

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básica e Ingeniería

Licenciatura en Ingeniería en Computación

Sistema de gestión de documentos de operaciones e inventario de la empresa

Experiencia profesional

Primera versión

Trimestre 2017 invierno

José Alberto Santos Flores

206302826

chevis42@gmail.com

Oxigen Air Pureza Para la Vida S.A. de C.V.

José de Jesús Santos Solís, Dulce María Hernandez Hernandez, José Alberto Santos Flores representantes legales de la empresa

purezaparalavida@hotmail.com

22 de Marzo de 2017




Declaratoria

Yo, José de Jesús Santos Solís, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi aprobación para su publicación en la Biblioteca Digital, así como el Repositorio Institucional de la Universidad Autónoma Metropolitana.

Yo, Dulce María Hernández Hernández, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi aprobación para su publicación en la Biblioteca Digital, así como el Repositorio Institucional de la Universidad Autónoma Metropolitana.

Yo, José Alberto Santos Flores, 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 la Universidad Autónoma Metropolitana.



José de Jesús Santos Solís



Dulce María Hernández Hernández



José Alberto Santos Flores

Resumen ejecutivo:

El presente documento contiene un resumen de las actividades que el alumno ha desarrollado durante su estancia laboral en Oxigen Air Pureza Para La Vida S.A. de C.V. detallando los proyectos y actividades en las que el alumno ha tomado parte.

El proyecto principal que ha desarrollado el alumno consiste en un sistema computacional que reemplaza todos los formatos físicos que manejaba la empresa que se llenaban a mano y de los que no quedaba respaldo alguno y genera los documentos requeridos automáticamente para el buen funcionamiento de la empresa, manejando además el inventario de productos.

En los siguientes apartados se detalla cada una de las etapas de desarrollo de cada uno de los proyectos realizados y el grado de participación que me fueron asignados, contiene tablas e imágenes de los resultados finales.

Tabla de contenido.

1.- Descripción de la empresa y actividades a las que se dedica

2.- Descripción del departamento o sección en la que labora el alumno

3.- Descripción técnica de las actividades asociadas al puesto

4.- Relación de proyectos en los que ha participado el alumno

4.1 Sistema de gestión documentos e inventarios de la empresa

4.1.1 Descripción técnica del proyecto

4.1.1.1 Módulo de registro de ventas y manejo de inventario

4.1.1.2 Módulo de impresión de tickets

4.1.1.3 Módulo de registro de gastos

4.1.1.4 Módulo de generación, almacenamiento y envío de reportes

4.1.1.5 Módulo de generación de contratos

4.1.1.6 Módulo de generación de pólizas de garantía

4.1.1.7 Control de acceso al sistema

4.1.1.8 Respaldo de información

4.1.2 Responsabilidades

4.2 Capacitación del personal

4.2.1 Descripción técnica del proyecto

4.2.2 Responsabilidades

4.3 Generación de códigos de barras

4.3.1 Descripción técnica del proyecto

4.3.2 Responsabilidades

4.4 Revisión, diagnóstico y reparación de tarjetas electrónicas de concentradores de oxígeno

4.4.1 Descripción técnica del proyecto

4.4.2 Responsabilidades

5.- Anexo

1.- Descripción de la empresa y actividades a las que se dedica

Oxigen Air Pureza Para La Vida S.A. de C.V. es una empresa familiar que desde hace siete años se dedica a la venta, renta, mantenimiento, distribución de equipos y productos especializados en terapia respiratoria con la finalidad de cubrir la necesidad de las personas que padecen distintos trastornos en las vías respiratorias ofreciendo una atención personalizada y de calidad a cada uno de nuestros clientes. Para cubrir dicha necesidad la empresa cuenta con los productos suficientes en cantidad y calidad en el momento en todo momento y cuenta con los asesores capacitados para brindar la atención al cliente, explicar las características y funcionamiento de los equipos por lo que se considera la atención al cliente como el producto fundamental que brinda la empresa con su filosofía “tratar a los demás como me gustaría ser tratado”.

Dentro de la empresa se realizan diversas actividades principales destacando las siguientes: atención y servicio al cliente, llenado de cilindros de oxígeno, venta de productos de oxigenoterapia (mascarillas de oxígeno, cánulas nasales, vasos humidificadores, cilindros de oxígeno medicinal, etc.) renta y venta de equipos de oxigenoterapia (cilindros de oxígeno medicinal, concentradores de oxígeno, oxímetros de pulso, aspiradores de secreciones, aparatos para el tratamiento de la apnea del sueño, reguladores de voltaje, etc.) servicio de prueba hidrostática a cilindros de oxígeno, mantenimiento preventivo y correctivo a concentradores de oxígeno.

2.- Departamento o sección en la que labora el alumno

Actualmente el alumno es encargado de todos los sistemas de cómputo, como son, computadoras, sistema de impresión, manejo del sistema de inventario, sistema de cámaras de seguridad y la capacitación del personal para el manejo de los sistemas que el alumno desarrolló.

3.- Descripción técnica de las actividades asociadas al puesto

- Diseño, normalización e implementación de una base de datos relacional en MySQL para el manejo del sistema de inventarios y documentos
- Configuración del servidor de base de datos para el acceso a diferentes usuarios con diferentes privilegios según su grado de responsabilidad en la empresa
- Elaboración del sistema de gestión de inventarios y documentos de la empresa
- Generación de reportes diarios con la información de los productos que se vendieron o rentaron así como su envío por correo electrónico
- Actualización y respaldo de la información de los productos existentes en inventario así como cada una de las operaciones de venta y renta que se generan
- Impresión de archivos fijos y publicidad que la empresa requiere para ciertos productos o servicios

4. Relación de proyectos en los que ha participado el alumno

4.1 Sistema de gestión de documentos e inventario de la empresa

4.1.1 Descripción técnica del proyecto

El proyecto que a continuación se menciona tiene como finalidad el reemplazo de todos los formatos físicos que anteriormente se llenaban a mano y de los que no quedaba respaldo alguno por lo que surgió la necesidad de tener un sistema que maneje el inventario de la empresa y los documentos que se necesitan de manera interna para el control de las operaciones de venta y renta que realiza la empresa tanto para los clientes como para los trabajadores de la misma.

El sistema fue creado con el lenguaje de programación Java para crear los objetos que fueron requeridos así como generar la interfaz gráfica con la que los usuarios interactúan con el sistema. La interfaz gráfica se creó utilizando la tecnología Swing que permite crear una interfaz que resulta muy familiar para los usuarios que están acostumbrados a usar interfaces del sistema operativo Windows.

Se utilizó el manejador de base de datos MySQL para guardar los datos de todos los usuarios, productos y operaciones que realiza la empresa además de que permite hacer un respaldo constante de dichos datos en caso de descompostura de la computadora principal o su disco duro.

Para la conexión de la base de datos con la interfaz y los objetos creados se utilizó el conector JDBC que permite la interacción de ambas tecnologías.

En la figura 1 se muestra un diagrama de la forma en la que interactúan las tres tecnologías antes mencionadas.



Figura1. Interacción del lenguaje de programación con la base de datos por medio del conector JDBC

Al utilizar el conector JDBC se permite al sistema trabajar en dos computadoras simultáneamente utilizando la arquitectura cliente-servidor y una conexión WiFi entre ellas. En la figura 2 se muestra la arquitectura cliente servidor que el alumno configuró para el funcionamiento de ambas computadoras con el sistema.



Figura 2. Configuración de computadoras según el modelo cliente servidor.

El sistema fue creado a partir del modelo vista-controlador. En la parte del controlador se crearon clases y métodos que brindaron acceso a los valores de la base de datos y en la parte de vista se utilizó la tecnología Swing para crear las interfaces gráficas con las que los trabajadores interactúan.

Ambas computadoras en las que el sistema funciona tienen las mismas características: procesador AMD A6 a 3.9 GHz, 8GB en memoria RAM, disco duro de 500GB y sistema operativo Windows 7 de 64 bits tal y como se muestra en la figura 3.

Procesador:	AMD A6-6400K APU with Radeon(tm) HD Graphics	3.90 GHz
Memoria instalada (RAM):	8.00 GB (7.19 GB utilizable)	
Tipo de sistema:	Sistema operativo de 64 bits	
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla	
Configuración de nombre, dominio y grupo de trabajo del equipo		
Nombre de equipo:	f2a55m-s1-PC	
Nombre completo de equipo:	f2a55m-s1-PC	
Descripción del equipo:		
Grupo de trabajo:	WORKGROUP	

Figura 3. Características de las computadoras utilizadas para el sistema.

4.1.1.1 Módulo de registro de ventas y manejo de inventario

En este módulo se controlan las ventas que la empresa realiza mediante el ingreso de los productos o servicios que hacen durante una jornada de trabajo. El ingreso de los productos fue diseñado para que el usuario del sistema tenga a su disposición varias opciones y utilice la que más le facilite el trabajo, estas opciones son:

1.- Utilizar el lector de código de barras: el alumno escogió y configuró un lector de códigos de barras con conexión USB para que los trabajadores puedan ingresar los productos de manera rápida, directa y sin confusión los productos que vayan requiriendo durante su jornada laboral.

2.- Ingresar directamente el identificador del producto: en caso de que el trabajador haya memorizado el identificador del producto lo puede ingresar mediante el teclado

numérico de la computadora y al final presiona la tecla ENTER para que el sistema determine de qué producto se trata.

3.- Búsqueda por palabra clave: si se utiliza una palabra clave en la búsqueda el sistema no solamente mostrará todos los productos relacionados con dicha palabra sino que mostrará para cada producto una imagen con la finalidad de que la identificación del producto sea aún más rápida, basta con que el trabajador haga doble clic sobre el producto que desea para que el sistema lo registre.

Cabe destacar que el sistema resta automáticamente los productos del inventario y muestra en tiempo real las cantidades de productos que están disponibles para se tomen las medidas en caso de que sea necesario realizar un pedido de algún producto.

Este módulo tiene la capacidad de modificar diferentes valores que son relevantes para la generación de reportes como son: forma de pago, cantidad de productos, manejo de productos de precio variable. Para la empresa es muy importante saber el monto de las operaciones que se realizaron por medio de tarjetas de bancarias ya que la cantidad ingresada debe coincidir con la cantidad que el banco deposita a la cuenta bancaria de la empresa al día siguiente.

El sistema ingresa por defecto las operaciones en modo de pago “efectivo” ya que son mayoría pero el usuario tiene siempre la opción de cambiar cualquier operación a pago con tarjeta bancaria.

Cabe destacar que el sistema no almacena ningún dato sobre las tarjetas bancarias de los clientes únicamente se cambia el valor de “efectivo” a “pago con tarjeta bancaria” para el control interno de la empresa.

El sistema tiene la capacidad de dividir un pago si el cliente requirió hacer un pago parcial en efectivo y un pago parcial con tarjeta bancaria, ya que, en caso de que el producto solicitado tenga un precio elevado algunos clientes optan por esta opción.

Los productos de precio variable se crearon para manejar una lista de productos más corta y evitar búsquedas confusas, un ejemplo son los pagos de renta que son de precio variable ya que el monto depende de los equipos que el cliente haya contratado. Una vez que se ingresado un producto de precio variable el sistema lo reconoce de manera automática y genera un cuadro de diálogo en el que el usuario ingresa el monto pagado.

Existen algunos equipos que una vez que se rentan o venden llevan algunos productos desechables consigo por lo que el alumno se dio a la tarea de registrar estos “grupos” de productos para que una vez que se ingrese al sistema una venta o renta de equipo el sistema descuente automáticamente los desechables que se

entregarán al cliente y de esta manera se lleve un control más preciso sobre el inventario.

En la figura 4 se muestra el resultado del módulo de registro de ventas y manejo de inventario donde se ve un botón de lupa para la búsqueda de productos por palabra clave y el espacio para la utilización del lector de códigos de barras, además de mostrar la forma en que se registran los productos y la actualización del inventario en la parte inferior indicando las existencias actuales del producto.

OXIGEN AIR PUREZA PARA LA VIDA
CONTRATO Edit

INGRESE EL CÓDIGO DE BARRAS: E

ID	PRODUCTO	MARCA	PRECIO U.	CANTIDAD	METODO DE PAGO	TOTAL	FACTURADO?
26484	CANULA NASAL ADULTO SILICON 2M	WESTMED	45.0	3	EF	135.0	NO
26485	CANULA EXTENSION ADULTO 4M	WESTMED	65.0	1	EF	65.0	NO
26486	CANULA EXTENSION ADULTO 7M	WESTMED	85.0	1	EF	85.0	NO
26487	VASO HUMIDIFICADOR	SALTER LABS	75.0	1	EF	75.0	NO
26488	MASCARILLA DE OXIGENO ADULTO	WESTMED	45.0	1	EF	45.0	NO
26489	INSPIROMETRO	RESPIFIX	175.0	1	EF	175.0	NO
26490	RECARGA CILINDRO 255 LTS	OXIGEN	45.0	1	EF	45.0	NO
26491	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	17	EF	935.0	NO
26492	LITRO DE AGUA BIDESTILADA	OXIGEN	20.0	1	EF	20.0	NO
26493	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	EF	55.0	NO
26494	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	EF	55.0	NO
26495	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	EF	55.0	NO
26496	PAGO DE RENTA MENSUAL	OXIGEN	1.0	1	EF	1200.0	NO
26498	DEPOSITO DE SEGURIDAD	OXIGEN	1.0	1	PT	3000.0	NO
26499	RENDA MENSUAL KIT 680 LTS	OXIGEN	650.0	1	PT	650.0	NO
26500	RECARGA CILINDRO 3500 LTS	OXIGEN	195.0	1	PT	195.0	NO
26501	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	PT	55.0	NO
26502	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	EF	55.0	NO
26503	PAQUETE DE AGUA BIDESTILADA	OXIGEN	80.0	1	EF	80.0	NO
26504	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	EF	55.0	NO
26505	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	1	EF	55.0	NO
26506	PAQUETE DE AGUA BIDESTILADA	OXIGEN	80.0	3	PT	240.0	NO
26507	RENDA MENSUAL CILINDRO 9500 LTS	OXIGEN	550.0	1	EF	550.0	NO
26508	RECARGA CILINDRO 9500 LTS CAMBIO	OXIGEN	550.0	1	EF	550.0	NO
26510	PAGO DE RENTA MENSUAL	OXIGEN	1.0	1	EF	1570.0	NO
26511	CAMBIO DE CILINDRO 680 LTS VENCIDO P...	CATALINA	1750.0	1	EF	1750.0	NO
26512	RECARGA CILINDRO 680 LTS	OXIGEN	55.0	4	EF	220.0	NO
26513	VASO HUMIDIFICADOR	SALTER LABS	75.0	4	EF	300.0	NO
26514	RECARGA CILINDRO 3500 LTS	OXIGEN	195.0	1	EF	195.0	NO
26515	RECARGA CILINDRO 1750 LTS	OXIGEN	85.0	1	EF	85.0	NO
26517	CONECTOR COLA DE RATON	OXIGEN	35.0	1	EF	35.0	NO

EXISTENCIAS ACTUALES DEL PRODUCTO ->CONECTOR COLA DE RATON OXIGEN: 39

Eviar Mail Generar Corte AUN NO SE HA ENVIADO EL CORTE mostrar VENTA TOTAL: ###

Figura 4. Implementación del módulo de registro de ventas y manejo de inventario.

En la figura 5 se muestra el diagrama del módulo de registro de operaciones y manejo de inventario

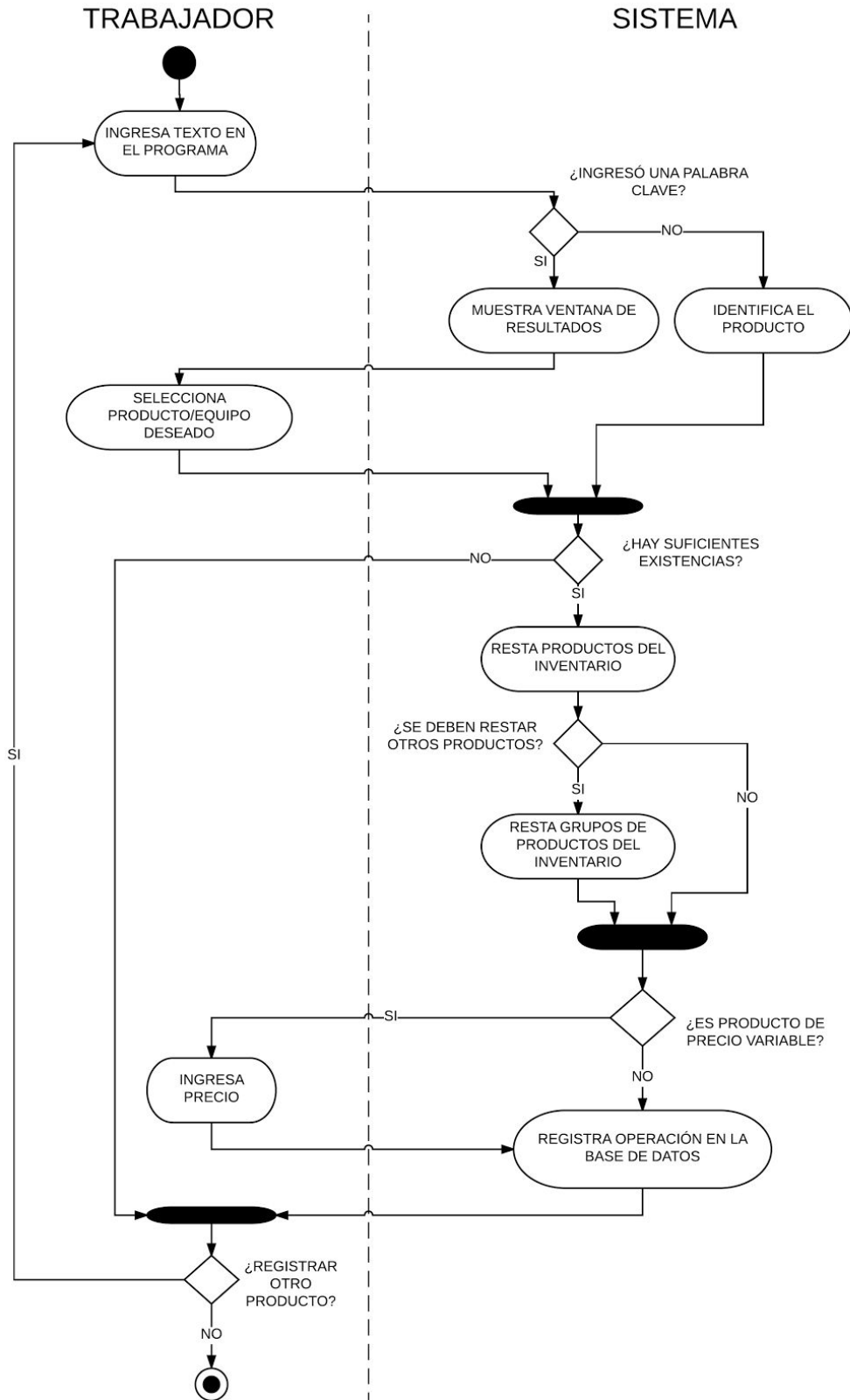


Figura 5. Diagrama del módulo de registro de operaciones

4.1.1.2 Módulo de impresión de tickets

Una vez que se ha completado el registro de los productos que se han vendido la empresa tiene la obligación de entregar a los clientes un comprobante de la compra que acaban de realizar por lo que el sistema es capaz de imprimir tickets con la información de la operación que el cliente ha realizado indicando datos importantes como el nombre de la empresa, teléfonos de contacto, los productos que se han vendido, fecha y hora de la operación, cantidad que el cliente pagó, cambio que recibió, entre otros como se muestra en la figura 6.



Figura 6. Impresión de un ticket.

Anteriormente los trabajadores de la empresa entregaban remisiones a mano que resultaban sumamente lentas de llenar y estaban sujetas a tener que repetir todo el proceso si se generaba algún error al llenar dicha remisión.

Cabe destacar que el usuario del sistema no tiene la necesidad de revisar las opciones de la impresora de tickets en ningún momento ya que fue configurada previamente para que la impresión de tickets fuera automática.

4.1.1.3 Módulo de generación, almacenamiento y envío de reportes

Una vez que la jornada de trabajo se ha completado el alumno se da a la tarea de generar, verificar y enviar el reporte que se genera con todas las operaciones del día incluyendo ingresos y gastos así como mostrando la cantidad vendida de cada producto. En el reporte se detalla la cantidad de dinero que queda al final del día tanto en efectivo como por pagos con tarjetas bancarias.

Una vez que se ha verificado que el reporte esté completo se envía por correo electrónico desde el sistema seleccionando la casilla de enviar.

Este reporte se guarda en el disco duro de la máquina y se respalda constantemente para evitar la pérdida de información. En la figura 7 se muestra una miniatura del reporte que se genera al día, en la primer hoja se resaltan las recargas de cilindro de oxígeno en color verde dada su importancia; en la segunda hoja se muestra una gráfica de las cantidades de recargas que se hicieron y en la tercer hoja se detallan los gastos generados durante el día y los totales de dinero en efectivo y en tarjeta de crédito.

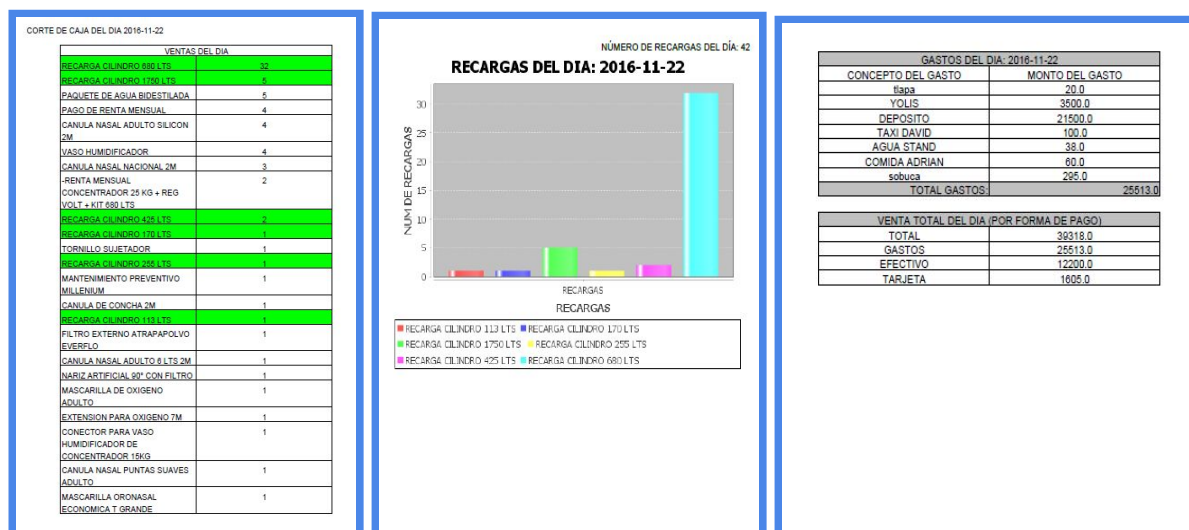


Figura 7. Resultado de la generación de reportes

En la figura 8 se muestra el diagrama del funcionamiento del módulo de generación y envío de reportes

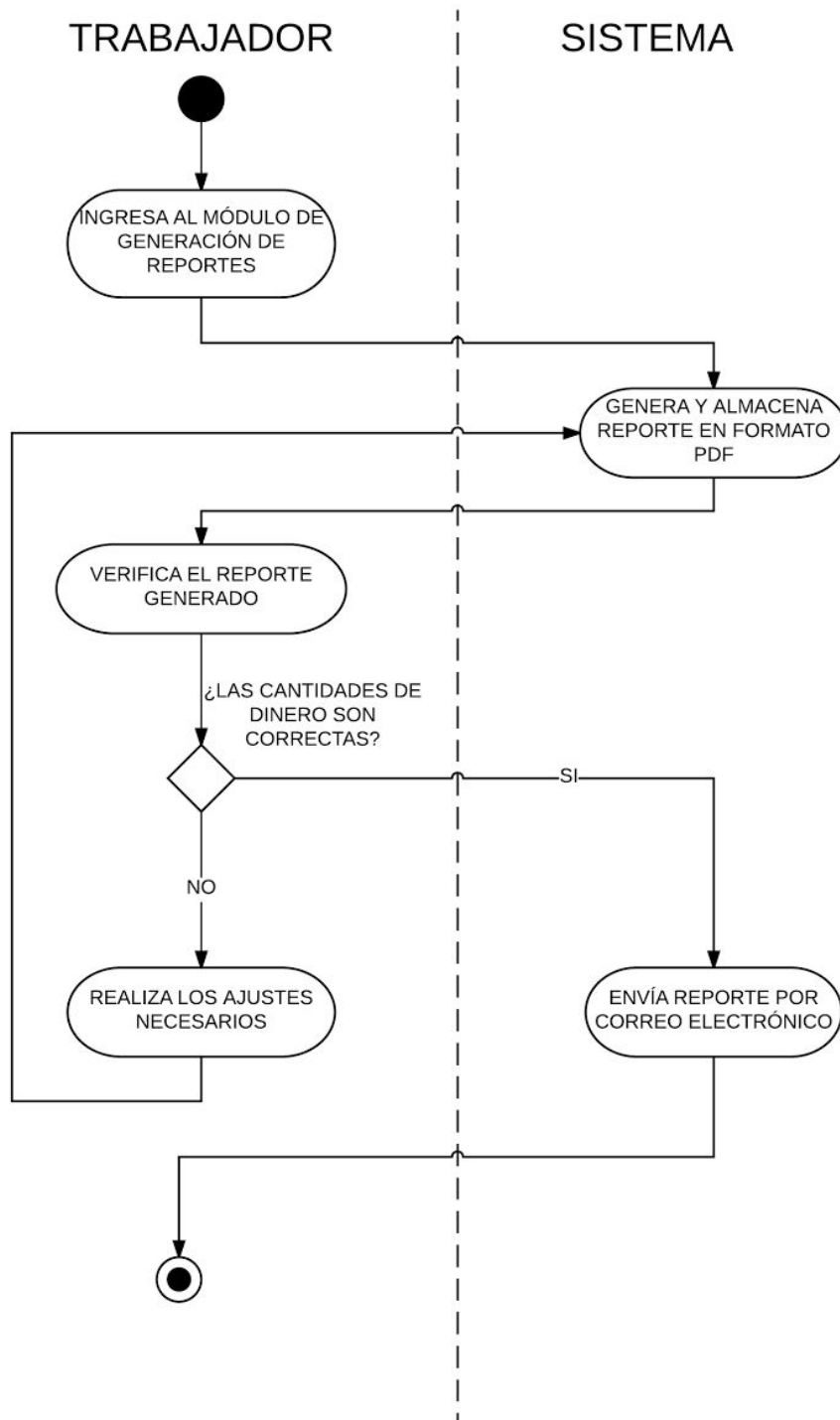


Figura 8. Diagrama del módulo de generación y envío de reportes

4.1.1.4 Módulo de registro de gastos

Durante la jornada de trabajo se generan diversos gastos para la empresa por lo que es necesario llevar un control preciso de las salidas de dinero generadas para que el reporte sea preciso además de servir como respaldo de las fechas en las que la empresa pagó por productos o servicios tal como se muestra en la figura 9.



The screenshot shows a software window titled "REGISTRO DE GASTOS". At the top, there are two input fields: "Concepto del gasto:" and "Monto: \$". Below these is a section titled "GASTOS DEL DIA DE HOY" with a scrollable list of expenses. At the bottom right of the window is a button labeled "Aceptar". At the bottom center, it displays "Total de gastos del dia: \$19191.0".

Concepto del gasto	Monto
1.-DEPOSITO -->	\$17000.0
2.-pila 9v -->	\$50.0
3.-AGUA -->	\$37.0
4.-COMIDA DAVID -->	\$34.0
5.-TAXI DAVID -->	\$100.0
6.-via publica -->	\$100.0
7.-comida adrian -->	\$60.0
8.-daniel -->	\$310.0

Figura 9. Implementación del módulo de registro de gastos

4.1.1.5 Módulo de generación de contratos

Un contrato de renta es muy importante para la empresa ya que representa el compromiso que la empresa tiene con el cliente y viceversa.

El contrato de renta debe ser generado sin errores para evitar conflictos con los clientes, además de que debe quedar un respaldo del mismo y no sólo un archivo físico sino también un respaldo digital en caso de extravío.

El sistema genera una plantilla que debe ser llenada con los datos del cliente y no permite avanzar en caso de falte algún dato o que esté incompleto como podría ser un número telefónico que por su importancia no puede quedar incompleto. Otros datos son proporcionados por el sistema como son la fecha y la vigencia del contrato. Una vez que los datos han sido completados se genera un archivo en formato PDF que se almacena y se imprime para que se firme por ambas partes.

El contrato debe ser impreso para que sea firmado tanto por los clientes como por los trabajadores de la empresa por lo que se mencionó la impresora como parte del hardware requerido para el funcionamiento adecuado del sistema. En la figura 10 se muestra el archivo PDF que se genera una vez que se han ingresado los datos.

4.1.1.5 Módulo de generación de garantías

A diferencia de la generación de contratos la generación de garantías no depende tanto de los datos que el cliente ha proporcionado sino de ciertos datos de equipos que son relevantes por ejemplo el tiempo de garantía con el cuenta un equipo así como el tipo de cláusulas que cubra dicha garantía. Algunos productos requieren de un cuidado especial como son los concentradores de oxígeno por lo que en caso de que el sistema detecte que se ha ingresado dicho equipo automáticamente generará una póliza de garantía con cláusulas adicionales que la mayoría de equipos no tiene debido a los cuidados que requiere el equipo. En la figura 12 se muestran una póliza de garantía en la que algunos de los datos son generados de forma automática como son fecha, tiempo de garantía y las cláusulas requeridas para el equipo en cuestión.



POLIZA DE GARANTIA DEL POLVO OXIGEN

OXYGEN AIR PURIFIER S.A. DE CV
 TRAY PEDRO DE SANTANA COL. DEL CARIÑO DOMINGUEZ SOCIEDAD CIVIL
 TULAPAL MEXICO D.F.
 TELEFONOS: 198-1903, 4808-4841, 8234-8747

FECHA: 2018-11-28

NOMBRE: José Roberto Santos
 CALLE: TRAY PEDRO DE SANTANA
 COL: DEL CARIÑO DOMINGUEZ SOCIEDAD CIVIL
 TEL: 198-1903-4841
 TEL: 198-1903-4841

NOMBRE DEL EQUIPO	MARCA	PRECIO	TIEMPO DE GARANTIA
JORNALCEROS FLOWAS	ORVIS	80000	1 AÑO

NÚMERO DE SERIE: 20181128

CLÁUSULAS

- 1.- NO AMBAYAN VALUOS POR PARTE DEL CLIENTE, SÓLO EN FALTA DE FABRICACIÓN
- 2.- CONSTA DE REPARACIÓN Y/O CAMBIO FÍSICO DIAGNOSTICADO POR EL PROVEEDOR DEL EQUIPO
- 3.- ES VÁLIDA EN NUESTRAS INSTALACIONES. NO APLICA SERVICIO A DOMICILIO SIN COSTO
- 4.- SE PAGA AL PUNTO DE ENTREGA DEL EQUIPO Y/O REPARACIÓN DEL EQUIPO SIN COSTO DE CAMBIO INMEDIATO
- 5.- LOS TIEMPOS DE RESPUESTA EN LA GARANTÍA DEPENDE DE LOS PROVEEDORES Y/O PROCEDIMIENTOS DEL EQUIPO
- 6.- UNA VEZ SALIDA LA MERCANCIA NO HAY DEVOLUCIÓN DEL DINERO

NOTA:

OXYGEN AIR PURIFIER PARA LA VIDA AL MEJOR	LEY ESTOY DE ACUERDO
--	----------------------

Figura 12. Póliza de garantía terminada con los datos del cliente.

En la figura 13 se muestra el diagrama que se realizó para el módulo de generación de garantías.

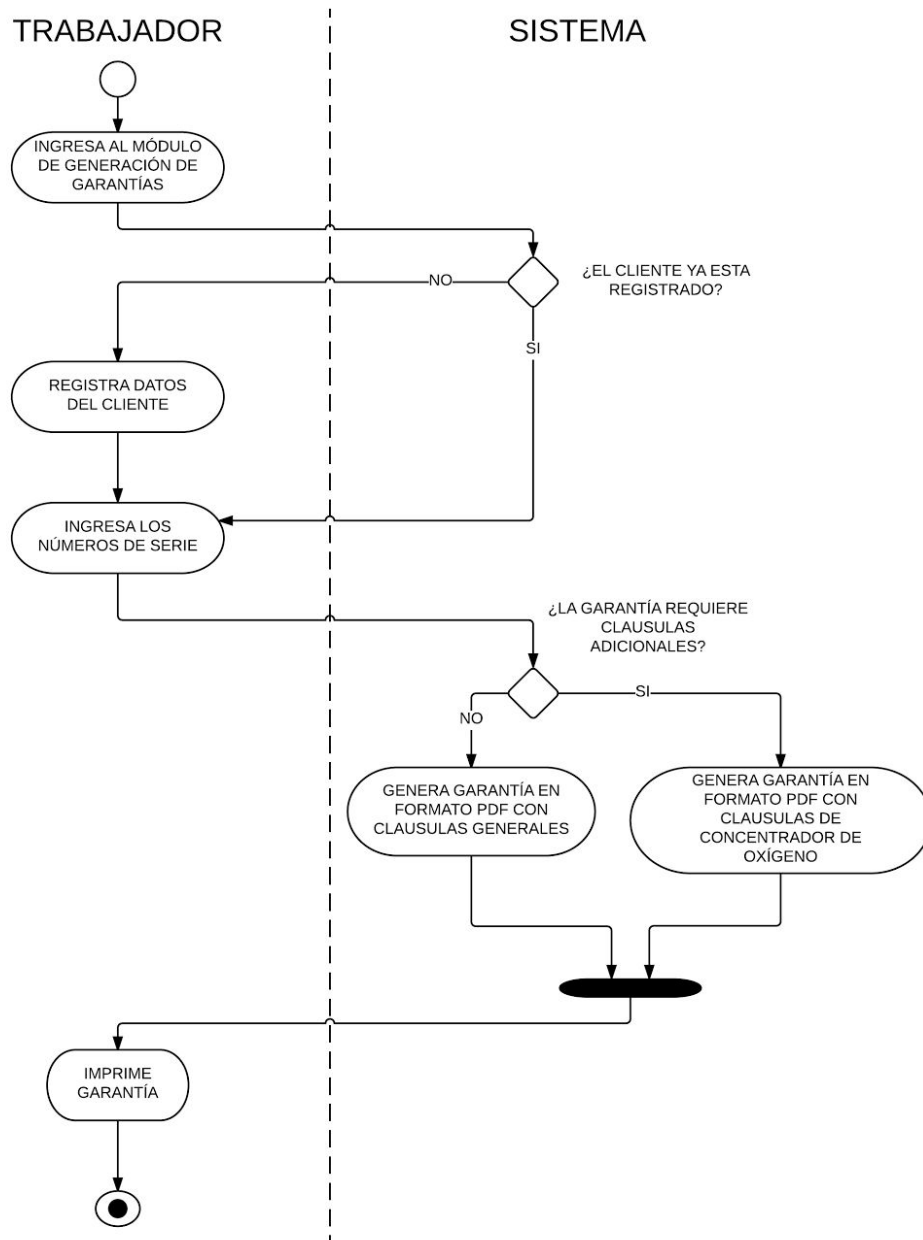


Figura 13. Diagrama del módulo de generación de garantías

Anteriormente los trabajadores de la empresa llenaban formatos a mano y constantemente surgía la duda de que tiempo de garantía tenía algún producto y se presentaron casos de errores al llenar los formatos mencionados.

4.1.1.7 Control de acceso al sistema

Existen en la empresa tres contraseñas: la primera se da a todos los trabajadores de la empresa y da acceso al sistema operativo de la computadora, la segunda es

para acceder al sistema de gestión de documentos e inventario de la empresa a la cual sólo tienen acceso los trabajadores encargados en turno y finalmente la tercer contraseña que sólo la tiene el alumno y es para realizar las operaciones de las básicas de la base de datos (altas, bajas y cambios).

Al tener un acceso restringido a la base de datos se garantiza la integridad de la información y se evita que haya modificaciones sin autorización.

Cabe destacar que la contraseña de acceso al sistema no está guardada en ninguna de las computadoras de trabajo para evitar que personas ajenas tengan acceso al sistema de gestión de documentos e inventario de la empresa.

4.1.1.8 Respaldo de la información

Debido a la gran cantidad de productos que maneja la empresa es de vital importancia que los datos no sufran de ninguna pérdida por lo que el alumno es encargado de hacer un respaldo periódicamente para asegurarse de que no haya pérdida de información. El alumno cuenta con una instrucción (query) que hace un respaldo de la base de datos completa tanto en su estructura como los datos almacenados que se guarda en un archivo de extensión .sql.

4.1.2 Responsabilidades

En la tabla 1 se muestran los segmentos del análisis realizado de acuerdo a la solicitud de la empresa planteando las fases para la realización del módulo de registro de ventas y manejo de inventario.

Tabla 1. Segmentos del análisis del módulo de registro de ventas y manejo de inventario

Análisis del módulo		Porcentaje de participación
Requerimientos de la empresa	Que el sistema maneje todas las operaciones de venta y renta además de ser capaz de manejar el inventario Que sea capaz de cambiar los pago a tarjeta bancaria Que sea capaz de dividir un pago una parte en efectivo y otra en tarjeta Que muestre el inventario del producto seleccionado en tiempo real Que reste productos de grupos	100%
Requerimientos del usuario	Que proporcione diversos métodos de búsqueda de equipos y productos para una rápida localización	100%

Requerimientos de hardware	Que se disponga de un lector de códigos de barras previamente configurado en el formato CODE-128	80%
Requerimientos de software	Librería SWING para la implementación de la interfaz de usuario	100%

En la tabla 2 se muestran los requerimientos para la elaboración del módulo de impresión de tickets.

Tabla 2. Análisis del módulo de impresión de tickets

Análisis del módulo		Porcentaje de participación
Requerimientos de la empresa	Que cada ticket contenga la información de la empresa así como los productos que el cliente ha solicitado, la fecha de la operación y el trabajador que lo realiza.	100%
Requerimientos del usuario	Que sea accesible, rápido y fácil de utilizar	100%
Requerimientos de hardware	Que se disponga de una impresora de térmica de alta velocidad	50%

En la tabla 3 se muestran los requerimientos para la elaboración del módulo de registro de gastos

Tabla 3. Análisis del módulo de registro de gastos

Análisis del módulo		Porcentaje de participación
Requerimientos de la empresa	Que guarde la información de cada registro de gastos para su posterior consulta especialmente la fecha del gasto	100%

En la tabla 4 se muestran los segmentos del análisis realizado para la elaboración del módulo de generación y almacenamiento de reportes

Tabla 4. Segmentos del análisis del módulo generación, almacenamiento y envío de reportes

Análisis del módulo		Porcentaje de participación
Requerimientos de la empresa	<p>Que genere un documento en formato PDF mostrando una lista detallada de los productos que se vendieron durante el día.</p> <p>Que genere una gráfica con información de la cantidad de cilindros de oxígeno que fueron recargados.</p> <p>Que muestre los ingresos del día de acuerdo al método de pago que se utilizó.</p> <p>Que se pueda enviar por correo electrónico sin la necesidad de adjuntar el archivo de forma manual.</p>	100%
Requerimientos del usuario	Que se pueda verificar y corregir previo al envío.	100%
Requerimientos de software	<p>Librería iText para la generación del documento en formato PDF.</p> <p>Librería JFreeChart para la creación de la gráfica de recargas de cilindros</p> <p>Librería JavaMail para el envío del reporte vía correo electrónico.</p>	100%
Requerimientos de hardware	Conexión a internet	100%

En la tabla 5 se muestran los segmentos del análisis del módulo de generación de contratos

Tabla 5. Segmentos del análisis del módulo de generación de contratos

Análisis del módulo		Porcentaje de participación
Requerimientos de la empresa	<p>Que sea capaz de detectar errores de escritura en caso de haber menos dígitos de los requeridos.</p> <p>Que proponga datos relevantes como fecha y vigencia.</p> <p>Que permita hacer cambios en caso de requerirse.</p> <p>Que tenga la capacidad de generar una</p>	100%

	hoja de cancelación una vez que el equipo ha sido devuelto.	
Requerimientos del usuario	Que se puedan ingresar los datos por etapas separando los datos del cliente y los de los equipos	100%
Requerimientos de software	Librería iText para la generación del archivo en formato PDF	100%
Requerimientos de hardware	Impresora	50%

En la tabla 6 se muestran los segmentos del análisis del módulo de generación de documentos

Tabla 6. Segmentos del análisis de generación de pólizas de garantía

Análisis del módulo		Porcentaje de participación
Requerimientos de la empresa	Que se a capaz de generar un documento con diferentes cláusulas dependiendo del equipo solicitado. Que proporcione automáticamente datos relevantes como son fecha de venta y tiempo de garantía.	100%
Requerimientos del usuario	Que tenga espacio para sellar la hoja	100%
Requerimientos de software	Librería iText para la generación del archivo en formato PDF	100%

4.2 Capacitación del personal

4.2.1 Descripción técnica del proyecto

Una vez que se completó la realización del sistema de gestión de documentos e inventario de la empresa el alumno capacitó al personal para la correcta utilización del mismo resolviendo todas las dudas que surgieron además se generó una guía en caso de que se añadiera un nuevo trabajador que requiriera aprender a utilizar el sistema.

4.2.2 Responsabilidades

En la tabla 7 se muestran los requerimientos para la capacitación del personal

Tabla 7. Análisis de la capacitación del personal

Análisis de la capacitación del personal		Porcentaje de participación
Requerimientos de la empresa	Que se explique a todos los trabajadores el funcionamiento del sistema Que se elabore una guía de funcionamiento	100%
Requerimientos del usuario	Que se resuelvan las dudas las veces que sea necesario antes, durante y después de la capacitación	100%

4.3 Generación de códigos de barras

4.3.1 Descripción técnica del proyecto

Algunos productos que maneja la empresa no cuentan con un código de barras generado por el fabricante del producto por lo que el alumno se dio a la tarea de generar códigos de barras para todos esos productos.

Se generó además una tabla con los productos más requeridos por los clientes que se mandó enmicar para que los trabajadores tengan fácil acceso a dichos productos incluyendo productos que no son físicos como son los llenados de cilindro de oxígeno a los que se les asignó un espacio en la tabla para acceso inmediato. La tabla antes mencionada puede ser actualizada y modificada en caso de que se requiera anexar nuevos productos o quitar algunos que ya no se requieran.

En la figura 14 se muestra la guía que el alumno elaboró para los productos que carecen de código de barras.

TABLA DE PRODUCTOS CON CÓDIGO DE BARRAS





 000001 CANULA NASAL ADULTO 6 LTS 2M Marca: SALTER LABS Precio \$40.0	 000003 CANULA NASAL ADULTO SILICON 2M Marca: WESTMED Precio \$45.0	 000010 CANULA NASAL SILICON ADULTO 7M Marca: WESTMED Precio \$105.0
 000014 CANULA EXTENSION ADULTO 7M Marca: WESTMED Precio \$85.0	 000016 VASO HUMIDIFICADOR Marca: SALTER LABS Precio \$75.0	 000025 CONECTOR COLA DE RATON Marca: OXIGEN Precio \$35.0
 000026 CONECTOR PARA EXTENSION Marca: OXIGEN Precio \$25.0	 000033 LITRO DE AGUA BIDESTILADA Marca: OXIGEN Precio \$20.0	 000034 PAQUETE DE AGUA BIDESTILADA Marca: OXIGEN Precio \$80.0

Figura 14. Guía de códigos de barras.

4.3.2 Responsabilidades

En la tabla 8 se muestra la segmentación del análisis de la generación de códigos de barras

Análisis de la generación de códigos de barras		Porcentaje de participación
Requerimientos de la empresa	Que todos los productos tengan un código de barras asignado	100%
Requerimientos del usuario	Que proporcione acceso inmediato a ciertos productos	100%
Requerimientos de software	Librería	100%

4.4 Revisión, diagnóstico y reparación de tarjetas electrónicas de concentradores de oxígeno

4.4.1 Descripción del proyecto

Las tarjetas electrónicas de los concentradores de oxígeno son muy importantes ya que controlan todo el accionar del equipo y la manera en que éste reacciona los diferentes cambios que se presentan durante la vida útil del equipo. La principal falla que presentan se debe a las descargas eléctricas que el aparato recibe por lo que el alumno y un trabajador de la empresa se dieron a la tarea de encontrar las piezas que estaban dañadas para su posterior reparación o cambio. Algunas de estas

piezas con switches de encendido, varistores, transformadores de corriente, fusibles, entre otros.

Se utilizó un multímetro y se midieron diferentes valores de voltaje de corriente directa para determinar en qué puntos había cortes de corriente y de esta manera saber que componentes estaban dañados y así reemplazarlos. En caso de que la pieza en cuestión no fuera reparable se sustituyó por un nueva. En la figura 15 se muestra un ejemplo de tarjeta electrónica de concentrador como las que el alumno ha ayudado a reparar.



Figura 15. Tarjeta electrónica de concentrador de oxígeno.

4.4.2 Responsabilidades

En la tabla 9 se muestran la segmentación del análisis del proyecto de revisión diagnóstico y reparación de tarjetas electrónicas de concentrador de oxígeno.

Análisis del proyecto de revisión, diagnóstico y reparación de tarjetas electrónicas		Porcentaje de participación
Requerimientos de la empresa	Que la tarjeta tenga la capacidad de operar todo el equipo nuevamente	50%

ANEXO

*CÓDIGO FUENTE DE LA CLASE PRODUCTO

```
public class Producto {
    private int idProducto;
    private String nombreProducto;
    private String marcaProducto;
    private double precioProducto;
    private String descripcionProducto;
    private int existenciasProdcuto;
    private String tieneGruposProducto;
    private String precioVariableProducto;
    private String tiempoGarantiaEquipo;
    private boolean sobre_pedido;

    public boolean isSobre_pedido() {
        return sobre_pedido;
    }

    public void setSobre_pedido(boolean sobre_pedido) {
        this.sobre_pedido = sobre_pedido;
    }

    public Producto(){

    }

    public Producto(String nombreProducto, String marcaProducto, double
precioProducto, String descripcionProducto, int existenciasProdcuto, String
tieneGruposProducto, String precioVariableProducto, String tiempoGarantiaEquipo)
{
    this.nombreProducto = nombreProducto;
    this.marcaProducto = marcaProducto;
    this.precioProducto = precioProducto;
    this.descripcionProducto = descripcionProducto;
    this.existenciasProdcuto = existenciasProdcuto;
    this.tieneGruposProducto = tieneGruposProducto;
    this.precioVariableProducto = precioVariableProducto;
    this.tiempoGarantiaEquipo = tiempoGarantiaEquipo;
}
```

```

public int getIdProducto() {
    return idProducto;
}

public void setIdProducto(int idProducto) {
    this.idProducto = idProducto;
}

public String getNombreProducto() {
    return nombreProducto;
}

public void setNombreProducto(String nombreProducto) {
    this.nombreProducto = nombreProducto;
}

public String getMarcaProducto() {
    return marcaProducto;
}

public void setMarcaProducto(String marcaProducto) {
    this.marcaProducto = marcaProducto;
}

public double getPrecioProducto() {
    return precioProducto;
}

public void setPrecioProducto(double precioProducto) {
    this.precioProducto = precioProducto;
}

public String getDescripcionProducto() {
    return descripcionProducto;
}

public void setDescripcionProducto(String descripcionProducto) {
    this.descripcionProducto = descripcionProducto;
}

public int getExistenciasProducto() {
    return existenciasProducto;
}

```

```

    }

    public void setExistenciasProducto(int existenciasProducto) {
        this.existenciasProducto = existenciasProducto;
    }

    public String isTieneGruposProducto() {
        return tieneGruposProducto;
    }

    public void setTieneGruposProducto(String tieneGruposProducto) {
        this.tieneGruposProducto = tieneGruposProducto;
    }

    public String isPrecioVariableProducto() {
        return precioVariableProducto;
    }

    public void setPrecioVariableProducto(String precioVariableProducto) {
        this.precioVariableProducto = precioVariableProducto;
    }

    public String getTiempoGarantiaEquipo() {
        return tiempoGarantiaEquipo;
    }

    public void setTiempoGarantiaEquipo(String tiempoGarantiaEquipo) {
        this.tiempoGarantiaEquipo = tiempoGarantiaEquipo;
    }

    public void imprimir(){
        System.out.println("PRODUCTO: "+this.getNombreProducto()+"\nMARCA: "+
            this.getMarcaProducto()+ "\nID: "+this.getIdProducto());
    }
}

```

*CÓDIGO FUENTE DE LA CLASE CLIENTE

```

public class Cliente {
    private int idCliente;
    private String nombre;
    private String calle;

```

```

private String colonia;
private String delegacion;
private String telefono;
private boolean rentando;

public String getDelegacion() {
    return delegacion;
}

public void setDelegacion(String delegacion) {
    this.delegacion = delegacion;
}

public boolean isRentando() {
    return rentando;
}

public void setRentando(boolean rentando) {
    this.rentando = rentando;
}

public Cliente(){

}

public Cliente(String nombre, String telefono) {
    this.nombre = nombre;
    this.telefono = telefono;
}

public Cliente(String nombre, String calle, String colonia, String telefono) {
    this.nombre = nombre;
    this.calle = calle;
    this.colonia = colonia;
    this.telefono = telefono;
}

public int getIdCliente() {
    return idCliente;
}

```

```

public void setIdCliente(int idCliente) {
    this.idCliente = idCliente;
}

public Cliente(String nombre) {
    this.nombre = nombre;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getCalle() {
    return calle;
}

public void setCalle(String calle) {
    this.calle = calle;
}

public String getColonia() {
    return colonia;
}

public void setColonia(String colonia) {
    this.colonia = colonia;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public void Imprimir(){
    System.out.println("Id cliente: "+this.getIdCliente());
}

```

```

        System.out.println("Nombre: "+this.getNombre());
        System.out.println("Calle: "+this.getCalle());
    }
}

```

*CÓDIGO FUENTE DE LA CLASE CONTRATO

```

public class Contrato {
    private String numero_contrato;
    private Cliente cliente_contrato;
    private Producto[] producto_contrato;
    private Date fecha;
    private int vigencia_contrato;

    public Contrato(String numero_contrato, Cliente cliente_contrato, Producto[]
producto_contrato, Date fecha, int vigencia_contrato) {
        this.numero_contrato = numero_contrato;
        this.cliente_contrato = cliente_contrato;
        this.producto_contrato = producto_contrato;
        this.fecha = fecha;
        this.vigencia_contrato = vigencia_contrato;
    }

    public String getNumero_contrato() {
        return numero_contrato;
    }

    public void setNumero_contrato(String numero_contrato) {
        this.numero_contrato = numero_contrato;
    }

    public Cliente getCliente_contrato() {
        return cliente_contrato;
    }

    public void setCliente_contrato(Cliente cliente_contrato) {
        this.cliente_contrato = cliente_contrato;
    }

    public Producto[] getProducto_contrato() {
        return producto_contrato;
    }
}

```



```
}  
CÓDIGO FUENTE DE LA CLASE PRINT NECESARIA PARA LA IMPRESIÓN DE  
TICKETS
```

```
package modelo;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.print.Doc;  
import javax.print.DocFlavor;  
import javax.print.DocPrintJob;  
import javax.print.PrintException;  
import javax.print.PrintService;  
import javax.print.PrintServiceLookup;  
import javax.print.SimpleDoc;  
import javax.print.attribute.HashPrintRequestAttributeSet;  
import javax.print.attribute.PrintRequestAttributeSet;  
import javax.print.attribute.standard.Copies;  
import javax.print.attribute.standard.MediaPrintableArea;  
import javax.print.attribute.standard.MediaSizeName;  
import javax.print.attribute.standard.OrientationRequested;  
import javax.swing.JFileChooser;  
import javax.swing.filechooser.FileNameExtensionFilter;  
  
/**  
 *  
 * @author f2 a55m-s1  
 */  
public class Print {  
    private JFileChooser fileChooser;  
        private FileNameExtensionFilter filter = new  
FileNameExtensionFilter("Imagen", "bmp", "gif", "png");  
    private FileInputStream fileInputStream = null;  
    private File file=null;  
  
    public Print(){}  
  
    public boolean Abrir()  
    {
```

```

fileChooser = new JFileChooser();
fileChooser.setFileFilter(filter);
int result = fileChooser.showOpenDialog(null);
if ( result == JFileChooser.APPROVE_OPTION ){
    try {
        this.fileInputStream = new FileInputStream( fileChooser.getSelectedFile()
);
        this.file = fileChooser.getSelectedFile();
        return true;
    } catch (IOException ex) {
        System.out.println("Error al abrir archivo " + ex);
    }
}
return false;

```

*FRAGMENTO DEL CÓDIGO FUENTE DE LA CLASE TICKET PARA EL MÓDULO DE GENERACIÓN DE TICKETS

```

public class Ticket {
//especificamos el contenido que llevará el ticket y se agregan espacios para que
dar presentacion
    public String contentTicket =" "+
        "OXIGEN AIR PUREZA PARA LA VIDA "+
        "    S.A. DE C.\n"+
        "    RFC: OAP1305244R1\n"
+ "    FRAY PEDRO DE GANTE #20 \n"
+ "COL SECCION XVI TLALPAN CP 14080\n"+
        "TELEFONOS: 6234-5747  6286-3990\n1988-1603  6234-7642
4608-4541\n"+
        "    {{fecha}} \n"
+ "=====\n"
+"ARTICULOS: \n"+
        "{{items}}"
+ "\n"+
        "=====\n"
+ "    GRACIAS POR SU COMPRA  \n"
+ "ESPERAMOS SU VISITA NUEVAMENTE\n"
+ " LE ATENDIO: {{vendedor}}"
+ "\n"
+ "\n\n\n \n";

```

```

//El constructor que setea los valores a la instancia
public Ticket(String items, String vendedor) {
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy
hh:mm:ss");
    String fech=(dateFormat.format(new Date()));
    this.contentTicket = this.contentTicket.replace("{{fecha}}", fech);
    this.contentTicket = this.contentTicket.replace("{{items}}", items);
    this.contentTicket = this.contentTicket.replace("{{vendedor}}", vendedor);
}

public void print() throws IOException, PrintException {

    PrintService[] services = PrintServiceLookup.lookupPrintServices(null, null);
//nos da el array de los servicios de impresión

    byte[] bytes = this.contentTicket.getBytes();
//Tipo de datos bytes; Subtipo: autodetectado
    DocFlavor flavor = DocFlavor.BYTE_ARRAY.AUTODENSENSE;

    Doc doc = new SimpleDoc(bytes, flavor, null);
//Creamos un trabajo de impresión
    DocPrintJob job = null;
    if (services.length > 0) {
        for (int i = 0; i < services.length; i++) {
            if (services[i].getName().equals("POS-58")) {
//aquí seleccionamos la impresora de tickets
                job = services[i].createPrintJob();
                System.out.println(i+": "+services[i].getName());
            }
        }
    }

//Imprimimos dentro de un try
    try {
        job.print(doc, null);
    } catch (PrintException ex) {
        System.out.println(ex);
    }
}

```

```

public static String texto(int numArticulos, int cantidades[], String productos[], double
precios[], double efectivo){
    String aux,salida="";
    int sumaN=0;
    for(int i=0;i<numArticulos;i++){
        sumaN+=precios[i];
        int n=productos[i].length();
        if(n>24){
            aux=productos[i].substring(0, 23);
        }else{
            aux=productos[i];
        }
        salida+=String.valueOf(cantidades[i])+"-"+aux+" $" +precios[i]+"\\n";
    }
    double resta=efectivo-sumaN;
    String suma="TOTAL"+" $" +sumaN+"\\n";
    String espacios="",espacios2="",espacios3="";
    String efec="EFECTIVO $" +efectivo+"\\n";
    String auxiliar="CAMBIO $" +resta+"\\n";
    for(int i=0;i<(28-suma.length());i++){
        espacios+=" ";
    }
    for(int i=0;i<(28-efec.length());i++){
        espacios2+=" ";
    }
    for(int i=0;i<(30-auxiliar.length());i++){
        espacios3+=" ";
    }
    salida+=espacios+suma;
    salida+=espacios2+efec;
    salida+=espacios3+resta;

    return salida;
}

```

*FRAGMENTO DE CÓDIGO DE LA PARTE DEL CONTROLADOR DE LA CLASE CLIENTE

```

package Controlador;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import java.sql.SQLException;
import java.sql.Statement;
import modelo.Cliente;
/**
 *
 * @author f2 a55m-s1
 */
public class ModeloCliente {
    private Connection conexion;
    private Cliente cliente;

    public ModeloCliente(Connection conexion, Cliente cliente) {
        this.conexion = conexion;
        this.cliente = cliente;
    }

    public Connection getConexion() {
        return conexion;
    }

    public void setConexion(Connection conexion) {
        this.conexion = conexion;
    }

    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    public Cliente guardarCliente(String nombre, String calle, String colonia, String
delegacion, String telefono){
        Cliente c=new Cliente();
        try {
            String sql="insert into
cliente(nombre_cliente,calle_cliente,colonia_cliente,delegacion_cliente,telefono_clie
nte) values(?,?,?,?);";
            PreparedStatement st=this.getConexion().prepareStatement(sql);
            st.setString(1,nombre);
            st.setString(2,calle);

```

```

        st.setString(3,colonia);
        st.setString(4,delegacion);
        st.setString(5,telefono);
        st.execute();
    } catch (SQLException ex) {
        System.out.println("Error en modelo guardarCliente "+ex);
    }
    return c;
}

public Cliente cargarCliente(int idCliente){
    try {
        Cliente n=new Cliente();
        String sql="select nombre_cliente, calle_cliente, colonia_cliente,
delegacion_cliente, telefono_cliente "
            + "from cliente where idCliente=?";
        PreparedStatement ps=this.getConexion().prepareStatement(sql);
        ps.setInt(1, idCliente);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            n.setNombre(rs.getString("nombre_cliente"));
            n.setCalle(rs.getString("calle_cliente"));
            n.setColonia(rs.getString("colonia_cliente"));
            n.setDelegacion(rs.getString("delegacion_cliente"));
            n.setTelefono(rs.getString("telefono_cliente"));
            n.setIdCliente(idCliente);
        }
        return n;
    } catch (SQLException ex) {
        System.out.println("Error al cargar cliente: "+ex);
        return null;
    }
}
}
}

```

*FRAGMENTO DEL CÓDIGO FUENTE DE LA PARTE CONTROLADORA DE LA CLASE PRODUCTO

```

public class ModeloProducto {
    private Producto producto;
    private Connection conexion;

```

```

public ModeloProducto(){

}

public ModeloProducto(Producto producto, Connection conexion) {
    this.producto = producto;
    this.conexion = conexion;
}

public Producto getProducto() {
    return producto;
}

public void setProducto(Producto producto) {
    this.producto = producto;
}

public Connection getConexion() {
    return conexion;
}

public void setConexion(Connection conexion) {
    this.conexion = conexion;
}

public Producto setDatos(int idProducto,Connection c){
    Producto producto1=new Producto();
    try{
        String sql="select
nombre_producto,marca_producto,precio_producto,tiempo_garantia,
precio_variable, "
        + "sobre_pedido,tiene_grupos from producto where idproducto=?";
        PreparedStatement ps=c.prepareStatement(sql);
        ps.setInt(1,idProducto);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            producto1.setNombreProducto(rs.getString("nombre_producto"));
            producto1.setMarcaProducto(rs.getString("marca_producto"));
            producto1.setPrecioProducto(rs.getDouble("precio_producto"));
            producto1.setPrecioVariableProducto(rs.getString("precio_variable"));
            producto1.setTiempoGarantiaEquipo(rs.getString("tiempo_garantia"));
            producto1.setTieneGruposProducto(rs.getString("tiene_grupos"));
        }
    }
}

```

```

        if(rs.getString("sobre_pedido").equalsIgnoreCase("si")){
            producto1.setSobre_pedido(true);
        }else{
            producto1.setSobre_pedido(false);
        }
        producto1.setIdProducto(idProducto);
    }
    return producto1;
}catch(Exception e){
    System.out.println("Error en setDatos productos: "+e);
}
return producto1;
}

```

*CÓDIGO FUENTE DE UNO DE LOS MÉTODOS QUE DA ACCESO A LOS DATOS DE UN OBJETO DE LA CLASE PRODUCTO

```

int getIdProducto(Connection c, int folio){
    int resultado=0;
    try{
        String sql42 = "select producto_idproducto from tabla where idtabla=?";
        PreparedStatement psl=c.prepareStatement(sql42);
        psl.setInt(1,folio);
        ResultSet rsl=psl.executeQuery();
        while(rsl.next()){
            resultado= rsl.getInt("producto_idProducto");
        }
    }catch(Exception e){
        System.out.println("Error en obtener el id del producto "+e);
    }
    return resultado;
}

```

*FRAGMENTO DEL CÓDIGO FUENTE DEL MÓDULO DE GENERACIÓN DE REGISTRO DE OPERACIONES Y MANEJO DE INVENTARIO

```

private void txt01KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try{
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {

```



```

int idProducto = Integer.parseInt(txt01.getText());
ModeloProducto mp = new ModeloProducto();
Producto pr1 = mp.setDatos(idProducto, conexion);
pr1.imprimir();
//si el producto es de precio variable
if(pr1.isPrecioVariableProducto().equalsIgnoreCase("si")){
    System.out.println("Producto de precio variable");
    //si el producto es sobre pedido
        String Scantidad=JOptionPane.showInputDialog(null,"INGRESE EL
MONTO DEL PAGO:", "MONTO:", JOptionPane.QUESTION_MESSAGE);
        Double Smonto=Double.parseDouble(Scantidad);
        ingresar(conexion,1,pr1.getIdProducto(),Smonto);
        if(pr1.isSobre_pedido()){
            System.out.println("Producto sobre pedido");
            cadenaAux=txtArea.getText();
            txtArea.setText(cadenaAux + "Prodcuto NO en almacen \n");
            //si el producto es de almacen
        }else{
            restar(conexion,idProducto,1);
            System.out.println("Producto de almacen");
            cadenaAux=txtArea.getText();
            txtArea.setText(cadenaAux +existencias(idProducto)+"\n");
        }
    //verificando grupos
    if(pr1.isTieneGruposProducto().equalsIgnoreCase("si")){
        System.out.println("se restarÃ¡n los grupos");
        restarGrupos(idProducto,false);
    }
//si el producto es de precio fijo
}else{
    System.out.println("Producto de precio NO variable");
    ingresar(conexion,1,pr1.getIdProducto(),1);
    if(pr1.isSobre_pedido()){
        System.out.println("Producto sobre pedido");
        cadenaAux=txtArea.getText();
        txtArea.setText(cadenaAux + "Prodcuto NO en almacen \n");
    //si el producto es de almacen
    }else{
        System.out.println("Producto de almacen");
        restar(conexion,idProducto,1);
        cadenaAux=txtArea.getText();
        txtArea.setText(cadenaAux +existencias(idProducto)+"\n");
    }
}

```

```

    }
    //verificando grupos
    if(pr1.isTieneGruposProducto().equalsIgnoreCase("si")){
        System.out.println("se restarÃ¡n los grupos");
        restarGrupos(idProducto,false);
    }
}
NuevaTabla();
moverTabla();
txt01.setText(null);
txt01.requestFocus();
}
}catch(NumberFormatException | HeadlessException e){
    System.out.println("Error txt01 key released: "+e);
    String texto = String.valueOf(txt01.getText());
    if (texto.equals("") == false) {
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            txt01.setText("");
            //muestra ventana de seleccion de productos
            segunda segunda = new segunda(this, true, conexion, texto);
            segunda.setVisible(true);
            if (segunda.EXIT_ON_CLOSE == 3) {
                if (txt01.getText().equals("") == false) {
                    Foco();
                    try {
                        //automaticamente ingresa el producto seleccionado
                        Robot robot = new Robot();
                        robot.keyRelease(KeyEvent.VK_ENTER);
                    } catch (Exception ex) {
                        System.out.println("error en robot: " + ex);
                    }
                }
            }
        }
        moverTabla();
    }
}
}
}
}

```

*FRAGMENTO DEL CÃDIGO FUENTE DE LA GENERACIÃN DE REPORTE
DIARIOS

```

        void generarCorte(Connection con,String nombreArchivo) throws
FileNotFoundException, DocumentException{
    try {
        FileOutputStream archivo = new FileOutputStream("E:\\CORTES\\" +
nombreArchivo + ".pdf"); //se crea el documento

        Document documento = new Document();
        PdfWriter.getInstance(documento, archivo);
        //se abre el documento para empezar a agregar texto
        documento.open();

        int numRec=0;

        documento.add(new Paragraph("CORTE DE CAJA DEL DIA
"+nombreArchivo));
        documento.add(new Paragraph("\n"));
        PdfPTable tabla1=new PdfPTable(2);

        PdfPCell celda=new PdfPCell(new Phrase("VENTAS DEL DIA"));
        celda.setColspan(2);
        celda.setHorizontalAlignment(Element.ALIGN_CENTER);
        tabla1.addCell(celda);

        String sql3="SELECT producto.nombre_producto AS NOMBRE,
SUM(tabla.CANTIDAD_PRODUCTO) as TOTALES from \n" +
                "producto, tabla, venta where
VENTA.idVENTA=TABLA.VENTA_idVENTA AND \n" +
                "PRODUCTO.idPRODUCTO=TABLA.PRODUCTO_idPRODUCTO
AND \n" +
                "venta.FECHA_VENTA=(SELECT CURDATE()) GROUP BY
producto.nombre_producto \n" +
                "order by TOTALES DESC ";

        try{
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery(sql3);
            while(rs.next()){
                String dato=rs.getString("NOMBRE");
                int dato2=rs.getInt("TOTALES");
                PdfPCell celdaC=new PdfPCell();
                PdfPCell celda2=new PdfPCell();

```

```

        if(dato.startsWith("RECARGA CILINDRO")){
            numRec+=dato2;
            celda2.setBackgroundColor(BaseColor.GREEN);
            celdaC.setBackgroundColor(BaseColor.GREEN);
        }else{
            celda2.setBackgroundColor(BaseColor.WHITE);
            celdaC.setBackgroundColor(BaseColor.WHITE);
        }
        celdaC.addElement(new Phrase(dato));
        tabla1.addCell(celdaC);
        Paragraph pa=new Paragraph(String.valueOf(dato2));
        pa.setAlignment(Element.ALIGN_CENTER);
        celda2.addElement(pa);
        celda2.setHorizontalAlignment(Element.ALIGN_CENTER);
        tabla1.addCell(celda2);
    }

}

}catch(Exception e){
    System.out.println("Error en tabla 1: "+e);
}

documento.add(tabla1);
documento.add(new Paragraph("\n"));

Paragraph p=new Paragraph("NMERO DE RECARGAS DEL DA:
"+numRec);
p.setAlignment(Element.ALIGN_RIGHT);
documento.add(p);

if (generarGraficaRecargas(conexion)) {
    try {
        Image imagen = Image.getInstance("grafica
"+getFechaActual()+".png");
        imagen.setAlignment(Element.ALIGN_CENTER);
        //imagen.scaleToFit(500, 400);
        documento.add(imagen);
        documento.add(new Paragraph("\n"));
    } catch (IOException | DocumentException e) {
        System.out.println("Error en la imagen " + e);
    }
}
}

```

```
documento.add(new Paragraph("\n"));
documento.add(new Paragraph("\n"));
```

```
documento.newPage();
```

```
PdfPTable tabla2=new PdfPTable(2);
PdfPCell celda2=new PdfPCell(new Phrase("VENTA TOTAL DEL DIA (POR
FORMA DE PAGO)"));
celda2.setColspan(2);
celda2.setHorizontalAlignment(Element.ALIGN_CENTER);
celda2.setBackgroundColor(BaseColor.LIGHT_GRAY);
tabla2.addCell(celda2);
```

```
Double pt=0.0;
Double ef=0.0;
Double valor;
for(int i=0;i<tabla.getRowCount();i++){
    valor=Double.parseDouble(tabla.getValueAt(i, 6).toString());
    if(tabla.getValueAt(i,5).toString().equals("EF")){
        ef+=valor;
    }else{
        pt+=valor;
    }
}
```

```
String sqlMG="select gastos.concepto_gastos,gastos.costos_gastos from
gastos,venta where \n" +
```

```
                "venta.idVENTA=gastos.venta_idventa and
venta.FECHA_VENTA=(select curdate());"
```

```
PdfPTable tablaG=new PdfPTable(2);
Paragraph pp=new Paragraph("GASTOS DEL DIA: "+getFechaActual());
PdfPCell del=new PdfPCell(pp);
del.setHorizontalAlignment(Element.ALIGN_CENTER);
del.setBackgroundColor(BaseColor.LIGHT_GRAY);
del.setColspan(2);
tablaG.addCell(del);
Paragraph p1=new Paragraph("CONCEPTO DEL GASTO");
PdfPCell titulo1=new PdfPCell(p1);
titulo1.setHorizontalAlignment(Element.ALIGN_CENTER);
Paragraph p2=new Paragraph("MONTO DEL GASTO");
PdfPCell titulo2=new PdfPCell(p2);
```

```

titulo2.setHorizontalAlignment(Element.ALIGN_CENTER);
tablaG.addCell(titulo1);
tablaG.addCell(titulo2);
try {
    Statement ss=conexion.createStatement();
    ResultSet rsMG=ss.executeQuery(sqlMG);
    while(rsMG.next()){
        String aux1=rsMG.getString("concepto_gastos");
        Double dou=rsMG.getDouble("costo_gastos");
        Paragraph p11 = new Paragraph(aux1);
        PdfPCell aux2 = new PdfPCell(p11);
        aux2.setHorizontalAlignment(Element.ALIGN_CENTER);
        Paragraph p22 = new Paragraph(String.valueOf(dou));
        PdfPCell titulo23 = new PdfPCell(p22);
        titulo23.setHorizontalAlignment(Element.ALIGN_CENTER);
        tablaG.addCell(aux2);
        tablaG.addCell(titulo23);
    }
    PdfPCell pdf=new PdfPCell(new Phrase("TOTAL GASTOS: "));
    PdfPCell pdf2=new PdfPCell(new
Phrase(String.valueOf(getGastos(conexion))));
    pdf.setBackgroundColor(BaseColor.LIGHT_GRAY);
    pdf.setHorizontalAlignment(Element.ALIGN_RIGHT);
    pdf2.setBackgroundColor(BaseColor.LIGHT_GRAY);
    pdf2.setHorizontalAlignment(Element.ALIGN_RIGHT);
    tablaG.addCell(pdf);
    tablaG.addCell(pdf2);

} catch (SQLException ex) {
    System.out.println("Error en la creaci3n de la tabla de gastos");
}

documento.add(new Paragraph("\n"));
documento.add(tablaG);

documento.add(new Paragraph("\n"));

PdfPCell celdaz1=new PdfPCell(new Phrase("TOTAL"));
celdaz1.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell celdaz2=new PdfPCell(new
Phrase(String.valueOf(truncar(suma))));
celdaz2.setHorizontalAlignment(Element.ALIGN_CENTER);

```

```

tabla2.addCell(celdaz1);
tabla2.addCell(celdaz2);

PdfPCell celdaz3=new PdfPCell(new Phrase("GASTOS"));
celdaz3.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell celdaz4=new PdfPCell(new
Phrase(String.valueOf(truncar(getGastos(conexion))));
celdaz4.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla2.addCell(celdaz3);
tabla2.addCell(celdaz4);

PdfPCell celdax3=new PdfPCell(new Phrase("EFECTIVO"));
celdax3.setHorizontalAlignment(Element.ALIGN_CENTER);

tabla2.addCell(celdax3);
PdfPCell celdax4=new PdfPCell(new
Phrase(String.valueOf(truncar(ef-getGastos(conexion))));
celdax4.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla2.addCell(celdax4);
PdfPCell celdax5=new PdfPCell(new Phrase("TARJETA"));
celdax5.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla2.addCell(celdax5);
PdfPCell celdax6=new PdfPCell(new Phrase(String.valueOf(pt)));
celdax6.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla2.addCell(celdax6);

documento.add(tabla2);

documento.close();

}catch(FileNotFoundException | DocumentException | NumberFormatException
e){
    System.out.println("error en pdf "+e);
}
}

```

*FRAGMENTO DE CÓDIGO FUENTE DEL MÓDULO DE GENERACIÓN DE GARANTÍAS

```
public void generarGarantia(Connection c,String nombreArchivo,Producto p, Cliente
cl, boolean check) throws BadElementException, IOException{
    try{
        FileOutputStream archivo=new
FileOutputStream("E:\\garantias\\"+nombreArchivo+".pdf");
        Document documento=new Document(PageSize.LETTER,30,30,20,20);
        PdfWriter.getInstance(documento, archivo);
        documento.open();

        Font letraGrande=new Font();
        letraGrande.setFamily("arial");
        letraGrande.setSize(16);
        //////////////////////////////////
        Font letraMediana=new Font();
        letraMediana.setFamily("arial");
        letraMediana.setSize(10);
        //////////////////////////////////
        Font letraChica=new Font();
        letraChica.setFamily("arial");
        letraChica.setSize(7);

        Paragraph par=new Paragraph("POLIZA DE GARANTIA #DE FOLIO:
"+folio);
```



```

par.setAlignment(Element.ALIGN_RIGHT);
documento.add(par);
Image imagen=Image.getInstance("logoFondoBlanco.png");
//imagen.setAlignment(Element.ALIGN_CENTER);
imagen.setAbsolutePosition(245,680);
imagen.scaleToFit(100,200);
documento.add(imagen);
    Paragraph par1=new Paragraph("\n\n OXYGEN AIR PUREZA PARA LA
VIDA S.A. DE C.V.");
par1.setAlignment(Element.ALIGN_CENTER);
par1.setFont(letraGrande);
documento.add(par1);
    Paragraph par2=new Paragraph("FRAY PEDRO DE GANTE #20, COL
BELISARIO DOMINGUEZ SECCIÓN XVI, \n TLALPAN, MÉXICO D.F.");
par2.setAlignment(Element.ALIGN_CENTER);
par2.setFont(letraMediana);
documento.add(par2);
    Paragraph par3=new Paragraph("TELEFONOS: 1988-1603, 4608-4541,
6234-5747");
par3.setAlignment(Element.ALIGN_CENTER);
par3.setFont(letraMediana);
documento.add(par3);

Paragraph salto=new Paragraph("\n");
//documento.add(salto);

Paragraph par4=new Paragraph("FECHA: "+TODO.getFechaActual());
par4.setAlignment(Element.ALIGN_RIGHT);
par4.setFont(letraMediana);
documento.add(par4);
//documento.add(salto);
Paragraph par5=new Paragraph("NOMBRE: "+cl.getNombre());
par5.setAlignment(Element.ALIGN_LEFT);
par5.setFont(letraChica);
documento.add(par5);
//documento.add(salto);
Paragraph par6=new Paragraph("CALLE: "+cl.getCalle());
par6.setAlignment(Element.ALIGN_LEFT);
par6.setFont(letraChica);
documento.add(par6);
//documento.add(salto);
Paragraph par7=new Paragraph("COLONIA: "+cl.getColonia());

```

```

par7.setAlignment(Element.ALIGN_LEFT);
par7.setFont(letraChica);
documento.add(par7);
//documento.add(salto);
        Paragraph par8=new Paragraph("DELEGACION/MUNICIPIO:
"+cl.getDelegacion());
par8.setAlignment(Element.ALIGN_LEFT);
par8.setFont(letraChica);
documento.add(par8);
//documento.add(salto);
Paragraph par9=new Paragraph("TELEFONO: "+cl.getTelefono());
par9.setAlignment(Element.ALIGN_LEFT);
par9.setFont(letraChica);
documento.add(par9);
documento.add(salto);

PdfPTable tabla=new PdfPTable(4);
PdfPCell celda11=new PdfPCell(new Phrase("NOMBRE DEL EQUIPO"));
celda11.setBackgroundColor(BaseColor.LIGHT_GRAY);
celda11.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla.addCell(celda11);
PdfPCell celda12=new PdfPCell(new Phrase("MARCA"));
celda12.setBackgroundColor(BaseColor.LIGHT_GRAY);
celda12.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla.addCell(celda12);
PdfPCell celda13=new PdfPCell(new Phrase("PRECIO"));
celda13.setBackgroundColor(BaseColor.LIGHT_GRAY);
celda13.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla.addCell(celda13);
PdfPCell celda14=new PdfPCell(new Phrase("TIEMPO DE GARANTÍA"));
celda14.setBackgroundColor(BaseColor.LIGHT_GRAY);
celda14.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla.addCell(celda14);
PdfPCell celda21=new PdfPCell(new Phrase(p.getNombreProducto()));
celda21.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla.addCell(celda21);
PdfPCell celda22=new PdfPCell(new Phrase(p.getMarcaProducto()));
celda22.setHorizontalAlignment(Element.ALIGN_CENTER);
tabla.addCell(celda22);
        PdfPCell celda23=new PdfPCell(new
Phrase("$"+String.valueOf(p.getPrecioProducto())));
        celda23.setHorizontalAlignment(Element.ALIGN_CENTER);

```

```

        tabla.addCell(celda23);
        PdfPCell celda24=new PdfPCell(new
Phrase(p.getTiempoGarantiaEquipo()));
        celda24.setHorizontalAlignment(Element.ALIGN_CENTER);
        tabla.addCell(celda24);
        documento.add(tabla);

        //documento.add(new Paragraph("\n"));
        PdfPCell celdaNumeros=new PdfPCell(new Phrase("Números de serie:
"+txtNumerosDeSerie.getText()));
        documento.add(celdaNumeros);
        documento.add(new Paragraph("Números de serie:
"+txtNumerosDeSerie.getText()));
        //documento.add(new Paragraph("\n"));

        documento.add(new Paragraph("CLAUSULAS"));

        if (check) {
            documento.add(new Paragraph("1.- USO OBLIGATORIO DE AGUA
BIDESTILADA"));
            documento.add(new Paragraph("2.- INDISPENSABLE USO DE
REGULADOR DE VOLTAJE DE 2000 WATTS"));
            documento.add(new Paragraph("3.- NO USAR EL FLUJO A MAS DE 3.5
LITROS POR MINUTO"));
            documento.add(new Paragraph("4.- MENTENIMIENTO PREVENTIVO
OBLIGATORIO CADA 6 MESES PARA NO PERDER GARANTIA"));
            documento.add(new Paragraph("5.- NO AMPARA MAL USO POR PARTE
DEL CLIENTE, SÓLO DEFECTO DE FABRICACIÓN "));
            documento.add(new Paragraph("6.- CONSTA DE REPARACIÓN Y/O
CAMBIO FÍSICO DIAGNOSTICADO POR EL PROVEEDOR DEL EQUIPO"));
            documento.add(new Paragraph("7.- ES VÁLIDA EN NUESTRAS
INSTALACIONES, NO APLICA SERVICIO A DOMICILIO SIN COSTO "));
            documento.add(new Paragraph("8.- NO ESTA SUJETA A PRÉSTAMO DE
EQUIPO Y/O RENTA DE EQUIPO SIN COSTO NI A CAMBIO INMEDIATO"));
            documento.add(new Paragraph("9.- LOS TIEMPOS DE RESPUESTA EN
LA GARANTÍA DEPENDE DE LOS PROVEEDORES Y/O PROCEDENCIAS DEL
EQUIPO"));
            documento.add(new Paragraph("10.- UNA VEZ SALIDA LA MERCANCIA
NO HAY DEVOLUCIÓN DEL DINERO"));
            documento.add(new Paragraph("11.- INDISPENSABLE CONSERVAR LA
REMISIÓN PARA CUALQUIER DUDA O ACLARACIÓN"));
        }

```

```

    } else {
        documento.add(new Paragraph("1.- NO AMPARA MAL USO POR PARTE
DEL CLIENTE, SÓLO DEFECTO DE FABRICACIÓN "));
        documento.add(new Paragraph("2.- CONSTA DE REPARACIÓN Y/O
CAMBIO FÍSICO DIAGNOSTICADO POR EL PROVEEDOR DEL EQUIPO"));
        documento.add(new Paragraph("3.- ES VÁLIDA EN NUESTRAS
INSTALACIONES, NO APLICA SERVICIO A DOMICILIO SIN COSTO "));
        documento.add(new Paragraph("4.- NO ESTA SUJETA A PRÉSTAMO DE
EQUIPO Y/O RENTA DE EQUIPO SIN COSTO NI A CAMBIO INMEDIATO"));
        documento.add(new Paragraph("5.- LOS TIEMPOS DE RESPUESTA EN
LA GARANTÍA DEPENDE DE LOS PROVEEDORES Y/O PROCEDENCIAS DEL
EQUIPO"));
        documento.add(new Paragraph("6.- UNA VEZ SALIDA LA MERCANCIA
NO HAY DEVOLUCIÓN DEL DINERO"));
    }
    documento.add(new Paragraph(notas.getText()));

    documento.add(new Paragraph("\n"));
    documento.add(new Paragraph("\n"));

    PdfPTable tablaF = new PdfPTable(2);
    float[] medidaCeldas = {1.5f, 1.5f};

    //ASIGNAS LAS MEDIDAS A LA TABLA (ANCHO)
    tablaF.setWidths(medidaCeldas);
    //Image imagen2=Image.getInstance("blanco.png");
    //imagen2.scaleToFit(30, 50);
    Paragraph psalto=new Paragraph("\n\n");
    PdfPCell espacio=new PdfPCell(psalto);
    tablaF.addCell(espacio);
    tablaF.addCell(espacio);
    Paragraph pf=new Paragraph("OXIGEN AIR PUREZA PARA LA VIDA SA.
DE CV.");
    pf.setFont(letraChica);
    PdfPCell celdaF=new PdfPCell(pf);
    celdaF.setHorizontalAlignment(Element.ALIGN_CENTER);
    Paragraph pfc=new Paragraph("LEÍ Y ESTOY DE ACUERDO");
    pfc.setFont(letraChica);
    PdfPCell celdaFC=new PdfPCell(pfc);
    celdaFC.setHorizontalAlignment(Element.ALIGN_CENTER);
    tablaF.addCell(celdaF);

```

```
tablaF.addCell(celdaFC);

documento.add(tablaF);

documento.close();
}catch(FileNotFoundException | DocumentException e){
    System.out.println("error en generar garantia: "+e);
}
}
```