><a
href="LoginCliente.html">Puntos</a></font>
                        </label>
                    </figure>
                </div>
            </div>
        </div>
    </div>
</body>
</html>

```

3.- loginCliente.jsp

```

<!-- Esta Vista permite la autenticación de un Cliente para poder acceder a la vista de
"Consultar Puntos" -->

<%@ page language="java" contentType="text/html; charset=iso-8859-1" pageEncoding="iso-8859-1"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <title>Login Cliente</title>
    </head>

```

```

<link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>"/>"/>
<link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>"/>"/>
<link rel="stylesheet" href="<c:url value="/recursos/css_propios/Login.css"/>"/>"/>

<script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
<script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
</head>
<body>
<div class="container">
  <header class="row encabezado">
    <div class="col-lg-4 col-md-4 col-sm-4 hidden-sm hidden-
xs col-lg-offset-3 col-md-offset-3 nombreNegocio">
      <font face="Vivaldi">Mi Negocio</font>
    </div>
    <div class="col-lg-2 col-md-2 col-sm-2 Logotipo">
      <figure>
        " /> />
      </figure>
    </div>
  </header>
  <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target=".migajas" aria-expanded="false" aria-controls="navbar">
    <span class="glyphicon glyphicon-menu-hamburger"></span>
  </button>
  <div class="row cuerpo">
    <div class="row migajas">
      <div class="navbar-collapse collapse col-Lg-12 col-md-12 hidden-sm hidden-
xs">
        <ol class="breadcrumb">
          <li><a href="Principal.html">Inicio</a></li>
          <li><a href="MenuCliente.html">Menú Cliente</a></li>
          <li class="active">Login</li>
        </ol>
      </div>
    </div>
    <div class="formularioLogin">
      <div class="Icon">
        <span class="glyphicon glyphicon-user"></span>
      </div>
      <div class="ContentForm">
        <tag:form action="validarCliente" method="POST"
modelAttribute="formularioCliente" class="form-inline" role="form">
          <div class="form-group input-group input-group-Lg centrado" >
            <span class="input-group-addon icono"><i class="glyphicon
glyphicon-user"></i></span>
            <tag:input path="usuario" class="form-control"
placeholder="Usuario" required="required"/>
          </div>
          <br/><br/>
          <div class="form-group input-group input-group-Lg centrado" >
            <span class="input-group-addon icono"><i class="glyphicon
glyphicon-Lock"></i></span>
            <tag:password path="contrasenia" class="form-control"
placeholder="Clave" required="required"/>
          </div>
          <tag:button class="btn">Ingresar</tag:button>
        </tag:form>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

4.- NuevaCuenta.jsp

```
<!-- Esta Vista muestra un formulario para que el Cliente ingrese sus datos y pueda
consultar los puntos acumulados con los que cuenta -->

<%@ page language="java" contentType="text/html; charset=iso-8859-1" pageEncoding="iso-
8859-1"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Nueva Cuenta</title>
<link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" />
<link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" />
<link rel="stylesheet" href="<c:url
value="/recursos/css_propios/nuevaCuenta.css"/>" />
<link rel="stylesheet" href="<c:url
value="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css"/>" />
<script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
<script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
<script src="<c:url value="http://code.jquery.com/ui/1.10.1/jquery-
ui.js"/>"></script>
<script>
    $(function()
    {
        $("#elDatePicker").datepicker
        ({
            dateFormat: "yy-mm-dd",
            showAnim: "slideDown",
            changeMonth: true,
            changeYear: true,
            firstDay: 1,
            maxDate: 0,
            maxYear: 2018,
            dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
            monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul",
"Ago", "Sep", "Oct", "Nov", "Dic"]
        });
    });
</script>
</head>

<body>
<div class="container">
<header class="row encabezado">
<div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
<font face="Vivaldi">Mi Negocio</font>
</div>
<div class="col-lg-2 col-md-2 col-sm-2 logotipo">
<figure>
" />
</figure>
</div>
</header>
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
<span class="glyphicon glyphicon-menu-hamburger"></span>
</button>
<div class="row cuerpo">
<div class="row migajas">
<div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
<ol class="breadcrumb">
<li><a href="Principal.html">Inicio</a></li>
</ol>
</div>
</div>
</div>
</body>
</html>
```



```

        <li><a href="MenuCliente.html">Menú Cliente</a></li>
        <li class="active">Nueva Cuenta</li>
    </ol>
</div>
</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 formularioPregunta">
    <div class="preregistro">
        <label><font face="Cambria" size="4px">¿Ya cuenta con el Pre-
Registro de su cuenta?</font></label>
        <tag:form action="pasarRfcCliente" method="POST"
modelAttribute="formularioCliente" class="form-inline">
            <label><font face="Cambria" size="4px">Ingrese su
RFC:</font></label>
            <tag:input path="rfcCliente" class="form-control"
placeholder="Número de RFC" size="16" style="font-size:15px;text-align:center"
required="required"/>
            <br/><br/><br/>
            <tag:button class="btn"><font>Buscar</font></tag:button>
        </tag:form>
    </div>
</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 usuarioLogin">
    <div class="formulario1">
        <tag:form action="guardarDatosCliente" name="formularioPrincipal"
modelAttribute="formularioCliente" method="POST" class="form-inline" id="formulario">
            <label><font face="Cambria"
size="4px">Usuario:</font></label>
            <tag:input path="usuario" class="form-control"
placeholder="Ingresar Usuario" required="required" size="12" style="font-size:15px;text-
align:center"/>
            <br/>
            <label><font face="Cambria" size="4px">Clave:</font></label>
            <tag:password path="contrasenia" class="form-control"
placeholder="Ingresar Clave" required="required" size="12" style="font-size:15px;text-
align:center"/>
            <br/>
            <label><font face="Cambria" size="4px">Fecha:</font></label>
            <tag:input path="fechaRegistro" class="form-control"
id="elDatePicker" placeholder="Click Aquí" size="10" style="font-size:15px;text-
align:center" value="{DatoRecuperado.fechaRegistro}" required="required"/>
            <br/>
            <label><font face="Cambria" size="4px">RFC: </font></label>
            <tag:input path="rfcCliente" class="form-control"
placeholder="Ingresar RFC" size="15" style="font-size:15px;text-align:center"
value="{DatoRecuperado.rfcCliente}" required="required"/>
            <br/>
            <label><font face="Cambria" size="4px">Nombre:
</font></label>
            <tag:input path="nombreCliente" id="elNombre" class="form-
control" placeholder="Nombre de Cliente" size="27" style="font-size:15px;text-
align:center" value="{DatoRecuperado.nombreCliente}" required="required"/>
            <br/>
            <label><font face="Cambria" size="4px">Dirección:
</font></label>
            <tag:input path="direccion" class="form-control"
placeholder="Ingresar Dirección" style="font-size:15px;text-
align:center;width:220px;height:100px" value="{DatoRecuperado.direccion}"
required="required"></tag:input>
            <br/><br/>
            <label><font face="Cambria" size="4px">Correo:
</font></label>
            <tag:input path="correo" class="form-control"
placeholder="correo@correo.com" size="32" style="font-size:15px;text-align:center"
value="{DatoRecuperado.correo}" required="required"/>
            <br/>
    </div>
</div>

```

```

        <label><font face="Cambria" size="4px">Teléfono:
</font></label>
        <tag:input path="telefono" class="form-control"
placeholder="55-55-55-55" size="12" style="font-size:15px;text-align:center"
value="{DatoRecuperado.telefono}" required="required"/>
        <br/>
        <tag:button id="boton" class="btn"><font >Agregar
Datos</font></tag:button>
    </tag:form>
</div>
</div>
</div>
</div>
</body>
</html>

```

5.-consultarPuntos.jsp

```

<!-- Esta Vista permite al Cliente consultar de manera remota los puntos acumulados,
pagos con puntos y sus compras realizadas -->

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Resgistro y Consulta de Puntos</title>
        <link rel="stylesheet" href="{c:url value="/recursos/css/bootstrap.min.css"/}>
        <link rel="stylesheet" href="{c:url value="/recursos/css_propios/base.css"/}>
        <link rel="stylesheet" href="{c:url value="/recursos/css_propios/puntos.css"/}>
        <link rel="stylesheet" href="{c:url
value="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css"/}>
        <script src="{c:url value="/recursos/js/jquery-3.2.1.min.js"/}></script>
        <script src="{c:url value="/recursos/js/bootstrap.min.js"/}></script>
        <script src="{c:url value="http://code.jquery.com/ui/1.10.1/jquery-
ui.js"/}></script>
    </head>
    <body>
        <div class="container">
            <header class="row encabezado">
                <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
                    <font face="Vivaldi">Mi Negocio</font>
                </div>
                <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
                    <figure>
                        
                    </figure>
                </div>
            </header>
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
                <span class="glyphicon glyphicon-menu-hamburger"></span>
            </button>
            <div class="row cuerpo">
                <div class="row migajas">
                    <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
                        <ol class="breadcrumb">
                            <li><a href="Principal.html">Inicio</a></li>
                            <li><a href="MenuCliente.html">Menú Cliente</a></li>

```

```

        <li class="active">Puntos</li>
    </ol>
</div>
</div>

<div class="col-lg-12 col-md-12 col-sm-12" action="traerVentas"
method="POST" modelAttribute="formularioCliente">
    <div class="formatoEtiquetas col-lg-5 col-md-5 col-sm-5 col-xs-5">
        <label path="rfcCliente"><font face="Cambria" size="4px">RFC:
        ${DatosRecuperados.cliente.rfcCliente}</font></label>
    </div>
    <div class="formatoEtiquetas col-lg-6 col-md-6 col-sm-6 col-xs-6">
        <label><font face="Cambria" size="4px">Nombre:
        ${DatosRecuperados.cliente.nombreCliente}</font></label>
    </div>
    <div class="centrado col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <label><font face="Cambria" size="4px">Total Puntos Acumulados:
        ${DatosRecuperados.cliente.puntosAcumulados}</font></label>
    </div>
</div>

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
    <table class="table table-bordered laTabla">
        <thead>
            <tr>
                <th>Fecha</th>
                <th>Ticket</th>
                <th>Importe</th>
                <th>Puntos Pagados</th>
                <th>Puntos Acumulados</th>
            </tr>
            <c:forEach items="${DatosRecuperados.ventas}" var="venta">
                <tr>
                    <td style="text-align:
                    center;">${venta.fechaVenta}</td>
                    <td style="text-align:
                    center;">${venta.numeroTransaccion}</td>
                    <td style="text-align: center;">${venta.total}</td>
                    <td style="text-align:
                    center;">${venta.puntosPagados}</td>
                    <td style="text-align:
                    center;">${venta.puntosAcumulados}</td>
                </tr>
            </c:forEach>
        </thead>
    </table>
</div>
</div>
</body>
</html>

```

6.- loginAdministrador.jsp

```

<!-- Esta Vista permite la autenticación de un Administrador o Encargado para poder
acceder a las vistas de consulta de "Ventas",
"Almacén" y "Proveedores" -->
<%@ page language="java" contentType="text/html; charset=iso-8859-1" pageEncoding="iso-
8859-1"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

```

```

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <title>Login Administrador</title>
  <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" /> />
  <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" /> />
  <link rel="stylesheet" href="<c:url value="/recursos/css_propios/login.css"/>" /> />
  <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>" /> </script>
  <script src="<c:url value="/recursos/js/bootstrap.min.js"/>" /> </script>
</head>
<body>
  <div class="container">
    <header class="row encabezado">
      <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
        <font face="Vivaldi">Mi Negocio</font>
      </div>
      <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
        <figure>
          " />
        </figure>
      </div>
    </header>
    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
      <span class="glyphicon glyphicon-menu-hamburger"></span>
    </button>
    <div class="row cuerpo">
      <div class="row migajas">
        <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
          <ol class="breadcrumb">
            <li><a href="Principal.html">Inicio</a></li>
            <li class="active">Login Administrador</li>
          </ol>
        </div>
      </div>
      <div class="formularioLogin">
        <div class="Icon">
          <span class="glyphicon glyphicon-user"></span>
        </div>
        <div class="ContentForm">
          <tag:form class="form-inline" role="form"
action="validarEmpleado" method="POST" modelAttribute="usuarioClaveEmpleado">
            <div class="form-group input-group input-group-lg centrado" >
              <span class="input-group-addon icono"><i class="glyphicon
glyphicon-user"></i></span>
              <tag:input type="text" path="usuario" class="form-
control" placeholder="Usuario" required="required"/>
            </div>
            <br/><br/>
            <div class="form-group input-group input-group-lg centrado" >
              <span class="input-group-addon icono"><i class="glyphicon
glyphicon-lock"></i></span>
              <tag:input type="password" path="contrasenia"
class="form-control" placeholder="Clave" required="required"/>
            </div>
            <tag:button class="btn">Ingresar</tag:button>
          </tag:form>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

```

7.- menuEncargado.jsp

```
<!-- Esta Vista muestra las 3 secciones a las que puede acceder un Administrador o Encargado que son: "Ventas", "Almacén" y "Proveedores" -->
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Menú Encargado</title>
    <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" /> />
    <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" /> />
    <link rel="stylesheet" href="<c:url
value="/recursos/css_propios/menuEncargado.css"/>" /> />
    <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>" /></script>
    <script src="<c:url value="/recursos/js/bootstrap.min.js"/>" /></script>
  </head>

  <body>
    <div class="container">
      <header class="row encabezado">
        <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
          <font face="Vivaldi">Mi Negocio</font>
        </div>
        <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
          <figure>
            " />
          </figure>
        </div>
      </header>
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
        <span class="glyphicon glyphicon-menu-hamburger"></span>
      </button>
      <div class="row cuerpo">
        <div class="row migajas">
          <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
            <ol class="breadcrumb">
              <li><a href="Principal.html">Inicio</a></li>
              <li class="active">Menú Administrador</li>
            </ol>
          </div>
          <div>
            <h1><font face="Cambria">Consultar:</font></h1>
            <div class="row iconosMenu">
              <div class="col-lg-offset-1 col-md-offset-1 col-sm-offset-1 col-lg-2
col-md-2 col-sm-2">
                <figure>
                  " />
                  <label>
                    <font face="Cambria"><a
href="ConsultarVentas.html">Ventas</a></font>
                  </label>
                </figure>
              </div>
              <div class="col-lg-offset-2 col-md-offset-2 col-sm-offset-2 col-lg-2
col-md-2 col-sm-2">
                <figure>
                  " />
                  <label class="puntos">

```

```

        <font face="Cambria"><a
href="ConsultarAlmacen.html">Almacén</a></font>
        </label>
    </figure>
</div>
<div class="col-lg-offset-2 col-md-offset-2 col-sm-offset-2 col-lg-2
col-md-2 col-sm-2">
    <figure>
        "/>
        <label class="puntos">
            <font face="Cambria"><a
href="ConsultarProveedores.html">Proveedores</a></font>
        </label>
    </figure>
</div>
</div>
</div>
</body>
</html>

```

8.- consultarAlmacen.jsp

```

<!-- Esta Vista permite al Administrador o Encargado consultar de manera remota los datos
de los productos por rango de fechas
unicamente o fechas, código de barras y/o proveedor -->

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Consultar Almacén</title>
        <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>"/>
        <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>"/>
        <link rel="stylesheet" href="<c:url
value="/recursos/css_propios/almacen.css"/>"/>
        <link rel="stylesheet" href="<c:url
value="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css"/>"/>
        <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
        <script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
        <script src="<c:url value="http://code.jquery.com/ui/1.10.1/jquery-
ui.js"/>"></script>
        <script>
            $(function()
            {
                $("#elDatePicker1").datepicker
                ({
                    dateFormat: "yy-mm-dd",
                    showAnim: "slideDown",
                    changeMonth: true,
                    changeYear: true,
                    firstDay: 1,
                    maxDate: 0,
                    maxYear: 2018,
                    dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
                    monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul",
"Ago", "Sep", "Oct", "Nov", "Dic"]
                });
            });
        </script>
        <script>
            $(function()

```

```

        {
            $("#elDatePicker2").datepicker
            (
                {
                    dateFormat: "yy-mm-dd",
                    showAnim: "slideDown",
                    changeMonth: true,
                    changeYear: true,
                    firstDay: 1,
                    maxDate: 0,
                    maxYear: 2018,
                    dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
                    monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul",
"Ago", "Sep", "Oct", "Nov", "Dic"]
                }
            );
        }
    </script>
</head>

<body>
    <div class="container">
        <header class="row encabezado">
            <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
                <font face="Vivaldi">Mi Negocio</font>
            </div>
            <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
                <figure>
                    "/>
                </figure>
            </div>
        </header>
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
            <span class="glyphicon glyphicon-menu-hamburger"></span>
        </button>
        <div class="row cuerpo">
            <div class="row migajas">
                <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
                    <ol class="breadcrumb">
                        <li><a href="Principal.html">Inicio</a></li>
                        <li><a href="MenuEncargado.html">Menú Encargado</a></li>
                        <li class="active">Almacén</li>
                    </ol>
                </div>
                <div class="col-lg-12 col-md-12 col-sm-12">
                    <tag:form class="filtrosConsulta" action="filtrosAlmacen"
method="POST" modelAttribute="filtrosBusquedaAlmacen">
                        <div class="form-inline col-lg-5 col-md-5 col-sm-5">
                            <fieldset class="rangoConsulta">
                                <legend><font face="Cambria">Rango de
Consulta:</font></legend>
                                <label><font face="Cambria">Desde:</font></label>
                                <tag:input path="desde" class="form-control" size="10"
id="elDatePicker1" placeholder="Click Aquí" required="required"/>
                                <br/><br/>
                                <label><font face="Cambria">Hasta:</font></label>
                                <tag:input path="hasta" class="form-control" size="10"
id="elDatePicker2" placeholder="Click Aquí" required="required"/>
                                <br/><br/>
                            </fieldset>
                        </div>
                        <div class="form-inline col-lg-offset-1 col-md-offset-1 col-sm-
offset-1 col-lg-6 col-md-6 col-sm-6">
                            <fieldset class="consultaAvanzada">

```

```

Avanzada:</font></legend>
Barras:</font></label>
placeholder="Ingresar Código"/>
Opción</tag:option>
items="{DatosRecuperados.listaProveedores}" itemValue="rfcProveedor"
itemLabel="nombreProveedor"/>
</tag:select>
<br/><br/>
</fieldset>
</div>
<div class="botonConsultar col-lg-12 col-md-12 col-sm-12">
<tag:button class="btn">Consultar</tag:button>
</div>
</tag:form>
</div>
<div class="tabla col-lg-12 col-md-12 col-sm-12 col-xs-12">
<table class="table table-bordered">
<thead>
<tr>
<th>Fecha</th>
<th>Código</th>
<th>Producto</th>
<th>Proveedor</th>
<th>Costo Compra</th>
<th>Costo Público</th>
<th>Puntos</th>
<th>Entrada</th> -->
<th>Salida</th> -->
<th>Existencia</th>
</tr>
</thead>
<c:forEach items="{DatosRecuperados.listaProductos}"
var="producto">
<tr>
<td style="text-align:
center;">${producto.fechaRegistro}</td>
<td style="text-align:
center;">${producto.codigoBarras}</td>
<td style="text-align:
center;">${producto.nombreProducto}</td>
<td style="text-align:
center;">${producto.proveedor}</td>
<td style="text-align:
center;">${producto.costoCompra}</td>
<td style="text-align:
center;">${producto.costoPublico}</td>
<td style="text-align:
center;">${producto.puntosAsignados}</td>
<td style="text-align:
center;">${producto.cantidadIngreso}</td>
</tr>
</c:forEach>
</thead>
</table>
<div class="botonSalir col-lg-12 col-md-12 col-sm-12 col-xs-12">
<a href="MenuEncargado.html"><button
class="btn">Salir</button></a>
</div>
</div>
</div>

```



```
</body>
</html>
```

9.- consultarProveedores.jsp

```
<!-- Esta Vista permite al Administrador o Encargado consultar de manera remota los datos
de uno o de todos los proveedores -->
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Consultar Proveedores</title>
    <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" />
    <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" />
    <link rel="stylesheet" href="<c:url
value="/recursos/css_propios/proveedores.css"/>" />
    <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
    <script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
  </head>

  <body>
    <div class="container">
      <header class="row encabezado">
        <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
          <font face="Vivaldi">Mi Negocio</font>
        </div>
        <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
          <figure>
            " />
          </figure>
        </div>
      </header>
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
        <span class="glyphicon glyphicon-menu-hamburger"></span>
      </button>
      <div class="row cuerpo">
        <div class="row migajas">
          <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
            <ol class="breadcrumb">
              <li><a href="Principal.html">Inicio</a></li>
              <li><a href="MenuEncargado.html">Menú Encargado</a></li>
              <li class="active">Proveedores</li>
            </ol>
          </div>
          <div class="col-lg-12 col-md-12 col-sm-12">
            <tag:form class="filtrosConsulta form-inline"
action="buscarProveedores" method="POST" modelAttribute="filtroProveedor">
              <div class="filtro col-lg-12 col-md-12 col-sm-12">
                <label><font face="Cambria">Proveedor:</font></label>
                <tag:select path="rfcProveedor" class="form-control">
                  <tag:option value="Ninguno">Seleccionar una
Opción</tag:option>
                  <tag:option value="Todos">Todos los
Proveedores</tag:option>
```

```

        <tag:options items="{proveedores.listaProveedores}"
itemValue="rfcProveedor" itemLabel="nombreProveedor"/>
        </tag:select>
    </div>
    <div class="botonConsultar col-lg-12 col-md-12 col-sm-12">
        <tag:button class="btn">Consultar</tag:button>
    </div>
</tag:form>
</div>
<div class="tabla col-lg-12 col-md-12 col-sm-12 col-xs-12">
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>RFC</th>
                <th>Fecha</th>
                <th>Nombre</th>
                <th>Dirección</th>
                <th>Teléfono</th>
                <th>Correo</th>
                <th>Contacto</th>
            </tr>
        </thead>
        <c:forEach items="{ProveedoresRecuperados.listaProveedores}"
var="proveedor">
            <tr>
                <td style="text-align:
center;">${proveedor.rfcProveedor}</td>
                <td style="text-align:
center;">${proveedor.fechaRegistro}</td>
                <td style="text-align:
center;">${proveedor.nombreProveedor}</td>
                <td style="text-align:
center;">${proveedor.direccion}</td>
                <td style="text-align:
center;">${proveedor.telefono}</td>
                <td style="text-align:
center;">${proveedor.correo}</td>
                <td style="text-align:
center;">${proveedor.nombreContacto}</td>
            </tr>
        </c:forEach>
        <tr>
                <td style="text-align:
center;">${ProveedorRecuperado.rfcProveedor}</td>
                <td style="text-align:
center;">${ProveedorRecuperado.fechaRegistro}</td>
                <td style="text-align:
center;">${ProveedorRecuperado.nombreProveedor}</td>
                <td style="text-align:
center;">${ProveedorRecuperado.direccion}</td>
                <td style="text-align:
center;">${ProveedorRecuperado.telefono}</td>
                <td style="text-align:
center;">${ProveedorRecuperado.correo}</td>
                <td style="text-align:
center;">${ProveedorRecuperado.nombreContacto}</td>
            </tr>
        </thead>
    </table>
    <div class="botonSalir col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <a href="MenuEncargado.html"><button
class="btn">Salir</button></a>
    </div>
</div>
</div>
</body>
</html>

```

10.- consultarVentas.jsp

```
<!-- Esta Vista permite al Administrador o Encargado consultar de manera remota las
Ventas por rango de fechas
unicamente o fechas, forma de pago y/o vendedor -->

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="tag" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Consultar Ventas</title>
    <link rel="stylesheet" href="<c:url value="/recursos/css/bootstrap.min.css"/>" />
    <link rel="stylesheet" href="<c:url value="/recursos/css_propios/base.css"/>" />
    <link rel="stylesheet" href="<c:url value="/recursos/css_propios/ventas.css"/>" />
    <link rel="stylesheet" href="<c:url
value="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css"/>" />
    <script src="<c:url value="/recursos/js/jquery-3.2.1.min.js"/>"></script>
    <script src="<c:url value="/recursos/js/bootstrap.min.js"/>"></script>
    <script src="<c:url value="http://code.jquery.com/ui/1.10.1/jquery-
ui.js"/>"></script>
    <script>
      $(function()
      {
        $("#elDatePicker1").datepicker
        ({
          dateFormat: "yy-mm-dd",
          showAnim: "slideDown",
          changeMonth: true,
          changeYear: true,
          firstDay: 1,
          maxDate: 0,
          maxYear: 2018,
          dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
          monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul",
"Ago", "Sep", "Oct", "Nov", "Dic"]
        });
      });
    </script>
    <script>
      $(function()
      {
        $("#elDatePicker2").datepicker
        ({
          dateFormat: "yy-mm-dd",
          showAnim: "slideDown",
          changeMonth: true,
          changeYear: true,
          firstDay: 1,
          maxDate: 0,
          maxYear: 2018,
          dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
          monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul",
"Ago", "Sep", "Oct", "Nov", "Dic"]
        });
      });
    </script>
  </head>

  <body>
    <div class="container">
      <header class="row encabezado">
        <div class="col-lg-4 col-md-4 col-sm-4 hidden-xs col-lg-offset-3 col-
md-offset-3 col-sm-offset-3 nombreNegocio">
          <font face="Vivaldi">Mi Negocio</font>
        </div>
      </header>
    </div>
  </body>
</html>
```

```

        </div>
        <div class="col-lg-2 col-md-2 col-sm-2 logotipo">
            <figure>
                " />
            </figure>
        </div>
    </header>
    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target=".migajas" aria-expanded="false" aria-controls="navbar">
        <span class="glyphicon glyphicon-menu-hamburger"></span>
    </button>
    <div class="row cuerpo">
        <div class="row migajas">
            <div class="navbar-collapse collapse col-lg-12 col-md-12 hidden-sm
hidden-xs">
                <ol class="breadcrumb">
                    <li><a href="Principal.html">Inicio</a></li>
                    <li><a href="MenuEncargado.html">Menú Encargado</a></li>
                    <li class="active">Ventas</li>
                </ol>
            </div>
        </div>

        <div class="col-lg-12 col-md-12 col-sm-12">
            <tag:form class="filtrosConsulta" action="filtros" method="POST"
modelAttribute="filtrosBusquedaVentas">
                <div class="form-inline col-lg-5 col-md-5 col-sm-5">
                    <fieldset class="rangoConsulta">
                        <legend><font face="Cambria">Rango de
Consulta:</font></legend>
                            <label><font face="Cambria">Desde:</font></label>
                            <tag:input path="desde" class="form-control" size="10"
id="elDatePicker1" placeholder="Click Aquí" required="required"/>
                            <br/><br/>
                            <label><font face="Cambria">Hasta:</font></label>
                            <tag:input path="hasta" class="form-control" size="10"
id="elDatePicker2" placeholder="Click Aquí" required="required"/>
                            <br/><br/>
                        </fieldset>
                    </div>
                    <div class="form-inline col-lg-offset-1 col-md-offset-1 col-sm-
offset-1 col-lg-6 col-md-6 col-sm-6">
                        <fieldset class="consultaAvanzada">
                            <legend><font face="Cambria">Consulta
Avanzada:</font></legend>
                                <label><font face="Cambria">Forma de Pago:</font></label>
                                <tag:select path="tipoPago" class="form-control">
                                    <tag:option value="Ninguno">Sleccionar una
Opción</tag:option>
                                    <tag:option value="Efectivo">Efectivo</tag:option>
                                    <tag:option value="Puntos">Puntos</tag:option>
                                </tag:select>
                                <br/><br/>
                                <label><font face="Cambria">Vendedor:</font></label>
                                <tag:select path="empleadoRFC" class="form-control">
                                    <tag:option value="Ninguno">Seleccionar una
Opción</tag:option>
                                    <tag:option value="CAMA831019NY1">Adriana Camal
Mendoza</tag:option>
                                    <tag:option value="DIFC840522F37">Carolina Díaz
Fernández</tag:option>
                                    <tag:option value="FEAJ540327R18">Juana Luz Fernández
Arreola</tag:option>
                                    <tag:option value="AARM770405GF4">Mayela del Carmen
Alba Renteria</tag:option>
                                </tag:select>
                                <br/><br/>
                            </fieldset>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="botonConsultar col-lg-12 col-md-12 col-sm-12">
            <tag:button class="btn">Consultar</tag:button>
        </div>
    </tag:form>
</div>
<div class="tabla col-lg-12 col-md-12 col-sm-12 col-xs-12">
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Venta</th>
                <th>Fecha</th>
                <th>Vendedor</th>
                <th>Tipo de Pago</th>
                <th>Sub-Total</th>
                <th>I.V.A.</th>
                <th>Total</th>
            </tr>
        </thead>
        <c:forEach items="\${VentasRecuperadas.listaVentas}"
var="venta">
            <tr>
                <td style="text-align:
center;">\${venta.numeroTransaccion}</td>
                <td style="text-align:
center;">\${venta.fechaVenta}</td>
                <td style="text-align:
center;">\${venta.empleadoRFC}</td>
                <td style="text-align:
center;">\${venta.tipoPago}</td>
                <td style="text-align:
center;">\${venta.subTotal}</td>
                <td style="text-align: center;">\${venta.iva}</td>
                <td style="text-align: center;">\${venta.total}</td>
            </tr>
        </c:forEach>
    </thead>
</table>
    <div class="botonSalir col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <a href="MenuEncargado.html"><button
class="btn">Salir</button></a>
    </div>
</div>
</div>
</body>
</html>

```