

# UNIVERSIDAD AUTÓNOMA METROPOLITANA

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Proyecto de Integración Terminal de Ingeniería en Computación I

**“Gestión de Información para el Manejo de Clientes de PyMEs  
CRM”**

Mendoza Manzano Gerardo  
204308177

Asesorado por:

M. en C. Rafaela Blanca Silva López  
17114

# Índice

## Contenido

Introducción .....	3
Antecedentes .....	5
Descripción Técnica.....	7
Desarrollo del Proyecto.....	11
Casos de Uso .....	11
Validar Usuario .....	11
Registrar Usuario.....	14
Ofrecer Servicios .....	17
Administrar Usuarios.....	20
Administrar Productos .....	22
Administrar Servicios.....	24
Administrar Foro .....	26
Diagraman de Clases .....	31
Estructura de la base de datos .....	34
Diccionario de datos.....	35
Anexo 1 Manual de usuario .....	40
ANEXO 2. Código Fuente .....	79
Resumen.....	500
Bibliografía .....	500
Referencias.....	500

## Introducción

CRM (*Customer Relationship Management*) es el conjunto de estrategias de negocio, *marketing*, comunicación e infraestructuras tecnológicas, diseñadas con el objetivo de construir una relación duradera con los clientes, identificando, comprendiendo y satisfaciendo sus necesidades. Permite a la empresa centrar su atención en el cliente para interactuar más efectivamente con él.

Las herramientas de la gestión de relaciones con los clientes (CRM), son las soluciones tecnológicas para conseguir desarrollar la “teoría” del *marketing* relacional, que es la “la estrategia de negocio centrada en anticipar, conocer y satisfacer las necesidades y deseos presentes y previsibles de los clientes”.

Cabe destacar que CRM tiene como objetivo atraer y retener a los clientes de manera más exitosa a través de un proceso lógico, soportado por la tecnología de la información.

Esto consiste en construir relaciones duraderas mediante la comprensión de las necesidades y preferencias individuales y de este modo añadir valor a la empresa y al cliente. Es conseguir que los clientes sean fieles. Eso supone conocerlos, saber quiénes son, cuáles son sus gustos, sus preferencias para, así, poder ofrecerles lo que quieren, cuando lo quieren y como lo quieren.

CRM es en los tiempos que corren la suma de dos elementos: ahorro de coste y, servicio al cliente. Para ello hay que poner en funcionamiento cuatro engranes básicos: información, actuación, resolución y atención. La adecuada implementación de filosofía CRM en las Pequeñas y Medianas Empresas (PyME's), permite el incremento de los ingresos y una mayor satisfacción en el servicio a clientes.

### **Diferencias entre la PyME's y una gran Empresa**

Como podemos observar, en una PyME's es más fácil instalar una filosofía como el CRM ya que al tener menor cantidad de empleados es más sencillo formalizar su conducta y concientizarlos de ella, con una resistencia al cambio mucho menor que una gran empresa.

Otra cosa para destacar, es que la pyme al poseer una estructura pequeña con respecto a la gran empresa, posee mayor flexibilidad y agilidad para responder a cambios en la demanda, adecuándose más rápidamente a las necesidades de los clientes. En cambio, en la gran empresa se cumple un proceso más burocrático, es mucho más lenta debido a que no es tan sencillo adaptar su gran estructura a la satisfacción de las necesidades de los clientes.

## **Ventajas y desventajas del CRM**

La principal ventaja que atrae aparejada la implementación de la Gestión de las relaciones con los clientes en una empresa es el aumento de la información que esta posee de sus actuales y potenciales clientes, lo que permite direccionar la oferta hacia sus deseos y necesidades, aumentando así el grado de satisfacción y optimizando su ciclo de vida.

Otras ventajas que trae aparejada la implementación del CRM son el aumento de las ventas y la reducción del ciclo de venta. Las desventajas están relacionadas con elevados costos que tiene la aplicación de CRM, tanto en términos de recursos económicos como así también humanos, y con la dificultad que tiene el manejo de la información dada la reticencia de algunos sectores a compartirla y también al riesgo de invadir la privacidad del cliente y exponerlo a situaciones indeseadas.

Las micro y pequeñas empresas tienen la ventaja de contar con menos clientes y menos personal lo que le facilita el manejo de la información y hace difícil la aparición de problemas para compartirla.

El hecho de manejar un bajo caudal de información también reduce los costos ya que muchas veces no es necesario adquirir un software especial para hacerlo sino que se puede llevar manualmente o en programas simples como el Access.

Un problema que puede presentar algunas veces en las empresas de menor tamaño es la falta de información de los procedimientos y la falta de interés de los empleados por interiorizarse en la importancia que tienen estos en la filosofía de trabajo de la empresa

## Antecedentes

En la actualidad existen diversos Sistemas de Información o herramientas que pretenden facilitar el manejo de información de los clientes CRM entre ellos:

**PeopleSoft:** Esta solución transforma datos complejos en información útil, lo que facilita tomar decisiones en tiempo real que incrementen la rentabilidad del cliente. Permite además que ejecutivos de mercadotecnia, ventas y soporte sean más estratégicos en la manera como manejan la interacción con sus clientes.

Por ella identifican a los clientes que compran o que no están interesados en hacerlo. Pueden además: incrementar los ingresos en el punto de interacción, reducir la carga laboral y aumentar el enfoque en los proyectos más estratégicos; así como enfocarse de manera efectiva a las inversiones y a los mensajes más importantes y reducir la longitud de los ciclos de venta. [I]

**Navision:** Permite adelantarse en la economía de *e-business* de hoy, para cuidar de las relaciones comerciales: con los clientes, prospectos, inversores, proveedores y otros. Ofrece un conjunto integrado de soluciones de administración para la gestión de las relaciones con los clientes, desde compañías que implementan un CRM por primera vez, hasta compañías con una larga experiencia en esta filosofía. Esta solución de CRM crece a medida que el negocio crece. [II]

**Solomon:** Está diseñado para que el mismo usuario pueda realizar adaptaciones en forma rápida y sencilla sin modificar el código fuente. Al día de hoy Solomon está presente en casi todos los tipos de industrias, ya que han encontrado en este producto un sistema empresarial único, por su capacidad para cubrir los requerimientos particulares de información que van surgiendo conforme la empresa se va desarrollando.

La funcionalidad de Solomon permite dar a los clientes un servicio de excelente calidad. Al acceder la base de datos de Solomon, es posible generar gráficas, reportes analíticos, proyecciones, cartas, etiquetas, solicitudes de pago, confirmación de pedidos o cualquier otro documento que requiera.

**MySAP CRM:** Ofrece una panorámica nítida sobre los aspectos de las estrategias asociadas al cliente, para identificar aquello que funciona bien, y saber por qué; proporciona una serie de aplicaciones que ayudan a la empresa a centrarse en el cliente y a aumentar su nivel de eficacia, entre la cuales cabe destacar:

- Aplicaciones CRM operativas: Estas aplicaciones aumentan la capacidad de sus empleados al proporcionarles *workplaces* basados en roles.
- Además, permiten la integración perfecta en tiempo real de la interacción *front-office* y del suministro *back-office*, al tiempo que sincronizan las interacciones con el cliente a través de todos los canales.

- Aplicaciones CRM analíticas: A partir de su almacén de datos y de otras fuentes, estas aplicaciones le ayudan a comprender lo que quieren sus clientes, así como su comportamiento. También le ayudan a adquirir nuevos clientes y a retener los ya existentes.
- Aplicaciones CRM cooperativas: Estas aplicaciones le ayudan a trabajar más estrechamente con sus clientes.

**Microsoft Dynamics CRM:** Microsoft Dynamics CRM 4.0 es una solución de gestión de relaciones con el cliente que proporciona las capacidades que las organizaciones necesitan para obtener y mantener la ventaja competitiva en el mundo empresarial actual.

Incluye la funcionalidad de “Autocompletar” para que no tener que escribir la misma información una y otra vez; búsqueda más rápida; compatibilidad con ejecución masiva y cierre de campañas; y combinación de correspondencia mejorada. Microsoft Dynamics CRM 4.0 también permite a los usuarios personalizar su panel de navegación para ofrecerles un acceso rápido a la información que necesitan y a los datos relevantes para su función de trabajo concreta.

**Hipergate 3.0:** Es una suite de aplicaciones de código abierto completamente gratuita basada en un servidor Web que es accesible desde Internet Explorer 6 o superior. Tiene la capacidad de trabajar a partir de múltiples bases de datos y sistemas operativos, ideal para ser utilizado en modalidad ASP, capacidad multi-idioma. Este programa tiene capacidad multi-entidad, pudiendo ser empleado para dar servicio a una empresa, a un grupo de empresas, o en modalidad ASP para alojar un número ilimitado de entidades usuarios diferentes.

## Descripción Técnica

El sistema de manejo de clientes, se plantea como un sistema de seguimiento que se ejecuta sobre ambiente Web, con el fin de facilitar la intercomunicación con los múltiples clientes que puede atender una empresa.

Se plantea desarrollar el sistema utilizando la plataforma Java, con el fin de que pueda ser instalado en servidores con características de sistema operativo y capacidad de procesamiento diferentes.

El sistema contará con los siguientes bloques o módulos:

- a) Foro
- b) Manejo de Clientes
- c) Manejo de Productos y Servicios
- d) Manejo de Ofertas
- e) Generación de lista de Servicios

**Foro:** Este bloque permitirá tener una comunicación asíncrona con el cliente, para ofrecerle de manera personalizada los productos o servicios convenientes. Dentro de este foro se mantendrá un histórico de las ofertas realizadas al cliente.

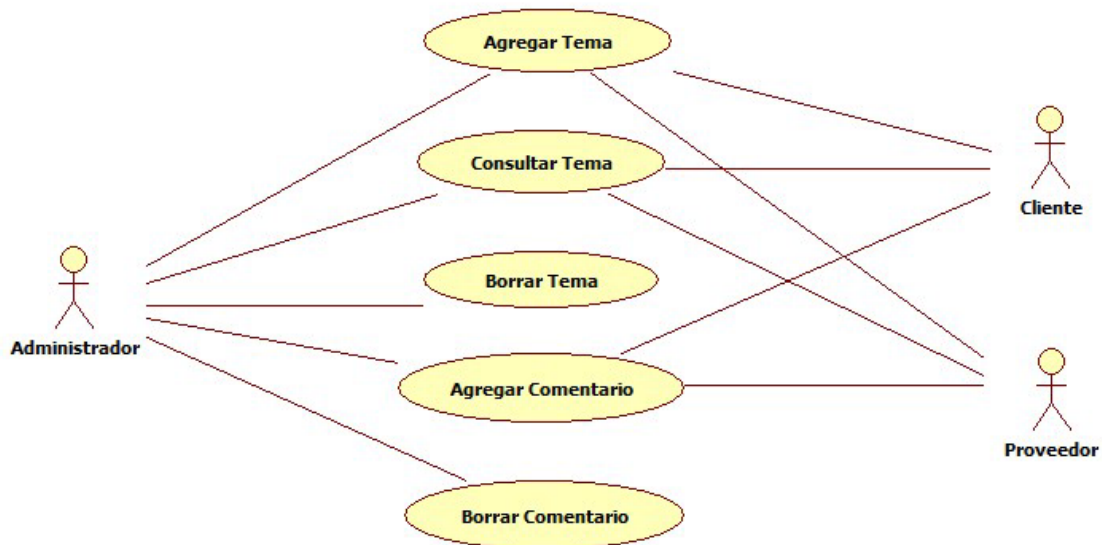


Diagrama de Casos de Uso del modulo foro

**Manejo de Clientes:** Dentro de este módulo se dará de alta los datos de los clientes de la empresa.

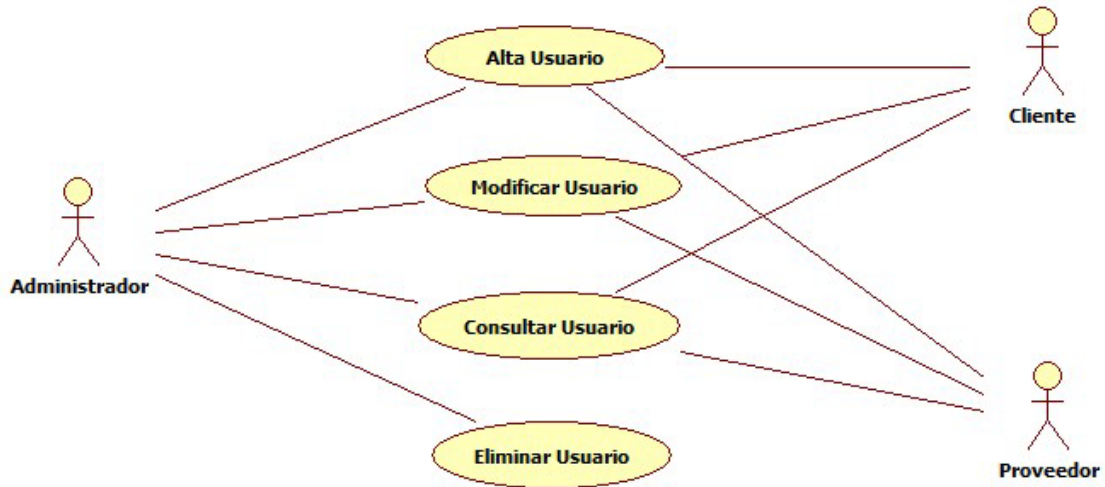


Diagrama de Casos De uso del modulo manejo clientes

**Manejo de Productos y Servicios:** Dentro de este bloque se manejarán los diferentes estados y características de los productos y servicios que manejan la empresa, así como las relaciones y dependencias entre los productos o servicios.

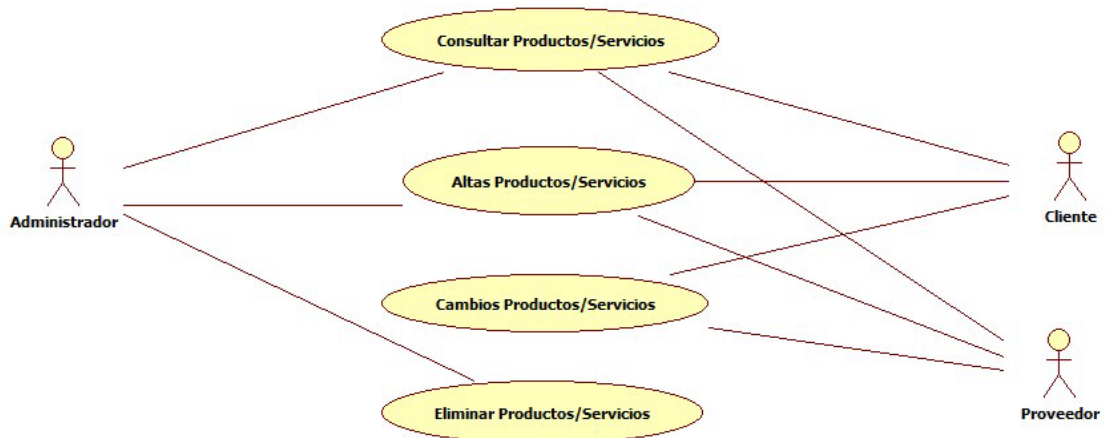
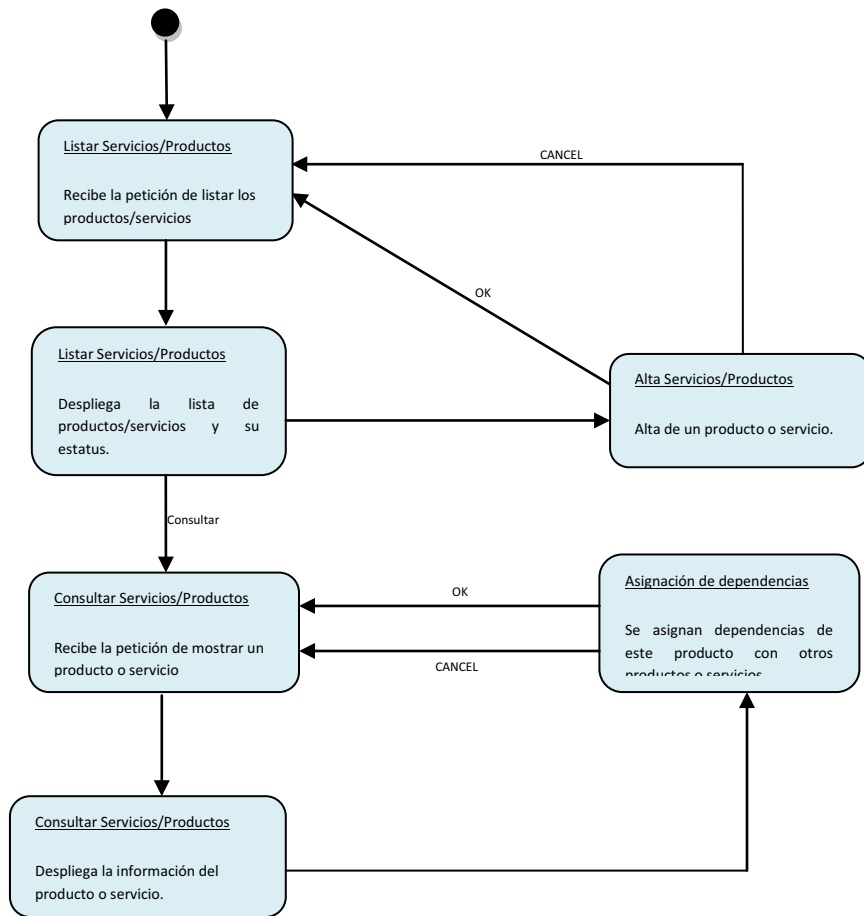


Diagrama de Casos de Uso del Modulo Manejo de Productos y Servicios



El siguiente diagrama muestra el flujo que se seguirá para dar de alta un producto o servicio de una empresa



**Manejo de Ofertas:** Este módulo manejará las ofertas de productos y servicios que se le pueden ofrecer al cliente, dependiendo de las necesidades que él tenga. Estas necesidades corresponderán a las que un determinado producto o servicio.

Este módulo podrá atender tanto a los clientes, para que ellos mismos realicen una búsqueda de las necesidades que pueden ser cubiertos por la empresa, como a las empresas para que se puedan ofrecer los servicios o productos a los clientes.

**Generación de lista de Servicios y Productos:** Es el módulo encargado de imprimir los servicios y productos que ofrece una empresa para las distintas necesidades de los clientes.



Diagrama De Caso de Uso del Modulo Generación de listas de Servicios y Productos

# Desarrollo del Proyecto

## Casos de Uso

### Validar Usuario

#### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Proveedor
- ✓ Base de Datos de Registro

#### Propósito:

- ✓ Verificar que un Usuario, esté dado de alta en el sistema

#### Precondiciones:

- ✓ El usuario debe estar dado de alta en el sistema (Ingresar al sistema).
- ✓ Ninguna (Registrarse por primera vez).

#### Post-condiciones:

- ✓ Se inicializa la sesión en el sistema dependiendo del tipo de usuario que se registró (Ingresar al Sistema).
- ✓ Se despliega el formulario con los datos necesarios para introducirlos en el sistema (Registrarse por primera vez).

#### Flujo Principal:

- ✓ Se muestra la pantalla principal (P-1), en donde el usuario puede elegir entre crear una nueva cuenta al presionar "Registrarse" (S-1), ingresar sus datos (E-1), y presionar "Ingresar" (S-2) o presionar "Limpiar" (S-3).

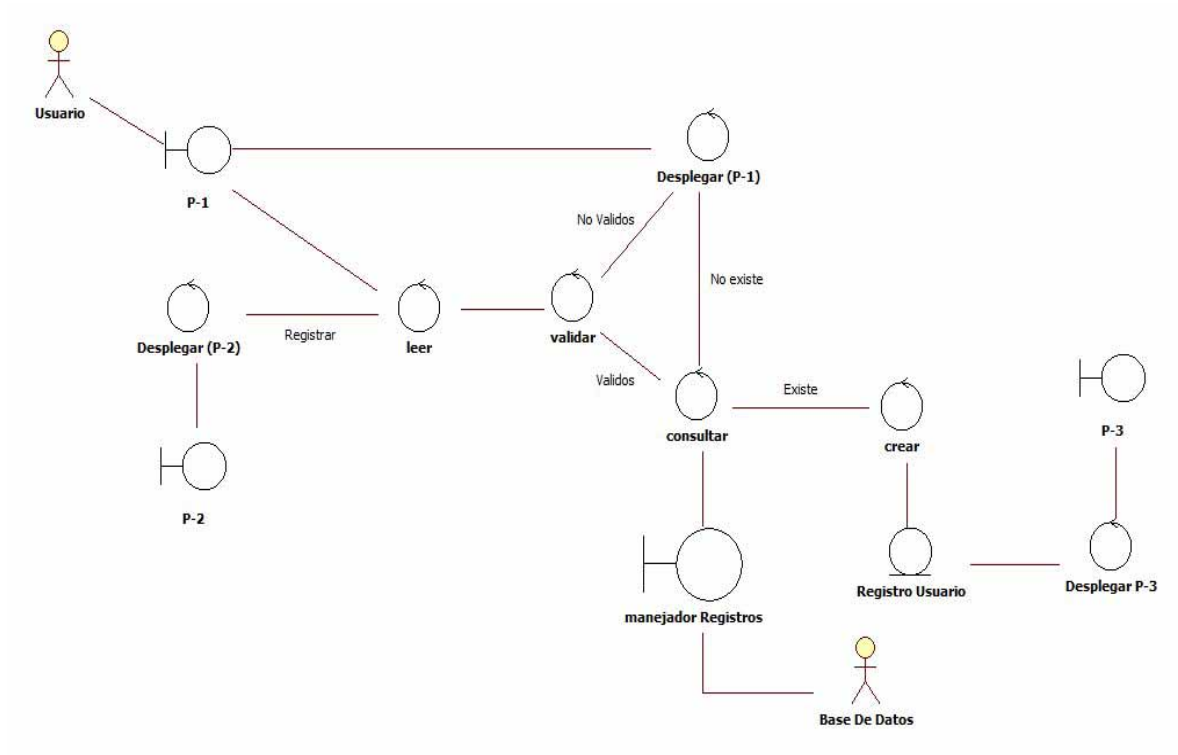
#### Sub-flujo:

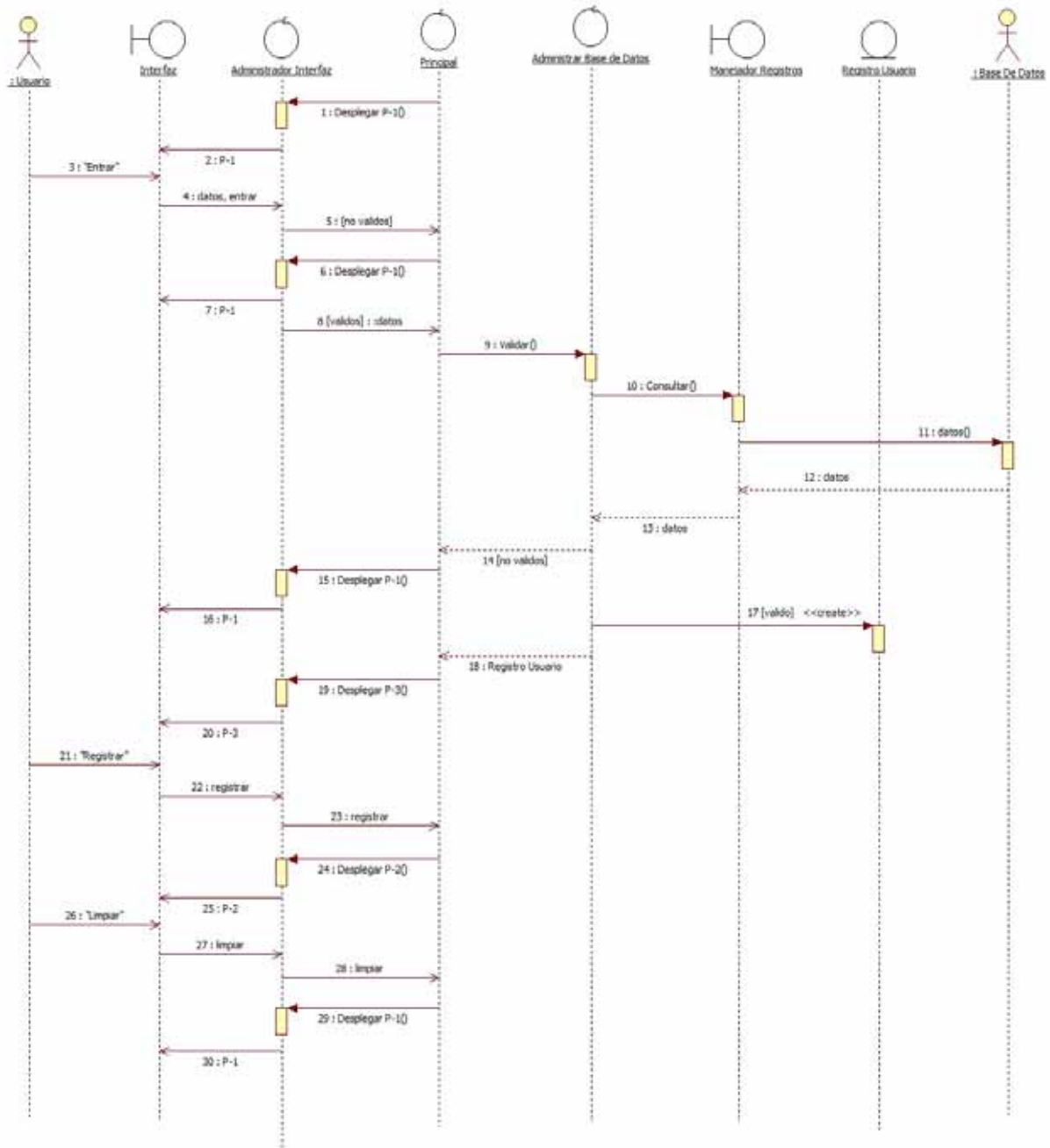
- ✓ (S-1) Se despliega la pantalla (P-2) y se ejecuta el caso de uso Registrar Usuario.
- ✓ (S-2) El sistema valida los datos introducidos por el usuario (E-2), se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-3) Se limpia los datos introducidos por el usuario

#### Excepciones:

- ✓ (E-1) En caso de que los datos no sean introducidos de manera correcta, se mostrara un mensaje en la misma pantalla (P-1).

- ✓ (E-2) En caso de que los datos introducidos no existan en la Base de Datos, se mostrara un mensaje en la misma pantalla (P-1)





## Registrar Usuario

### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Proveedor
- ✓ Base de Datos de Registro

### Propósito:

- ✓ Que el usuario introduzca sus datos para obtener una nueva cuenta en el sistema.

### Precondiciones:

- ✓ El usuario no debe existir en la base de datos de Registro.

### Post-condiciones:

- ✓ Se almacenarán en la Base de datos los datos del nuevo usuario.

### Flujo Principal:

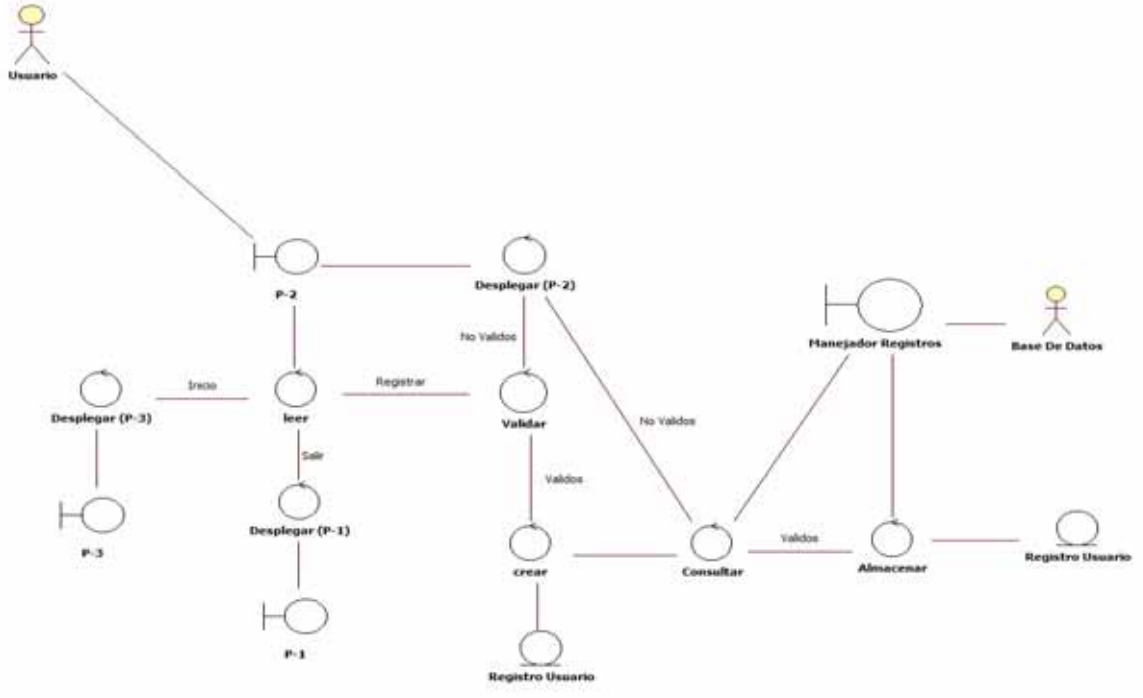
- ✓ Se muestra la pantalla (P-2), en donde el usuario llena los campos correspondientes y elige “Inicio” (S-1), “Salir” (S-2), “Registrar” (S-3) o “Cancelar” (S-4).

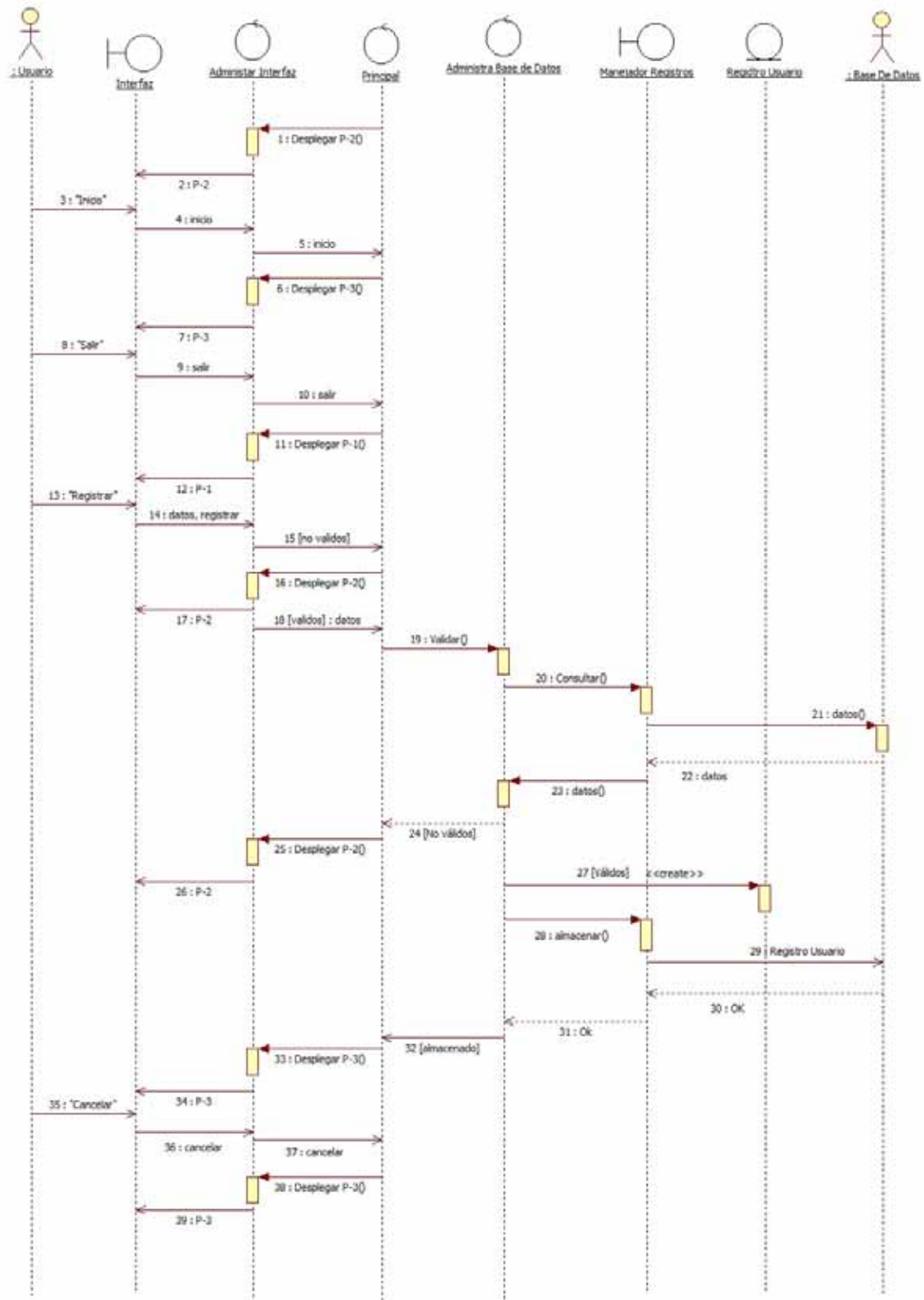
### Sub-flujo:

- ✓ (S-1) Se despliega la pantalla (P-1) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-2) Se despliega la pantalla (P-2) y se ejecuta el caso de uso Validar Usuario.
- ✓ (S-3) Se Verifica que los datos sean válidos (E-1, E-2), Se almacenan en la Base de Datos de Registros se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-4) Se despliega la pantalla principal (P-1) se ejecuta el caso de Uso Validar Usuario.

### Excepciones:

- ✓ (E-1) En caso de que alguno de los campos no sea llenado o sea llenado de manera incorrecta, se mostrara un mensaje en la misma pantalla (P-2).
- ✓ (E-2) Se debe verificar que el login elegido por el usuario no se encuentre ya registrado en el sistema. En caso de estarlo, es necesario indicar al usuario que introduzca otro login







## Ofrecer Servicios

### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Proveedor

### Propósito:

- ✓ Que el usuario elija el servicio que desea

### Precondiciones:

- ✓ El usuario debe haber sido validado previamente en el sistema

### Post-condiciones:

- ✓ Se presentara un pantalla diferente dependiendo de la acción

### Flujo Principal:

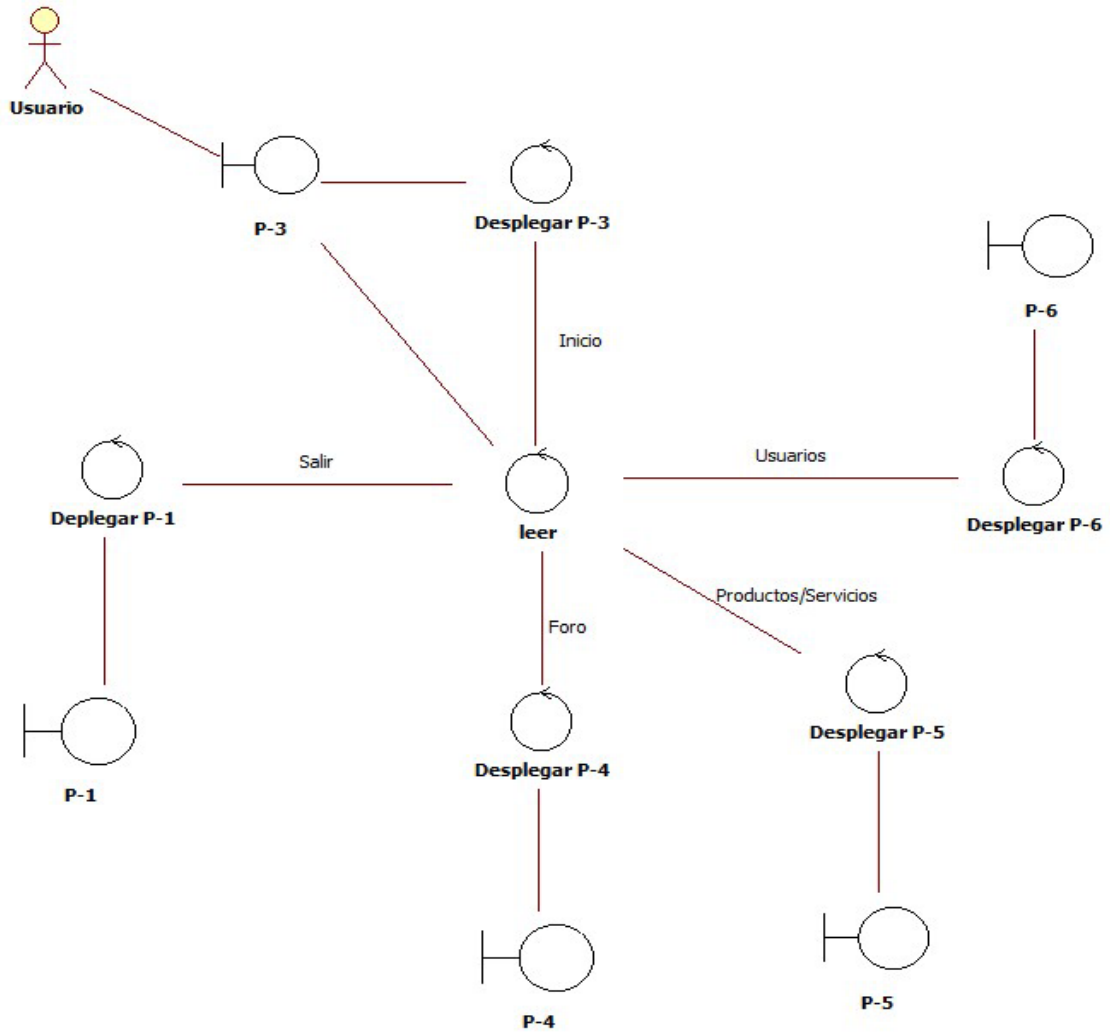
- ✓ Se muestra la pantalla (P-3), en donde el usuario puede elegir entre "Inicio" (S-1), "Salir" (S-2) "Contactos" (S-3), "Productos y Servicios" (S-4) o "Foro" (S-5),

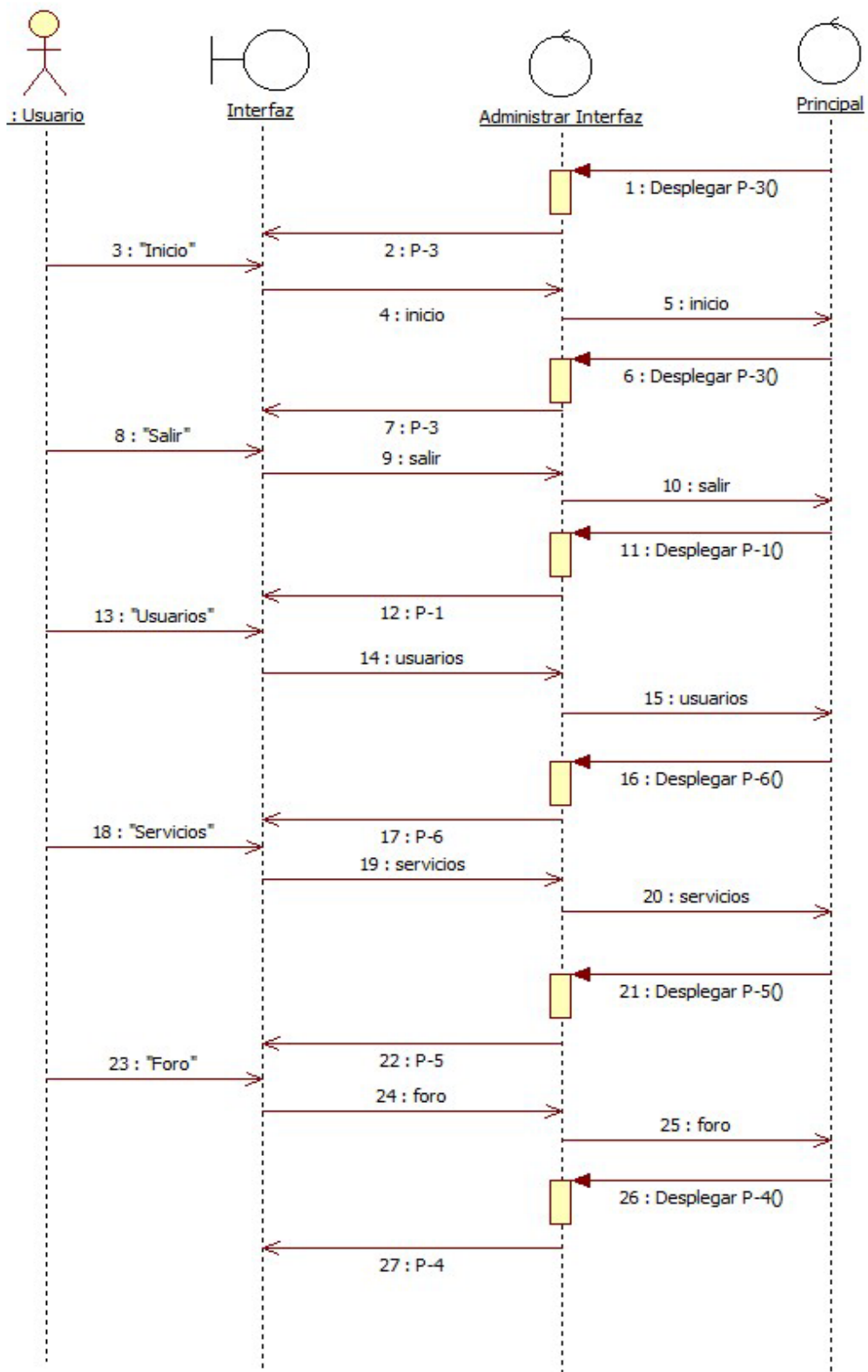
### Sub-flujo:

- ✓ (S-1) Se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios
- ✓ (S-2) Se despliega la pantalla (P-1) y se ejecuta el caso de uso Validar Usuario.
- ✓ (S-3) Se despliega la pantalla (P-6) y se ejecuta el caso de uso Administra Usuarios
- ✓ (S-4) Se despliega la pantalla (P-5) y se ejecuta el caso de uso Administrar Productos y Servicios.
- ✓ (S-5) Se despliega la pantalla (P-4), y se ejecuta el caso de uso Administrar Foro.

### Excepciones:

- ✓ Ninguna





## Administrar Usuarios

### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Vendedor
- ✓ Base de Datos de Registro

### Propósito:

- ✓ Que el usuario pueda verificar, modificar sus datos o, eliminar a un usuario.

### Precondiciones:

- ✓ El usuario debe existir en el sistema y haber sido validado previamente

### Post-condiciones:

- ✓ Se almacenaran los datos del usuario, ya sea que se hayan sido modificados o no

### Flujo Principal:

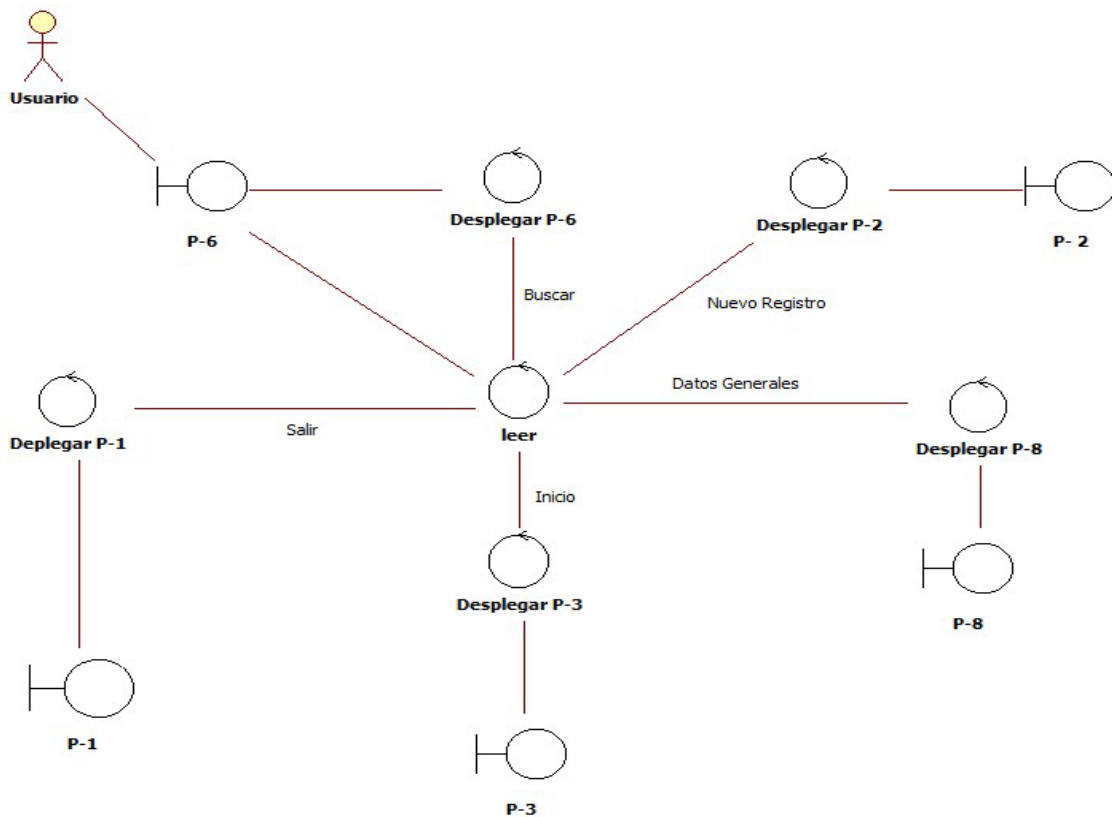
- ✓ Se muestra la pantalla (P-6), con un listado de usuarios que se encuentran en el sistema. El usuario puede elegir entre "Inicio" (S-1), "Salir" (S-2), "Datos Generales" (S-3), ingresar los datos (E-1) y presionar Buscar (S-4), Seleccionar un Usuario (S-5) o "Agregar Usuario" (S-6).

### Sub-flujo:

- ✓ (S-1) Se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-2) Se despliega la pantalla (P-1) y se ejecuta el caso de uso Validar Usuario.
- ✓ (S-3) Se despliega la pantalla (P-8). El usuario puede modificar sus datos en el sistema. Puede seleccionar entre "Actualizar" (S-7) o "Cancelar" (S-8).
- ✓ (S-4) El sistema valida los datos introducidos por el usuario (E-2), se despliega la pantalla (P-6)
- ✓ (S-5) Se despliega a pantalla (P-8). El usuario puede ver o modificar los datos de un usuario. Puede seleccionar entre "Actualizar" (S-7), "Cancelar" (S-8) o "Eliminar" (S-9)
- ✓ (S-6) Se despliega la pantalla (P-2) y se ejecuta el caso de uso Registrar Usuario.
- ✓ (S-7) Se verifican que los datos sean validos (E-1), se almacenan en la Base de Datos de registro, se despliega la pantalla (P-6)
- ✓ (S-8) Se despliega la pantalla (P-6).
- ✓ (S-9) Los datos del usuario son eliminados del sistema

Excepciones:

- ✓ (E-1) En caso de que el campo no sea llenado o sea llenado de manera incorrecta, se mostrara un mensaje en la misma pantalla (P-6).
- ✓ (E-2) En caso de que los datos introducidos no existan en la Base de Datos, se mostrara un mensaje en la misma pantalla (P-6).



## Administrar Productos

### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Vendedor
- ✓ Base de Datos

### Propósito:

- ✓ Que el usuario puede verificar, modificar los datos de un producto, o eliminar un producto.

### Precondiciones:

- ✓ El usuario debe existir en el sistema y haber sido validado previamente

### Post-condiciones:

- ✓ Se almacenan los datos del producto, ya sea que hayan sido modificados o no

### Flujo Principal:

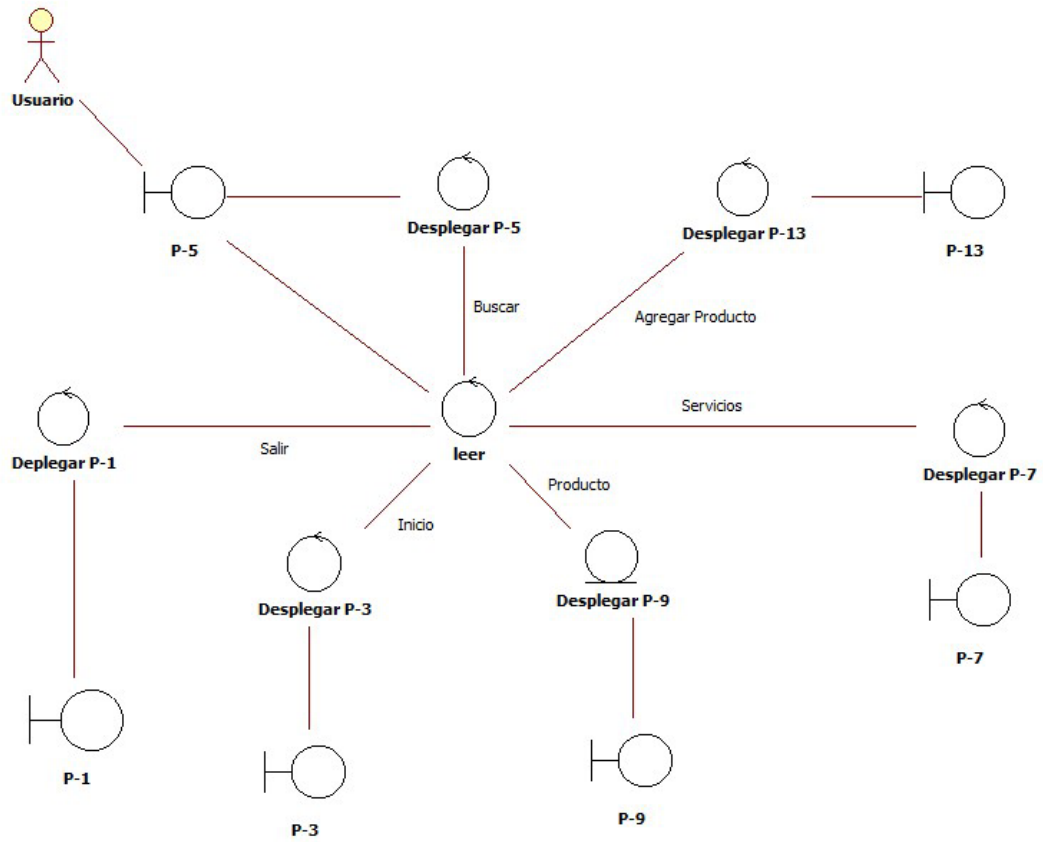
- ✓ Se muestra la pantalla (P-5), con un listado de productos que se encuentran en el sistema. El usuario puede entre "Inicio" (S-1), "Salir" (S-2), "Servicios" (S-3), ingresar los datos (E-1) y presionar Buscar (S-4), Seleccionar un Producto (S-5) o "Agregar Producto" (S-6).

### Sub-flujo:

- ✓ (S-1) Se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-2) Se despliega la pantalla (P-1) y se ejecuta el caso de uso Validar Usuario.
- ✓ (S-3) Se despliega la pantalla (P-7) y se ejecuta el caso de uso Administrar Servicios
- ✓ (S-4) El sistema valida los datos introducidos por el usuario (E-2), se despliega la pantalla (P-5)
- ✓ (S-5) Se despliega la pantalla (P-9), el usuario puede ver, modificar o eliminar los datos del producto. Puede seleccionar entre "Actualizar" (S-7), "Eliminar" (S-8), "Imprimir" (S-9) o "Cancelar" (S-10)
- ✓ (S-6) Se despliega la pantalla (P-13) y se ejecuta el caso de uso Registrar Producto
- ✓ (S-7) Se verifican que los datos sean validos (E-2), se almacenan en la Base de Datos, se despliega la pantalla (P-5)
- ✓ (S-8) Los datos del producto son eliminados del sistema
- ✓ (S-9) Se imprime la información detalla del Producto
- ✓ (S-10) Se despliega la pantalla (P-5), y se ejecuta el caso de uso Administrar Productos

### Excepciones:

- ✓ (E-1) En caso de que el campo no sea llenado o sea llenado de manera incorrecta, se mostrara un mensaje en la misma pantalla (P-5).
- ✓ (E-2) En caso de que los datos introducidos no existan en la Base de Datos, se mostrara un mensaje en la misma pantalla (P-5).



## Administrar Servicios

### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Vendedor
- ✓ Base de Datos de Registro

### Propósito:

- ✓ Que el usuario pueda verificar, modificar los datos de un servicio, o dar por terminado un servicio.

### Precondiciones:

- ✓ El usuario debe existir en el sistema y haber sido validado previamente

### Post-condiciones:

- ✓ Se almacenaran los datos del servicio, ya sea que se hayan sido modificados o no

### Flujo Principal:

- ✓ Se muestra la pantalla (P-7), con un listado de usuarios que se encuentran en el sistema. El usuario puede elegir entre "Inicio" (S-1), "Salir" (S-2), "Productos" (S-3), ingresar los datos (E-1) y presionar Buscar (S-4), Seleccionar un Servicio (S-5) o "Agregar Servicio" (S-6).

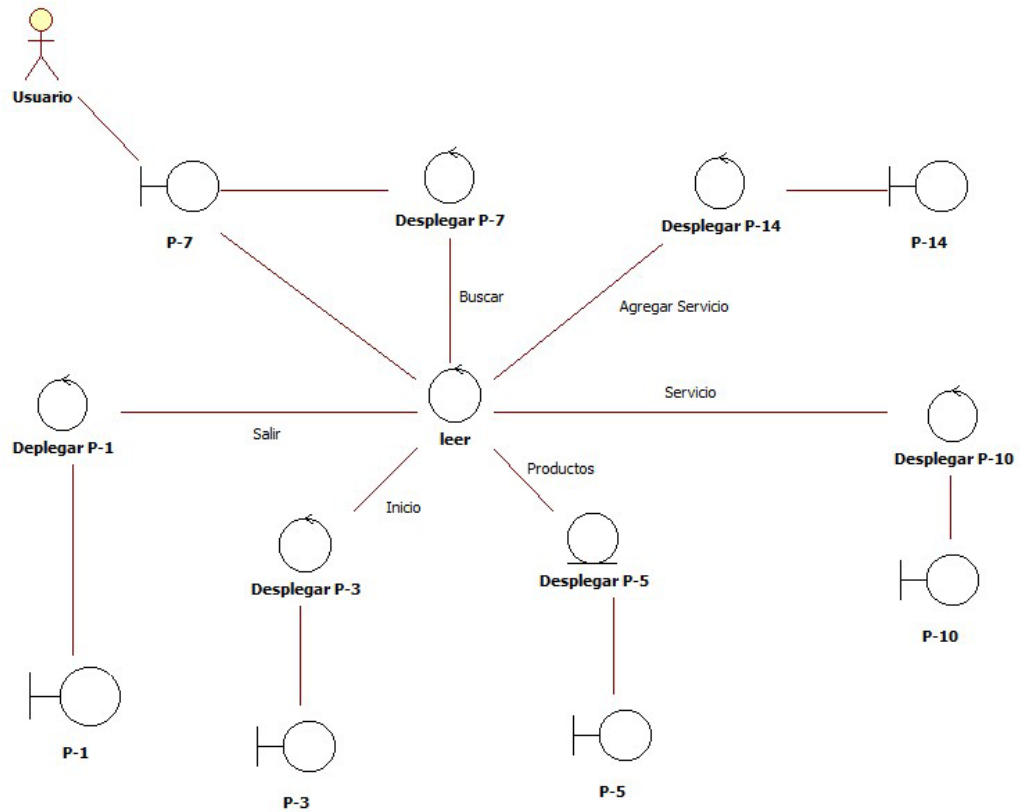
### Sub-flujo:

- ✓ (S-1) Se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-2) Se despliega la pantalla (P-1) y se ejecuta el caso de uso Validar Usuario.
- ✓ (S-3) Se despliega la pantalla (P-5) y se ejecuta el caso de uso Administrar Productos.
- ✓ (S-4) El sistema valida los datos introducidos por el usuario (E-2), se despliega la pantalla (P-7)
- ✓ (S-5) Se despliega la pantalla (P-10), el usuario puede ver, modificar o eliminar los datos del servicio. Puede seleccionar entre "Actualizar" (S-7), "Eliminar" (S-8), "Imprimir" (S-9) o "Cancelar" (S-10)
- ✓ (S-6) Se despliega la pantalla (P-14) y se ejecuta el caso de uso Registrar Servicio
- ✓ (S-7) Se verifican que los datos sean validos (E-2), se almacenan en la Base de Datos, se despliega la pantalla (P-7)
- ✓ (S-8) Los datos del servicio son eliminados del sistema
- ✓ (S-9) Se imprime la información detallada del Servicio
- ✓ (S-10) Se despliega la pantalla (P-7), y se ejecuta el caso de uso Administrar Servicio



Excepciones:

- ✓ (E-1) En caso de que el campo no sea llenado o sea llenado de manera incorrecta, se mostrara un mensaje en la misma pantalla (P-7).
- ✓ (E-2) En caso de que los datos introducidos no existan en la Base de Datos, se mostrara un mensaje en la misma pantalla (P-7).



## Administrar Foro

### Actores:

- ✓ Administrador
- ✓ Cliente
- ✓ Vendedor
- ✓ Base de Datos

### Propósito:

- ✓ Que el usuario pueda comunicarse de forma asíncrona con diferentes usuarios

### Precondiciones:

- ✓ El usuario debe existir en el sistema y haber sido validado en el sistema

### Post-condiciones:

- ✓ El usuario creara un nuevo tema en el foro, o agregara un nuevo comentario a un tema ya existente

### Flujo Principal

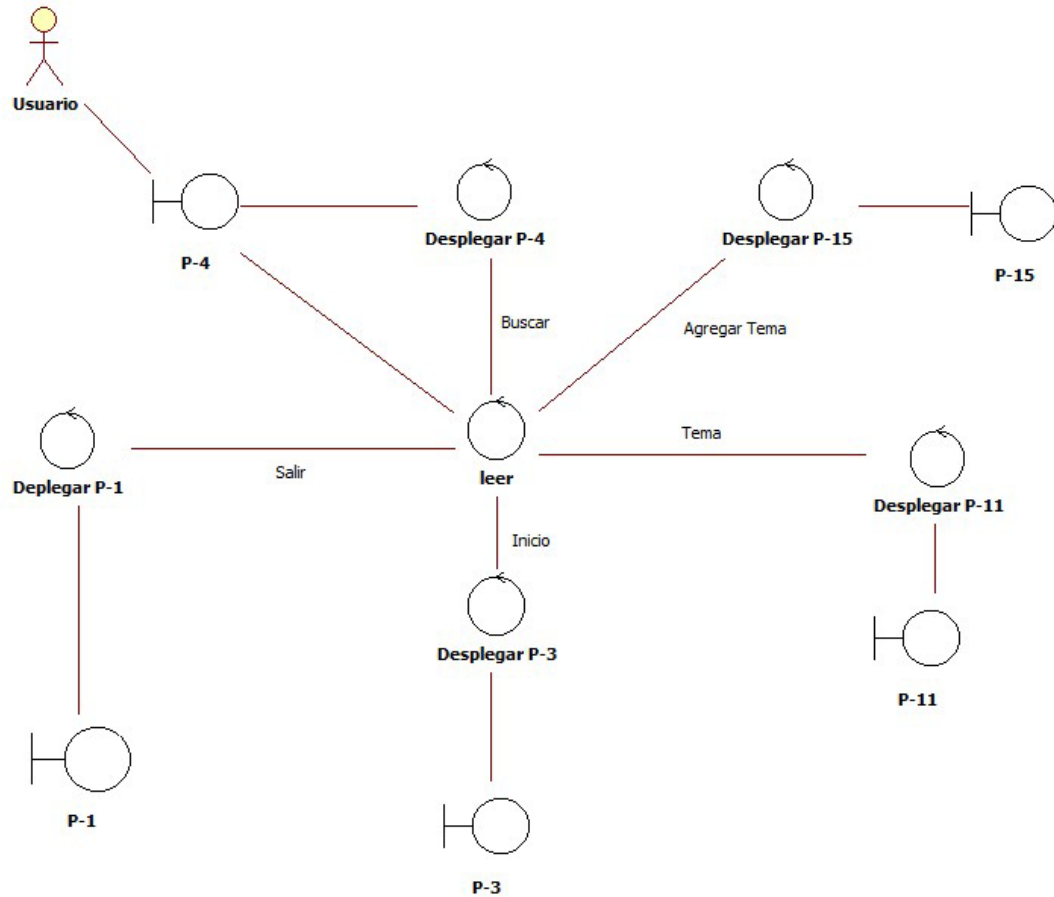
- ✓ Se muestra la pantalla (P-4), con un listado de los temas que se encuentran el sistema. El usuario puede elegir entre "Inicio" (S-1), "Salir" (S-2), Ingresar los datos (E-1) y presionar buscar (S-3), Seleccionar un Tema (S-4) o "Agregar Tema" (S-5).

### Sub-flujo:

- ✓ (S-1) Se despliega la pantalla (P-3) y se ejecuta el caso de uso Ofrecer Servicios.
- ✓ (S-2) Se despliega la pantalla (P-1) y se ejecuta el caso de uso Validar Usuario.
- ✓ (S-3) El sistema valida los datos introducidos por el usuario (E-2), se despliega la pantalla (P-4)
- ✓ (S-4) Se despliega la pantalla (P-11), el usuario puede ver, modificar, agregar comentario o eliminar los datos de un tema del foro. Puede seleccionar entre "Actualizar" (S-6), "Eliminar" (S-7), "Agregar Comentario" (S-8) o "Cancelar" (S-9)
- ✓ (S-5) Se despliega la pantalla (P-15) y se ejecuta el caso de uso Registrar Tema
- ✓ (S-6) Se verifican que los datos sean validos (E-2), se almacenan en la Base de Datos, se despliega la pantalla (P-4)
- ✓ (S-7) El tema es eliminado del sistema
- ✓ (S-8) Se despliega la pantalla (P-16). El usuario ingresa los datos (E-1). Puede seleccionar entre "Agregar" (S-5) y Cancelar (S-5)
- ✓ (S-9) Se despliega la pantalla (P-7), y se ejecuta el caso de uso Administrar Foro

Excepciones:

- ✓ (E-1) En caso de que el campo no sea llenado o sea llenado de manera incorrecta, se mostrara un mensaje en la misma pantalla.
- ✓ (E-2) En caso de que los datos introducidos no existan en la Base de Datos, se mostrara un mensaje en la misma pantalla (P-7).




## Pantalla Validar Usuario

Inicio de Sesión	
Usuario:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Aceptar"/> <input type="button" value="Reset"/>	

## Pantalla Principal

Inicio | Usuario | Productos | Servicios | Filtros | Cerrar Sesión

CRM (Customer Relationship Management) es el conjunto de estrategias de negocio, marketing, comunicación e infraestructuras tecnológicas, diseñadas con el objetivo de construir una relación duradera con los clientes, identificando, comprendiendo y satisfaciendo sus necesidades.



CUSTOMER RELATIONSHIP  
MANAGEMENT

## Pantalla Clientes

Inicio Usuarios Productos Servicios Fact. Carr. Sección

Clientes

Nombre	Apellido Paterno	Apellido Materno	Tipo Usuario	Mail	Username	RFC	Celular
Gerardo	Hernandez	Hernandez	Administrador	gerardom@gmail.com	gerardom	8851084081896	344551984712
Juan Francisco	Rojas	Sanchez	Proveedor	gerardom@gmail.com	rojas	8851084081896	344551984712
Daniel	Castro	Monter	Cliente	gerardom@gmail.com	castro	8851084081896	344551984712
Haydee	Corzo	Zamudio	Cliente	haydey1@gmail.com	zamudio	022485218	12345678

Page 1 of 1

## Pantalla Productos

Inicio Usuarios Productos Servicios Fact. Carr. Sección

Listado Productos

Producto	Código	Precio	Descuento	Tipo venta	Fabricante	Proveedor
Producto con nombre	Producto con id	\$0	0%	Pista	Hp	Juan Francisco Rojas Sanchez
Producto 1	9846	\$100	10%	Pista	Hp	Juan Francisco Rojas Sanchez
Producto 5	123456	\$200	25%	Pista	Hp	Juan Francisco Rojas Sanchez
Producto 2 Actualizar	123456	\$0	0%	Pista	Dell	Juan Francisco Rojas Sanchez

Page 1 of 1

## Pantalla Servicios

Inicio Usuarios Productos Servicios Foro Control de Acceso

Lista de Servicios

Cliente	Proveedor	Estatus	Prioridad	Fecha	Servicio	Modificado por	Ultima Modifica...
Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	07/21/2012	Baja de Usuario	Gerardo Mendoza Manzano	07/21/2012
Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Nuevo	Baja	09/04/2013	Cambio de PC	Gerardo Mendoza Manzano	09/04/2013
Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	09/04/2013	Baja de Usuario	Gerardo Mendoza Manzano	11/05/2013
Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Asignado	Alta	10/03/2013	Cambio de PC	Juan Francisco Rojas Sanchez	12/03/2013
Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	07/21/2012	Alta de Usuario	Gerardo Mendoza Manzano	10/03/2013
Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Resuelto	Medio	07/28/2012	Cambio de PC	Gerardo Mendoza Manzano	12/07/2013
Héctor Carlos Zamudio	Juan Francisco Rojas Sanchez	Asignado	Alta	12/06/2013	Cambio de PC	Gerardo Mendoza Manzano	12/06/2013
Daniel Castellón Morales	Gerardo Mendoza Manzano	Nuevo	Alta	12/03/2013	Baja de Usuario	Daniel Castellón Morales	12/03/2013

Page 1 of 2

## Pantalla Foro

Inicio Usuarios Productos Servicios Foro Control de Acceso

Temas

Tema	Categoría	Fecha	Usuario
Actualización 3	Hardware	02/14/2012	Juan Francisco Rojas Sanchez
Nueva Prueba	Hardware	05/08/2012	Daniel Castellón Morales
Prueba HTSL	Hardware	10/03/2013	Gerardo Mendoza Manzano
Tema 5	Hardware	02/08/2012	Daniel Castellón Morales
Tema 8	Hardware	12/07/2013	Gerardo Mendoza Manzano

Page 1 of 1

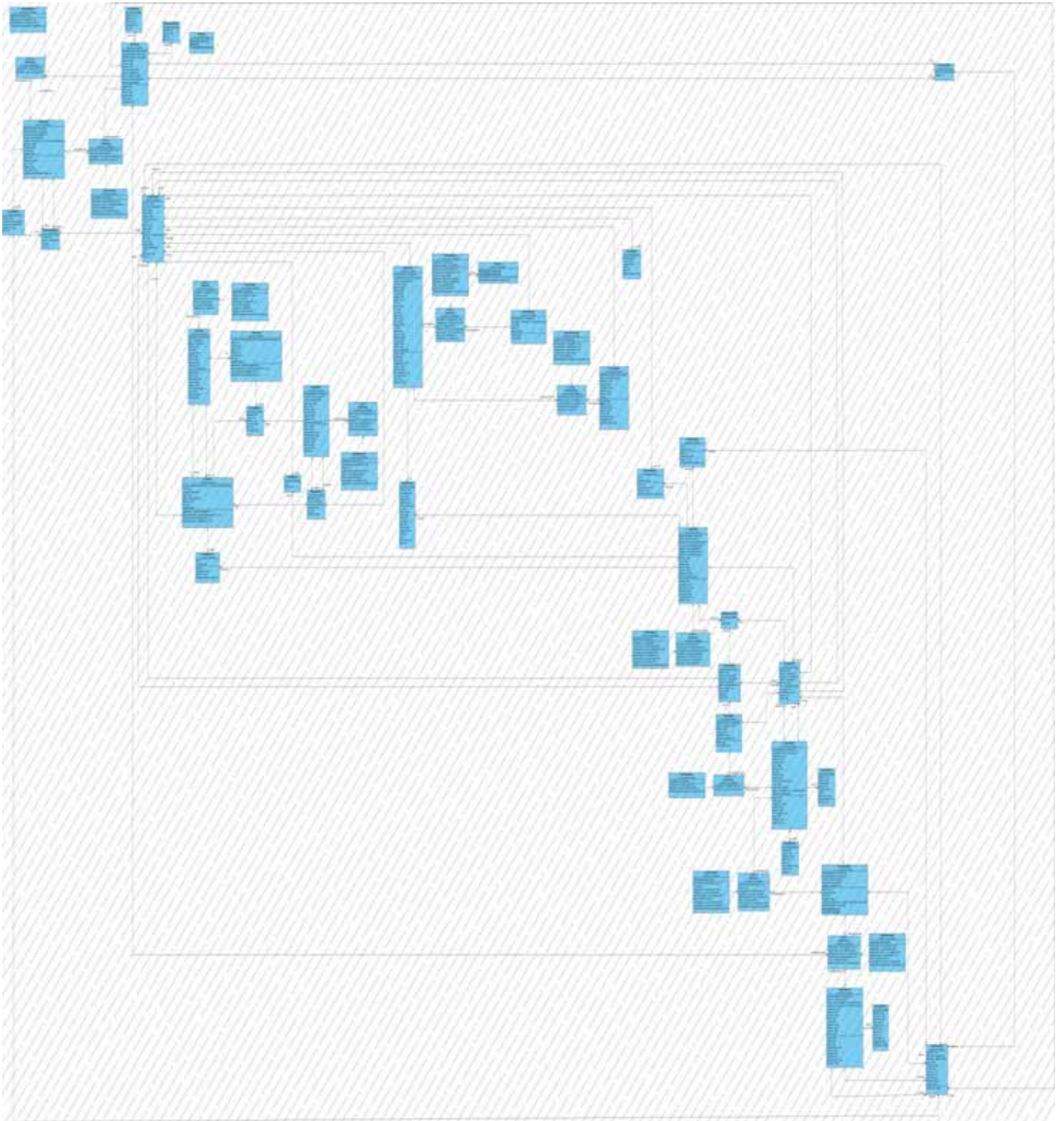
## Diagraman de Clases

- ❖ mx.uam.azc.cbi.action
  - CarritoAction.java
  - CombosAction.java
  - CredencialAction.java
  - DocumentosAction.java
  - GestionCMRAction.java
  - MensajesAction.java
  - OrdenesAction.java
  - ProductosAction.java
  - ServiciosAction.java
  - TemasAction.java
  - TiempoAction.java
  - UsuariosActios.java
  
- ❖ mx.uam.azc.cbi.data
  - CarritoComprasDTO.java
  - CatCategoriasDTO.java
  - CatEstadoServiciosDTO.java
  - CatEstadosDTO.java
  - CatFabricanteDTO.java
  - CatServiciosDTO.java
  - CatTipoUsuarioDTO.java
  - CompraSaveDTO.java
  - CompraJoinDTO.java
  - CredencialDTO.java
  - MensajeJoinDTO.java
  - MensajeSaveDTO.java
  - OrdenComprasSave.java
  - OrdenComprasJoin.java
  - ProductoJoinDTO.java
  - ProductoSaveDTO.java
  - ServicioJoinDTO.java
  - ServicioSaveDTO.java
  - TiemposJoinDTO.java
  - TiemposSaveDTO.java
  - TemaJoinDTO.java
  - TemaSaveDTO.java
  - UsuarioJoinDTO.java
  - UsuarioSaveDTO.java
  
- ❖ mx.uam.azc.cbi.interfaces
  - AdminCombos.java
  - AdminCompras.java
  - AdminCredenciales.java
  - AdminMensajes.java
  - AdminOrdenes.java
  - AdminProductos.java
  - AdminServicios.java
  - AdminTemas.java
  - AdminTiempos.java
  - AdminUsuarios.java
  
- ❖ mx.uam.azc.cbi.logica
  - Encryption.java
  - Timestamp.java
  
- ❖ mx.uam.azc.cbi.metodos
  - AdminCombosImpl.java
  - AdminComprasImpl.java

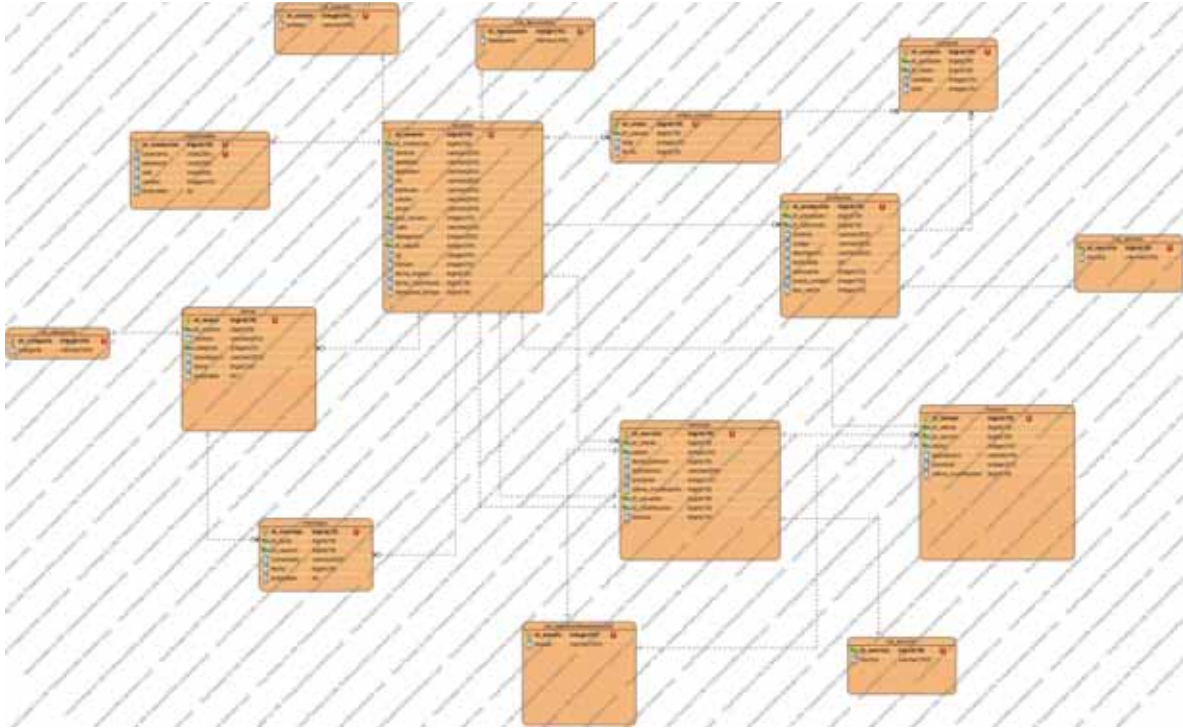
- AdminCredencialesImpl.java
- AdminMensajesImpl.java
- AdminOrdenesImpl.java
- AdminProductosImpl.java

- AdminServiciosImpl.java
- AdminTemasImpl.java
- AdminTiemposImpl.java
- AdminUsuariosImpl.java





## Estructura de la base de datos



## Diccionario de datos

Tabla: credenciales

Descripción: Contiene las credenciales de acceso de los usuarios

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_credencial	bigint	Llave primaria autoincremental	Identificador	
username	varchar(255)		Nombre de usuario	
password	varchar(255)		Contraseña	
salt	varchar(255)			
versión	integer			
isobsolete	boolean		Indica si un usuario se encuentra activo	

Tabla: usuarios

Descripción: Contiene los datos generales de los usuarios

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_usuario	bigint	Llave primaria autoincremental	Identificador	
id_credencial	bigint	Llave foránea	Id credencial	credenciales
nombre	varchar(255)		Nombre del usuario	
apellidop	varchar(255)		Apellido paterno	
apellidom	varchar(255)		Apellido Materno	
rfc	varchar(255)		Rfc del usuario	
particular	varchar(255)		Numero de teléfono particular	
celular	varchar(255)		Numero de celular del usuario	
email	varchar(255)		Email del usuario	
tipo_usuario	integer	Llave foránea	Tipo de usuario 1. Administrador 2. Cliente 3. Proveedor	cat_tipousuario
calle	varchar(255)		Calle	
colonia	varchar(255)		Colonia	
delegación	varchar(255)		Delegación	
cp	integer		Código postal	
entidad	integer	Llave foránea	Estado de la república	cat_estados
versión	integer			
fecha_ingreso	bigint		Fecha en la que fue dado de alta el usuario en el sistema	
fecha_nacimiento	bigint		Fecha de nacimiento del usuario	

Tabla: temas

Descripción: Contiene los temas del módulo de foros

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_temas	bigint	Llave primaria autoincremental	Identificador	
id_usuario	bigint	Llave foránea	Id del usuario	usuarios
nombre			Nombre del tema	
categoría	integer	Llave foránea	Categoría: 1. General 2. Hardware 3. Software	cat_categorias
descripción	varchar(255)		Descripción del tema	
fecha	bigint		Fecha que se agrego el tema	
isobsolete	boolean		Indica si el tema fue eliminado	

Tabla: mensajes

Descripción: Contiene los mensajes de los que se publican en los temas

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_mensaje	bigint	Llave primaria autoincremental	Identificador	
id_temas	bigint	Llave foránea	Id temas	temas
id_usuario	bigint	Llave foránea	Id usuario	usuarios
comentario	varchar(255)		Comentario	
fecha	bigint		Fecha en que fue agregado el comentario	
isobsolete	boolean		Indica si el comentario fue eliminado	

Tabla: productos

Descripción: Contiene el catálogo de los productos dados de alta en el sistema

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_productos	bigint	Llave primaria autoincremental	Identificador	
id_proveedor	bigint	Llave foránea	Id proveedor	usuarios
id_fabricante	bigint	Llave foránea	Id fabricante	cat_fabricantes
nombre	varchar(255)		Nombre del producto	
código	varchar(255)		Código del producto	
descripción	varchar(255)		Descripción del producto	
isobsolete	boolean		Indica si el producto fue	

			eliminado	
descuento	integer		Descuento del producto	
precio_unitario	integer		Precio unitario del producto	
tipo_venta	integer		Tipo de venta del producto	

Tabla: orden\_compra

Descripción: Contiene las órdenes de compra de los usuarios

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_orden	bigint	Llave primaria autoincremental	Identificador	
id_usuario	bigint	Llave foránea	Id usuario	usuarios
total	double		Total de compra	
fecha	bigint		Fecha de compra	

Tabla: compras

Descripción: Contiene los productos de las órdenes de compra de los usuarios

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_compra	bigint	Llave primaria autoincremental	Identificador	
id_producto	bigint	Llave foránea	Id producto	productos
id_orden	bigint	Llave foránea	Id orden	orden_compra
cantidad	integer		Número de productos	
total	double		Total por productos	

Tabla: servicios

Descripción: Contiene el estatus del

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_servicio	bigint	Llave primaria autoincremental	Identificador	
id_cliente	bigint	Llave foránea	Id cliente	usuarios
id_proveedor	bigint	Llave foránea	Id proveedor	usuarios
id_modificacion	bigint	Llave foránea	Indica quien realizo la última modificación	usuarios
id_tipo_servicio	bigint	Llave foránea		cat_servicio
status	integer	Llave foránea	Estatus del servicio	cat_estadoservicio
fecha_peticion	bigint		Fecha en la que se solicito el servicio	
descripcion	varchar(255)		Descripción del servicio	

prioridad	integer		Prioridad del servicio	
ultima_modificacion	bigint		Fecha de la última modificación	

Tabla: tiempos

Descripción: Contiene un historial de los servicios proporcionados a los usuarios

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_tiempo	bigint	Llave primaria autoincremental	Identificador	
id_servicio	bigint	Llave foránea	Id servicio	servicios
id_proveedor	bigint	Llave foránea	Id proveedor	usuarios
id_modificacion	bigint	Llave foránea	Indica quien realizo la última modificación	usuarios
Status	integer	Llave foránea	Estatus del servicio	cat_estadoservicio
Descripción	varchar(255)		Descripción del servicio	
Prioridad	integer		Prioridad del servicio	
ultima_modificacion	bigint		Fecha de modificación del servicio	

Tabla: cat\_categorias

Descripción: Catalogo de categorías de los temas

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_categoria	integer	Llave primaria autoincremental	Identificador	
Categoría	varchar(100)		Descripción	

Tabla: cat\_estadoservicio

Descripción: Catalogo de los estados de servicio

Campo	Tipo	Propiedad	Descripción	Tabla foránea
id_estado	integer	Llave primaria autoincremental	Identificador	
Estados	varchar(100)		Descripción	

Tabla: cat\_estados

Descripción: Catalogo de estados

<b>Campo</b>	<b>Tipo</b>	<b>Propiedad</b>	<b>Descripción</b>	<b>Tabla foránea</b>
id_estado	integer	Llave primaria autoincremental	Identificador	
Estado	varchar(100)		Descripción	

Tabla: cat\_fabricantes

Descripción: Catalogo de fabricantes

<b>Campo</b>	<b>Tipo</b>	<b>Propiedad</b>	<b>Descripción</b>	<b>Tabla foránea</b>
id_fabricante	integer	Llave primaria autoincremental	Identificador	
Fabricante	varchar(100)		Descripción	

Tabla: cat\_tipousuario

Descripción: Catalogo tipo de usuarios

<b>Campo</b>	<b>Tipo</b>	<b>Propiedad</b>	<b>Descripción</b>	<b>Tabla foránea</b>
id_tipousuario	integer	Llave primaria autoincremental	Identificador	
Tipousuario	varchar(100)		Descripción	

Tabla: cat\_servicios

Descripción: Catálogo de servicios ofrecidos

<b>Campo</b>	<b>Tipo</b>	<b>Propiedad</b>	<b>Descripción</b>	<b>Tabla foránea</b>
id_servicio	integer	Llave primaria autoincremental	Identificador	
servicio	varchar(100)		Descripción	

## Anexo 1 Manual de usuario

En la pantalla del navegador se mostrará un formulario solicitando un nombre de usuario y contraseña, ingresar el nombre de usuario y contraseña y dar clic en “Aceptar”



The image shows a login form with a blue header bar containing the text "Inicio de Sesión". Below the header, there are two input fields: "Usuario:" and "Password:". Below the input fields, there are two buttons: "Aceptar" and "Reset".

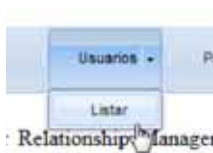
### Administrar Usuarios

Si los datos ingresados son correctos se desplegara la pantalla principal (), en caso contrario deberás volver ingresar tus datos ()



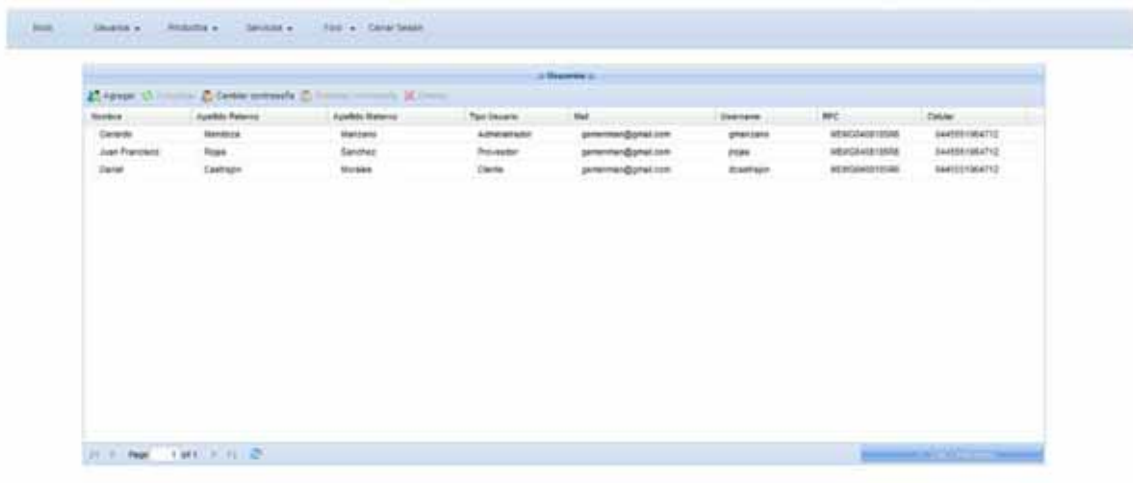
### Alta de usuarios.

Desde la pantalla principal damos clic en “Usuarios” -> “Listar” ()





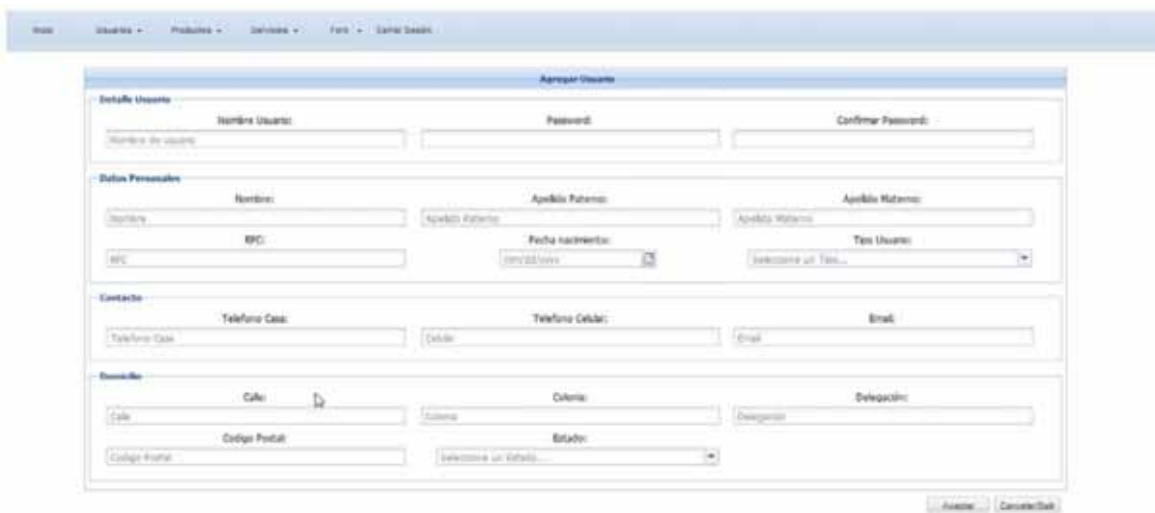
El navegador mostrará en pantalla una lista con información de los usuarios dados de alta en el sistema ()



En la parte superior de la lista hacemos clic en el botón de “Agregar” ()



El navegador mostrará en pantalla un formulario para dar de alta al usuario ().



**Nombre de usuario:** Nombre de usuario para el inicio de sesión

**Contraseña:** Contraseña para ingresar al sistema

**Confirmar Contraseña:** Confirmación de la contraseña para ingresar al sistema

**Nombre:** Nombre del recurso

**Apellido paterno:** Apellido paterno del usuario

**Apellido materno:** Apellido materno del usuario

**RFC:** RFC del usuario

**Fecha de nacimiento:** Fecha de nacimiento del usuario

**Tipo de usuario:** Tipo de usuario del sistema

**Teléfono casa:** Teléfono de casa del usuario

**Teléfono celular:** Teléfono celular del usuario

**Email:** Email del usuario

**Calle:** Domicilio del usuario

**Colonia:** Colonia del domicilio del usuario

**Delegación:** Delegación o municipio del domicilio del usuario

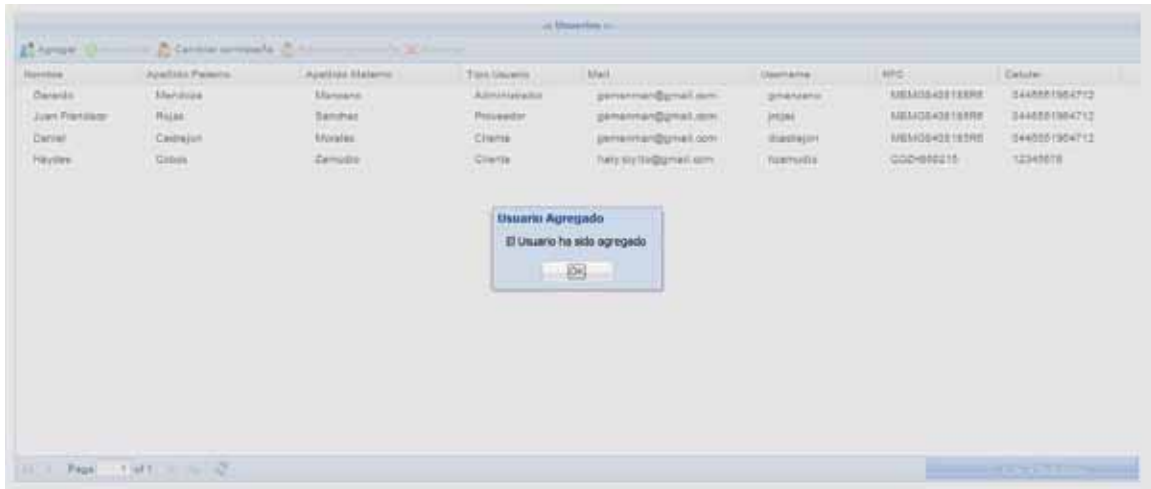
**Código postal:** Código postal del domicilio del usuario

**Entidad:** Entidad federativa del domicilio del usuario

Después de ingresar los datos solicitados por el sistema, hacemos clic en Aceptar para dar de alta al usuario ().

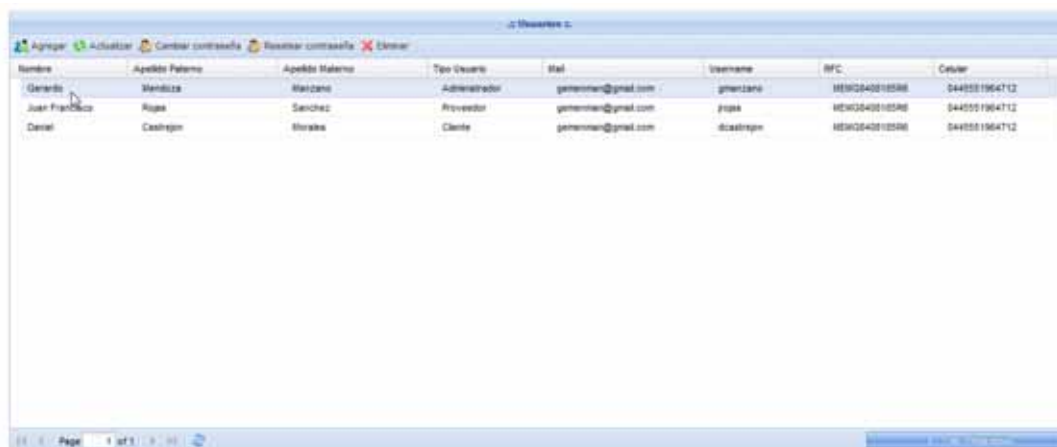


Si los datos ingresados en el sistema son correctos el navegador mostrara un mensaje diciendo que el usuario fue dado de alta, en caso contrario nos mandara un mensaje de error ()



### Actualizar usuarios

Para actualizar los datos de un usuario seleccionamos al usuario que deseamos actualizar la información (); presionamos el botón "Actualizar" de la parte superior de la pantalla que se activará al momento de seleccionar a un usuario ().



Al hacer clic en “Actualizar” el sistema nos mostrara en pantalla con un formulario con los datos del usuario que deseamos actualizar, realizamos los cambios y hacemos clic en “Aceptar” para guardar los cambios o “Cancelar” para regresar a la lista de usuarios ().

**Actualizar**

**Datos Personales**

Nombre: Gerardo Apellido Paterno: Mendosa Apellido Materno: Manzano  
RFC: HEM044018586 Fecha nacimiento: 06/06/1985 Tipo Usuario: Administrador

**Contacto**

Telefono Casa: 51127371 Telefono Celular: 9445551064712 Email: gemenman@gmail.com

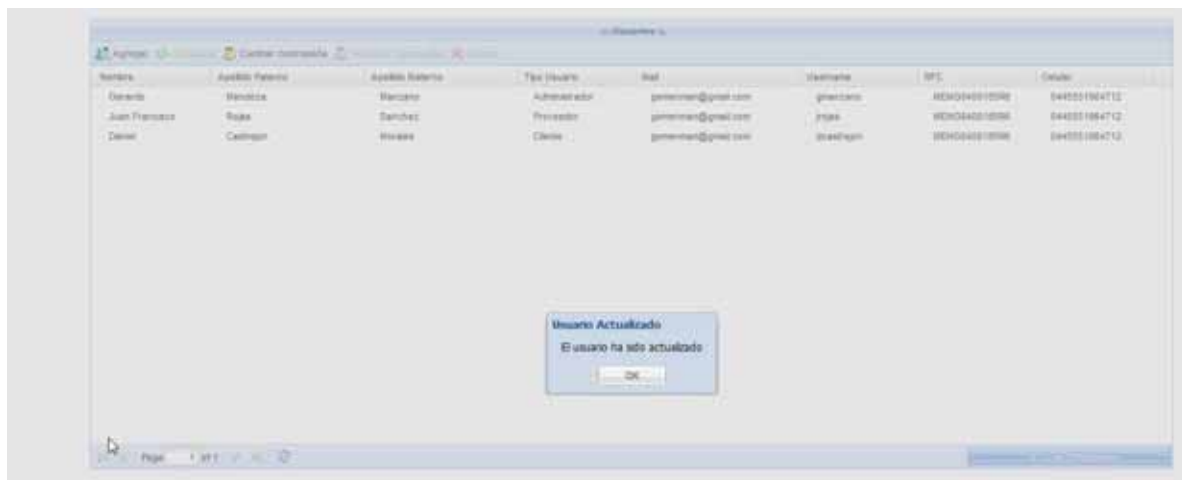
**Dirección**

Calle: Fincho López 329 Colonia: Benito Juárez Delegación: C.C. Netz  
Codigo Postal: 57000 Estado: Campeche

Aceptar Cancelar/Salir

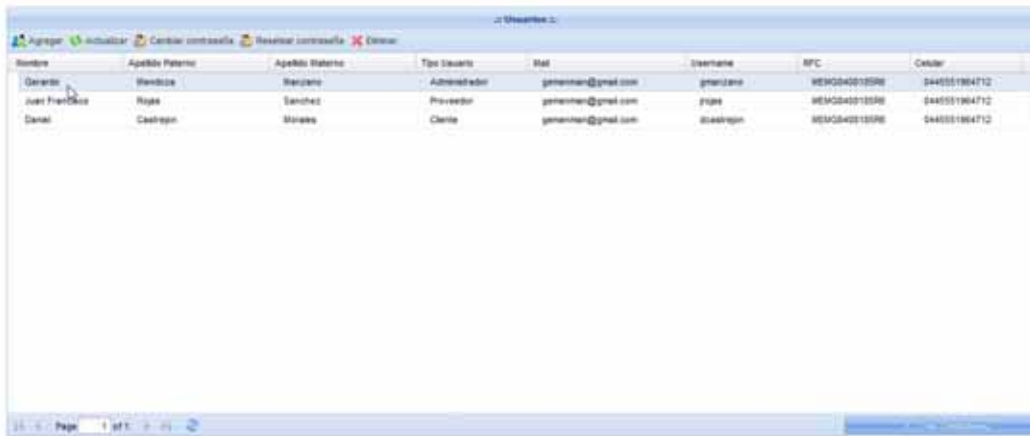


Si los datos fueron ingresados correctamente el sistema mostrará en la pantalla del navegador un mensaje de que el usuario fue actualizado ().

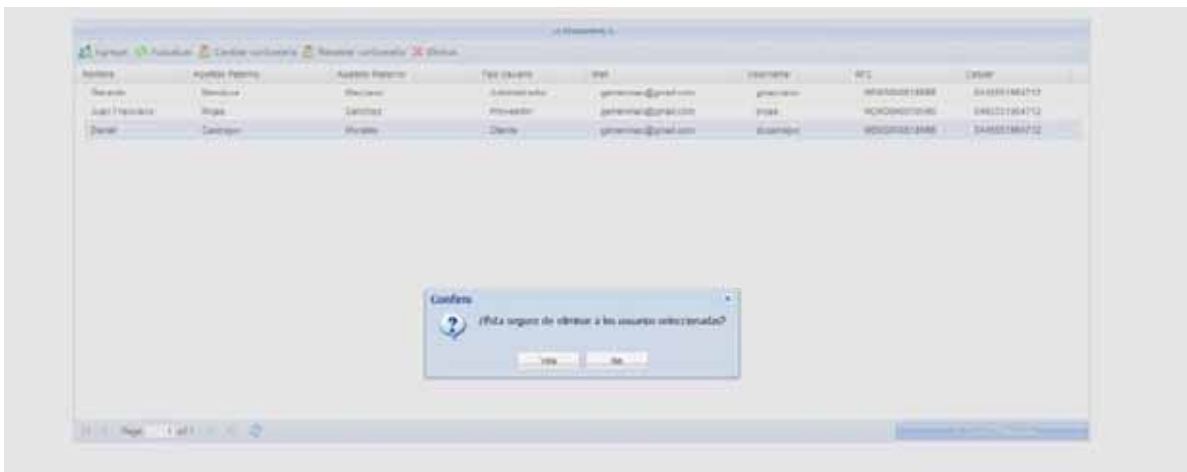


## Eliminar usuarios

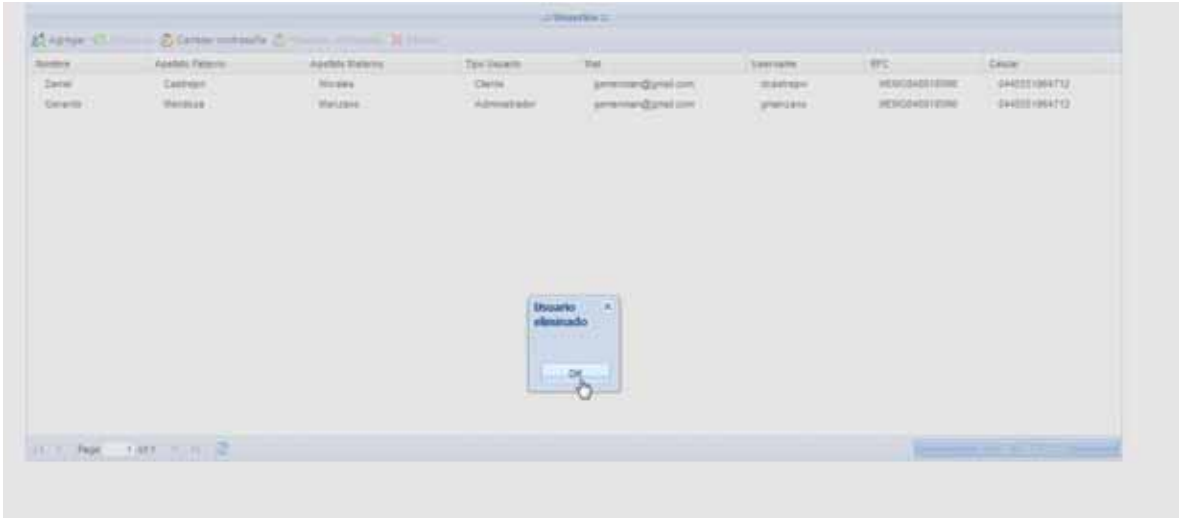
Para eliminar un usuario, seleccionamos al usuario que deseamos eliminar (); presionamos el botón de “Eliminar” de la parte superior de la pantalla que se activará al momento de seleccionar a un usuario ().



Al hacer clic en el botón de “Eliminar”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para eliminar y “No” para cancelar ().

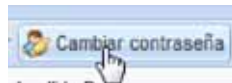


Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que el usuario fue eliminado ().



### **Cambiar Contraseña**

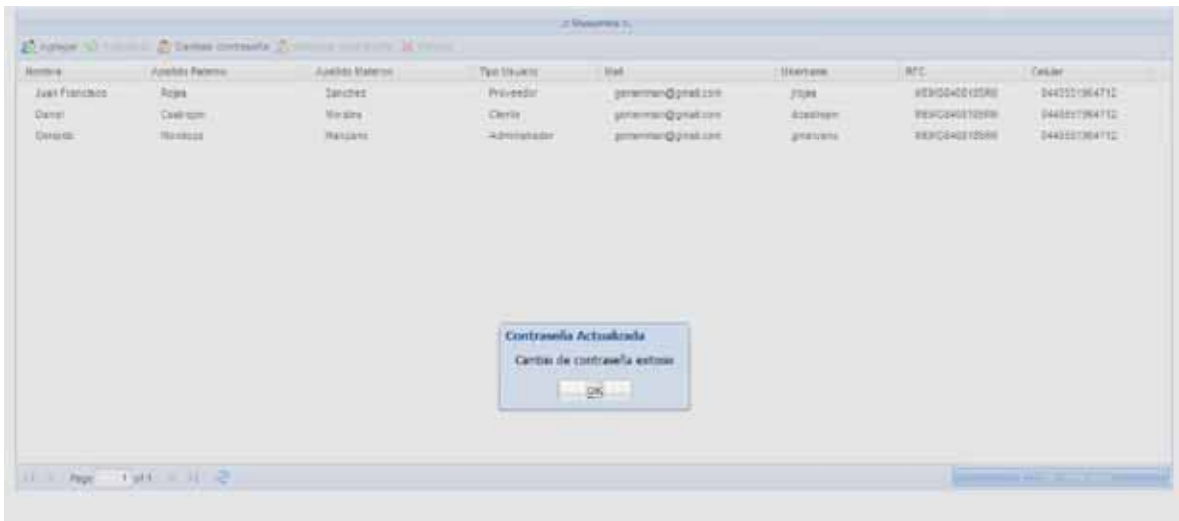
Para cambiar la contraseña del usuario que accedió al sistema, presionamos el botón “Cambiar contraseña” de la parte superior de la pantalla ().



Al hacer clic en el botón el sistema nos mostrara un formulario ingresamos la nueva contraseña y le damos clic en “Aceptar” ()

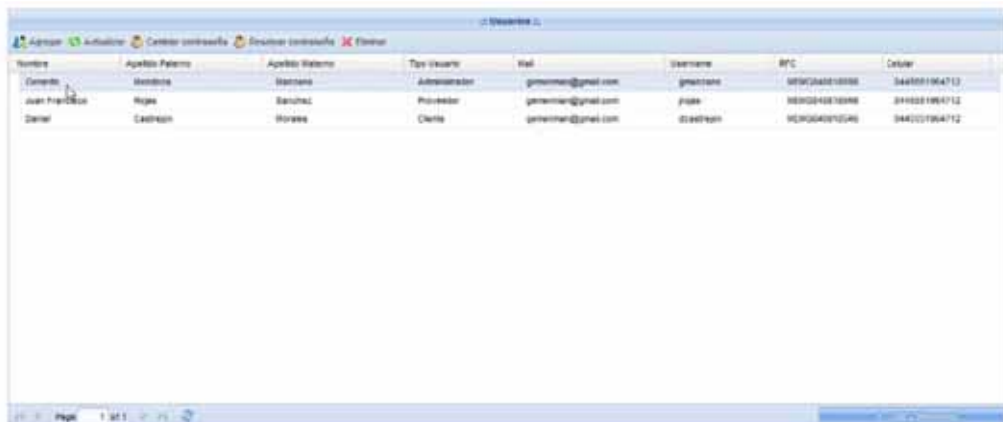


Si la información fue correcta el sistema nos mostrara en pantalla un mensaje que el cambio de contraseña fue exitoso ().

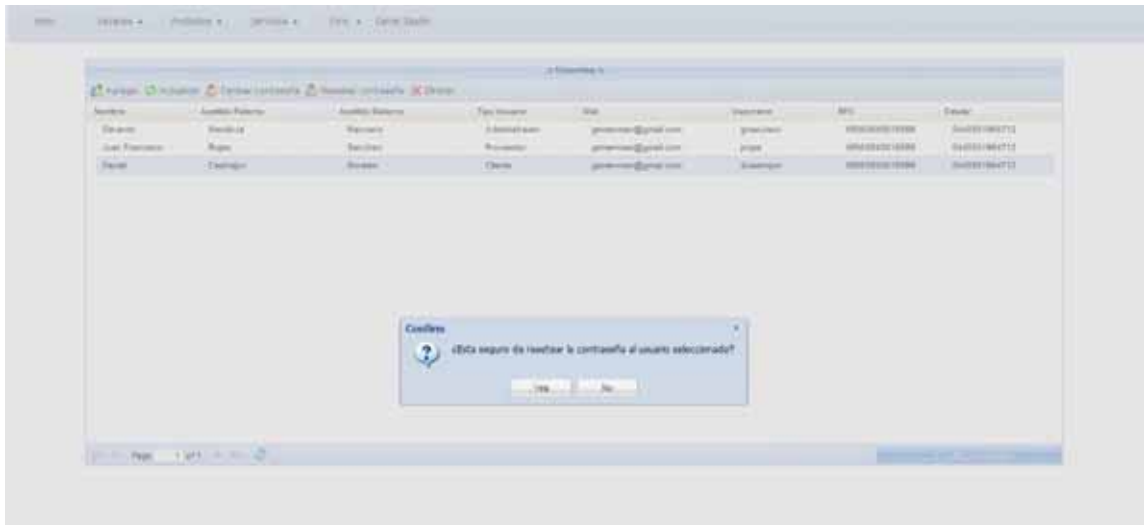


## Resetear contraseña

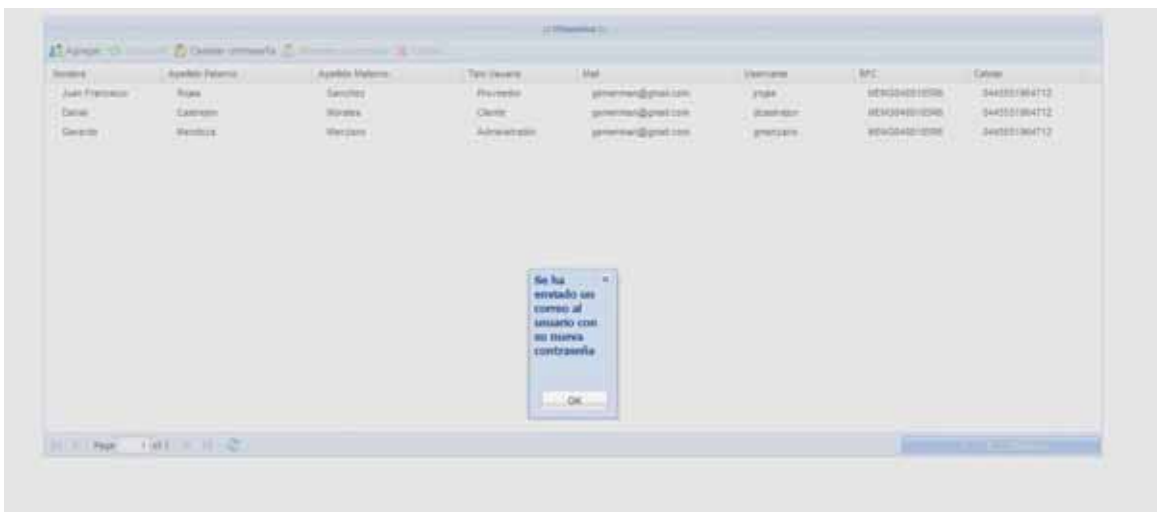
Para resetear la contraseña de un usuario seleccionamos al usuario que deseamos resetear la contraseña (); presionamos el botón "Resetear contraseña" de la parte superior de la pantalla que se activará al momento de seleccionar a un usuario ().



Al hacer clic en el botón de “Resetear contraseña”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para resetear la contraseña y “No” para cancelar ()



Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que se le envió un correo al usuario con su nueva contraseña ().





## Administrar Productos

### Alta de productos.

Desde la pantalla principal damos clic en “Productos” -> “Listar” ()

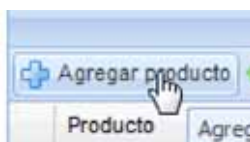


El navegador mostrará en pantalla una lista con información de los productos dados de alta en el sistema ()



Producto	Código	Precio	Descuento	Tipo venta	Fabricante	Proveedor
Producto 2 Actualizar	123456	\$0	0%	Peza	Del	Juan Francisco Rojas Sanchez
Producto save nombre	Producto save código	\$0	0%	Peza	Hp	Juan Francisco Rojas Sanchez
Producto 1	save	\$100	10%	Peza	Hp	Juan Francisco Rojas Sanchez

En la parte superior de la lista hacemos clic en el botón de “Agregar producto” ()



El navegador mostrará en pantalla un formulario para dar de alta un producto ().

The screenshot shows a web browser window with a form titled "Agregar producto". The form is titled "Detalle Producto" and contains the following fields:

- Nombre:** A text input field.
- Código:** A text input field.
- Precio Unitario:** A text input field.
- Tpo de venta:** A dropdown menu with the text "Seleccione un Tipo de Venta...".
- Descuento:** A text input field.
- Fabricante:** A dropdown menu with the text "Seleccione un Fabricante...".
- Proveedor:** A dropdown menu with the text "Seleccione un Proveedor...".
- Descripción:** A large text area for entering the product description.

At the bottom right of the form, there are two buttons: "Aceptar" and "Cancelar/Salir".

**Nombre:** Nombre del producto

**Código:** Código del producto

**Precio Unitario:** Precio de compra del producto

**Tipo de Venta:** Tipo de venta del producto

**Descuento:** Descuento del producto

**Fabricante:** Fabricante del producto

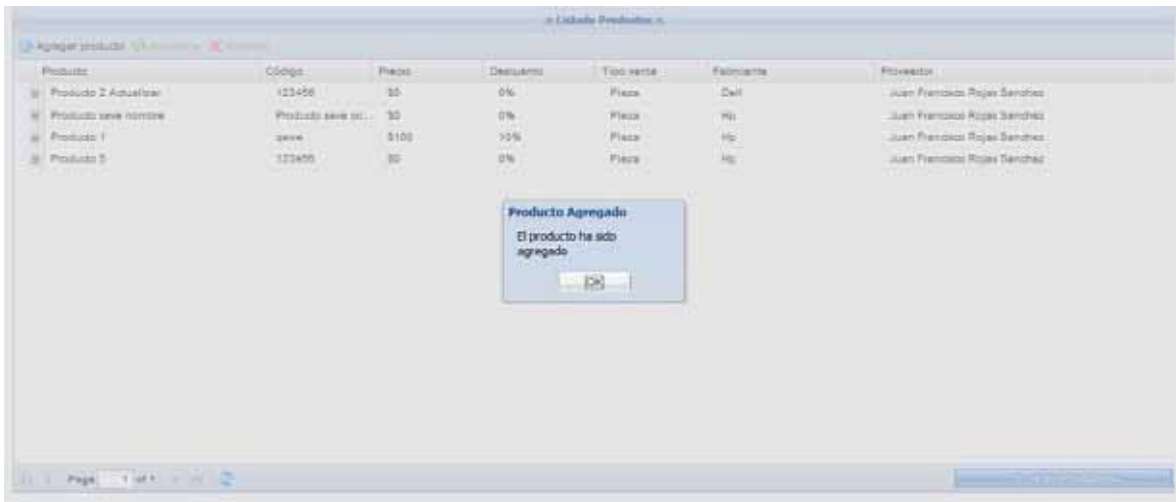
**Proveedor:** Proveedor que ofrece el producto

**Descripción:** Descripción del producto

Después de ingresar los datos solicitados por el sistema, hacemos clic en Aceptar para dar de alta al producto ().

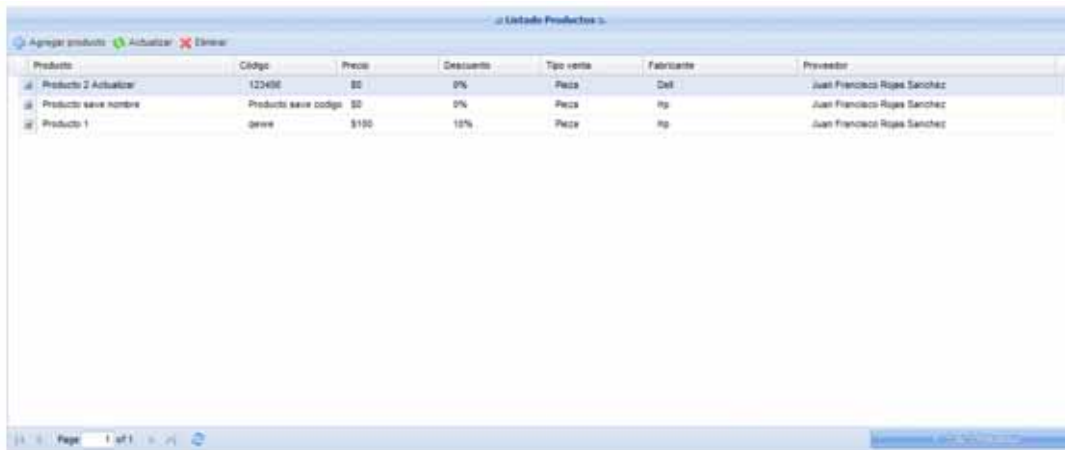


Si los datos ingresados en el sistema son correctos el navegador mostrara un mensaje diciendo que el producto fue dado de alta, en caso contrario nos mandara un mensaje de error ( )



### Actualizar productos

Para actualizar los datos de un producto seleccionamos el producto que deseamos actualizar ( ); presionamos el botón "Actualizar" de la parte superior de la pantalla que se activará al momento de seleccionar a un producto ( ).



Al hacer clic en “Actualizar” el sistema nos mostrara en pantalla con un formulario con los datos del producto que deseamos actualizar, realizamos los cambios y hacemos clic en “Aceptar” para guardar los cambios o “Cancelar” para regresar a la lista de productos ( )

Actualizar Producto

Detalle Producto

Nombre: Producto 2 Actualizar      Código: 123456      Precio Unitario: 0

Tipo de venta: Pieza      Descuento: 0      Fabricante: Del

Proveedor: Juan Francisco Rojas Sanchez

Descripción: producto 2

Aceptar      Cancelar/Salir



Si los datos fueron ingresados correctamente el sistema mostrará en la pantalla del navegador un mensaje de que el producto fue actualizado ( ).

Unidad Productiva

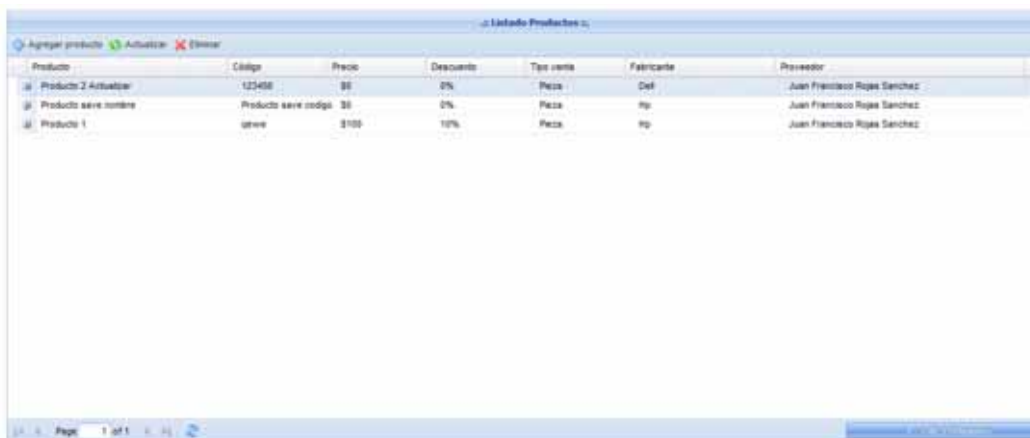
Producto	Código	Precio	Descuento	Tipo venta	Fabricante	Proveedor
Producto save nombre	Producto save código	30	0%	Pieza	Hr	Juan Francisco Rojas Sanchez
Producto 1	save	3100	10%	Pieza	Hr	Juan Francisco Rojas Sanchez
Producto 2 Actualizar	123456	30	0%	Pieza	Del	Juan Francisco Rojas Sanchez

Producto Agregado  
El producto ha sido actualizado  
OK

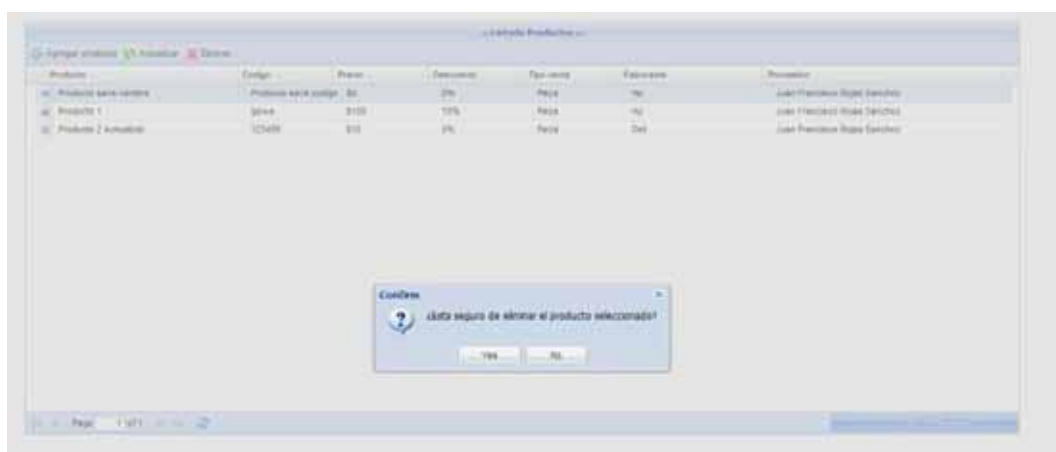
Page 1 of 1

## Eliminar productos

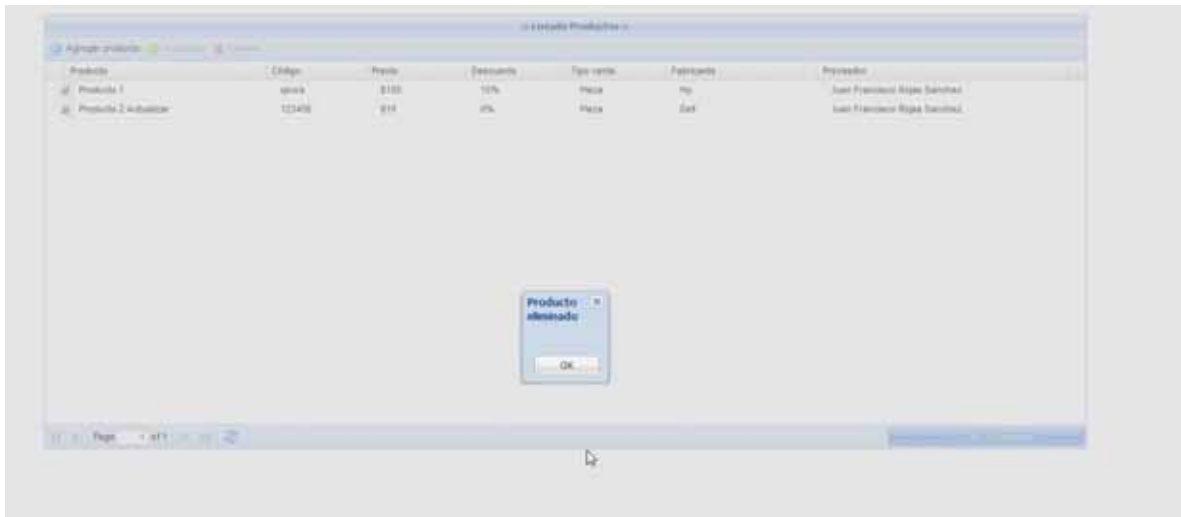
Para eliminar un producto, seleccionamos al producto que deseamos dar de baja (); presionamos el botón de “Eliminar” de la parte superior de la pantalla que se activará al momento de seleccionar un producto ().



Al hacer clic en el botón de “Eliminar”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para eliminar y “No” para cancelar ().



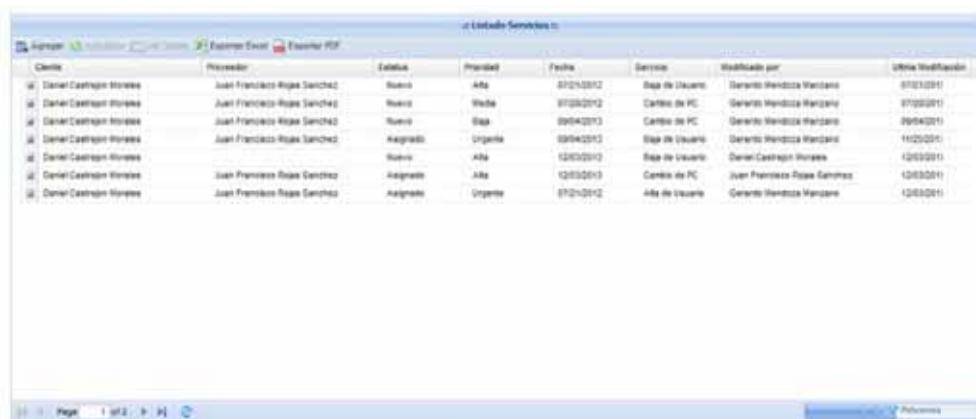
Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que el producto fue eliminado ().



## Administrar Servicios

### Alta servicios.

Desde la pantalla principal damos clic en “Foro” -> “Listar” ()



Cliente	Proveedor	Estado	Prioridad	Fecha	Servicio	Modificado por	Ultima Modificación
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	07/21/2012	Asa de Usuario	Gerardo Hernandez Mancera	07/21/2011
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Nuevo	Medio	07/20/2012	Cambio de PC	Gerardo Hernandez Mancera	07/20/2011
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Nuevo	Baja	09/04/2013	Cambio de PC	Gerardo Hernandez Mancera	09/04/2011
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	09/04/2013	Asa de Usuario	Gerardo Hernandez Mancera	11/20/2011
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	12/03/2011	Asa de Usuario	Daniel Castellón Morales	12/03/2011
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Asignado	Alta	12/03/2011	Cambio de PC	Juan Francisco Rojas Sanchez	12/03/2011
✓ Daniel Castellón Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	07/21/2012	Asa de Usuario	Gerardo Hernandez Mancera	12/03/2011

En la parte superior de la lista hacemos clic en el botón de “Agregar producto” ()



El navegador mostrará en pantalla un formulario para dar de alta un servicio ().



**Nuevo Servicio**

Detalle Usuario

Cliente:	Seleccione un Cliente...	Servicio:	Seleccione un Servicio...
Proveedor:	Seleccione un Proveedor...	Prioridad:	Seleccione una Prioridad...
Descripción:			

... Aceptar ... Cancelar/Salir

**Cliente:** Nombre del cliente a quien pertenece el servicio

**Servicio:** Tipo de servicio que se le realizará al cliente

**Proveedor:** Proveedor que realizara el servicio

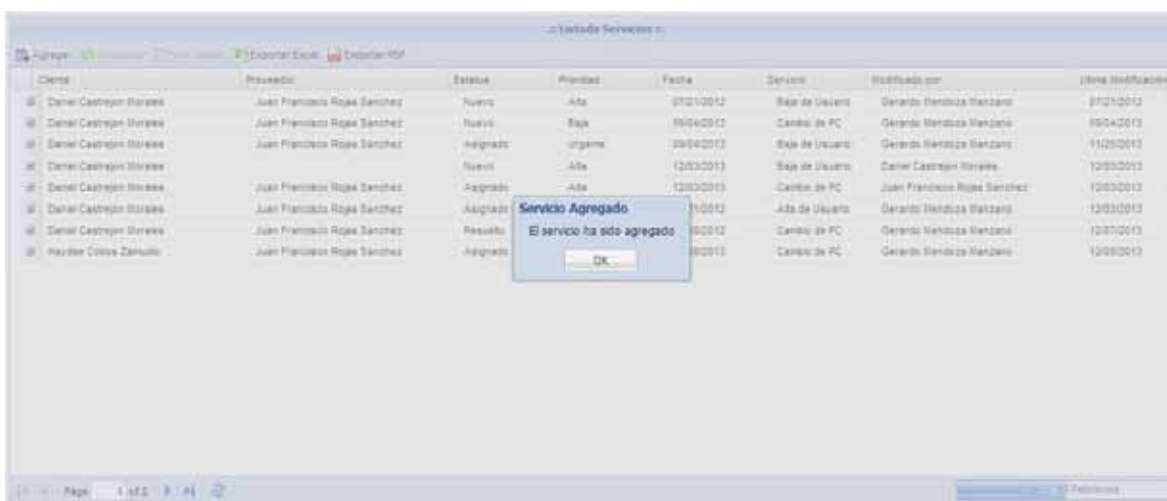
**Prioridad:** Prioridad del servicio

**Descripción:** Descripción del servicio

Después de ingresar los datos solicitados por el sistema, hacemos clic en Aceptar para dar de alta el servicio ().



Si los datos ingresados en el sistema son correctos el navegador mostrara un mensaje diciendo que el servicio fue dado de alta, en caso contrario nos mandara un mensaje de error ().



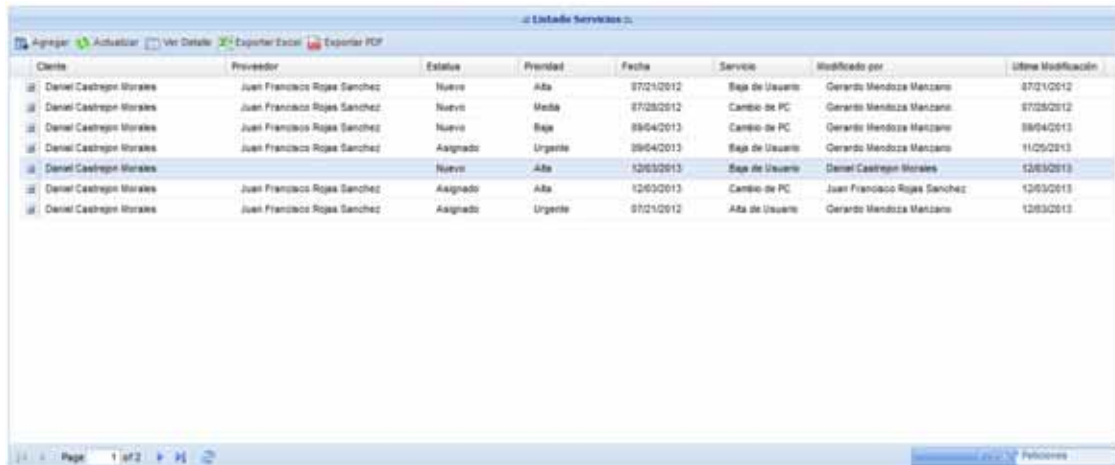
The screenshot shows a window titled 'Estado Servicios'. It contains a table with the following columns: Cliente, Proveedor, Estado, Prioridad, Fecha, Servicio, Modificado por, and Última Modificación. A dialog box titled 'Servicio Agregado' is overlaid on the table, displaying the message 'El servicio ha sido agregado' and an 'OK' button.

Cliente	Proveedor	Estado	Prioridad	Fecha	Servicio	Modificado por	Última Modificación
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	07/21/2012	Baja de Usuario	Gerardo Mendoza Manzana	07/21/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Baja	09/04/2012	Cambio de PC	Gerardo Mendoza Manzana	09/04/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Aprobado	Urgente	09/04/2012	Baja de Usuario	Gerardo Mendoza Manzana	11/25/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	12/03/2012	Baja de Usuario	Daniel Castrejon Morales	12/03/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Aprobado	Alta	12/03/2012	Cambio de PC	Juan Francisco Rojas Sanchez	12/03/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Aprobado	Alta	11/09/12	Alta de Usuario	Gerardo Mendoza Manzana	12/03/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Resuelto	Urgente	09/04/12	Cambio de PC	Gerardo Mendoza Manzana	12/07/2012
Haydee Colon Zamudio	Juan Francisco Rojas Sanchez	Aprobado	Urgente	09/04/12	Cambio de PC	Gerardo Mendoza Manzana	12/09/2012



## Actualizar productos

Para actualizar los datos de un servicio seleccionamos el servicio que deseamos actualizar (); y presionamos el botón “Actualizar” de la parte superior de la pantalla que se activará al momento de seleccionar a un servicio ().



The screenshot shows a web application window titled "Estado Servicios". At the top, there are navigation buttons: "Agregar", "Actualizar", "Ver Detalle", "Exportar Excel", and "Exportar PDF". Below these is a table with the following columns: Cliente, Proveedor, Estado, Prioridad, Fecha, Servicio, Modificado por, and Última Modificación. The table contains seven rows of service records.

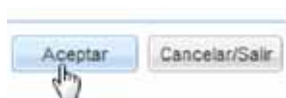
Cliente	Proveedor	Estado	Prioridad	Fecha	Servicio	Modificado por	Última Modificación
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	07/21/2012	Baja de Usuario	Gerardo Mendoza Mancano	07/21/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Medio	07/25/2012	Cambio de PC	Gerardo Mendoza Mancano	07/25/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Baja	09/04/2012	Cambio de PC	Gerardo Mendoza Mancano	09/04/2012
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	09/04/2013	Baja de Usuario	Gerardo Mendoza Mancano	11/25/2013
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	12/03/2013	Baja de Usuario	Daniel Castrejon Morales	12/03/2013
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Asignado	Alta	12/03/2013	Cambio de PC	Juan Francisco Rojas Sanchez	12/03/2013
Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	01/21/2012	Alta de Usuario	Gerardo Mendoza Mancano	12/03/2013



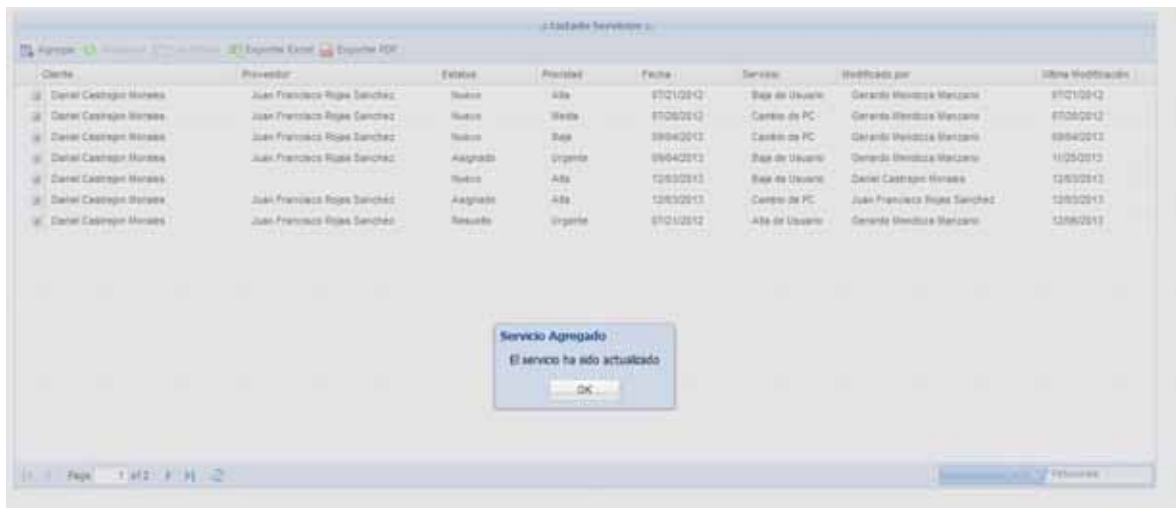
Al hacer clic en “Actualizar” el sistema nos mostrara en pantalla con un formulario con los datos del servicio que deseamos actualizar, realizamos los cambios y hacemos clic en “Aceptar” para guardar los cambios o “Cancelar” para regresar a la lista de servicios ().



The screenshot shows the "Actualizar Servicio" form. It contains several input fields and dropdown menus: "Detalle Usuario" (text field), "Cliente" (text field), "Proveedor" (dropdown menu), "Estado Servicio" (dropdown menu), "Prioridad" (dropdown menu), "Servicio" (dropdown menu), and "Descripción" (text area). The "Descripción" field contains the text "Ejemplo de alta de un servicio". At the bottom right, there are two buttons: "Aceptar" and "Cancelar/Salir".



Si los datos fueron ingresados correctamente el sistema mostrará en la pantalla del navegador un mensaje de que el servicio fue actualizado ().

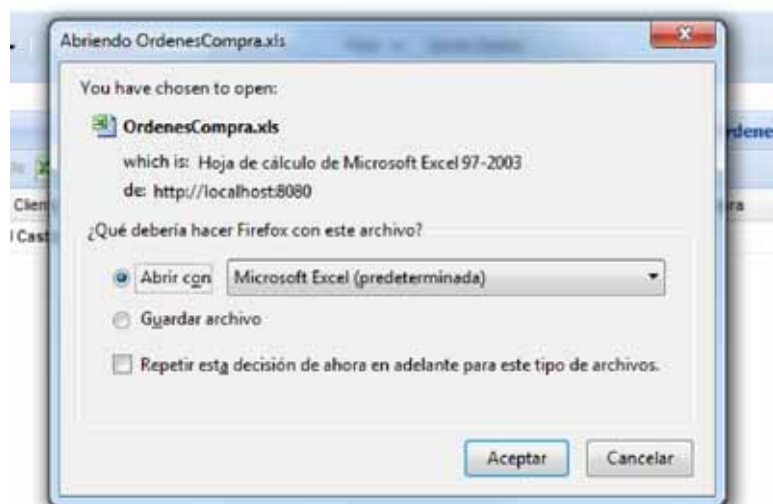


## Exportar Servicios

Para exportar la información de los servicios en archivos de Excel o PDF, en la parte superior de la pantalla hacemos clic en botón del formato al que deseamos exportar la información ().

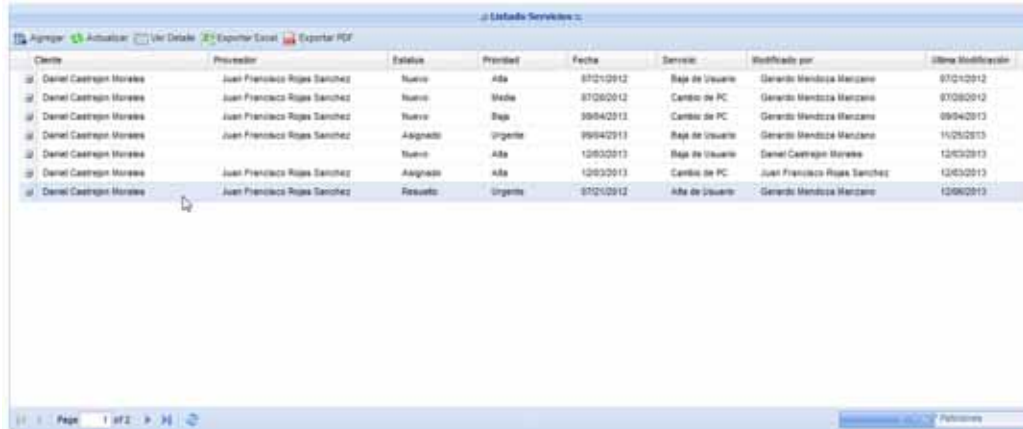


Al hacer clic en el botón el navegador mostrara una ventana con las opciones de Abrir o Guardar el archivo, seleccionamos una y le hacemos clic en "Aceptar" ().



## Ver detalles

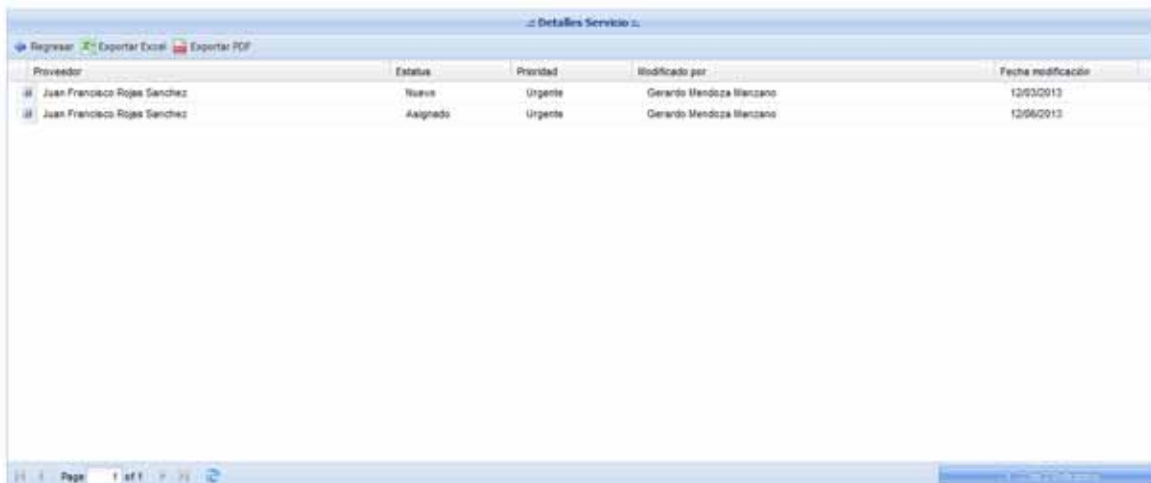
Para ver los detalles de un servicio, seleccionamos el servicio que deseamos ver los detalles (); y presionamos el botón de “Ver comentarios” de la parte superior de la pantalla que se activará al momento de seleccionar un servicio ().



Cliente	Proveedor	Estado	Prioridad	Fecha	Servicio	Modificado por	Última Modificación
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	07/01/2012	Baja de Usuario	Gerardo Mendoza Mancano	07/01/2012
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Medio	07/08/2012	Cambio de PC	Gerardo Mendoza Mancano	07/08/2012
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Baja	08/04/2012	Cambio de PC	Gerardo Mendoza Mancano	08/04/2012
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Asignado	Urgente	09/04/2012	Baja de Usuario	Gerardo Mendoza Mancano	11/25/2012
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Nuevo	Alta	12/03/2012	Baja de Usuario	Daniel Castrejon Morales	12/03/2012
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Asignado	Alta	12/03/2012	Cambio de PC	Juan Francisco Rojas Sanchez	12/03/2012
<input type="checkbox"/> Daniel Castrejon Morales	Juan Francisco Rojas Sanchez	Resuelto	Urgente	07/01/2012	Alta de Usuario	Gerardo Mendoza Mancano	12/08/2012



Al hacer clic en “Ver detalle” el sistema nos mostrara en pantalla una lista con todos los estados por los que ha pasado el servicio ().



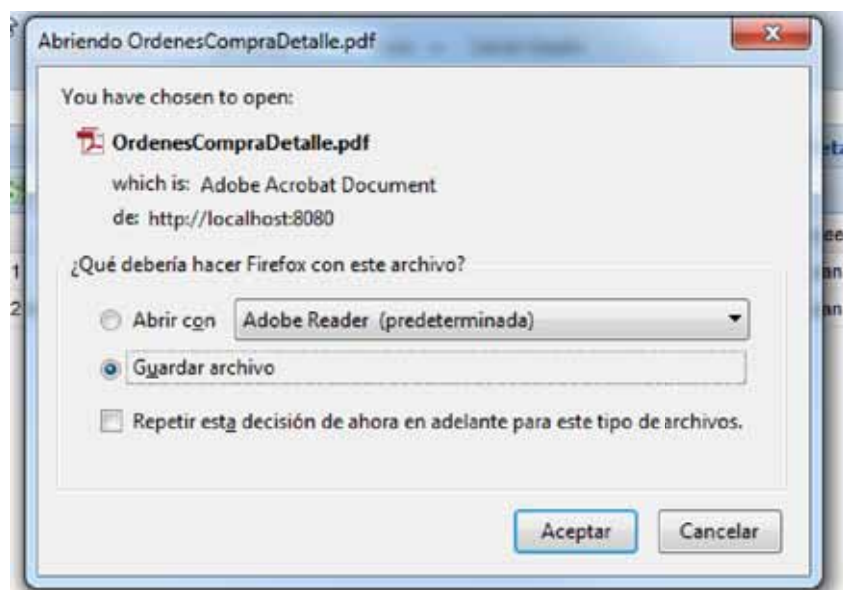
Proveedor	Estado	Prioridad	Modificado por	Fecha modificación
<input type="checkbox"/> Juan Francisco Rojas Sanchez	Nuevo	Urgente	Gerardo Mendoza Mancano	12/03/2012
<input type="checkbox"/> Juan Francisco Rojas Sanchez	Asignado	Urgente	Gerardo Mendoza Mancano	12/06/2012

## Exportar detalles

Para exportar la información de los detalles de un servicio en formato de Excel o PDF, en la parte superior de la pantalla hacemos clic en botón del formato al que deseamos exportar la información ()



Al hacer clic en el botón el navegador mostrara una ventana con las opciones de Abrir o Guardar el archivo, seleccionamos una y le hacemos clic en "Aceptar" ().



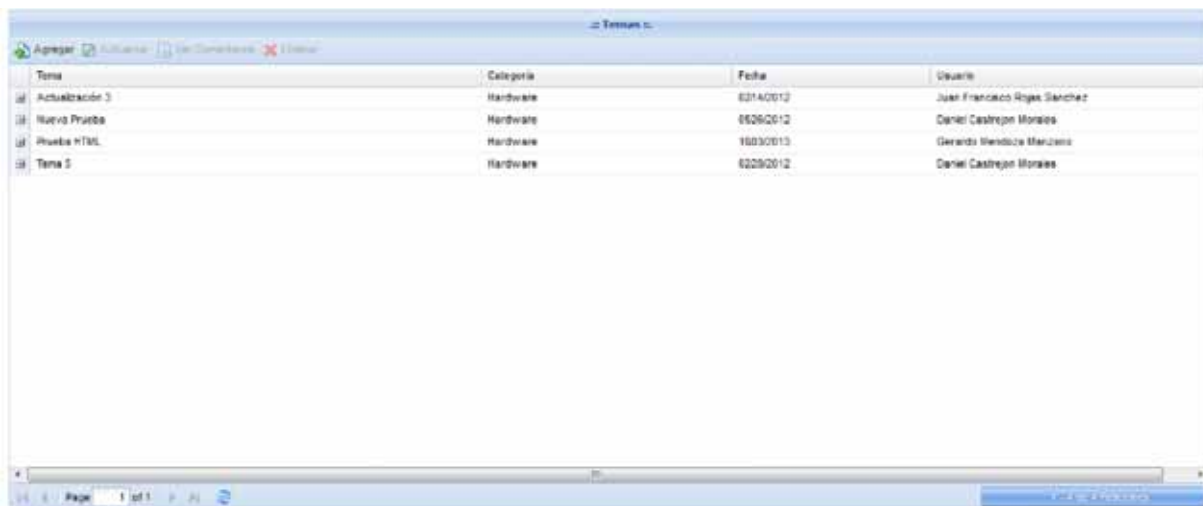
## Administrar Foro

### Alta temas.

Desde la pantalla principal damos clic en “Foro” -> “Listar” ()



El navegador mostrará en pantalla una lista con información de los temas abiertos en el sistema ().

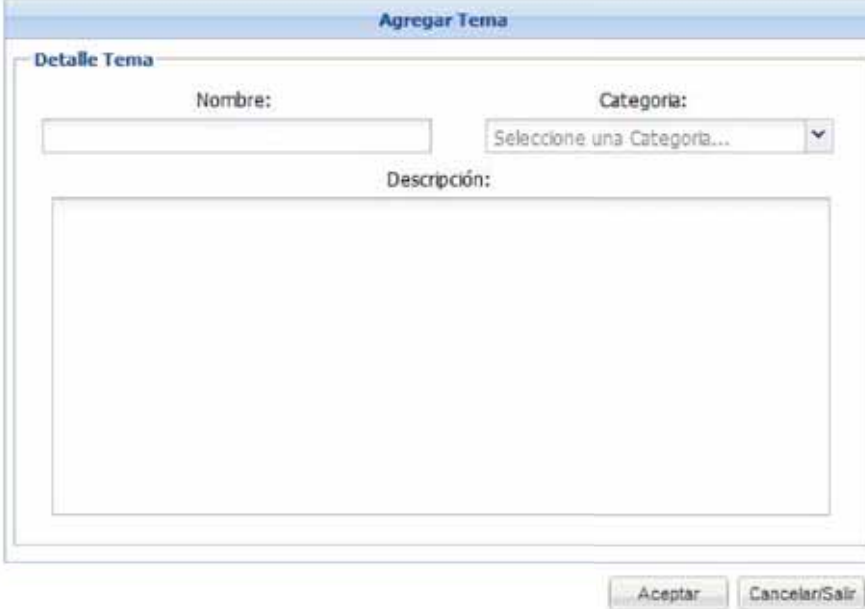


Tema	Categoría	Fecha	Usuario
Actualización 3	Hardware	03/14/2012	Juan Francisco Rojas Sanchez
Nuevo Prueba	Hardware	05/06/2012	Daniel Castrejon Morales
Prueba #123	Hardware	16/03/2013	Gerardo Mendoza Merizola
Tema 5	Hardware	02/08/2012	Daniel Castrejon Morales

En la parte superior de la lista hacemos clic en el botón de “Agregar” ()



El navegador mostrará en pantalla un formulario para dar de alta al usuario ().



Agregar Tema

Detalle Tema

Nombre:

Categoría:

Descripción:

Aceptar Cancelar/Salir

**Nombre:** Nombre del tema

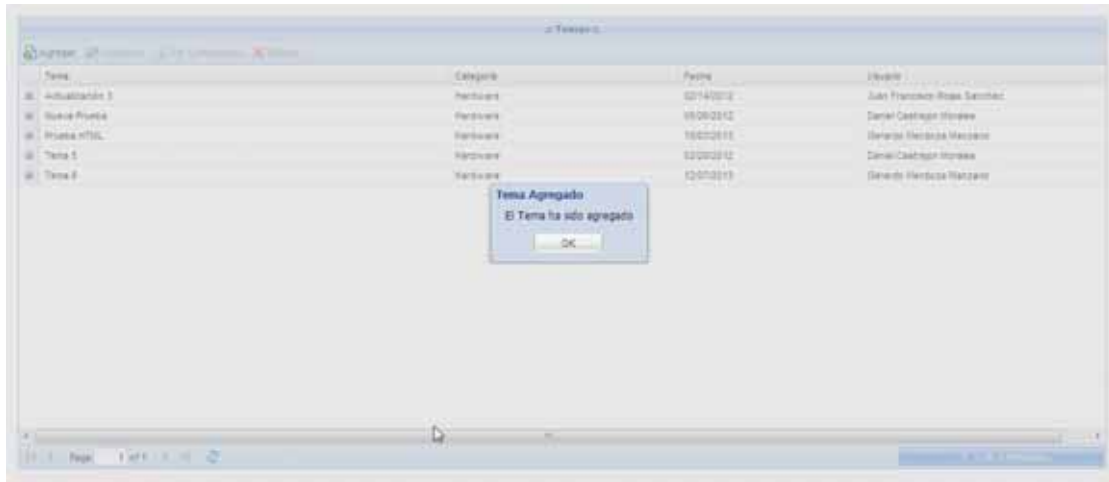
**Categoría:** Categoría del tema

**Descripción:** Descripción del tema

Después de ingresar los datos solicitados por el sistema, hacemos clic en Aceptar para dar de alta el tema ().

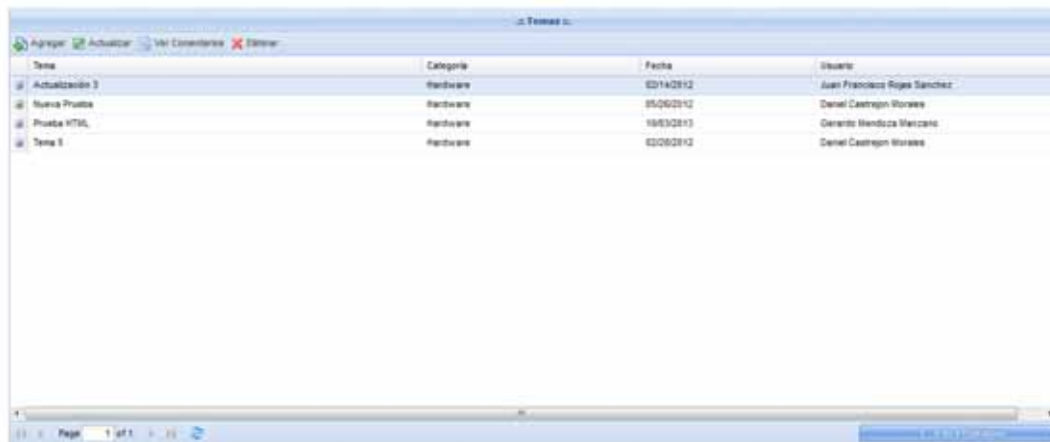


Si los datos ingresados en el sistema son correctos el navegador mostrara un mensaje diciendo que el tema fue dado de alta, en caso contrario nos mandara un mensaje de error ()



### Actualizar temas

Para actualizar los datos de un tema seleccionamos el tema que deseamos actualizar (); presionamos el botón “Actualizar” de la parte superior de la pantalla que se activará al momento de seleccionar a un tema ().



Al hacer clic en “Actualizar” el sistema nos mostrara en pantalla con un formulario con los datos del producto que deseamos actualizar, realizamos los cambios y hacemos clic en “Aceptar” para guardar los cambios o “Cancelar” para regresar a la lista de productos ().

Actualizar Tema

Detalle Tema

Nombre: Actualización 3

Categoria: Hardware

Descripción: Se actualizan los campos después de modificación

Aceptar Cancelar/Salir



Si los datos fueron ingresados correctamente el sistema mostrará en la pantalla del navegador un mensaje de que el mensaje fue actualizado ().

Temas	Categoria	Fecha	Usuario
Nueva Prueba	Hardware	05/26/2012	Daniel Castañón Rosales
Prueba VTDL	Hardware	10/31/2012	Daniel Castañón Rosales
Tema 2	Hardware	03/30/2012	Daniel Castañón Rosales
Actualización free	Hardware	02/14/2012	Juan Francisco Rojas Sanchez

Temas Agregado

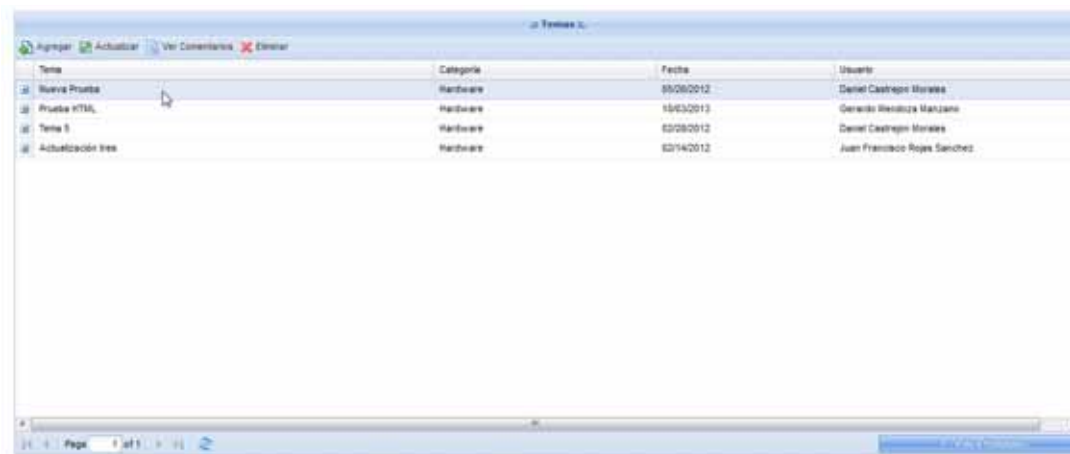
El Tema ha sido actualizado

OK

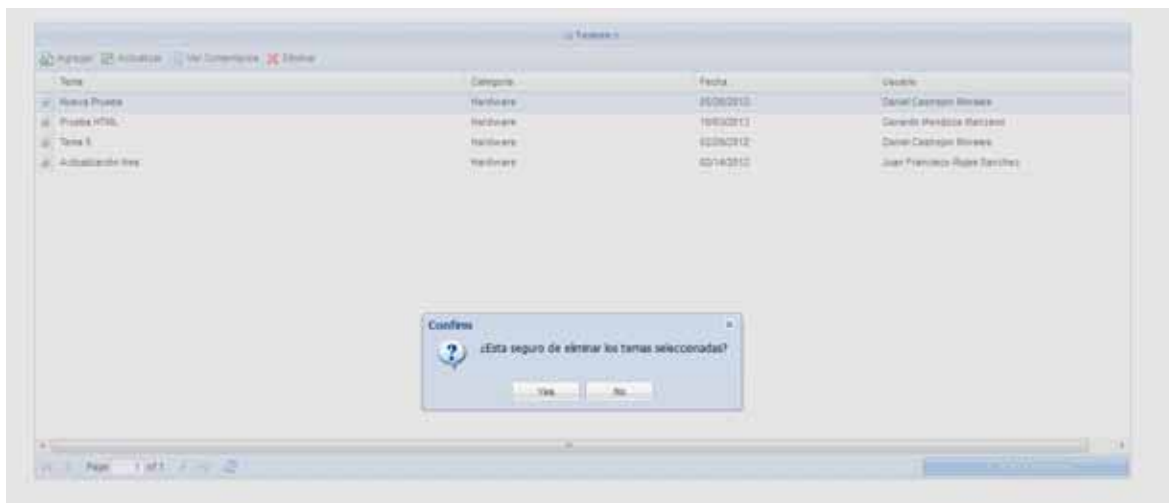


## Eliminar temas

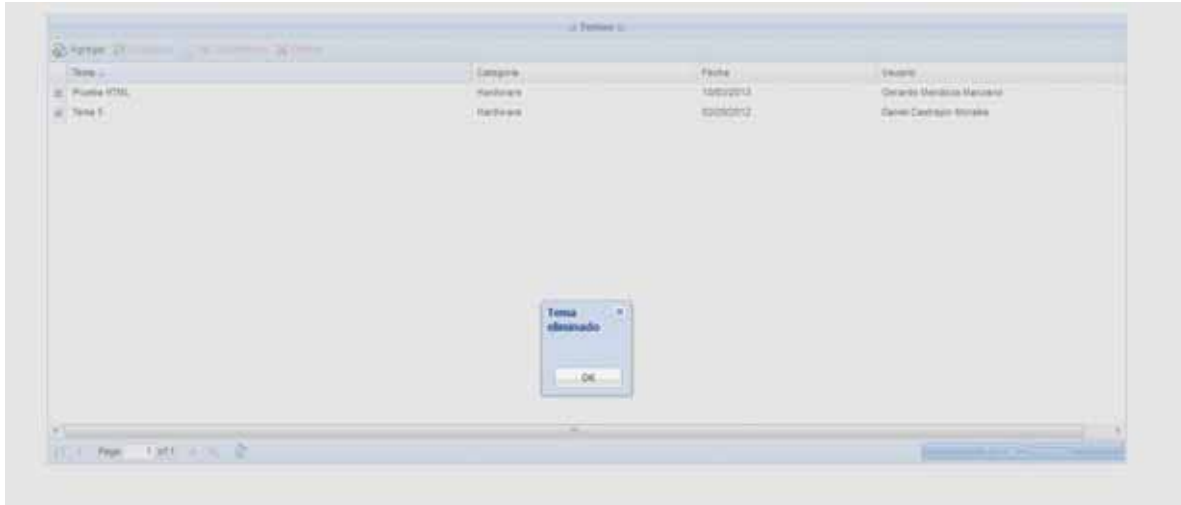
Para eliminar un tema, seleccionamos el tema que deseamos dar eliminar (); presionamos el botón de “Eliminar” de la parte superior de la pantalla que se activará al momento de seleccionar un tema ().



Al hacer clic en el botón de “Eliminar”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para eliminar y “No” para cancelar ().

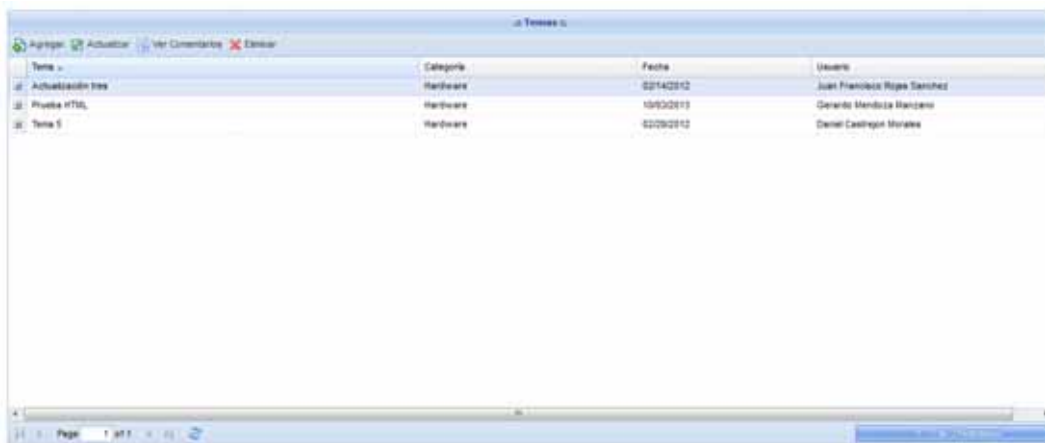


Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que el tema fue eliminado ().



### Ver comentarios

Para ver los comentarios de un tema, seleccionamos el tema que deseamos ver los comentarios (); y presionamos el botón de “Ver comentarios” de la parte superior de la pantalla que se activará al momento de seleccionar un tema ().



Al hacer clic en “Ver comentarios” el sistema nos mostrara en pantalla una lista con todos los comentarios del tema ().

Mensaje	Fecha
Gerardo Mendoza Mancano	02/14/2012
Gerardo Mendoza Mancano	02/14/2012
Gerardo Mendoza Mancano	02/14/2012
Gerardo Mendoza Mancano	02/14/2012
Juan Francisco Rojas Sanchez	03/01/2012
Daniel Castrejon Morales	08/19/2013
Daniel Castrejon Morales	08/19/2013

### Agregar comentario

Para agregar un comentario, en la lista de comentarios del tema (); presionamos el botón de “Agregar comentario” ubicado en la parte superior de la pantalla ().



Al hacer clic en “Agregar comentario” el sistema nos mostrara en pantalla con un formulario para ingresar el comentario ().

Agregar Mensaje

Comentario:

Aceptar Cancelar/Salir

**Comentario:** Comentario hecho sobre el tema

Después de ingresar los datos solicitados por el sistema, hacemos clic en Aceptar para dar de alta el comentario ().

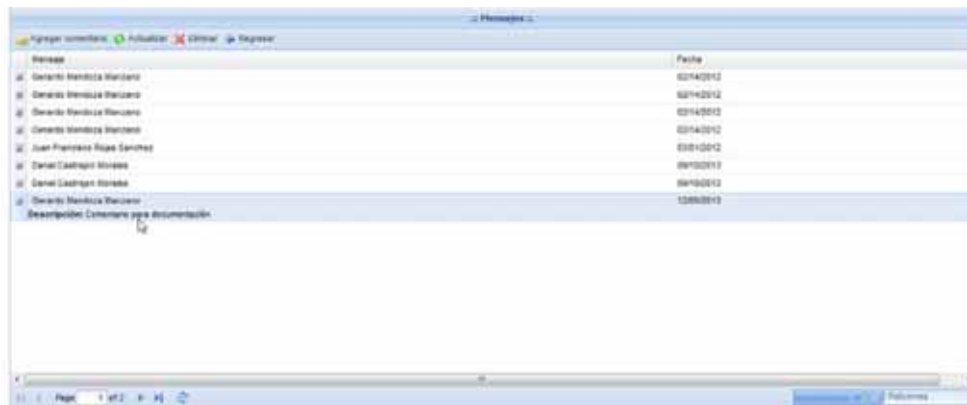


Si los datos ingresados en el sistema son correctos el navegador mostrara un mensaje diciendo que el comentario fue agregado, en caso contrario nos mandara un mensaje de error ().



### **Modificar comentario**

Para modificar el comentario de un tema seleccionamos el comentario que deseamos actualizar (); presionamos el botón “Actualizar” de la parte superior de la pantalla que se activará al momento de seleccionar a un tema ().



Al hacer clic en “Actualizar” el sistema nos mostrara en pantalla con un formulario con los datos del comentario que deseamos actualizar, realizamos los cambios y hacemos clic en “Aceptar” para guardar los cambios o “Cancelar” para regresar a la lista de comentarios ( )

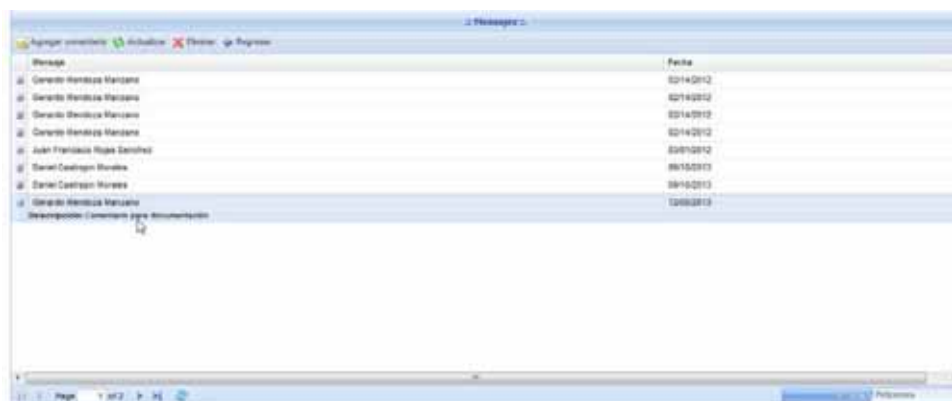


Si los datos fueron ingresados correctamente el sistema mostrará en la pantalla del navegador un mensaje de que el mensaje fue actualizado ( ).



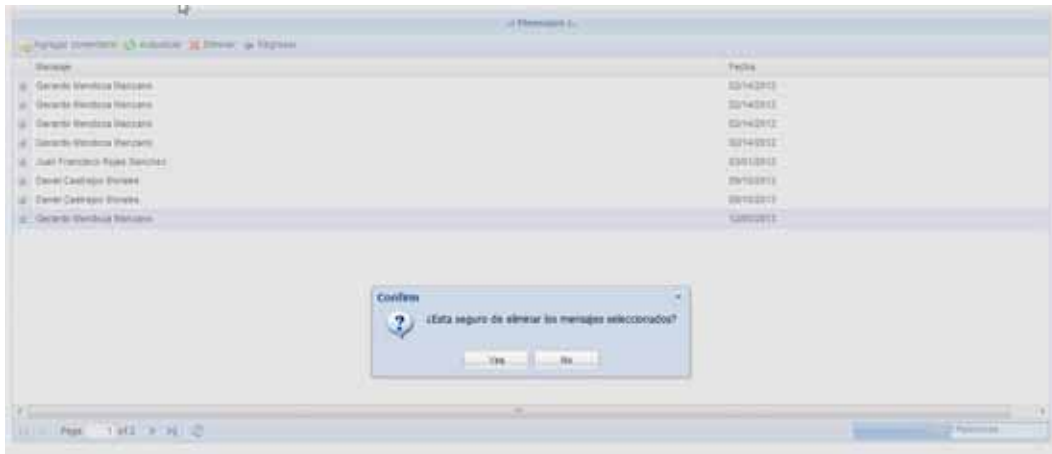
### **Eliminar comentario**

Para eliminar un tema, seleccionamos el tema que deseamos dar eliminar ( ); presionamos el botón de “Eliminar” de la parte superior de la pantalla que se activará al momento de seleccionar un tema ( ).

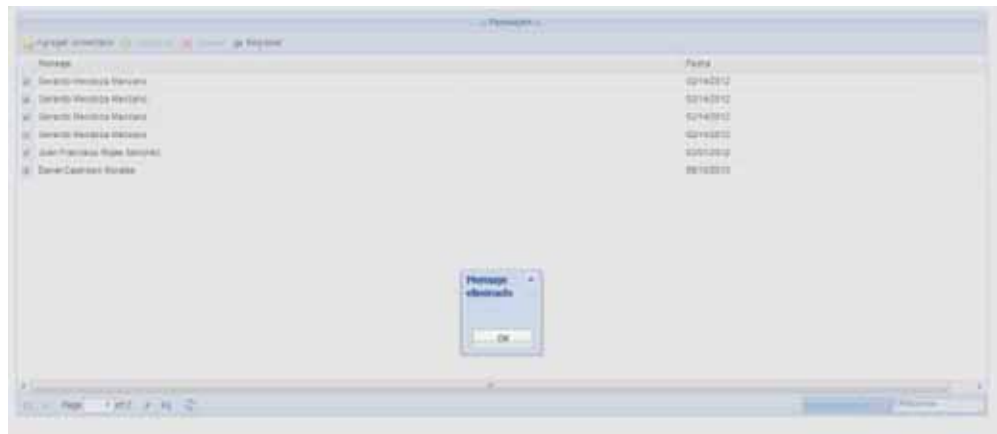




Al hacer clic en el botón de “Eliminar”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para eliminar y “No” para cancelar ().



Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que el mensaje fue eliminado ().



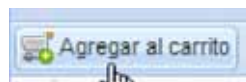
## Carrito de compras

### Agregar productos

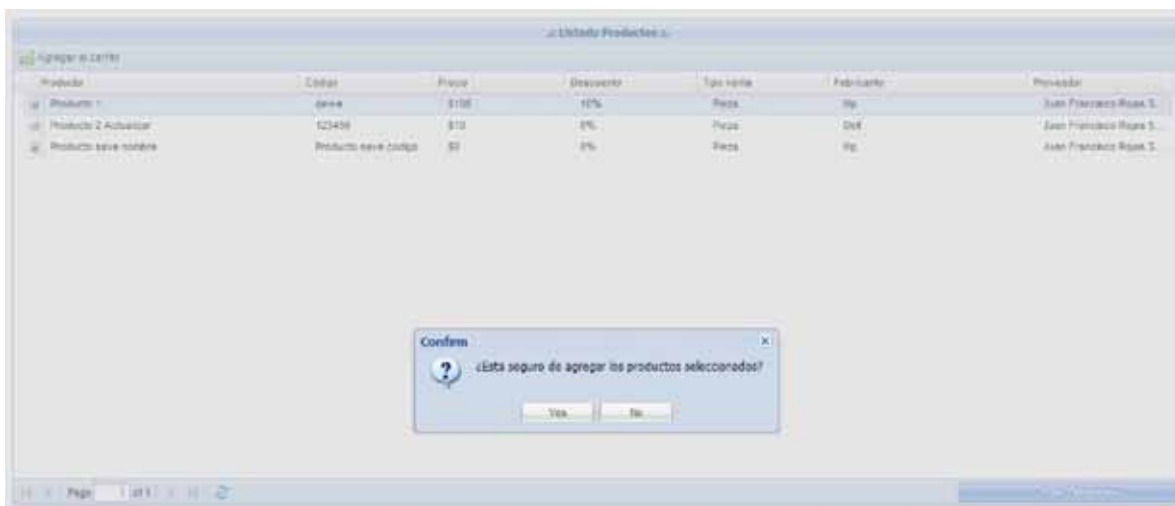
Para agregar productos al carrito de compras desde la página de listado de productos, seleccionamos el producto que deseamos agregar al carrito de compras y hacemos clic en “Agregar al carrito”



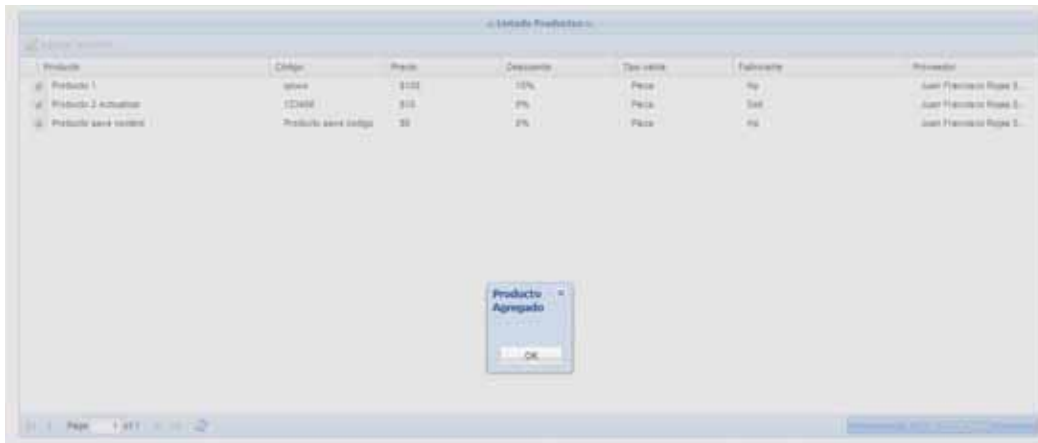
Producto	Código	Precio	Descuento	Tipo venta	Fabricante	Proveedor
Producto 1	9999	\$100	10%	Peza	Hp	Juan Francisco Rojas S...
Producto 2 Actualizar	123456	\$10	0%	Peza	Dell	Juan Francisco Rojas S...
Producto save nombre	Producto save código	\$0	0%	Peza	Hp	Juan Francisco Rojas S...



Al hacer clic en el botón de “Agregar al carrito”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para agregar y “No” para cancelar ( )



Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que el producto fue agregado al carrito () (Para agregar más productos al carrito de compras repetimos los pasos anteriores).



Para ver los productos que tenemos en la pantalla principal damos clic en “Carrito de compras” ().



El navegador mostrará en pantalla una lista con información de los productos que hemos agregado al carrito de compras ().





## Comprar productos

Para comprar los productos que tenemos en carrito de compras damos clic en el botón “Comprar” ubicado en la parte superior de la pantalla



Al hacer clic en el botón de “Comprar”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para comprar y “No” para cancelar ()

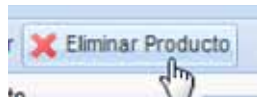
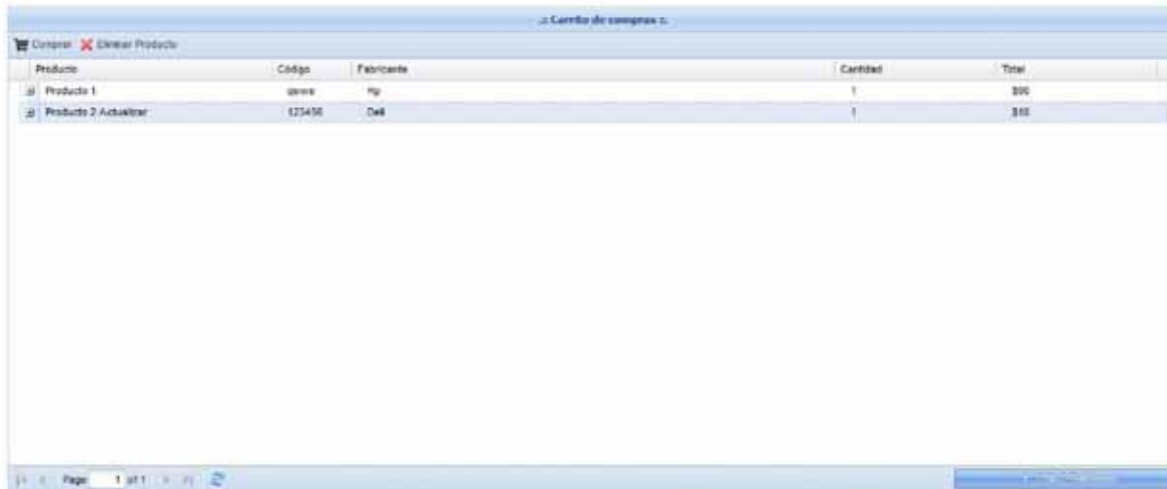


Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que los productos fueron comprados y nos enviara a la página del listado de ordenes ().

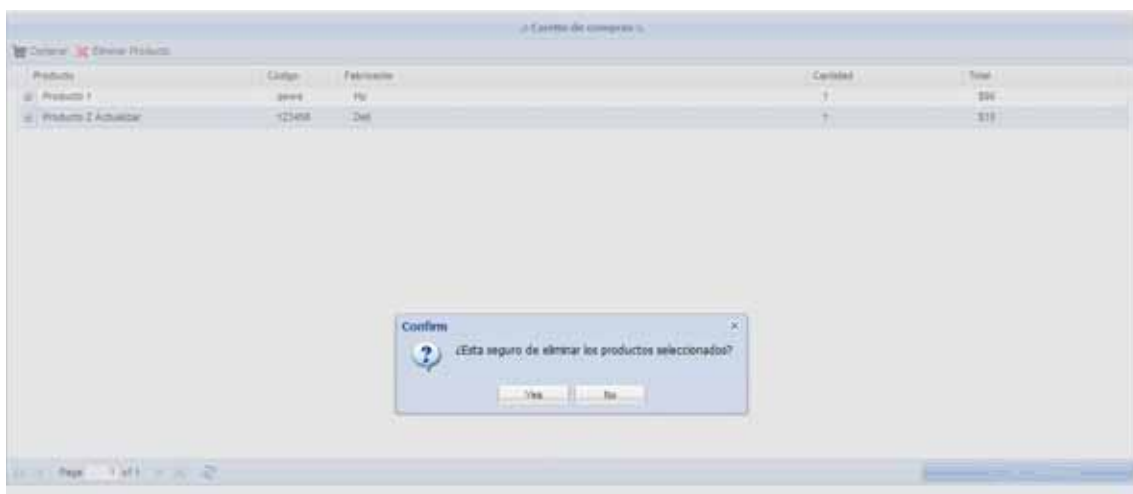


## Eliminar productos

Para eliminar un producto del carrito de compras, seleccionamos el producto que deseamos eliminar (); y presionamos el botón de “Eliminar” de la parte superior de la pantalla que se activará al momento de seleccionar un producto de la lista ()



Al hacer clic en el botón de “Eliminar Producto”, el sistema nos mostrara en pantalla un mensaje de confirmación hacemos clic “Yes” para eliminar y “No” para cancelar ().

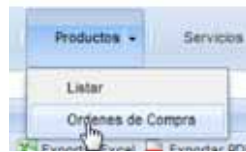


Al hacer clic en “Yes” el sistema nos mostrará en pantalla un mensaje que indica que el producto fue eliminado del carrito ( ) (Para eliminar más productos al carrito de compras repetimos los pasos anteriores).



## Orden de compras

Desde la pantalla principal damos clic en “Productos” -> “Órdenes de compra” ()



El navegador mostrará en pantalla una lista con información de las órdenes de compra del usuario temas abiertos en el sistema ()



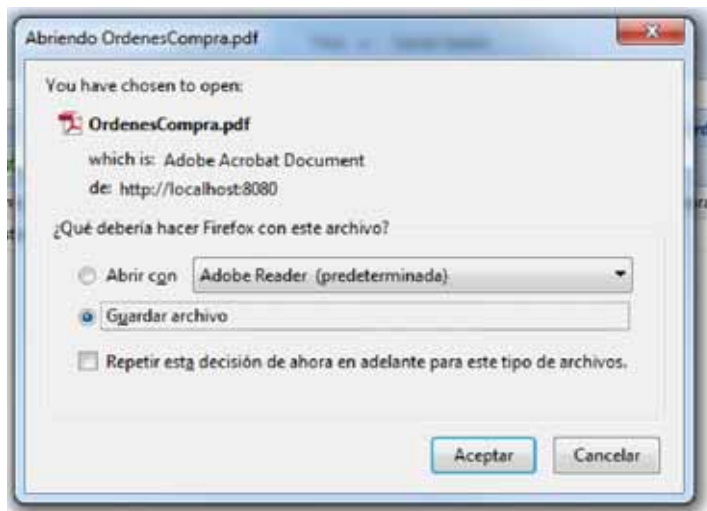
Nombre Cliente	Fecha de Compra	Total Compra
Daniel Castrejon Morales	11/27/2013	00

## Exportar órdenes de compra

Para exportar la información de las órdenes de compra en formato de Excel o PDF, en la parte superior de la pantalla hacemos clic en botón del formato al que deseamos exportar la información ()

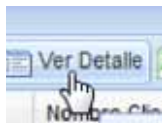


Al hacer clic en el botón el navegador mostrara una ventana con las opciones de Abrir o Guardar el archivo, seleccionamos una y le hacemos clic en “Aceptar” ().



### Ver comentarios

Para ver los detalles de una orden de compra, seleccionamos la orden de compra que deseamos ver los detalles (); y presionamos el botón de “Ver detalle” de la parte superior de la pantalla que se activará al momento de seleccionar una orden de compra ().



Al hacer clic en “Ver detalle” el sistema nos mostrara en pantalla una lista con todos los todos los artículos que conforman el orden de compra ().



Producto	Cantidad	Proveedor	Cantidad	Total
Producto 1	10	Juan Francisco Rojas Sanchez	1	10
Producto 2 Inicial	10	Juan Francisco Rojas Sanchez	1	10

### **Exportar artículos órdenes de compra**

Para exportar la información que contiene una orden de compra en formato de Excel o PDF, en la parte superior de la pantalla hacemos clic en botón del formato al que deseamos exportar la información ().



## Anexo 2. Código Fuente

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package mx.uam.azc.cbi.action;  
  
import com.opensymphony.xwork2.ActionContext;  
import com.opensymphony.xwork2.ActionSupport;  
import java.util.List;  
import java.util.Map;  
  
import mx.uam.azc.cbi.data.TemaJoinDTO;  
import mx.uam.azc.cbi.data.TemaSaveDTO;  
import mx.uam.azc.cbi.data.UsuarioJoinDTO;  
import mx.uam.azc.cbi.logica.TimeStamp;  
import mx.uam.azc.cbi.interfaces.AdminTemas;  
import org.springframework.beans.factory.annotation.Autowired;  
  
/**  
 *  
 * @author manzano  
 */  
  
public class TemasAction extends ActionSupport{  
  
    @Autowired
```

```

private AdminTemas administradorTema;

private TemaSaveDTO tema;

private TemaJoinDTO loadtopic;

private String listatemas;

private String topic;

private String respuesta;

private String idTema;

private String nombre;

private String categoria;

private String descripcion;

private List<TemaJoinDTO> temas;

public String cargar(){
    try{
        Map session = ActionContext.getContext().getSession();

        loadtopic = (TemaJoinDTO) session.get("loadtopic");//
        administradorTema.load(Long.parseLong("2"));

        topic = "{success: true, data: { \"nombre\": \"\" + loadtopic.getNombre() + "\", \"categoria\": \"\"
            + loadtopic.getCategoria().getCategoria() + "\", \"descripcion\": \"\" +
loadtopic.getDescripcion() + \"\"}}";
    }
    catch(Exception e){

    }

    return "loadtopic";
}

public String listar(){

```



```
Map session = ActionContext.getContext().getSession();  
session.remove("loadtopic");  
return "listar";  
}
```

```
public String listartemas(){  
    temas = administradorTema.listar();  
    listatemas = "{\"success\":true, \"totalCount\": \"\" + temas.size() + "\", \"data\":[";  
        for(int i = 0; i<temas.size();i++) {  
            listatemas = listatemas + "{\"idTema\":\""+ temas.get(i).getIdTema() +  
                "\", \"nombre\":\""+temas.get(i).getNombre()+  
                "\", \"categoria\": \"\" + temas.get(i).getCategoria().getCategoria()+ "\",  
\"descripcion\": \"\" + temas.get(i).getDescripcion() +  
                "\", \"fecha\": \"\" + Timestamp.Timestamp(temas.get(i).getDate()+  
                "\", \"usuario\": \"\" + temas.get(i).getId_usuario().getNombre() +  
                \" \" + temas.get(i).getId_usuario().getApelliop() +  
                \" \" + temas.get(i).getId_usuario().getApellidom() +  
                "\" },";  
        }  
        listatemas = listatemas.substring(0, listatemas.length()-1);  
        listatemas = listatemas + "]}";  
    return "listartemas";  
}
```

```
public String insertar(){  
    return "insertar";  
}
```

```
}
```

```
public String agregar(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    tema = new TemaSaveDTO();  
    tema.setNombre(nombre);  
    tema.setCategoria(Integer.parseInt(categoria));  
    tema.setDescripcion(descripcion);  
    tema.setDate(TimeStamp.Fecha());  
    tema.setIsobsolete(false);  
    tema.setId_usuario(userlogin.getId_usuario());  
    administradorTema.insertar(tema);  
    tema = null;  
    respuesta = "{success:true,result:\\"El Tema ha sido agregado\\"}";  
    return "respuesta";  
}
```

```
public String form_actualizar(){  
    return "form_actualizar";  
}
```

```
public String load(){  
    try{  
        long id = Long.parseLong(idTema);  
        loadtopic = administradorTema.loadtopic(id);  
        Map session = ActionContext.getContext().getSession();
```

```

    session.put("loadtopic",loadtopic);

    respuesta ="{success:true}";

    return "respuesta";

}

catch (NumberFormatException e){

    System.out.println(e);

    respuesta ="{failure:true}";

    return respuesta;

}

}

```

```

public String actualizar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    loadtopic = (TemaJoinDTO)session.get("loadtopic");

    tema = administradorTema.load(loadtopic.getIdTema());

    int cat;

    if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario().longValue() ==
tema.getId_usuario().longValue()){

        tema.setNombre(nombre);

        try{

            cat = Integer.parseInt(categoria);

        }

        catch (NumberFormatException e){

            cat = tema.getCategoria();

        }

        tema.setCategoria(cat);

        tema.setDescripcion(descripcion);

```

```

    administradorTema.actualizar(tema);

    respuesta = "{success:true,result:\'El Tema ha sido actualizado\'}";

    return "respuesta";
}

else{

    respuesta = "{failure:true,result:\'No tienes permisos para actualizar\'}";

    return "respuesta";

}

}

```

```

public String eliminar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    System.out.println("idTema: " + idTema);

    try{

        long id = Long.parseLong(idTema);

        tema = administradorTema.load(id);

    }

    catch (NumberFormatException e){

        System.out.println(e);

        respuesta ="{failure:true}";

    }

    if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario().longValue() ==
tema.getId_usuario().longValue()){

        tema.setIsobsolete(true);

        administradorTema.actualizar(tema);

```

```
        respuesta = "{success:true,result:\\"El tema fue eliminado\\"}";
        return "respuesta";
    }
    else{
        respuesta = "{failure:true,result:\\"No tienes permisos para eliminar\\"}";
        return "respuesta";
    }
}
```

```
public List<TemaJoinDTO> getTemas() {
    return temas;
}
```

```
public void setTemas(List<TemaJoinDTO> temas) {
    this.temas = temas;
}
```

```
public String getListatemas() {
    return listatemas;
}
```

```
public void setListatemas(String listatemas) {
    this.listatemas = listatemas;
}
```

```
public String getRespuesta() {
```

```
        return respuesta;
    }

    public void setRespuesta(String respuesta) {
        this.respuesta = respuesta;
    }

    public String getIdTema() {
        return idTema;
    }

    public void setIdTema(String idTema) {
        this.idTema = idTema;
    }

    public String getCategoria() {
        return categoria;
    }

    public void setCategoria(String categoria) {
        this.categoria = categoria;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
```

```
        this.descripcion = descripcion;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getTopic() {
        return topic;
    }

    public void setTopic(String topic) {
        this.topic = topic;
    }

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package mx.uam.azc.cbi.action;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import java.util.List;
import java.util.Map;
import mx.uam.azc.cbi.data.CompraJoinDTO;
import mx.uam.azc.cbi.data.OrdenCompraJoinDTO;
import mx.uam.azc.cbi.data.UsuarioJoinDTO;
import mx.uam.azc.cbi.interfaces.AdminCompras;
import mx.uam.azc.cbi.interfaces.AdminOrdenes;
import mx.uam.azc.cbi.logica.TimeStamp;
import org.springframework.beans.factory.annotation.Autowired;

/**
 *
 * @author gmendoza
 */

public class OrdenesAction extends ActionSupport{

    @Autowired
    private AdminOrdenes administradorOrdenes;

    @Autowired
    private AdminCompras administradorCompras;

    private List<OrdenCompraJoinDTO> ordenes;
```



```
private List<CompraJoinDTO> compras;

private OrdenCompraJoinDTO ordenload = new OrdenCompraJoinDTO();

private String listarordenes;

private String listardetalles;

private String idOrden;

private String respuesta;
```

```
public String listar(){

    return "listar";

}
```

```
public String listarOrdenes(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    if(userlogin.getTipo_usuario().getId() == 1){

        ordenes = administradorOrdenes.listar();

    }

    else

        ordenes = administradorOrdenes.listar(userlogin.getId_usuario());

    listarordenes = "{\"sucess\":true, \"totalCount\":" + "\"" + ordenes.size() + "\", "+ "\"data\":[";

    for(int i = 0; i < ordenes.size(); i++){

        listarordenes = listarordenes + "{\"idOrden\":\\"" + ordenes.get(i).getIdCompra() +

            "\", \"cliente\": \"" +

ordenes.get(i).getIdUsuario().getNombre() +

            " " + ordenes.get(i).getIdUsuario().getApellidoop()

+

```

```

                                " " +
ordenes.get(i).getIdUsuario().getApellido() +
                                "\", \"total\": \"" + ordenes.get(i).getTotal() +
                                "\", \"fecha\": \"" +
TimeStamp.Timestamp(ordenes.get(i).getFecha())+ "\", ";

    }

    listarordenes = listarordenes.substring(0, listarordenes.length()-1);

    listarordenes = listarordenes + "]]";

    return "listarOrdenes";

}

```

```

public String load(){
    try{
        long id = Long.parseLong(idOrden);
        ordenload = administradorOrdenes.load(id);
        Map session = ActionContext.getContext().getSession();
        session.put("ordenload",ordenload);
        respuesta ="{success:true}";
        return "respuesta";
    }
    catch (NumberFormatException e){
        System.out.println(e);
        respuesta ="{failure:true}";
        return respuesta;
    }
}

```

```

public String detalles(){

```

```

Map session = ActionContext.getContext().getSession();

UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

ordenload = (OrdenCompraJoinDTO) session.get("ordenload");

if( userlogin.getTipo_usuario().getId()== 1 || userlogin.getId_usuario().longValue() ==
ordenload.getIdUsuario().getId_usuario().longValue() ){

    compras = administradorCompras.listar(ordenload.getIdCompra());

    listardetalles = "{\"sucess\":true, \"totalCount\":" + "" + compras.size() + "\", "+ "\"data\":[\"";

    for(int i = 0; i < compras.size(); i++){

        listardetalles = listardetalles + "{\"producto\":" +
compras.get(i).getIdProducto().getNombre() +

                                "\", \"proveedor\":" +
compras.get(i).getIdProducto().getIdProveedor().getNombre() +

                                "\" +
compras.get(i).getIdProducto().getIdProveedor().getApelliop() +

                                "\" +
compras.get(i).getIdProducto().getIdProveedor().getApellidom() +

                                "\", \"fabricante\":" +
compras.get(i).getIdProducto().getIdFabricante().getFabricante() +

                                "\", \"catidad\":" + compras.get(i).getTotalproductos()

+

                                "\", \"total\":" + compras.get(i).getTotal()+ "\" },";

    }

    listardetalles = listardetalles.substring(0, listardetalles.length()-1);

    listardetalles = listardetalles + "]}";

}

return "detalles";

}

public String listarCompra(){

```

```
        return "listarCompra";
    }

    public String getListarordenes() {
        return listarordenes;
    }

    public void setListarproductos(String listarordenes) {
        this.listarordenes = listarordenes;
    }

    public String getIdOrden() {
        return idOrden;
    }

    public void setIdOrden(String idOrden) {
        this.idOrden = idOrden;
    }

    public String getRespuesta() {
        return respuesta;
    }

    public void setRespuesta(String respuesta) {
        this.respuesta = respuesta;
    }

    public String getListardetalles() {
```

```
        return listardetalles;
    }

    public void setListardetalles(String listardetalles) {
        this.listardetalles = listardetalles;
    }
}
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.action;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import java.util.List;
import java.util.Map;
import mx.uam.azc.cbi.data.ServicioJoinDTO;
import mx.uam.azc.cbi.data.ServicioSaveDTO;
import mx.uam.azc.cbi.data.TiemposSaveDTO;
import mx.uam.azc.cbi.data.UsuarioJoinDTO;
import mx.uam.azc.cbi.interfaces.AdminServicios;
import mx.uam.azc.cbi.interfaces.AdminTiempos;
import mx.uam.azc.cbi.logica.TimeStamp;
import org.springframework.beans.factory.annotation.Autowired;

/**
```

\*

\* @author manzano

\*/

```
public class ServiciosAction extends ActionSupport{
```

```
    @Autowired
```

```
    private AdminServicios administradorServicios;
```

```
    @Autowired
```

```
    private AdminTiempos administradorTiempos;
```

```
    private String listarservicios;
```

```
    private String respuesta;
```

```
    private String descripcion;
```

```
    private String estado;
```

```
    private String idServicio;
```

```
    private String tipo;
```

```
    private String cliente;
```

```
    private String proveedor;
```

```
    private String prioridad;
```

```
    private String services;
```

```
    private List<ServicioJoinDTO> servicios;
```

```
    private String service;
```

```
    private ServicioSaveDTO servicio = new ServicioSaveDTO();
```

```
    private final TiemposSaveDTO tiempos = new TiemposSaveDTO();
```

```
    private ServicioJoinDTO loadservice;
```

```
    public String cargar(){
```

```

try{

    Map session = ActionContext.getContext().getSession();

    loadservice = (ServicioJoinDTO) session.get("serviceload");

    System.out.println("Después del get");

    service = "{success: true, data: { \"cliente\":\":" + loadservice.getIdCliente().getNombre() + " "
+ loadservice.getIdCliente().getApelliop() + " " + loadservice.getIdCliente().getApellidom());

        if(loadservice.getIdProveedor() == null){

            service = service + "\", \"proveedor\":\":" ";

        }

        else{

            service = service + "\", \"proveedor\":\":" +
loadservice.getIdProveedor().getNombre() + " " + loadservice.getIdProveedor().getApelliop() + " " +
loadservice.getIdProveedor().getApellidom());

        }

        service = service + " " + "\", \"estado\":\":" + loadservice.getStatus().getEstado()

            + " " + "\", \"prioridad\":\":" + loadservice.getPriority()

            + " " + "\", \"services\":\":" +
loadservice.getServicio().getServicios()

            + "\", \"descripcion\": \":" + loadservice.getDescripcion() +
"\":}}";

        System.out.println("Cargar");

        System.out.println(service);

    }

    catch(Exception e){

    }

    return "loadservice";

}

```

```

public String listar(){
    Map session = ActionContext.getContext().getSession();
    session.remove("service");
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    if(userlogin.getTipo_usuario().getId() != 2)
        return "listar";
    else
        return "listaruser";
}

```

```

public String listarservicios(){
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    if(userlogin.getTipo_usuario().getId() == 1){
        servicios = administradorServicios.listar();
    }
    if(userlogin.getTipo_usuario().getId() == 2){
        servicios = administradorServicios.misservicios(userlogin.getId_usuario());
    }
    if(userlogin.getTipo_usuario().getId() == 3){
        servicios = administradorServicios.misserviciosp(userlogin.getId_usuario());
    }
}

```

```

listarservicios = "{\"sucess\":true, \"totalCount\":" + "\"" + servicios.size() + "\", "+ "\"data\":[";
    for(int i = 0; i<servicios.size();i++) {

```



```

listarservicios = listarservicios + "{\idServicio\:\\" + servicios.get(i).getIdServicio() +
    "\", \cliente\:\\" + servicios.get(i).getIdCliente().getNombre() + " " +
    " " + servicios.get(i).getIdCliente().getApelliop() +
    " " + servicios.get(i).getIdCliente().getApellidom());

if(servicios.get(i).getIdProveedor() == null ){
    listarservicios = listarservicios + "\", \proveedor\:\\" ";
}

else{
    listarservicios = listarservicios + "\", \proveedor\:\\" +
servicios.get(i).getIdProveedor().getNombre() + " " +
    " " + servicios.get(i).getIdProveedor().getApelliop() +
    " " + servicios.get(i).getIdProveedor().getApellidom());
}

listarservicios = listarservicios + "\", \modificado\:\\" +
servicios.get(i).getIdUltima().getNombre() + " " +
    " " + servicios.get(i).getIdUltima().getApelliop() +
    " " + servicios.get(i).getIdUltima().getApellidom() +
    "\", \estado\:\\" + servicios.get(i).getStatus().getEstado()+ "\",
\descripcion\:\\" + servicios.get(i).getDescripcion() +
    "\", \servicio\:\\" + servicios.get(i).getServicio().getServicios()+
    "\", \prioridad\:\\" + servicios.get(i).getPriority()+ "\", \ultima\:\\" +
TimeStamp.Timestamp(servicios.get(i).getUltima_modificacion()) +
    "\", \Fecha\:\\" +
TimeStamp.Timestamp(servicios.get(i).getFecha_peticion()+ "\", },";
}

listarservicios = listarservicios.substring(0, listarservicios.length()-1);

listarservicios = listarservicios + "]}";

return "listarservicios";
}

```

```

public String insertar(){
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    if(userlogin.getTipo_usuario().getId() == 1){
        return "insertar_admin";
    }
    if(userlogin.getTipo_usuario().getId() == 2){
        return "insertar_user";
    }
    if(userlogin.getTipo_usuario().getId() == 3){
        return "insertar_proveedor";
    }
    return "insertar_user";
}

```

```

public String agregar(){
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    servicio.setDescripcion(descripcion);
    servicio.setFecha_peticion(TimeStamp.Fecha());
    servicio.setUltima(userlogin.getId_usuario());
    servicio.setUltima_modificacion(TimeStamp.Fecha());
    try{
        servicio.setServicio(Integer.parseInt(services));
        servicio.setPrioridad(Integer.parseInt(prioridad));
    }
    catch (NumberFormatException e){

```

```
        System.out.println(e);

        respuesta = "{failure:true}";

        return respuesta;

    }

    if(userlogin.getTipo_usuario().getId() == 1){

        servicio.setStatus(2);

        try{

            servicio.setIdCliente(Long.parseLong(cliente));

            servicio.setIdProveedor(Long.parseLong(proveedor));

        }

        catch (NumberFormatException e){

            System.out.println(e);

            respuesta = "{failure:true}";

            return respuesta;

        }

    }

}
```

```
if(userlogin.getTipo_usuario().getId() == 2){

    servicio.setIdCliente(userlogin.getId_usuario());

    servicio.setIdProveedor(11);

    servicio.setStatus(2);

}
```

```
if(userlogin.getTipo_usuario().getId() == 3 ){

    servicio.setStatus(2);

    try{

        servicio.setIdCliente(Long.parseLong(cliente));

        servicio.setIdProveedor(userlogin.getId_usuario());

    }
```

```

    }

    catch (NumberFormatException e){
        System.out.println(e);
        respuesta = "{failure:true}";
        return respuesta;
    }
}

administradorServicios.insertar(servicio);

respuesta = "{success:true,result:\'El servicio ha sido agregado\'}";
return "respuesta";
}

public String form_actualizar(){
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    if(userlogin.getTipo_usuario().getId() == 1)
        return "form_actualizar";
    else
        return "form_actualizar_proveedor";
}

public String load(){
    try{
        long id = Long.parseLong(idServicio);
        ServicioJoinDTO serviceload = administradorServicios.loadservice(id);
    }
}

```

```

    Map session = ActionContext.getContext().getSession();

    session.put("serviceload",serviceload);

    respuesta ="{success:true}";

    return "respuesta";

}

catch (NumberFormatException e){

    System.out.println(e);

    respuesta ="{failure:true}";

    return respuesta;

}

}

}

public String actualizar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    loadservice = (ServicioJoinDTO) session.get("serviceload");

    try{

        servicio = administradorServicios.load(loadservice.getIdServicio().longValue());

    }

    catch(ExceptionInInitializerError e){

        System.out.println(e.toString());

    }

    System.out.println("despues de servicio load");

    int state;

    long cl;

    long pr;

    if (userlogin.getTipo_usuario().getId() == 1){

        try{

```

```
        cl = Long.parseLong(cliente);
    }
    catch (NumberFormatException e){
        cl = servicio.getIdCliente();
    }
    servicio.setIdCliente(cl);

    try{
        pr = Long.parseLong(proveedor);
    }
    catch (NumberFormatException e){
        pr = servicio.getIdProveedor();
    }
    servicio.setIdProveedor(pr);
}

if(userlogin.getTipo_usuario().getId() == 2){
    try{
        cl = Long.parseLong(cliente);
    }
    catch (NumberFormatException e){
        cl = servicio.getIdCliente();
    }
    servicio.setIdCliente(cl);
}

try{
    state = Integer.parseInt(estados);
}
}
```

```
catch (NumberFormatException e){
    state = servicio.getStatus();
}

servicio.setStatus(state);
servicio.setUltima(userlogin.getId_usuario().longValue());
servicio.setUltima_modificacion(TimeStamp.Fecha());
servicio.setDescripcion(descripcion);

tiempos.setIdServicio(loadservice.getIdServicio().longValue());
tiempos.setIdProveedor(loadservice.getIdProveedor().getId_usuario().longValue());
tiempos.setIdUltima(userlogin.getId_usuario().longValue());
tiempos.setEstado(loadservice.getStatus().getId());
tiempos.setFecha_modificacion(TimeStamp.Fecha());
tiempos.setResumen(loadservice.getDescripcion());
tiempos.setPrioridad(loadservice.getPrioridad());

if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario().longValue() ==
servicio.getIdProveedor()){
    administradorServicios.actualizar(servicio);
    administradorTiempos.insertar(tiempos);
    respuesta = "{success:true,result:\\"El servicio ha sido actualizado\\"}";
    return "respuesta";
}
else{
    respuesta = "{failure:true,result:\\"No tienes permisos para agregar\\"}";
    return "respuesta";
}
```





```
        return servicios;
    }

    public void setServicios(List<ServicioJoinDTO> servicios) {
        this.servicios = servicios;
    }

    public String getIdServicio() {
        return idServicio;
    }

    public void setIdServicio(String idServicio) {
        this.idServicio = idServicio;
    }

    public String getEstado() {
        return estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public String getService() {
        return service;
    }

    public void setService(String service) {
```

```
        this.service = service;
    }

    public String getTipo() {
        return tipo;
    }

    public void setTipo(String tipo) {
        this.tipo = tipo;
    }

    public String getCliente() {
        return cliente;
    }

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

    public String getProveedor() {
        return proveedor;
    }

    public void setProveedor(String proveedor) {
        this.proveedor = proveedor;
    }

    public String getPrioridad() {
```

```
        return prioridad;
    }

    public void setPrioridad(String prioridad) {
        this.prioridad = prioridad;
    }

    public ServicioSaveDTO getServicio() {
        return servicio;
    }

    public void setServicio(ServicioSaveDTO servicio) {
        this.servicio = servicio;
    }

    public String getServices() {
        return services;
    }

    public void setServices(String services) {
        this.services = services;
    }

}

/*
```

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package mx.uam.azc.cbi.action;
```

```
import com.opensymphony.xwork2.ActionContext;
```

```
import com.opensymphony.xwork2.ActionSupport;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
import mx.uam.azc.cbi.data.ProductoJoinDTO;
```

```
import mx.uam.azc.cbi.data.ProductoSaveDTO;
```

```
import mx.uam.azc.cbi.data.UsuarioJoinDTO;
```

```
import mx.uam.azc.cbi.interfaces.AdminProductos;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
public class ProductosAction extends ActionSupport{
```

```
    @Autowired
```

```
    private AdminProductos administradorProductos;
```

```
    private List<ProductoJoinDTO> productos;
```

```
    private ProductoSaveDTO producto = new ProductoSaveDTO();
```

```

private ProductoJoinDTO productload = new ProductoJoinDTO();

private String listarproductos;

private String respuesta;

private String idProducto;

private String nombre;

private String codigo;

private String descripcion;

private String fabricante;

private String proveedor;

private String product;

private String precio;

private String descuento;

private String tipo;

public String cargar(){
    try{
        Map session = ActionContext.getContext().getSession();
        productload = (ProductoJoinDTO) session.get("productload");
        product = "{success: true, data: { \"nombre\": \"\" + productload.getNombre() +
            \"\", \"codigo\": \"\" + productload.getCodigo() +
            \"\", \"fabricante\": \"\" + productload.getIdFabricante().getFabricante() +
            \"\", \"proveedor\": \"\" + productload.getIdProveedor().getNombre() + \" \" +
productload.getIdProveedor().getApelliop()+ \" \" + productload.getIdProveedor().getApellidom()+
            \"\", \"descuento\": \"\" + productload.getDescuento() +
            \"\", \"tipo\": \"\" + productload.getTipoVenta() +
            \"\", \"precio\": \"\" + productload.getPrecio_unitario() +
            \"\", \"descripcion\": \"\" + productload.getDescripcion() +\"\"}}";
    }
}

```

```
}  
catch(Exception e){  
  
}  
return "loadproducto";  
}
```

```
public String listar(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    if(userlogin.getTipo_usuario().getId() == 1)  
        return "admin";  
    else  
        return "user";  
}
```

```
public String load(){  
    try{  
        long id = Long.parseLong(idProducto);  
        productload = administradorProductos.loadproduct(id);  
        Map session = ActionContext.getContext().getSession();  
        session.put("productload",productload);  
        respuesta ="{success:true}";  
        return "respuesta";  
    }  
    catch (NumberFormatException e){  
        System.out.println(e);  
    }  
}
```

```

        respuesta = "{failure:true}";

        return respuesta;

    }

}

public String listarproductos(){

    productos = administradorProductos.listar();

    listarproductos = "{\"sucess\":true, \"totalCount\": \"\" + \"\" + productos.size() + \"\", \"+ \"data\":[\";

        for(int i = 0; i<productos.size();i++) {

            listarproductos = listarproductos + \"{idProducto\": \"\"+
productos.get(i).getIdProducto() +

                "\", \"nombre\": \"\"+productos.get(i).getNombre()+

                "\",

\"fabricante\": \"\"+productos.get(i).getIdFabricante().getFabricante()+

                "\", \"proveedor\": \"\" +

productos.get(i).getIdProveedor().getNombre() +

                \" \" +

productos.get(i).getIdProveedor().getApelliop() +

                \" \" +

productos.get(i).getIdProveedor().getApellidom() +

                "\", \"codigo\": \"\" + productos.get(i).getCodigo() +

                "\", \"descuento\": \"\" + productos.get(i).getDescuento() +

\"%\" +

                "\", \"tipo\": \"\" + productos.get(i).getTipoVenta() +

                "\", \"precio\": \"\" + \"$\" +

productos.get(i).getPrecio_unitario() +

                "\", \"descripcion\": \"\" + productos.get(i).getDescripcion()+

                \"\", \"\",

            }

            listarproductos = listarproductos.substring(0, listarproductos.length()-1);

            listarproductos = listarproductos + \"]}";

```

```
    return "listarproductos";  
}
```

```
public String insertar(){  
    return "insertar";  
}
```

```
public String agregar(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    if(userlogin.getTipo_usuario().getId() != 2 ){  
        int des;  
        long pre;  
        int tip;  
        try{  
            des = Integer.parseInt(descuento);  
        }  
        catch (NumberFormatException e){  
            respuesta = "{failure:true,result:\'El descuento no es valido\'}";  
            return "respuesta";  
        }  
        try{  
            pre = Long.parseLong(precio);  
        }  
        catch (NumberFormatException e){  
            respuesta = "{failure:true,result:\'El precio no es valido\'}";  
            return "respuesta";  
        }  
    }  
}
```



```

try{
    tip = Integer.parseInt(tipo);
}
catch (NumberFormatException e){
    respuesta = "{failure:true,result:\\"El tipo no es valido\\"}";
    return "respuesta";
}

producto.setIdProveedor(Long.parseLong(proveedor));
producto.setIdFabricante(Long.parseLong(fabricante));
producto.setCodigo(codigo);
producto.setNombre(nombre);
producto.setDescripcion(descripcion);
producto.setDescuento(des);
producto.setPrecio_unitario(pre);
producto.setTipo_venta(tip);
administradorProductos.insertar(producto);
respuesta = "{success:true,result:\\"El producto ha sido agregado\\"}";
return "respuesta";
}
else{
    respuesta = "{failure:true,result:\\"No tienes permisos para agregar\\"}";
    return "respuesta";
}
}

public String eliminar(){
    Map session = ActionContext.getContext().getSession();

```

```

UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

try{

long id = Long.parseLong(idProducto);

producto = administradorProductos.load(id);

}

catch (NumberFormatException e){

    System.out.println(e);

    respuesta ="{failure:true}";

}

if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario().longValue() ==
producto.getIdFabricante().longValue()){

    producto.setIsObsolete(true);

    administradorProductos.actualizar(producto);

    respuesta ="{success:true,result:\"El producto fue eliminado\"}";

    return "respuesta";

}

else{

    respuesta = "{failure:true,result:\"No tienes permisos para eliminar\"}";

    return "respuesta";

}

}

public String form_actualizar(){

    return "form_actualizar";

}

public String actualizar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

```

```
productload = (ProductoJoinDTO)session.get("productload");

try{

producto = administradorProductos.load(productload.getIdProducto().longValue());

}

catch (NumberFormatException e){

    System.out.println(e);

    respuesta = "{failure:true}";

}

long pro;

long fab;

long pre;

int des;

int tip;

if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario().longValue() ==
producto.getIdFabricante().longValue()){

    try{

        pro = Integer.parseInt(proveedor);

    }

    catch (NumberFormatException e){

        pro = producto.getIdProveedor();

    }

    try{

        fab = Integer.parseInt(fabricante);

    }

    catch (NumberFormatException e){

        fab = producto.getIdFabricante();

    }

    try{
```

```
        des = Integer.parseInt(descuento);
    }
    catch (NumberFormatException e){
        respuesta = "{failure:true,result:\'El descuento no es valido\'}";
        return "respuesta";
    }
    try{
        pre = Long.parseLong(precio);
    }
    catch (NumberFormatException e){
        respuesta = "{failure:true,result:\'El precio no es valido\'}";
        return "respuesta";
    }
    try{
        tip = Integer.parseInt(tipo);
    }
    catch (NumberFormatException e){
        respuesta = "{failure:true,result:\'El tipo no es valido\'}";
        return "respuesta";
    }
    producto.setIdProveedor(pro);
    producto.setIdFabricante(fab);
    producto.setPrecio_unitario(pre);
    producto.setDescuento(des);
    producto.setTipo_venta(tip);
    producto.setCodigo(codigo);
    producto.setNombre(nombre);
    producto.setDescripcion(descripcion);
```



```
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getFabricante() {
        return fabricante;
    }

    public void setFabricante(String fabricante) {
        this.fabricante = fabricante;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getProveedor() {
        return proveedor;
    }

    public void setProveedor(String proveedor) {
```

```
    this.proveedor = proveedor;
}

public String getListarproductos() {
    return listarproductos;
}

public void setListarproductos(String listarproductos) {
    this.listarproductos = listarproductos;
}

public String getIdProducto() {
    return idProducto;
}

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

public String getProduct() {
    return product;
}

public void setProduct(String product) {
    this.product = product;
}

public String getPrecio() {
```

```
        return precio;
    }

    public void setPrecio(String precio) {
        this.precio = precio;
    }

    public String getDescuento() {
        return descuento;
    }

    public void setDescuento(String descuento) {
        this.descuento = descuento;
    }

    public String getTipo() {
        return tipo;
    }

    public void setTipo(String tipo) {
        this.tipo = tipo;
    }
}

/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
```



```
*/
```

```
package mx.uam.azc.cbi.action;
```

```
import com.opensymphony.xwork2.ActionContext;
```

```
import com.opensymphony.xwork2.ActionSupport;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
import mx.uam.azc.cbi.data.MensajeJoinDTO;
```

```
import mx.uam.azc.cbi.data.MensajeSaveDTO;
```

```
import mx.uam.azc.cbi.data.TemaJoinDTO;
```

```
import mx.uam.azc.cbi.data.UsuarioJoinDTO;
```

```
import mx.uam.azc.cbi.logica.TimeStamp;
```

```
import mx.uam.azc.cbi.interfaces.AdminMensajes;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
public class MensajesAction extends ActionSupport {
```

```
    @Autowired
```

```
    private AdminMensajes administradorMensaje;
```

```
    private List<MensajeJoinDTO> mensajes;
```

```
    private MensajeSaveDTO mensaje;
```

```
    private String listarmensaje;
```

```

private String idTema;

private String idMensaje;

private String comentario;

private String respuesta;

private String message;

private MensajeJoinDTO loadmensaje;

public String listar(){

    return "listar";

}

public String cargar(){

    try{

        Map session = ActionContext.getContext().getSession();

        loadmensaje = (MensajeJoinDTO) session.get("loadmensaje");

        message = "{success: true, data: { \"comentario\": \"\" + loadmensaje.getComentario() +
\"\"}}";

    }

    catch(Exception e){

    }

    return "loadmensaje";

}

public String listarmensajes(){

    Map session = ActionContext.getContext().getSession();

    TemaJoinDTO topic = (TemaJoinDTO)session.get("loadtopic");

```

```

mensajes = administradorMensaje.listar(topic.getIdTema());

listarmensaje = "{\"sucess\":true, \"totalCount\":" + "\"" + mensajes.size() + "\", "+ "\"data\":[\"";

    for(int i = 0; i<mensajes.size();i++) {

        listarmensaje = listarmensaje + "{\"idMensaje\":\":" + mensajes.get(i).getIdMensaje() +
        "\", \"idTema\":\":" + mensajes.get(i).getIdTema().getIdTema() +

            "\", \"Tema\":\":" + mensajes.get(i).getIdTema().getNombre()+

            "\", \"usuario\": \":" + mensajes.get(i).getIdUsuario().getNombre() +

            " " + mensajes.get(i).getIdUsuario().getApelliop() +

            " " + mensajes.get(i).getIdUsuario().getApellidom() +

            "\", \"comentario\": \":" + mensajes.get(i).getComentario() +

            "\", \"fecha\": \":" + TimeStamp.Timestamp(mensajes.get(i).getDate())+ "\""

    },\"";

    }

    listarmensaje = listarmensaje.substring(0, listarmensaje.length()-1);

    listarmensaje = listarmensaje + "]}\"";

return "listarmensajes";

}

```

```

public String form_actualizar(){

    return "form_actualizar";

}

public String actualizar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    loadmensaje = (MensajeJoinDTO) session.get("loadmensaje");

    mensaje = administradorMensaje.load(loadmensaje.getIdMensaje());

    if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario() ==
mensaje.getIdUsuario()){

        mensaje.setComentario(comentario);
    }
}

```

```
administradorMensaje.actualizar(mensaje);

respuesta = "{success:true,result:\'El Mensaje ha sido actualizado\'}";

return "respuesta";

}

else{

    respuesta = "{failure:true,result:\'No tienes permisos para actualizar\'}";

    return "respuesta";

}

}
```

```
public String eliminar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    try{

        long id = Long.parseLong(idMensaje);

        mensaje = administradorMensaje.load(id);

    }

    catch (NumberFormatException e){

        System.out.println(e);

        respuesta ="{failure:true}";

    }

    if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario() ==
mensaje.getIdUsuario()){

        mensaje.setIsobsolete(true);

        administradorMensaje.actualizar(mensaje);

        respuesta ="{success:true,result:\'El Mensaje fue eliminado\'}";

        return "respuesta";

    }

}
```

```
}  
else{  
    respuesta = "{failure:true,result:\\"No tienes permisos para eliminar\\"}";  
    return "respuesta";  
}  
}
```

```
public String insertar(){  
    respuesta ="{success:true,result:\\"El comentario fue agregado\\"}";  
    return "insertar";  
}
```

```
public String agregar(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    TemaJoinDTO topic = (TemaJoinDTO)session.get("loadtopic");  
    mensaje = new MensajeSaveDTO();  
    mensaje.setComentario(comentario);  
    mensaje.setIIdTema(topic.getIIdTema());  
    mensaje.setIsobsolete(false);  
    mensaje.setDate(TimeStamp.Fecha());  
    mensaje.setIIdUsuario(userlogin.getIId_usuario());  
    administradorMensaje.insertar(mensaje);  
    mensaje = null;  
    respuesta ="{success:true,result:\\"El comentario fue agregado\\"}";  
    return "respuesta";  
}
```

```
public String load(){  
    try{  
        long id = Long.parseLong(idMensaje);  
        loadmensaje = administradorMensaje.loadmessage(id);  
        Map session = ActionContext.getContext().getSession();  
        session.put("loadmensaje",loadmensaje);  
        respuesta ="{success:true}";  
        return "respuesta";  
    }  
    catch (NumberFormatException e){  
        System.out.println(e);  
        respuesta ="{failure:true}";  
        return respuesta;  
    }  
}
```

```
public String getIdTema() {  
    return idTema;  
}
```

```
public void setIdTema(String idTema) {  
    this.idTema = idTema;  
}
```

```
public List<MensajeJoinDTO> getMensajes() {  
    return mensajes;  
}
```

```
public void setMensajes(List<MensajeJoinDTO> mensajes) {  
    this.mensajes = mensajes;  
}
```

```
public String getListarmensaje() {  
    return listarmensaje;  
}
```

```
public void setListarmensaje(String listarmensaje) {  
    this.listarmensaje = listarmensaje;  
}
```

```
public String getRespuesta() {  
    return respuesta;  
}
```

```
public void setRespuesta(String respuesta) {  
    this.respuesta = respuesta;  
}
```

```
public String getIdMensaje() {  
    return idMensaje;  
}
```

```
public void setIdMensaje(String idMensaje) {  
    this.idMensaje = idMensaje;  
}
```

```
public String getComentario() {
    return comentario;
}

public void setComentario(String comentario) {
    this.comentario = comentario;
}

public String getMessage() {
    return message;
}

public void setMessage(String message) {
    this.message = message;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.action;

import com.opensymphony.xwork2.ActionContext;
```



```
import com.opensymphony.xwork2.ActionSupport;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import mx.uam.azc.cbi.data.CarritoComprasDTO;

import mx.uam.azc.cbi.data.UsuarioJoinDTO;

import mx.uam.azc.cbi.interfaces.AdminUsuarios;

import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
public class GestionCRMAction extends ActionSupport {
```

```
    @Autowired
```

```
    private AdminUsuarios administradorUsuario;
```

```
    private UsuarioJoinDTO usuarioLogin = new UsuarioJoinDTO();
```

```
    private String chart;
```

```
    private String user;
```

```
    private String password;
```

```

public String valida() throws IOException{
    if(user.trim().equals("") || password.trim().equals("")){
        chart="{failure:true}";
    }
    else{
        String [] cadena = {user, password};
        usuarioLogin = administradorUsuario.login(cadena);
        if (usuarioLogin != null){
            Map session = ActionContext.getContext().getSession();
            session.put("user",usuarioLogin);
            chart="{success:true,result:\\"Usuario Valido\\"}";
        }
        else{
            chart="{failure:true}";
        }
    }
    return "valida";
}

```

```

public String index(){
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    if(userlogin.getTipo_usuario().getIId() == 1){
        userlogin = null;
        return "indexAdmin";
    }
}

```

```
else{
    userlogin = null;
    return "indexUser";
}
}
```

```
// Metodos Getters and Setters
```

```
public String getPassword() {
    return password;
}
```

```
public void setPassword(String password) {
    this.password = password;
}
```

```
public String getUser() {
    return user;
}
```

```
public void setUser(String user) {
    this.user = user;
}
```

```
public String getChart() {
    return chart;
}
```

```
public void setChart(String chart) {
    this.chart = chart;
}

public UsuarioJoinDTO getUsuarioLogin() {
    return usuarioLogin;
}

public void setUsuarioLogin(UsuarioJoinDTO usuarioLogin) {
    this.usuarioLogin = usuarioLogin;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.action;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import java.io.BufferedOutputStream;

import java.io.File;
import java.io.FileInputStream;
```

```
import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.math.BigDecimal;

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import mx.uam.azc.cbi.data.CompraJoinDTO;

import mx.uam.azc.cbi.data.OrdenCompraJoinDTO;

import mx.uam.azc.cbi.data.ProductoJoinDTO;

import mx.uam.azc.cbi.data.ServicioJoinDTO;

import mx.uam.azc.cbi.data.TiemposJoinDTO;

import mx.uam.azc.cbi.data.UsuarioJoinDTO;

import mx.uam.azc.cbi.interfaces.AdminCompras;

import mx.uam.azc.cbi.interfaces.AdminOrdenes;

import mx.uam.azc.cbi.interfaces.AdminProductos;

import mx.uam.azc.cbi.interfaces.AdminServicios;

import mx.uam.azc.cbi.interfaces.AdminTiempos;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperExportManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;
```

```
import net.sf.jxls.exception.ParsePropertyException;
import net.sf.jxls.transformer.XLSTransformer;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author Manzano
```

```
*/
```

```
public class DocumentosAction extends ActionSupport{
```

```
private OrdenCompraJoinDTO ordenload = new OrdenCompraJoinDTO();
```

```
@Autowired
```

```
private AdminProductos administradorProductos;
```

```
@Autowired
```

```
private AdminServicios administradorServicios;
```

```
@Autowired
```

```
private AdminTiempos administradorTiempos;
```

```
@Autowired
```

```
private AdminOrdenes administradorOrdenes;
```

```
@Autowired
```

```
private AdminCompras administradorCompras;
```

```
private List<ProductoJoinDTO> productos;
```

```
private List<ServicioJoinDTO> servicios;
```

```
private List<OrdenCompraJoinDTO> ordenes;
```

```
private List<CompraJoinDTO> compras;
```

```
private List<TiemposJoinDTO> tiempos;
```

```
private ServicioJoinDTO loadservice;
```

```
private InputStream inputStream = null;
```

```
/*public String excel() throws IOException, ParsePropertyException, InvalidFormatException {  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    productos = administradorProductos.listar();  
    String archivo = CrearArchivoExcel(productos, userlogin.getId_credencial().getUsername(),  
    "");  
    inputStream = new FileInputStream(new File(archivo));  
    return "excel";  
}*/
```

```
public String excelOrden() throws IOException, ParsePropertyException, InvalidFormatException  
{
```

```

Map session = ActionContext.getContext().getSession();

UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

if(userlogin.getTipo_usuario().getId() == 1){

    ordenes = administradorOrdenes.listar();

}

else {

    ordenes = administradorOrdenes.listar(userlogin.getId_usuario().longValue());

}

String archivo = CrearArchivoExcel(ordenes, userlogin.getId_credencial().getUsername(),
"Orden");

InputStream = new FileInputStream(new File(archivo));

return "excelOrden";

}

```

```

public String excelordendetalle() throws IOException, ParsePropertyException,
InvalidFormatException {

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    ordenload = (OrdenCompraJoinDTO) session.get("ordenload");

    compras = administradorCompras.listar(ordenload.getIdCompra());

    String archivo = CrearArchivoExcel(compras, userlogin.getId_credencial().getUsername(),
"Detalle");

    InputStream = new FileInputStream(new File(archivo));

    return "excelordendetalle";

}

```



```

public String excelservice() throws IOException, ParsePropertyException,
InvalidFormatException {

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    if(userlogin.getTipo_usuario().getId() == 1){

        servicios = administradorServicios.listar();

    }

    if(userlogin.getTipo_usuario().getId() == 2){

        servicios = administradorServicios.misservicios(userlogin.getId_usuario());

    }

    if(userlogin.getTipo_usuario().getId() == 3){

        servicios = administradorServicios.misserviciosp(userlogin.getId_usuario());

    }

    String archivo = CrearArchivoExcel(servicios, userlogin.getId_credencial().getUsername(),
"Servicio");

    inputStream = new FileInputStream(new File(archivo));

    return "excelservice";

}

```

```

public String excelestados() throws IOException, ParsePropertyException,
InvalidFormatException {

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    loadservice = (ServicioJoinDTO) session.get("serviceload");

    tiempos = administradorTiempos.listar(loadservice.getIdServicio().longValue());

    String archivo = CrearArchivoExcel(tiempos, userlogin.getId_credencial().getUsername(),
"Estados");

    inputStream = new FileInputStream(new File(archivo));

    return "excelestados";

}

```

```
/* public String pdf() throws Exception{  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    inputStream = new FileInputStream(new File("C:\\tempo\\reporte2.pdf"));  
    System.out.println("Despues de crear Documento en PDF");  
    return "pdf";  
} */
```

```
public String pdfOrden() throws Exception{  
    System.out.println("PDForden");  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    String archivo = CrearArchivoPdf(userlogin.getId_credencial().getUsername(), "Orden");  
    inputStream = new FileInputStream(new File("C:\\tempo\\Orden"+archivo));  
    System.out.println("Despues de crear Documento en PDF");  
    return "pdfOrden";  
}
```

```
public String pdfOrdendetalle() throws Exception{  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    String archivo = CrearArchivoPdf(userlogin.getId_credencial().getUsername(),  
"OrdenDetalle");  
    inputStream = new FileInputStream(new File("C:\\tempo\\OrdenDetalle"+archivo));
```

```
//System.out.println("Despues de crear Documento en PDF");  
return "pdfOrdendetalle";  
}
```

```
public String pdfservice() throws Exception{  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    String archivo = CrearArchivoPdf(userlogin.getId_credencial().getUsername(), "Servicio");  
    inputStream = new FileInputStream(new File("C:\\tempo\\Servicios"+archivo));  
    //System.out.println("Despues de crear Documento en PDF");  
    return "pdfservice";  
}
```

```
public String pdfestados() throws Exception{  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    String archivo = CrearArchivoPdf(userlogin.getId_credencial().getUsername(),  
"ServicioDetalles");  
    inputStream = new FileInputStream(new File("C:\\tempo\\DetallesServicio"+archivo));  
    //System.out.println("Despues de crear Documento en PDF");  
    return "pdfestados";  
}
```

```
public String CrearArchivoExcel(List objetos, String name, String plantilla)throws IOException,  
ParsePropertyException, InvalidFormatException {  
    if(plantilla.equals("Orden")){
```

```

    Map<String, List<OrdenCompraJoinDTO>> beans = new HashMap<String,
List<OrdenCompraJoinDTO>>();

    beans.put("ordenes",objetos);

    name = name.concat(".xls");

    InputStream istream = new FileInputStream("C:/tempo/Ordenes.xls");

    XLSTransformer transformer = new XLSTransformer();

    String archivo = "C:/tempo/"+name;

    HSSFWorkbook workbook = (HSSFWorkbook) transformer.transformXLS(istream, beans);

    OutputStream ostream = new BufferedOutputStream(new FileOutputStream(archivo));

    workbook.write(ostream);

    ostream.flush();

    ostream.close();

    return archivo;

}

```

```

if(plantilla.equals("Detalle")){

    Map<String, List<CompraJoinDTO>> beans = new HashMap<String,
List<CompraJoinDTO>>();

    beans.put("detalles",objetos);

    name = name.concat(".xls");

    InputStream istream = new FileInputStream("C:/tempo/Detalles.xls");

    XLSTransformer transformer = new XLSTransformer();

    String archivo = "C:/tempo/"+name;

    HSSFWorkbook workbook = (HSSFWorkbook) transformer.transformXLS(istream, beans);

    OutputStream ostream = new BufferedOutputStream(new FileOutputStream(archivo));

    workbook.write(ostream);

    ostream.flush();

    ostream.close();

    return archivo;

}

```

```

}

if(plantilla.equals("Servicio")){

    Map<String, List<ServicioJoinDTO>> beans = new HashMap<String,
List<ServicioJoinDTO>>();

    beans.put("servicios",objetos);

    name = name.concat(".xls");

    InputStream istream = new FileInputStream("C:/tempo/Servicios.xls");

    XLSXTransformer transformer = new XLSXTransformer();

    String archivo = "C:/tempo/"+name;

    HSSFWorkbook workbook = (HSSFWorkbook) transformer.transformXLS(istream, beans);

    OutputStream ostream = new BufferedOutputStream(new FileOutputStream(archivo));

    workbook.write(ostream);

    ostream.flush();

    ostream.close();

    return archivo;

}

if(plantilla.equals("Estados")){

    Map<String, List<TiemposJoinDTO>> beans = new HashMap<String,
List<TiemposJoinDTO>>();

    beans.put("estados",objetos);

    name = name.concat(".xls");

    InputStream istream = new FileInputStream("C:/tempo/Estados.xls");

    XLSXTransformer transformer = new XLSXTransformer();

    String archivo = "C:/tempo/"+name;

    HSSFWorkbook workbook = (HSSFWorkbook) transformer.transformXLS(istream, beans);

    OutputStream ostream = new BufferedOutputStream(new FileOutputStream(archivo));

    workbook.write(ostream);

```

```

        ostream.flush();

        ostream.close();

        return archivo;
    }

    return null;
}

public String CrearArchivoPdf(String name, String reporte) throws Exception{
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    ordenload = (OrdenCompraJoinDTO) session.get("ordenload");
    loadservice = (ServicioJoinDTO) session.get("serviceload");
    InputStream istream;
    JasperDesign design;
    JasperReport report;
    name = name.concat(".pdf");

    if(reporte.equals("Orden")){
        Map params = new HashMap();

        if(userlogin.getTipo_usuario().getId() == 1){
            istream = new FileInputStream("C:/tempo/OrdenAdmin.jrxml"); //Sin parametros
            design = JRXmlLoader.load(istream);
            report = JasperCompileManager.compileReport(design);

        }
        else{

```

```

    istream = new FileInputStream("C:/tempo/OrdenCliente.jrxml"); //Recibe parametro Id
cliente
    design = JRXmlLoader.load(istream);

    report = JasperCompileManager.compileReport(design);

    BigDecimal b = new BigDecimal(userlogin.getId_usuario().longValue());

    params.put("id_usuario", b);

    }

    istream.close();

    Connection connection = getConnection();

    try{

        JasperPrint print = JasperFillManager.fillReport( report, params, connection );

        JasperExportManager.exportReportToPdfFile( print, "c:\\tempo\\Orden"+name );

        System.out.println("REPORTES LISTOS");

    }

    finally{

        connection.close();

    }

    return name;

}

```

```

    if(reporte.equals("OrdenDetalle")){

        istream = new FileInputStream("C:/tempo/OrdenDetalle.jrxml"); //Recibe el parametro Id
Orden
        design = JRXmlLoader.load(istream);

        report = JasperCompileManager.compileReport(design);

        istream.close();

        report = JasperCompileManager.compileReport(design);
    }
}

```

```

        BigDecimal b = new BigDecimal(ordenload.getIdCompra());

        Map params = new HashMap();

        params.put("id_orden", b );

        Connection connection = getConnection();

    try{

        JasperPrint print = JasperFillManager.fillReport( report, params, connection );

        JasperExportManager.exportReportToPdfFile( print, "c:\\tempo\\OrdenDetalle"+name );

        System.out.println("REPORTES LISTOS");

    }

    finally{

        connection.close();

    }

    return name;

}

if(reporte.equals("Servicio")){

    Map params = new HashMap();

    if(userlogin.getTipo_usuario().getId() == 1){

        istream = new FileInputStream("C:/tempo/ServicioAdmin.jrxml"); //Sin parametros

        design = JRXmlLoader.load(istream);

        report = JasperCompileManager.compileReport(design);

    }

    else{

        cliente

        istream = new FileInputStream("C:/tempo/ServicioClientes.jrxml"); //Recibe parametro Id

        design = JRXmlLoader.load(istream);

```



```

report = JasperCompileManager.compileReport(design);

BigDecimal b = new BigDecimal(userlogin.getId_usuario());
params.put( "id_cliente", b );
}
istream.close();
Connection connection = getConnection();
try{
    JasperPrint print = JasperFillManager.fillReport( report, params, connection );
    JasperExportManager.exportReportToPdfFile( print, "c:\\tempo\\Servicios"+name );
    System.out.println("REPORTES LISTOS");
}
finally{
    connection.close();
}

return name;

}

```

```

if(reporte.equals("ServicioDetalles")){
    istream = new FileInputStream("C:/tempo/ServiciosDetalles.jrxml"); //Recibe Id servicio
    design = JRXmlLoader.load(istream);
    report = JasperCompileManager.compileReport(design);
    BigDecimal b = new BigDecimal(loadservice.getIdServicio());
}

```

```

Map params = new HashMap();

params.put("id_servicio", b );

Connection connection = getConnection();

try{

    JasperPrint print = JasperFillManager.fillReport( report, params, connection );

    JasperExportManager.exportReportToPdfFile( print, "c:\\tempo\\DetallesServicio"+name );

    System.out.println("REPORTES LISTOS");

}

finally{

    connection.close();

}

return name;

}

//ClassLoader loader = GenerarPDF.class.getClassLoader();

    /*istream = new FileInputStream("C:/tempo/ListaServicios.jrxml");
//loader.getResourceAsStream("mx/uam/azc/cbi/util/ListaServicios.jrxml");

System.out.println(istream);

design = JRXmlLoader.load(istream);

report = JasperCompileManager.compileReport(design);

    BigDecimal b = new BigDecimal("50000");

    Map params = new HashMap();

    params.put( "anio", "2007" );

    params.put( "creditoMinimo", b );

Connection connection = getConnection();

try{

```

```

    JasperPrint print = JasperFillManager.fillReport( report, params, connection );
    JasperExportManager.exportReportToPdfFile( print, "c:\\tempo\\reporte2.pdf" );
    System.out.println("REPORTES LISTOS");
}
finally{
    connection.close();
}*/
return "";
}

```

```

private static void setSystemProperties(){
    System.setProperty( "jasper.reports.compile.keep.java.file", "false" );
}

```

```

private static Connection getConnection() throws Exception{
    Class.forName( "org.postgresql.Driver" );
    String url = "jdbc:postgresql://localhost:5432/GestionCRM?autoReconnect=true";
    return DriverManager.getConnection( url, "postgres", "g3r4rd0m4nz4n0!");
}

```

```

public InputStream getInputStream() {
    return inputStream;
}

```

```

public void setInputStream(InputStream inputStream) {
    this.inputStream = inputStream;
}

```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.action;
```

```
import com.opensymphony.xwork2.ActionContext;
```

```
import com.opensymphony.xwork2.ActionSupport;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
import mx.uam.azc.cbi.data.CredencialDTO;
```

```
import mx.uam.azc.cbi.data.UsuarioJoinDTO;
```

```
import mx.uam.azc.cbi.logica.Encryption;
```

```
import mx.uam.azc.cbi.interfaces.AdminCredenciales;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```

public class CredencialAction extends ActionSupport{

    @Autowired

    private AdminCredenciales administradorCredencial;

    private List<UsuarioJoinDTO> usuarios;

    private String listausuarios;

    private String respuesta;

    private String idCredencial;

    private String password;

    private String passcfm;

    private static String id;

    public String insertar(){

        return "insertar";

    }

    public String agregar(){

        respuesta = "{success:true,result:\\"El Usuario ha sido agregado\\"}";

        return "respuesta";

    }

    public String actualizar(){

        Map session = ActionContext.getContext().getSession();

        UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

```

```

CredencialDTO credencial =
administradorCredencial.load(userlogin.getId_credencial().getIdCredencial());

if(password.equals(passcfrm)){

    Encryption encryption = new Encryption();

    String pass [] = encryption.getEncryptedPassword(password);

    credencial.setPassword(pass[0]);

    credencial.setSalt(pass[1]);

    administradorCredencial.actualizar(credencial);

    encryption = null;

    credencial = null;

    respuesta = "{success:true,result:\\"Cambio de contraseña exitoso\\"}";

    return "respuesta";

}

respuesta = "{failure:true,result:\\"Las contraseñas no conciden\\"}";

return "respuesta";

}

```

```

public String eliminar(){

    Map session = ActionContext.getContext().getSession();

    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");

    id = idCredencial;

    if(userlogin.getTipo_usuario().getId() == 1){

        CredencialDTO credencial = administradorCredencial.load(Long.parseLong(idCredencial));

        credencial.setIsobsolete(true);

        administradorCredencial.actualizar(credencial);

        respuesta ="{success:true,result:\\"El usuario fue eliminado\\"}";

        return "respuesta";

    }

}

```

```

else{
    respuesta = "{failure:true,result:\\"No tienes permisos para eliminar\\"}";
    return "respuesta";
}
}

public String borrar(){
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    System.out.println(userlogin.getId_credencial().getIdCredencial());
    CredencialDTO credencial =
administradorCredencial.load(userlogin.getId_credencial().getIdCredencial());
    credencial.setIsobsolete(true);
    respuesta ="{success:true,result:\\"El usuario fue eliminado\\"}";
    return "respuesta";
}

public String cambiarpass(){
    return "cambiarpass";
}

public String resetpass(){
    //System.out.println("Mandar mail con nuevas credenciales");
    //System.out.println(idCredencial);
    respuesta ="{success:true,result:\\"El usuario fue eliminado\\"}";
    return "respuesta";
}

```

```
public List<UsuarioJoinDTO> getUsuarios() {  
    return usuarios;  
}
```

```
public void setUsuarios(List<UsuarioJoinDTO> usuarios) {  
    this.usuarios = usuarios;  
}
```

```
public String getListausuarios() {  
    return listausuarios;  
}
```

```
public void setListausuarios(String listausuarios) {  
    this.listausuarios = listausuarios;  
}
```

```
public String getRespuesta() {  
    return respuesta;  
}
```

```
public void setRespuesta(String respuesta) {  
    this.respuesta = respuesta;  
}
```

```
public String getsetIdCredencial() {  
    return idCredencial;  
}
```



```
public void setIdCredencial(String idCredencial) {  
    this.idCredencial = idCredencial;  
}
```

```
public static String getId() {  
    return id;  
}
```

```
public static void setId(String id) {  
    CredencialAction.id = id;  
}
```

```
public String getPasscfrm() {  
    return passcfrm;  
}
```

```
public void setPasscfrm(String passcfrm) {  
    this.passcfrm = passcfrm;  
}
```

```
public String getPassword() {  
    return password;  
}
```

```
public void setPassword(String password) {  
    this.password = password;  
}
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.action;
```

```
import com.opensymphony.xwork2.ActionContext;
```

```
import com.opensymphony.xwork2.ActionSupport;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
import mx.uam.azc.cbi.data.CatCategoriasDTO;
```

```
import mx.uam.azc.cbi.data.CatEstadoServiciosDTO;
```

```
import mx.uam.azc.cbi.data.CatEstadosDTO;
```

```
import mx.uam.azc.cbi.data.CatFabricanteDTO;
```

```
import mx.uam.azc.cbi.data.CatServiciosDTO;
```

```
import mx.uam.azc.cbi.data.CatTipoUsuarioDTO;
```

```
import mx.uam.azc.cbi.data.ServicioJoinDTO;
```

```
import mx.uam.azc.cbi.data.UsuarioSaveDTO;
```

```
import mx.uam.azc.cbi.interfaces.AdminCombos;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```

public class CombosAction extends ActionSupport{

    @Autowired

    private AdminCombos administradorCombos;

    private List<CatEstadosDTO> estados;

    private List<CatEstadoServiciosDTO> estadoservicio;

    private List<CatTipoUsuarioDTO> tipousuarios;

    private List<CatCategoriasDTO> categorias;

    private List<CatFabricanteDTO> fabricantes;

    private List<UsuarioSaveDTO> proveedores;

    private List<CatServiciosDTO> servicios;

    private String tipousuario;

    private String estado;

    private String categoria;

    private String fabricante;

    private String proveedor;

    private String estadoservicios;

    private String servicio;

    private ServicioJoinDTO loadservice;

    public String estados() {

        estados = administradorCombos.leerEstados();

        estado = "{\"estados\":[" ;

        for(int i=0; i<estados.size(); i++){

            estado = estado + "{\"id\": \"" + estados.get(i).getId() + "\", \"estado\": \"" +
estados.get(i).getEstado() + "\"},";

```

```

    }

    estado = estado.substring(0, estado.length() - 1);

    estado = estado + "]]";

    return "estados";

}

```

```

public String servicios() {

    servicios = administradorCombos.leerservicios();

    servicio = "{\"servicios\":[" ;

    for(int i=0; i<servicios.size(); i++){

        servicio = servicio + "{\"id\": \"\" + servicios.get(i).getId() + "\", \"services\": \"\" +
servicios.get(i).getServicios() + "\"},";

    }

    servicio = servicio.substring(0, servicio.length() - 1);

    servicio = servicio + "]]";

    return "servicios";

}

```

```

public String tipousuario(){

    tipousuarios = administradorCombos.leerTipoUsuario();

    tipousuario = "{\"tipousuario\":[";

    for(int i=0; i<tipousuarios.size(); i++){

        tipousuario = tipousuario + "{\"id\": \"\" + tipousuarios.get(i).getId() + "\", \"tipo\": \"\" +
tipousuarios.get(i).getTipoUsuario() + "\"},";

    }

    tipousuario = tipousuario.substring(0, tipousuario.length() - 1);

    tipousuario = tipousuario + "]]";

}

```

```
return "tipousuario";
}

public String estadosservicio() {
    Map session = ActionContext.getContext().getSession();
    loadservice = (ServicioJoinDTO) session.get("serviceload");
    estadosservicio = administradorCombos.leerEstadosServicio();

    if(loadservice.getStatus().getId() == 1){
        estadosservicio.remove(0).getId();
    }

    if(loadservice.getStatus().getId() == 2){
        estadosservicio.remove(0).getId();
        estadosservicio.remove(1).getId();
        estadosservicio.remove(4).getId();
    }

    if(loadservice.getStatus().getId() == 3){
        estadosservicio.remove(0).getId();
        estadosservicio.remove(1).getId();
        estadosservicio.remove(2).getId();
    }

    if(loadservice.getStatus().getId() == 4){
        estadosservicio.remove(0).getId();
        estadosservicio.remove(1).getId();
        estadosservicio.remove(3).getId();
    }
}
```

```
}  
  
if(loadservice.getStatus().getId() == 5){  
    estadoservicio.remove(0).getId();  
    estadoservicio.remove(1).getId();  
    estadoservicio.remove(4).getId();  
}  
  
if(loadservice.getStatus().getId() == 6){  
    estadoservicio.remove(0);  
    estadoservicio.remove(1);  
    estadoservicio.remove(2);  
    estadoservicio.remove(3);  
    estadoservicio.remove(4);  
    estadoservicio.remove(5);  
    estadoservicio.remove(7);  
}  
  
if(loadservice.getStatus().getId() == 7){  
    estadoservicio.remove(0);  
    estadoservicio.remove(1);  
    estadoservicio.remove(2);  
    estadoservicio.remove(3);  
    estadoservicio.remove(4);  
    estadoservicio.remove(5);  
    estadoservicio.remove(6);  
}  
  
estadoservicios = "{\"estados\":[" ;  
for(int i=0; i<estadoservicio.size(); i++){
```

```
        estadoservicios = estadoservicios + "{\"id\": \"" + estadoservicio.get(i).getId() + "\",  
        \"estado\": \"" + estadoservicio.get(i).getEstado() + "\"},";
```

```
    }
```

```
    estadoservicios = estadoservicios.substring(0, estadoservicios.length() - 1);
```

```
    estadoservicios = estadoservicios + "]]";
```

```
    return "estadoservicio";
```

```
}
```

```
public String categorias() {
```

```
    categorias = administradorCombos.leerCategorias();
```

```
    categoria = "{\"categorias\":[\" ;
```

```
    for(int i=0; i<categorias.size(); i++){
```

```
        categoria = categoria + "{\"id\": \"" + categorias.get(i).getId() + "\", \"categoria\": \"" +  
        categorias.get(i).getCategoria() + "\"},";
```

```
    }
```

```
    categoria = categoria.substring(0, categoria.length() - 1);
```

```
    categoria = categoria + "]]";
```

```
    return "categorias";
```

```
}
```

```
public String fabricantes(){
```

```
    fabricantes = administradorCombos.leerfabricantes();
```

```
    fabricante = "{\"fabricantes\":[\" ;
```

```
    for(int i=0; i<fabricantes.size(); i++){
```

```
        fabricante = fabricante + "{\"id\": \"" + fabricantes.get(i).getId() + "\", \"fabricante\": \"" +  
        fabricantes.get(i).getFabricante() + "\"},";
```

```
    }
```

```
    fabricante = fabricante.substring(0, fabricante.length() - 1);
```

```
    fabricante = fabricante + "]]";
```

```

        return "fabricantes";
    }

    public String proveedores(){
        proveedores = administradorCombos.leerproveedores();
        proveedor = "{\"proveedores\":[" ;
        for(int i=0; i<proveedores.size(); i++){
            proveedor = proveedor + "{\"id\": \"\" + proveedores.get(i).getId_usuario() + "\",
            \"proveedor\": \"\" + proveedores.get(i).getNombre() + \" \" +
            proveedores.get(i).getApelliop() + \" \" +
            proveedores.get(i).getApellidom() + \"\"},";
        }
        proveedor = proveedor.substring(0, proveedor.length() - 1);
        proveedor = proveedor + "]}";
        return "proveedores";
    }

    public String clientes(){
        proveedores = administradorCombos.leerclientes();
        proveedor = "{\"clientes\":[" ;
        for(int i=0; i<proveedores.size(); i++){
            proveedor = proveedor + "{\"id\": \"\" + proveedores.get(i).getId_usuario() + "\", \"cliente\":
            \"\" + proveedores.get(i).getNombre() + \" \" +
            proveedores.get(i).getApelliop() + \" \" +
            proveedores.get(i).getApellidom() + \"\"},";
        }
    }

```



```
    proveedor = proveedor.substring(0, proveedor.length() - 1);  
    proveedor = proveedor + "]]";  
    return "clientes";  
}
```

```
public String getFabricante() {  
    return fabricante;  
}
```

```
public void setFabricante(String fabricante) {  
    this.fabricante = fabricante;  
}
```

```
public String getProveedor() {  
    return proveedor;  
}
```

```
public void setProveedor(String proveedor) {  
    this.proveedor = proveedor;  
}
```

```
public String getEstado() {  
    return estado;  
}
```

```
public void setEstado(String estado) {  
    this.estado = estado;  
}
```

```
public String getTipousuario() {  
    return tipousuario;  
}
```

```
public void setTipousuario(String tipousuario) {  
    this.tipousuario = tipousuario;  
}
```

```
public String getEstadoservicios() {  
    return estadoservicios;  
}
```

```
public void setEstadoservicios(String estadoservicios) {  
    this.estadoservicios = estadoservicios;  
}
```

```
public String getCategoria() {  
    return categoria;  
}
```

```
public void setCategoria(String categoria) {  
    this.categoria = categoria;  
}
```

```
public String getServicio() {  
    return servicio;  
}
```

```
public void setServicio(String servicio) {  
    this.servicio = servicio;  
}  
}  
  
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
*/  
  
package mx.uam.azc.cbi.action;  
  
import com.opensymphony.xwork2.ActionContext;  
import com.opensymphony.xwork2.ActionSupport;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Map;  
import mx.uam.azc.cbi.data.CarritoComprasDTO;  
import mx.uam.azc.cbi.data.CompraSaveDTO;  
import mx.uam.azc.cbi.data.OrdenCompraSaveDTO;  
import mx.uam.azc.cbi.data.ProductoJoinDTO;  
import mx.uam.azc.cbi.data.UsuarioJoinDTO;  
import mx.uam.azc.cbi.interfaces.AdminCompras;  
import mx.uam.azc.cbi.interfaces.AdminOrdenes;  
import mx.uam.azc.cbi.interfaces.AdminProductos;  
import mx.uam.azc.cbi.logica.TimeStamp;  
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
 *
 * @author Manzano
 */
public class CarritoAction extends ActionSupport {

    @Autowired
    private AdminProductos administradorProductos;

    @Autowired
    private AdminOrdenes administradorOrdenes;

    @Autowired
    private AdminCompras administradorCompras;

    private String idProducto;
    private String respuesta;
    private String listarcarrito;
    private CarritoComprasDTO carrito;
    private ProductoJoinDTO producto;
    private List<CarritoComprasDTO> lista;
    private OrdenCompraSaveDTO ordensave;

    public String agregar(){
        carrito = new CarritoComprasDTO();
```

```
int cantidad = 0;

Map session = ActionContext.getContext().getSession();

lista = (List<CarritoComprasDTO>)session.get("lista");

producto = administradorProductos.loadproduct(Long.parseLong(idProducto));

if (lista == null){

    lista = new ArrayList<CarritoComprasDTO>();

    carrito.setIdProducto(producto);

    lista.add(carrito);

    session.put("lista", lista);

    respuesta = "{success:true}";

    return "respuesta";

}

for(int i = 0 ; i < lista.size(); i++){

    if(lista.get(i).getIdProducto().getIdProducto().longValue() == Long.parseLong(idProducto)){

        cantidad = lista.get(i).getCantidad() + 1;

        lista.get(i).setCantidad(cantidad);

        session.put("lista", lista);

        respuesta = "{success:true}";

    }

    return "respuesta";

}

carrito.setIdProducto(producto);

lista.add(carrito);

session.put("lista", lista);

carrito = null;

respuesta = "{success:true}";

return "respuesta";
```

```
}
```

```
public String listar(){  
    return "listar";  
}
```

```
public String comprar(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO usuario = (UsuarioJoinDTO)session.get("user");  
    lista = (List<CarritoComprasDTO>)session.get("lista");  
    if(lista != null){  
        ordensave = new OrdenCompraSaveDTO();  
        ordensave.setIdUsuario(usuario.getId_usuario().longValue());  
        ordensave.setTotal(50);  
        ordensave.setFecha(TimeStamp.Fecha());  
        long orden = administradorOrdenes.insertar(ordensave);  
        CompraSaveDTO compras = new CompraSaveDTO();  
        long total = 0;  
        for(int i = 0 ; i < lista.size(); i++){  
            total = ((lista.get(i).getIdProducto().getPrecio_unitario()-  
(lista.get(i).getIdProducto().getPrecio_unitario() * lista.get(i).getIdProducto().getDescuento())/100) *  
lista.get(i).getCantidad());  
            compras.setIdOrden(orden);  
            compras.setIdProducto(lista.get(i).getIdProducto().getIdProducto().longValue());  
            compras.setTotalproductos(lista.get(i).getCantidad());  
            compras.setTotal(total);  
            administradorCompras.insertar(compras);  
        }  
    }  
}
```

```

        lista.clear();

        session.put("lista", lista);

        respuesta = "{success:true,result:\\"Los productos ha sido comprados\\"}";

        return "respuesta";
    }
    else{

        respuesta = "{failure:true,result:\\"No tiene productos para comprar\\"}";

        return "respuesta";
    }

}

public String eliminar(){

    Map session = ActionContext.getContext().getSession();

    lista = (List<CarritoComprasDTO>)session.get("lista");

    try{

        int index = Integer.parseInt(idProducto);

        lista.remove(index);

        session.put("lista", lista);

    }

    catch(NumberFormatException e){

        System.out.println(e);

    }

    respuesta = "{success:true,result:\\"La producto ha sido borrado\\"}";

    return "respuesta";

}

```

```

public String limpiar(){
    Map session = ActionContext.getContext().getSession();
    lista = (List<CarritoComprasDTO>)session.get("lista");
    lista.clear();
    session.put("lista", lista);
    respuesta = "{success:true,result:\\"La lista ha sido borrada\\"}";
    return "respuesta";
}

```

```

public String listarcarrito(){
    lista = new ArrayList<CarritoComprasDTO>();
    Map session = ActionContext.getContext().getSession();
    lista = (List<CarritoComprasDTO>)session.get("lista");
    listarcarrito = "{\\"sucess\\":true,\\"data\\":[";
    if(lista != null){
        for(int i = 0 ; i < lista.size(); i++){
            listarcarrito = listarcarrito + "{\\"idProducto\\":\\" + i +
                "\\", \\"nombre\\":\\" + lista.get(i).getIdProducto().getNombre() +
                "\\", \\"descripcion\\":\\" + lista.get(i).getIdProducto().getDescripcion() +
                "\\", \\"codigo\\":\\" + lista.get(i).getIdProducto().getCodigo() +
                "\\", \\"fabricante\\":\\" + lista.get(i).getIdProducto().getIdFabricante().getFabricante() +
                "\\", \\"cantidad\\":\\" + lista.get(i).getCantidad() +
                "\\", \\"total\\":\\" $" + ((lista.get(i).getIdProducto().getPrecio_unitario()-
                (lista.get(i).getIdProducto().getPrecio_unitario() * lista.get(i).getIdProducto().getDescuento())/100) *
                lista.get(i).getCantidad()) + "\\", \\"}";
        }
        listarcarrito = listarcarrito.substring(0, listarcarrito.length()-1);
        listarcarrito = listarcarrito + "]}";
    }
}

```



```
}  
lista = null;  
return "listarcarrito";  
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
public String getRespuesta() {  
    return respuesta;  
}
```

```
public void setRespuesta(String respuesta) {  
    this.respuesta = respuesta;  
}
```

```
public String getIdProducto() {  
    return idProducto;  
}
```

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

```
public String getListarcarrito() {  
    return listarcarrito;  
}
```

```
public void setListarcarrito(String listarcarrito) {  
    this.listarcarrito = listarcarrito;  
}  
}  
  
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
*/  
  
package mx.uam.azc.cbi.action;  
  
import com.opensymphony.xwork2.ActionContext;  
import com.opensymphony.xwork2.ActionSupport;  
  
import java.util.List;  
import java.util.Map;  
import mx.uam.azc.cbi.data.CredencialDTO;  
import mx.uam.azc.cbi.data.UsuarioJoinDTO;  
import mx.uam.azc.cbi.data.UsuarioSaveDTO;  
import mx.uam.azc.cbi.logica.Encryption;  
import mx.uam.azc.cbi.logica.TimeStamp;  
import mx.uam.azc.cbi.interfaces.AdminCredenciales;  
import mx.uam.azc.cbi.interfaces.AdminUsuarios;  
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
 *
 * @author manzano
 */
public class UsuariosAction extends ActionSupport{

    @Autowired
    private AdminUsuarios administradorUsuario;

    @Autowired
    private AdminCredenciales administradorCredencial;

    private List<UsuarioJoinDTO> usuarios;

    private String listausuarios;
    private String respuesta;

    private String idUsuario;
    private String nombre;
    private String apellidop;
    private String apellidom;
    private String rfc;
    private String telefono;
    private String tipo;
    private String mail;
    private String celular;
    private String colonia;
```

```

private String calle;

private String delegacion;

private String cp;

private String estado;

private String username;

private String nacimiento;

private String password;

private String passcfm;

private String loaduser;

UsuarioSaveDTO user;

public String cargar(){
    try{
        Map session = ActionContext.getContext().getSession();

        UsuarioJoinDTO usuario = (UsuarioJoinDTO) session.get("userload");

        loaduser = "{success: true, data: { \"nombre\": \"\" + usuario.getNombre() + "\", \"apellidop\":
        \"\"
            + usuario.getApelliop() + "\", \"apellidom\": \"\" + usuario.getApellidom() + "\",
        \"apellidom\": \"\" + usuario.getApellidom() + \"\"
            + "\", \"rfc\": \"\" + usuario.getRfc() + "\", \"telefono\": \"\" + usuario.getTelefono() + \"\"
            + "\", \"tipo\": \"\" + usuario.getTipo_usuario().getTipoUsuario() + "\", \"mail\": \"\" +
        usuario.getEmail() + \"\"
            + "\", \"celular\": \"\" + usuario.getCelular() + "\", \"colonia\": \"\" + usuario.getColonia() +
        \"\"
            + "\", \"calle\": \"\" + usuario.getCalle() + "\", \"delegacion\": \"\" + usuario.getDelegacion()
        + \"\"
            + "\", \"cp\": \"\" + usuario.getCp() + "\", \"estado\": \"\" + usuario.getEntidad().getEstado()
        + \"\"
            + "\", \"nacimiento\": \"\" + Timestamp.Timestamp(usuario.getFecha_nacimiento()) +
        \"\"}}";
    }
}

```

```

catch(Exception e){

}

return "loaduser";

}

public String perfil(){
try{
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO usuario = (UsuarioJoinDTO) session.get("user");
    loaduser = "{success: true, data: { \"nombre\": \"\" + usuario.getNombre() + "\", \"apellidop\": \"\"
+ usuario.getApelliop() + "\", \"apellidom\": \"\" + usuario.getApellidom() + "\",
\"apellidom\": \"\" + usuario.getApellidom() + \"\"
+ ", \"rfc\": \"\" + usuario.getRfc() + "\", \"telefono\": \"\" + usuario.getTelefono() + \"\"
+ ", \"tipo\": \"\" + usuario.getTipo_usuario().getTipoUsuario() + "\", \"mail\": \"\" +
usuario.getEmail() + \"\"
+ ", \"celular\": \"\" + usuario.getCelular() + "\", \"colonial\": \"\" + usuario.getColonia() +
\"\"
+ ", \"calle\": \"\" + usuario.getCalle() + "\", \"delegacion\": \"\" + usuario.getDelegacion()
+ \"\"
+ ", \"cp\": \"\" + usuario.getCp() + "\", \"estado\": \"\" + usuario.getEntidad().getEstado()
+ \"\"
+ ", \"nacimiento\": \"\" + TimeStamp.Timestamp(usuario.getFecha_nacimiento()) +
\"\"}}";
}
catch(Exception e){

}

return "loaduser";
}

```

```
}
```

```
public String listar(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    if(userlogin.getTipo_usuario().getId() == 1)  
        return "listaradmin";  
    else  
        return "listaruser";  
}
```

```
public String listarusuarios(){  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    session.remove("userload");  
    if(userlogin.getTipo_usuario().getId() == 1)  
        usuarios = administradorUsuario.listar();  
    else  
        usuarios = administradorUsuario.perfil(userlogin.getId_credencial().getUsername());  
  
    listausuarios = "{\"sucess\":true, \"totalCount\":" + "\"" + usuarios.size() + "\", "+ "\"data\":[\"";  
        for(int i = 0; i<usuarios.size();i++) {  
            listausuarios = listausuarios + "{\"idCredencial\":\","+  
usuarios.get(i).getId_credencial().getIdCredencial() +
```

```

        "\, \"idUsuario\":\":" + usuarios.get(i).getId_usuario() +
        "\, \"nombre\":\":" + usuarios.get(i).getNombre() + "\,
\"apellidop\": \":" + usuarios.get(i).getApelliop() +
        "\, \"apellidom\": \":" + usuarios.get(i).getApellidom() +
        "\, \"rfc\": \":" + usuarios.get(i).getRfc() + "\, \"telefono\": \":" +
usuarios.get(i).getTelefono()+
        "\, \"tipo\": \":" +
usuarios.get(i).getTipo_usuario().getTipoUsuario() + "\, \"mail\": \":" + usuarios.get(i).getEmail()+
        "\, \"celular\": \":" + usuarios.get(i).getCelular() + "\, \"colonia\":
\":" + usuarios.get(i).getColonia()+
        "\, \"calle\": \":" + usuarios.get(i).getCalle() + "\, \"delegacion\":
\":" + usuarios.get(i).getDelegacion()+
        "\, \"cp\": \":" + usuarios.get(i).getCp()+ "\, \"estado\": \":"
+ usuarios.get(i).getEntidad().getEstado()+
        "\, \"nacimiento\": \":" +
TimeStamp.Timestamp(usuarios.get(i).getFecha_nacimiento())+ "\, \"ingreso\": \":" +
TimeStamp.Timestamp(usuarios.get(i).getFecha_ingreso()+
        "\, \"username\":
\":" + usuarios.get(i).getId_credencial().getUsername()+"\}";
    }
    listausuarios = listausuarios.substring(0, listausuarios.length()-1);
    listausuarios = listausuarios + "]}";

    userlogin = null;

    return "listarusuarios";
}

public String insertar(){
    return "insertar";
}

public String agregar(){
    boolean valida = administradorCredencial.buscar(username);

```

```
if(username.equals("")){
    respuesta = "{failure:true,result:\\"Debe ingresar un nombre de usuario\\"}";
    return "respuesta";
}
if(valida == true){
    respuesta = "{failure:true,result:\\"El nombre de usuario ya existe\\"}";
    return "respuesta";
}
CredencialDTO credencial = new CredencialDTO();
Encryption e = new Encryption();
credencial.setUsername(username);
if(password.equals("")){
    respuesta = "{failure:true,result:\\"Debe ingresar una contraseña de usuario\\"}";
    return "respuesta";
}
if(!password.equals(passcfrm)){
    respuesta = "{failure:true,result:\\"Las contraseñas no coinciden\\"}";
    return "respuesta";
}
String pass [] = e.getEncryptedPassword(password);
credencial.setPassword(pass[0]);
credencial.setSalt(pass[1]);
long idCredencial = administradorCredencial.insertar(credencial);
user = new UsuarioSaveDTO();
user.setId_credencial(idCredencial);

user.setNombre(nombre);
```



```
user.setApelliop(apellidop);

user.setApellidom(apellidom);

user.setRfc(rfc);

if(tipo.equals("")){
    respuesta = "{failure:true,result:\\"Elija un tipo de usuario\\"}";
}

if(estado.equals("")){
    respuesta = "{failure:true,result:\\"Debe de elegir un estado\\"}";
}

user.setTipo_usuario(Integer.parseInt(tipo));

user.setFecha_nacimiento(TimeStamp.FechaNacimiento(nacimiento));

user.setFecha_ingreso(TimeStamp.Fecha());

user.setTelefono(telefono);

user.setCelular(celular);

user.setEmail(mail);

user.setCalle(calle);

user.setColonia(colonia);

user.setDelegacion(delegacion);
```

```
user.setCp(Integer.parseInt(cp));

user.setEntidad(Integer.parseInt(estado));

administradorUsuario.insertar(user);
credencial = null;
user = null;
e = null;
respuesta = "{success:true,result:\'El Usuario ha sido agregado\'}";
return "respuesta";
}
```

```
public String form_actualizar(){
    return "form_actualizar";
}
```

```
public String form_perfil(){
    return "form_perfil";
}
```

```
public String load(){
    try{
        long id = Long.parseLong(idUsuario);
        UsuarioJoinDTO userload = administradorUsuario.loaduser(id);
        Map session = ActionContext.getContext().getSession();
        session.put("userload",userload);
        respuesta = "{success:true}";
    }
```

```
return "respuesta";  
}  
catch (NumberFormatException e){  
    System.out.println(e);  
    respuesta = "{failure:true}";  
    return respuesta;  
}  
}
```

```
public String actualizar(){  
  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    UsuarioJoinDTO userload =(UsuarioJoinDTO) session.get("userload");  
  
    int type = 0;  
    int estate;  
  
    user = administradorUsuario.load(userload.getId_usuario());  
  
    if(userlogin.getTipo_usuario().getId() == 1 || userlogin.getId_usuario().longValue() ==  
user.getId_usuario().longValue()){  
        user.setNombre(nombre);  
        user.setApelliop(apellidop);  
        user.setApellidom(apellidom);  
        user.setRfc(rfc);  
  
        try{  
            if(userlogin.getTipo_usuario().getId() == 1)  
                type = Integer.parseInt(tipo);  
            else
```

```
        type =user.getTipo_usuario();
    }
    catch(NumberFormatException e){
        type =user.getTipo_usuario();
    }
    user.setTipo_usuario(type);
    System.out.println("*****");
    System.out.println(nacimiento);
    System.out.println("*****");
    user.setFecha_nacimiento(TimeStamp.FechaNacimiento(nacimiento));
    user.setTelefono(telefono);
    user.setCelular(celular);
    user.setEmail(mail);
    user.setCalle(calle);
    user.setColonia(colonia);
    user.setDelegacion(delegacion);
    user.setCp(Integer.parseInt(cp));
    try{
        estate = Integer.parseInt(estado);
    }
    catch(NumberFormatException e){
        estate = user.getEntidad();
    }
    user.setEntidad(estate);
    administradorUsuario.actualizar(user);
    user = null;
    respuesta = "{success:true,result:\\"El usuario ha sido actualizado\\"}";
    return "respuesta";
```

```
    }  
    else{  
        respuesta = "{failure:true,result:\\"No tienes permisos para actualizar al usuario\\"}";  
        return "respuesta";  
    }  
}
```

```
public List<UsuarioJoinDTO> getUsuarios() {  
    return usuarios;  
}
```

```
public void setUsuarios(List<UsuarioJoinDTO> usuarios) {  
    this.usuarios = usuarios;  
}
```

```
public String getListausuarios() {  
    return listausuarios;  
}
```

```
public void setListausuarios(String listausuarios) {  
    this.listausuarios = listausuarios;  
}
```

```
public String getRespuesta() {  
    return respuesta;  
}
```

```
public void setRespuesta(String respuesta) {  
    this.respuesta = respuesta;  
}
```

```
public String getIdUsuario() {  
    return idUsuario;  
}
```

```
public void setIdUsuario(String idUsuario) {  
    this.idUsuario = idUsuario;  
}
```

```
public String getUsername() {  
    return username;  
}
```

```
public void setUsername(String username) {  
    this.username = username;  
}
```

```
public String getNacimiento() {  
    return nacimiento;  
}
```

```
public void setNacimiento(String nacimiento) {  
    this.nacimiento = nacimiento;  
}
```

```
public String getApellido() {  
    return apellido;  
}
```

```
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}
```

```
public String getApellido() {  
    return apellido;  
}
```

```
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}
```

```
public String getCalle() {  
    return calle;  
}
```

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

```
public String getCelular() {  
    return celular;  
}
```

```
public void setCelular(String celular) {  
    this.celular = celular;  
}
```

```
public String getColonia() {  
    return colonia;  
}
```

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

```
public String getCp() {  
    return cp;  
}
```

```
public void setCp(String cp) {  
    this.cp = cp;  
}
```

```
public String getDelegacion() {  
    return delegacion;  
}
```

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



```
public String getEstado() {  
    return estado;  
}
```

```
public void setEstado(String estado) {  
    this.estado = estado;  
}
```

```
public String getMail() {  
    return mail;  
}
```

```
public void setMail(String mail) {  
    this.mail = mail;  
}
```

```
public String getNombre() {  
    return nombre;  
}
```

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

```
public String getRfc() {  
    return rfc;  
}
```

```
public void setRfc(String rfc) {  
    this.rfc = rfc;  
}
```

```
public String getTelefono() {  
    return telefono;  
}
```

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

```
public String getTipo() {  
    return tipo;  
}
```

```
public void setTipo(String tipo) {  
    this.tipo = tipo;  
}
```

```
public String getPassword() {  
    return password;  
}
```

```
public void setPassword(String password) {  
    this.password = password;  
}
```

```
public String getLoaduser() {
    return loaduser;
}

public void setLoaduser(String loaduser) {
    this.loaduser = loaduser;
}

public String getPasscfrm() {
    return passcfrm;
}

public void setPasscfrm(String passcfrm) {
    this.passcfrm = passcfrm;
}
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.action;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import java.util.List;
import java.util.Map;
```

```
import mx.uam.azc.cbi.data.ServicioJoinDTO;

import mx.uam.azc.cbi.data.TiemposJoinDTO;
import mx.uam.azc.cbi.interfaces.AdminTiempos;
import mx.uam.azc.cbi.logica.TimeStamp;
import org.springframework.beans.factory.annotation.Autowired;

/**
 *
 * @author Manzano
 */
public class TiemposAction extends ActionSupport{

    @Autowired
    private AdminTiempos administradorTiempos;

    private List<TiemposJoinDTO> tiempos;
    private String idServicio;
    private String respuesta;
    private String listardetalles;
    private ServicioJoinDTO loadservice;

    public String listar(){
        return "listar";
    }
}
```

```

public String listardetalles(){

    Map session = ActionContext.getContext().getSession();

    loadservice = (ServicioJoinDTO) session.get("serviceload");

    tiempos = administradorTiempos.listar(loadservice.getIdServicio().longValue());

    listardetalles = "{\"sucess\":true, \"totalCount\":" + "\"" + tiempos.size() + "\", "+ "\"data\":";

    for(int i = 0; i<tiempos.size();i++) {

        listardetalles = listardetalles + "{\"idServicio\":" + tiempos.get(i).getIdServicio() +

            "\", \"proveedor\":" + tiempos.get(i).getIdProveedor().getNombre() + " " +

                " " + tiempos.get(i).getIdProveedor().getApelliop() +

                " " + tiempos.get(i).getIdProveedor().getApellidom() +

            "\", \"modificado\":" + tiempos.get(i).getIdUltima().getNombre() + " " +

                " " + tiempos.get(i).getIdUltima().getApelliop() +

                " " + tiempos.get(i).getIdUltima().getApellidom() +

            "\", \"estado\":" + tiempos.get(i).getEstado().getEstado()+ "\",

    \"descripcion\":" + tiempos.get(i).getResumen() +

            "\", \"prioridad\":" + tiempos.get(i).getPriority()+ "\", \"ultima\":" +

    TimeStamp.Timestamp(tiempos.get(i).getFecha_modificacion()) + "\" },";

    }

    listardetalles = listardetalles.substring(0, listardetalles.length()-1);

    listardetalles = listardetalles + "]}";

    return "listardetalles";

}

public String getIdServicio() {

    return idServicio;

}

```

```
public void setIdServicio(String idServicio) {
    this.idServicio = idServicio;
}

public String getRespuesta() {
    return respuesta;
}

public void setRespuesta(String respuesta) {
    this.respuesta = respuesta;
}

public String getListardetalles() {
    return listardetalles;
}

public void setListardetalles(String listardetalles) {
    this.listardetalles = listardetalles;
}
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.util;
```

```
import java.io.*;

import java.util.*;

import net.sf.jxls.exception.ParsePropertyException;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import net.sf.jxls.transformer.XLSTransformer;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;

/**
 *
 * @author Manzano
 */
public class GenerarExcel {

    public void CrearArchivo()throws IOException, ParsePropertyException, InvalidFormatException
    {

        Map beans = new HashMap();

        beans.put("empresas","Hola Mundo");

        InputStream istream = new FileInputStream("C:/tempo/Prueba.xls");//
        GenerarExcel.class.getClassLoader().getResourceAsStream("mx/uam/azc/cbi/util/Prueba.xls");

        XLSTransformer transformer = new XLSTransformer();

        HSSFWorkbook workbook = (HSSFWorkbook) transformer.transformXLS(istream, beans);

        OutputStream ostream = new BufferedOutputStream(new
        FileOutputStream("C:/tempo/reporte.xls"));

        workbook.write(ostream);

        ostream.flush();

        ostream.close();
    }
}
```

```
}  
}  
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
*/
```

```
package mx.uam.azc.cbi.action;
```

```
import com.opensymphony.xwork2.ActionContext;  
import com.opensymphony.xwork2.ActionSupport;  
import java.io.BufferedOutputStream;
```

```
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.math.BigDecimal;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import mx.uam.azc.cbi.data.ProductoJoinDTO;  
import mx.uam.azc.cbi.data.ServicioJoinDTO;
```



```
import mx.uam.azc.cbi.data.UsuarioJoinDTO;

import mx.uam.azc.cbi.interfaces.AdminProductos;

import mx.uam.azc.cbi.interfaces.AdminServicios;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperExportManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

import net.sf.jxls.exception.ParsePropertyException;

import net.sf.jxls.transformer.XLSTransformer;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
/**
```

```
*
```

```
* @author Manzano
```

```
*/
```

```
public class DocumentosAction extends ActionSupport{
```

```
    @Autowired
```

```
    private AdminProductos administradorProductos;
```

@Autowired

private AdminServicios administradorServicios;

private List<ProductoJoinDTO> productos;

private List<ServicioJoinDTO> servicios;

private InputStream inputStream = null;

```
public String excel() throws IOException, ParsePropertyException, InvalidFormatException {  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    productos = administradorProductos.listar();  
    String archivo = CrearArchivoExcel(productos, userlogin.getId_credencial().getUsername(),  
    "");  
    inputStream = new FileInputStream(new File(archivo));  
    return "excel";  
}
```

```
public String excelservice() throws IOException, ParsePropertyException,  
InvalidFormatException {  
    Map session = ActionContext.getContext().getSession();  
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");  
    servicios = administradorServicios.misservicios(userlogin.getId_usuario());  
    String archivo = CrearArchivoExcel(servicios, userlogin.getId_credencial().getUsername(), "");  
    inputStream = new FileInputStream(new File(archivo));  
    return "excelservice";  
}
```

```

public String pdf() throws Exception{
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    inputStream = new FileInputStream(new File("C:\\temp\\reporte2.pdf"));
    System.out.println("Despues de crear Documento en PDF");
    return "pdf";
}

```

```

public String pdfservice() throws Exception{
    Map session = ActionContext.getContext().getSession();
    UsuarioJoinDTO userlogin =(UsuarioJoinDTO) session.get("user");
    inputStream = new FileInputStream(new File("C:\\temp\\reporte2.pdf"));
    System.out.println("Despues de crear Documento en PDF");
    return "pdfservice";
}

```

```

public String CrearArchivoExcel(List productos, String name, String plantilla)throws IOException,
ParsePropertyException, InvalidFormatException {

```

```

    Map<String, List<ProductoJoinDTO>> beans = new HashMap<String,
List<ProductoJoinDTO>>();

```

```

    beans.put("productos",productos);

```

```

    name = name.concat(".xls");

```

```

    InputStream istream = new FileInputStream("C:/tempo/Prueba.xls");

```

```

    XLSTransformer transformer = new XLSTransformer();

```

```

    HSSFWorkbook workbook = (HSSFWorkbook) transformer.transformXLS(istream, beans);

```

```

String archivo = "C:/tempo/"+name;

    OutputStream ostream = new BufferedOutputStream(new FileOutputStream(archivo));

    workbook.write(ostream);

    ostream.flush();

ostream.close();

return archivo;

}

```

```

public String CrearArchivoPdf(String name) throws Exception{

    setSystemProperties();

    name = name.concat(".pdf");

    //ClassLoader loader = GenerarPDF.class.getClassLoader();

    InputStream istream = new FileInputStream("C:/temp/ListaServicios.jrxml");
    //loader.getResourceAsStream("mx/uam/azc/cbi/util/ListaServicios.jrxml");

    System.out.println(istream);

    JasperDesign design = JRXmlLoader.load(istream);

    JasperReport report = JasperCompileManager.compileReport(design);

    BigDecimal b = new BigDecimal("50000");

    Map params = new HashMap();

    params.put( "anio", "2007" );

    params.put( "creditoMinimo", b );

    Connection connection = getConnection();

    try{

        JasperPrint print = JasperFillManager.fillReport( report, params, connection );

        JasperExportManager.exportReportToPdfFile( print, "c:\\temp\\reporte2.pdf" );

        System.out.println("REPORTES LISTOS");

    }

    finally{

```

```

        connection.close();
    }
    return name;
}

private static void setSystemProperties(){
    System.setProperty( "jasper.reports.compile.keep.java.file", "false" );
}

private static Connection getConnection() throws Exception{
    Class.forName( "org.postgresql.Driver" );
    String url = "jdbc:postgresql://localhost:5432/GestionCRM?autoReconnect=true";
    return DriverManager.getConnection( url, "postgres", "g3r4rd0m4nz4n0!");
}

public InputStream getInputStream() {
    return inputStream;
}

public void setInputStream(InputStream inputStream) {
    this.inputStream = inputStream;
}
}

/*

```

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
import org.hibernate.annotations.Proxy;
```

```
import static javax.persistence.GenerationType.IDENTITY;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
@Entity @Table(name="mensajes")
```

```
@Proxy (lazy=false)
```

```
public class MensajeSaveDTO implements Serializable {
```

```
    private Long idMensaje;
```

```
    private long idTema;
```

```
private long idUsuario;

private String comentario;

private long date;

private Boolean isobsolete;

//private Calendar _fecha = Calendar.getInstance();
```

```
@Id @Column(name="id_mensaje")
@GeneratedValue(strategy=IDENTITY)
public Long getIdMensaje() {
    return idMensaje;
}
```

```
@Column(name="comentario")
public String getComentario() {
    return comentario;
}
```

```
@Column(name="fecha")
public long getDate() {
    return date;
}
```

```
@Column(name="id_temas")
public long getIdTema() {
    return idTema;
}
```

```
@Column(name="id_usuario")
```

```
public long getIdUsuario() {  
    return idUsuario;  
}
```

```
@Column(name="isobsolete")  
public Boolean getIsobsolete() {  
    return isobsolete;  
}
```

```
public void setComentario(String comentario) {  
    this.comentario = comentario;  
}
```

```
public void setIsobsolete(Boolean isobsolete) {  
    this.isobsolete = isobsolete;  
}
```

```
public void setDate(long date) {  
    this.date = date;  
}
```

```
public void setIdMensaje(Long idMensaje) {  
    this.idMensaje = idMensaje;  
}
```

```
public void setIdTema(long idTema) {  
    this.idTema = idTema;  
}
```



```
}

public void setIdUsuario(long idUsuario) {
    this.idUsuario = idUsuario;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

/**
 *
```

```
* @author manzano
```

```
*/
```

```
@Entity @Table(name="mensajes")
```

```
@Proxy (lazy=false)
```

```
public class MensajeJoinDTO implements Serializable {
```

```
    private Long idMensaje;
```

```
    private TemaJoinDTO idTema;
```

```
    private UsuarioJoinDTO idUsuario;
```

```
    private String comentario;
```

```
    private long date;
```

```
    private Boolean isobsolete;
```

```
    //private Calendar _fecha = Calendar.getInstance();
```

```
    @Id @Column(name="id_mensaje")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public Long getIdMensaje() {
```

```
        return idMensaje;
```

```
    }
```

```
    @Column(name="comentario")
```

```
    public String getComentario() {
```

```
        return comentario;
```

```
    }
```

```
@Column(name="fecha")
```

```
public long getDate() {
```

```
    return date;
```

```
}
```

```
//@Column(name="id_temas")
```

```
@OneToOne @JoinColumn (name="id_temas")
```

```
public TemaJoinDTO getIdTema() {
```

```
    return idTema;
```

```
}
```

```
//@Column(name="id_usuario")
```

```
@OneToOne @JoinColumn (name="id_usuario")
```

```
public UsuarioJoinDTO getIdUsuario() {
```

```
    return idUsuario;
```

```
}
```

```
@Column(name="isobsolete")
```

```
public Boolean getIsobsolete() {
```

```
    return isobsolete;
```

```
}
```

```
public void setComentario(String comentario) {
```

```
    this.comentario = comentario;
```

```
}
```

```
public void setIsobsolete(Boolean isobsolete) {
```

```
        this.isobsolete = isobsolete;
    }

    public void setDate(long date) {
        this.date = date;
    }

    public void setIdMensaje(Long idMensaje) {
        this.idMensaje = idMensaje;
    }

    public void setIdTema(TemaJoinDTO idTema) {
        this.idTema = idTema;
    }

    public void setIdUsuario(UsuarioJoinDTO idUsuario) {
        this.idUsuario = idUsuario;
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
```

```
import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.persistence.Version;

import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

/**
 *
 * @author Manzano
 */

@Entity @Table(name="credenciales")

@Proxy (lazy=false)

public class CredencialDTO implements Serializable {

    private Long idCredencial;

    private String username;

    private String password;

    private String salt;

    private int version;

    private Boolean isobsolete = false;
```

```
@Id @Column(name="id_credencial")
@GeneratedValue(strategy=IDENTITY)
public Long getIdCredencial() {
    return idCredencial;
}
```

```
@Column(name="username")
public String getUsername() {
    return username;
}
```

```
@Column(name="password")
public String getPassword() {
    return password;
}
```

```
@Column(name="salt")
public String getSalt() {
    return salt;
}
```

```
@Version
@Column(name="version")
public int getVersion() {
    return version;
}
```

```
@Column(name="isobsolete")
```

```
public Boolean getIsobsolete() {
```

```
    return isobsolete;
```

```
}
```

```
public void setIdCredencial(Long idCredencial) {
```

```
    this.idCredencial = idCredencial;
```

```
}
```

```
public void setIsobsolete(Boolean isobsolete) {
```

```
    this.isobsolete = isobsolete;
```

```
}
```

```
public void setPassword(String password) {
```

```
    this.password = password;
```

```
}
```

```
public void setUsername(String username) {
```

```
    this.username = username;
```

```
}
```

```
public void setVersion(int version) {
```

```
    this.version = version;
```

```
}
```

```
public void setSalt(String salt) {
```

```
    this.salt = salt;
```

```
}
```

```
}
```

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import static javax.persistence.GenerationType.IDENTITY;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
import org.hibernate.annotations.Proxy;
```

```
/**
```

```
*
```

```
* @author gmendoza
```

```
*/
```

```
@Entity @Table(name="compras")
```



@Proxy (lazy=false)

```
public class CompraSaveDTO implements Serializable {
```

```
    private long idCompra;
```

```
    private long idOrden;
```

```
    private long idProducto;
```

```
    private long totalproductos;
```

```
    private long total;
```

```
    @Id @Column(name="id_compra")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public long getIdCompra() {
```

```
        return idCompra;
```

```
    }
```

```
    @Column(name="id_orden")
```

```
    public long getIdOrden() {
```

```
        return idOrden;
```

```
    }
```

```
    @Column(name="id_producto")
```

```
    public long getIdProducto() {
```

```
        return idProducto;
```

```
    }
```

```
    @Column(name="cantidad")
```

```
public long getTotalproductos() {  
    return totalproductos;  
}
```

```
@Column(name="total")  
public long getTotal() {  
    return total;  
}
```

```
public void setTotalproductos(long totalproductos) {  
    this.totalproductos = totalproductos;  
}
```

```
public void setIdCompra(long idCompra) {  
    this.idCompra = idCompra;  
}
```

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

```
public void setIdOrden(long idOrden) {  
    this.idOrden = idOrden;  
}
```

```
public void setTotal(long total) {
```

```
        this.total = total;
    }
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import org.hibernate.annotations.Proxy;

/**
 *
 * @author gmendoza
 */
```

```
@Entity @Table(name="compras")
```

```
@Proxy (lazy=false)
```

```
public class CompraJoinDTO implements Serializable{
```

```
    private long idCompra;
```

```
    private OrdenCompraJoinDTO idOrden;
```

```
    private ProductoJoinDTO idProducto;
```

```
    private long totalproductos;
```

```
    private long total;
```

```
    @Id @Column(name="id_compra")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public long getIdCompra() {
```

```
        return idCompra;
```

```
    }
```

```
    @OneToOne @JoinColumn (name="id_orden")
```

```
    public OrdenCompraJoinDTO getIdOrden() {
```

```
        return idOrden;
```

```
    }
```

```
    @OneToOne @JoinColumn (name="id_producto")
```

```
    public ProductoJoinDTO getIdProducto() {
```

```
        return idProducto;
```

```
    }
```

```
@Column(name="cantidad")
```

```
public long getTotalproductos() {
```

```
    return totalproductos;
```

```
}
```

```
@Column(name="total")
```

```
public long getTotal() {
```

```
    return total;
```

```
}
```

```
public void setIdCompra(long idCompra) {
```

```
    this.idCompra = idCompra;
```

```
}
```

```
public void setIdOrden(OrdenCompraJoinDTO idOrden) {
```

```
    this.idOrden = idOrden;
```

```
}
```

```
public void setIdProducto(ProductoJoinDTO idProducto) {
```

```
    this.idProducto = idProducto;
```

```
}
```

```
public void setTotalproductos(long totalproductos) {
```

```
    this.totalproductos = totalproductos;
```

```
}
```

```
public void setTotal(long total) {
```

```
    this.total = total;
```

```
    }  
  
}  
  
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
*/  
  
package mx.uam.azc.cbi.data;  
  
import java.io.Serializable;  
import javax.persistence.*;  
import org.hibernate.annotations.Proxy;  
import static javax.persistence.GenerationType.IDENTITY;  
  
/**  
 *  
 * @author Manzano  
 */  
  
@Entity @Table (name="cat_tipousuario")  
@Proxy(lazy=false)  
public class CatTipoUsuarioDTO implements Serializable{  
  
    private int _id;  
  
    private String _tipoUsuario;
```

```

@Id @Column(name="id_tipousuario")
@GeneratedValue(strategy=IDENTITY)
public int getId() {
    return _id;
}

@Column(name="tipousuario")
public String getTipoUsuario() {
    return _tipoUsuario;
}

public void setId(int id) {
    _id = id;
}

public void setTipoUsuario(String tipoUsuario) {
    _tipoUsuario = tipoUsuario;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;

```

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import org.hibernate.annotations.Proxy;
```

```
/**
 *
 * @author gmendoza
 */
```

```
@Entity @Table (name="cat_fabricantes")
```

```
@Proxy(lazy=false)
```

```
public class CatFabricanteDTO implements Serializable{
```

```
    private int _id;
```

```
    private String _fabricante;
```

```
    @Id @Column(name="id_fabricante")
```

```
    public int getId() {
```

```
        return _id;
```

```
    }
```

```
    @Column(name="fabricante")
```

```
    public String getFabricante() {
```



```
        return _fabricante;
    }

    public void setId(int id) {
        _id = id;
    }

    public void setFabricante(String fabricante) {
        _fabricante = fabricante;
    }
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import org.hibernate.annotations.Proxy;

import java.io.Serializable;
import javax.persistence.*;

/**
 *
 */
```

```
* @author manzano
*/

@Entity @Table (name="cat_estadoservicio")
@Proxy(lazy=false)
public class CatEstadoServiciosDTO implements Serializable{

    private int id;
    private String estado;

    @Id @Column(name="id_estado")
    public int getId() {
        return id;
    }

    @Column(name="estado")
    public String getEstado() {
        return estado;
    }

    public void setId(int id) {
        this.id = id;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.interfaces;
```

```
import java.util.List;
```

```
import mx.uam.azc.cbi.data.ProductoJoinDTO;
```

```
import mx.uam.azc.cbi.data.ProductoSaveDTO;
```

```
import org.springframework.transaction.annotation.Transactional;
```

```
/**
```

```
*
```

```
* @author Manzano
```

```
*/
```

```
@Transactional(rollbackFor=Exception.class)
```

```
public interface AdminProductos {
```

```
    public long insertar(ProductoSaveDTO producto);
```

```
    public ProductoSaveDTO load(Long idProducto);
```

```
    public ProductoJoinDTO loadproduct(Long idProducto);
```

```
    public void actualizar(ProductoSaveDTO producto);
```

```

public List<ProductoJoinDTO> listar();

public List<ProductoJoinDTO> misproductos(Long user);

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.interfaces;

import java.util.List;
import mx.uam.azc.cbi.data.OrdenCompraSaveDTO;
import mx.uam.azc.cbi.data.OrdenCompraJoinDTO;
import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author gmendoza
 */

@Transactional(rollbackFor=Exception.class)
public interface AdminOrdenes {

public long insertar(OrdenCompraSaveDTO Orden);

```

```
public List<OrdenCompraJoinDTO> listar ();

public List<OrdenCompraJoinDTO> listar (long idUsuario);

public OrdenCompraJoinDTO load(Long idProducto);

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.interfaces;

import java.util.List;
import mx.uam.azc.cbi.data.MensajeJoinDTO;
import mx.uam.azc.cbi.data.MensajeSaveDTO;
import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author Manzano
 */

@Transactional(rollbackFor=Exception.class)
```

```
public interface AdminMensajes {

    public long insertar(MensajeSaveDTO mensaje);

    public void eliminar();

    public void actualizar(MensajeSaveDTO mensaje);

    public List<MensajeJoinDTO> listar(long id);

    public MensajeSaveDTO load(long id);

    public MensajeJoinDTO loadmessage(long id);
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.interfaces;

import mx.uam.azc.cbi.data.CredencialDTO;
import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author Manzano
 */
```

```
@Transactional(rollbackFor=Exception.class)

public interface AdminCredenciales {

    public long insertar (CredencialDTO credencial);

    public void eliminar (CredencialDTO credencial);

    public void actualizar(CredencialDTO credencial);

    public boolean buscar (String username);

    public CredencialDTO load(long id);

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.interfaces;

import java.util.List;

import mx.uam.azc.cbi.data.CompraJoinDTO;

import mx.uam.azc.cbi.data.CompraSaveDTO;

import org.springframework.transaction.annotation.Transactional;
```

```
/**
```

```
*
```

```
* @author gmendoza
```

```
*/
```

```
@Transactional(rollbackFor=Exception.class)
```

```
public interface AdminCompras {
```

```
    public long insertar(CompraSaveDTO compra);
```

```
    public List<CompraJoinDTO> listar (long idOrden);
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.interfaces;
```

```
import java.util.List;
```

```
import mx.uam.azc.cbi.data.CatCategoriasDTO;
```

```
import mx.uam.azc.cbi.data.CatEstadoServiciosDTO;
```

```
import mx.uam.azc.cbi.data.CatEstadosDTO;
```

```
import mx.uam.azc.cbi.data.CatFabricanteDTO;
```

```
import mx.uam.azc.cbi.data.CatServiciosDTO;
```



```
import mx.uam.azc.cbi.data.CatTipoUsuarioDTO;

import mx.uam.azc.cbi.data.UsuarioSaveDTO;

import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author Manzano
 */

@Transactional(rollbackFor=Exception.class)

public interface AdminCombos {

    //@@Transactional(propagation=Propagation.SUPPORTS, readOnly=true)
    public List<CatEstadosDTO> leerEstados();

    //@@Transactional(propagation=Propagation.SUPPORTS, readOnly=true)
    public List<CatTipoUsuarioDTO> leerTipoUsuario();

    public List<CatEstadoServiciosDTO> leerEstadosServicio();

    //@@Transactional(propagation=Propagation.SUPPORTS, readOnly=true)
    public List<CatCategoriasDTO> leerCategorias();

    public List<CatFabricanteDTO> leerfabricantes();

    public List<UsuarioSaveDTO> leerproveedores();

    public List<UsuarioSaveDTO> leerclientes();
```

```
        public List<CatServiciosDTO> leerservicios();

    }

    /*
    * To change this template, choose Tools | Templates
    * and open the template in the editor.
    */

    package mx.uam.azc.cbi.interfaces;

    import java.io.IOException;

    import java.util.List;

    import mx.uam.azc.cbi.data.UsuarioJoinDTO;

    import mx.uam.azc.cbi.data.UsuarioSaveDTO;

    import org.springframework.transaction.annotation.Transactional;

    /**
    *
    * @author manzano
    */

    @Transactional(rollbackFor=Exception.class)
    public interface AdminUsuarios {
```

```
public long insertar (UsuarioSaveDTO usuario);

public void actualizar(UsuarioSaveDTO usuario);

//@Transactional(propagation = Propagation.SUPPORTS, readOnly=true)
public List<UsuarioJoinDTO> listar ();

public List<UsuarioJoinDTO> perfil(String usuario);

public UsuarioJoinDTO login (String [] username) throws IOException;

public UsuarioSaveDTO load(long id);

public UsuarioJoinDTO loaduser(long id);

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.interfaces;

import java.util.List;
```

```
import mx.uam.azc.cbi.data.TiemposJoinDTO;

import mx.uam.azc.cbi.data.TiemposSaveDTO;

import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author gmendoza
 */

@Transactional(rollbackFor=Exception.class)
public interface AdminTiempos {

    public long insertar (TiemposSaveDTO tiempos);

    public List<TiemposJoinDTO> listar (long idServicio);

}

/**
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.interfaces;

import java.util.List;

import mx.uam.azc.cbi.data.TemaSaveDTO;
```

```
import mx.uam.azc.cbi.data.TemaJoinDTO;
import org.springframework.transaction.annotation.Transactional;
```

```
/**
```

```
*
```

```
* @author Manzano
```

```
*/
```

```
@Transactional(rollbackFor=Exception.class)
```

```
public interface AdminTemas {
```

```
    public long insertar(TemaSaveDTO temas);
```

```
    public void eliminar(long id);
```

```
    public void actualizar(TemaSaveDTO tema);
```

```
    public List<TemaJoinDTO> listar();
```

```
    public TemaSaveDTO load(long id);
```

```
    public TemaJoinDTO loadtopic(long id);
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/  
  
package mx.uam.azc.cbi.interfaces;  
  
  
import java.util.List;  
import mx.uam.azc.cbi.data.ServicioJoinDTO;  
import mx.uam.azc.cbi.data.ServicioSaveDTO;  
import org.springframework.transaction.annotation.Transactional;  
  
/**  
 *  
 * @author Manzano  
 */  
  
@Transactional(rollbackFor=Exception.class)  
public interface AdminServicios {  
  
    public long insertar(ServicioSaveDTO servicio);  
  
    public void actualizar(ServicioSaveDTO servicio);  
  
    public ServicioSaveDTO load(Long idServicio);  
  
    public ServicioJoinDTO loadservice(Long idServicio);  
  
    public List<ServicioJoinDTO> listar();  
  
    public List<ServicioJoinDTO> misservicios(long user);  
}
```

```
public List<ServicioJoinDTO> misserviciosp(long user);

}

package mx.uam.azc.cbi.logica;

import org.apache.commons.codec.binary.Base64;

import java.io.IOException;
import java.security.MessageDigest;
import java.security.SecureRandom;

/**
 * @author manzano
 */

public class Encryption {

    public static byte[] base64ToByte(String data) throws IOException {

        return Base64.decodeBase64(data);
    }

    public static String byteToBase64(byte[] data) {

        return Base64.encodeBase64String(data);
    }
}
```

```
public byte[] getHash(int iterationNb, String password, byte[] salt) {  
    byte[] input = null;  
    try {  
        MessageDigest digest = MessageDigest.getInstance("SHA-1");  
        digest.reset();  
        digest.update(salt);  
        input = digest.digest(password.getBytes());  
        for (int i = 0; i < iterationNb; i++) {  
            digest.reset();  
            input = digest.digest(input);  
        }  
    } catch (Exception ex) {  
    }  
    return input;  
}
```

```
public String[] getEncryptedPassword(String password) {  
  
    String[] passwordArray = new String[2];  
    try {  
        SecureRandom random = SecureRandom.getInstance("SHA1PRNG");  
        byte[] bSalt = new byte[8];  
        random.nextBytes(bSalt);  
        byte[] bDigest = getHash(1000, password, bSalt);  
        passwordArray[0] = byteToBase64(bDigest);  
        passwordArray[1] = byteToBase64(bSalt);  
    } catch (Exception ex) {
```



```
        System.out.print(ex.toString());
    }

    return passwordArray;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.logica;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

/**
 *
 * @author manzano
 */
public class TimeStamp {

    public static String Timestamp(long time) {
        if(time == 0)
```

```
        return "Sin Fecha";

        Date fecha = new Date();

        fecha.setTime(time*1000);

        SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy");

        return sdf.format(fecha);

    }
}
```

```
public static long Fecha() {

    Calendar fecha = Calendar.getInstance();

    return fecha.getTimeInMillis() / 1000;

}
```

```
public static long FechaNacimiento(String fecha) {

    String [] nacimiento = fecha.split("/");

    Calendar date = Calendar.getInstance();

    date.set(Calendar.YEAR, Integer.parseInt(nacimiento[2])); // Modificar fecha de inicio
    para el rango de los reportes

    date.set(Calendar.MONTH, (Integer.parseInt(nacimiento[0])-1));

    date.set(Calendar.DAY_OF_MONTH, Integer.parseInt(nacimiento[1]));

    date.set(Calendar.HOUR_OF_DAY, 0);

    date.set(Calendar.MINUTE, 0);

    date.set(Calendar.SECOND, 0);

    return date.getTimeInMillis() / 1000;

}
```

```
}
```

```
/*
```

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package mx.uam.azc.cbi.metodos;
```

```
import java.util.List;
```

```
import mx.uam.azc.cbi.data.MensajeJoinDTO;
```

```
import mx.uam.azc.cbi.data.MensajeSaveDTO;
```

```
import mx.uam.azc.cbi.interfaces.AdminMensajes;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
public class AdminMensajesImpl implements AdminMensajes{
```

```
    private SessionFactory _sessionFactory;
```

```
    public long insertar(MensajeSaveDTO mensaje){
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        session.save(mensaje);
```

```
        return mensaje.getIdMensaje();
```

```
}
```

```
public void eliminar(){
```

```
}
```

```
public void actualizar(MensajeSaveDTO mensaje){
```

```
    Session session = _sessionFactory.getCurrentSession();
```

```
    session.update(mensaje);
```

```
}
```

```
public List<MensajeJoinDTO> listar(long id){
```

```
    Session session = _sessionFactory.getCurrentSession();
```

```
    Query query = session.createQuery("select m From MensajeJoinDTO m where  
        m.isobsolete='false' and m.idTema = :idTema");
```

```
    query.setLong("idTema", id);
```

```
    return query.list();
```

```
}
```

```
public MensajeSaveDTO load(long id){
```

```
    Session session = _sessionFactory.getCurrentSession();
```

```
    MensajeSaveDTO mensaje = (MensajeSaveDTO)session.load(MensajeSaveDTO.class, id);
```

```
    return mensaje;
```

```
}
```

```
public MensajeJoinDTO loadmessage(long id){
```

```
    Session session = _sessionFactory.getCurrentSession();
```

```

    MensajeJoinDTO mensaje = (MensajeJoinDTO)session.load(MensajeJoinDTO.class, id);
    return mensaje;
}

    public final void setSessionFactory(SessionFactory sessionFactory) {
        _sessionFactory = sessionFactory;
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.cbi.metodos;

import mx.uam.azc.cbi.data.CredencialDTO;
import mx.uam.azc.cbi.interfaces.AdminCredenciales;
import org.hibernate.Query;
import org.hibernate.SessionFactory;
import org.hibernate.Session;

/**
 *
 * @author Manzano
 */
public class AdminCredencialesImpl implements AdminCredenciales{

```

```
private SessionFactory _sessionFactory;
```

```
public long insertar(CredencialDTO credencial){  
    Session session = _sessionFactory.getCurrentSession();  
    session.save(credencial);  
    return credencial.getIdCredencial();  
}
```

```
public void actualizar (CredencialDTO credencial){  
    Session session = _sessionFactory.getCurrentSession();  
    session.update(credencial);  
}
```

```
public void eliminar (CredencialDTO credencial){  
    Session session = _sessionFactory.getCurrentSession();  
    session.update(credencial);  
}
```

```
public boolean buscar (String username){  
    Session session = _sessionFactory.getCurrentSession();  
    Query query = session.createQuery("select c From CredencialDTO c where c.username =  
    :username");  
    query.setString( "username", username);  
    if(query.list().size()>0){
```

```

        return true;
    }

    return false;
}

public CredencialDTO load(long id){
    Session session = _sessionFactory.getCurrentSession();
    CredencialDTO credencial = (CredencialDTO) session.load(CredencialDTO.class, id);
    return credencial;
}

public final void setSessionFactory(SessionFactory sessionFactory) {
    _sessionFactory = sessionFactory;
}
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.metodos;

import java.util.List;
import mx.uam.azc.cbi.data.CompraJoinDTO;
import mx.uam.azc.cbi.data.CompraSaveDTO;

```

```
import mx.uam.azc.cbi.interfaces.AdminCompras;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

/**
 *
 * @author gmendoza
 */
public class AdminComprasImpl implements AdminCompras {

    private SessionFactory _sessionFactory;

    public long insertar(CompraSaveDTO compra){

        Session session = _sessionFactory.getCurrentSession();

        session.save(compra);

        return compra.getIdCompra();

    }

    public List<CompraJoinDTO> listar (long idOrden){

        Session session = _sessionFactory.getCurrentSession();

        Query query = session.createQuery("select c From CompraJoinDTO c where c.idOrden =
        :idOrden");

        query.setLong("idOrden", idOrden);

        return query.list();

    }

    public final void setSessionFactory(SessionFactory sessionFactory) {
```



```
        _sessionFactory = sessionFactory;
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.cbi.metodos;

import java.util.List;

import mx.uam.azc.cbi.data.CatCategoriasDTO;
import mx.uam.azc.cbi.data.CatEstadoServiciosDTO;

import mx.uam.azc.cbi.data.CatEstadosDTO;
import mx.uam.azc.cbi.data.CatFabricanteDTO;
import mx.uam.azc.cbi.data.CatServiciosDTO;
import mx.uam.azc.cbi.data.CatTipoUsuarioDTO;
import mx.uam.azc.cbi.data.UsuarioSaveDTO;
import mx.uam.azc.cbi.interfaces.AdminCombos;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

/**
 *
 */
```

```
* @author Manzano
```

```
*/
```

```
public class AdminCombosImpl implements AdminCombos {
```

```
    private SessionFactory _sessionFactory;
```

```
    /////////////////////////////////////////////////////////////////// Estados \\\\/////////////////////////////////////////////////////////////////
```

```
    public List<CatEstadosDTO> leerEstados() {
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        Query query = session.createQuery("FROM CatEstadosDTO e ");
```

```
        return query.list();
```

```
    }
```

```
    /////////////////////////////////////////////////////////////////// Usuarios \\\\/////////////////////////////////////////////////////////////////
```

```
    @SuppressWarnings("unchecked")
```

```
    public List<CatTipoUsuarioDTO> leerTipoUsuario() {
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        Query query = session.createQuery("FROM CatTipoUsuarioDTO u ");
```

```
        return query.list();
```

```
    }
```

```
    @SuppressWarnings("unchecked")
```

```
    public List<CatEstadoServiciosDTO> leerEstadosServicio() {
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        Query query = session.createQuery("FROM CatEstadoServiciosDTO s ORDER BY s.id ");
```

```
        return query.list();
```

```
    }
```



```
    Session session = _sessionFactory.getCurrentSession();

    Query query = session.createQuery("FROM CatServiciosDTO s ");

    return query.list();

}
```

```
public final void setSessionFactory(SessionFactory sessionFactory) {

    _sessionFactory = sessionFactory;

}

}
```

```
/*

* To change this template, choose Tools | Templates

* and open the template in the editor.

*/
```

```
package mx.uam.azc.cbi.metodos;
```

```
import java.io.IOException;
```

```
import java.util.List;
```

```
import java.util.Arrays;
```

```
import mx.uam.azc.cbi.data.UsuarioJoinDTO;
```

```
import mx.uam.azc.cbi.data.UsuarioSaveDTO;
```

```
import mx.uam.azc.cbi.logica.Encryption;
```

```
import mx.uam.azc.cbi.interfaces.AdminUsuarios;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
public class AdminUsuariosImpl implements AdminUsuarios {
```

```
    private SessionFactory _sessionFactory;
```

```
    Encryption encryption = new Encryption();
```

```
    public long insertar(UsuarioSaveDTO usuario) {
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        session.save(usuario);
```

```
        return usuario.getId_usuario();
```

```
    }
```

```
    public void actualizar(UsuarioSaveDTO usuario) {
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        session.update(usuario);
```

```
    }
```

```
    public List<UsuarioJoinDTO> listar() {
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        Query query = session.createQuery("select u From UsuarioJoinDTO u Join u.id_credencial c  
        where c.isobsolete = 'false'");
```

```
        return query.list();
```

```
}
```

```
public List<UsuarioJoinDTO> perfil(String usuario) {  
    Session session = _sessionFactory.getCurrentSession();  
    Query query = session.createQuery("select u From UsuarioJoinDTO u Join u.id_credencial c  
        where c.isobsolete = 'false' and c.username = :username");  
    query.setString( "username", usuario);  
    return query.list();  
}
```

```
public UsuarioJoinDTO login(String[] username) throws IOException {  
    Session session = _sessionFactory.getCurrentSession();  
    Query query = session.createQuery("select u From UsuarioJoinDTO u Join u.id_credencial c  
        where c.username = :username");  
    query.setString( "username", username[0]);  
    UsuarioJoinDTO user = (UsuarioJoinDTO) query.uniqueResult();  
    if(user != null){  
        byte[] bDigest = Encryption.base64ToByte(user.getId_credencial().getPassword());  
        byte[] bSalt = Encryption.base64ToByte(user.getId_credencial().getSalt());  
        byte[] proposedDigest = encryption.getHash(1000, username[1], bSalt);  
        if (Arrays.equals(proposedDigest, bDigest)) {  
            if(user.getId_credencial().getIsobsolete()==true)  
                return null;  
            return user;  
        }  
        else  
            return null;  
    }  
    else
```

```
        return null;
    }

    public UsuarioSaveDTO load(long id){
        Session session = _sessionFactory.getCurrentSession();
        UsuarioSaveDTO usuario = (UsuarioSaveDTO)session.load(UsuarioSaveDTO.class, id);
        return usuario;
    }

    public UsuarioJoinDTO loaduser(long id){
        Session session = _sessionFactory.getCurrentSession();
        UsuarioJoinDTO usuario = (UsuarioJoinDTO)session.load(UsuarioJoinDTO.class, id);
        return usuario;
    }

    public final void setSessionFactory(SessionFactory sessionFactory) {
        _sessionFactory = sessionFactory;
    }
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.metodos;
```

```
import java.util.List;

import mx.uam.azc.cbi.data.TiemposJoinDTO;

import mx.uam.azc.cbi.data.TiemposSaveDTO;

import mx.uam.azc.cbi.interfaces.AdminTiempos;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

/**
 *
 * @author gmendoza
 */

public class AdminTiemposImpl implements AdminTiempos{

    private SessionFactory _sessionFactory;

    public long insertar (TiemposSaveDTO tiempos){

        Session session = _sessionFactory.getCurrentSession();

        session.save(tiempos);

        return 2;

    }

    public List<TiemposJoinDTO> listar (long idServicio){

        Session session = _sessionFactory.getCurrentSession();

        Query query = session.createQuery("select t From TiemposJoinDTO t where t.idServicio =
            :idServicio");

        query.setLong("idServicio", idServicio);

        return query.list();

    }

}
```



```
}

public final void setSessionFactory(SessionFactory sessionFactory) {
    _sessionFactory = sessionFactory;
}
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.metodos;

import java.util.ArrayList;
import java.util.List;
import mx.uam.azc.cbi.data.TemaSaveDTO;
import mx.uam.azc.cbi.data.TemaJoinDTO;
import mx.uam.azc.cbi.interfaces.AdminTemas;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

/**
 *
 * @author manzano
 */
```

```

public class AdminTemasImpl implements AdminTemas{

    private SessionFactory _sessionFactory;

    public long insertar(TemaSaveDTO tema){

        Session session = _sessionFactory.getCurrentSession();

        session.save(tema);

        return tema.getIdTema();

    }

    public void eliminar(long id){

    }

    public void actualizar(TemaSaveDTO tema){

        Session session = _sessionFactory.getCurrentSession();

        session.update(tema);

    }

    public List<TemaJoinDTO> listar(){

        List<TemaJoinDTO> topics = new ArrayList<TemaJoinDTO>();

        Session session = _sessionFactory.getCurrentSession();

        Query query = session.createQuery("select t From TemaJoinDTO t where t.isobsolete='false'
");

        List<TemaJoinDTO> temas = query.list();

        for(int i = 0; i< temas.size(); i++){

            if(temas.get(i).getId_usuario().getId_credencial().getIsobsolete() != true){

```

```

        topics.add(temas.get(i));
    }
}
return topics;
}

public TemaSaveDTO load(long id){
    Session session = _sessionFactory.getCurrentSession();
    TemaSaveDTO tema = (TemaSaveDTO)session.load(TemaSaveDTO.class, id);
    return tema;
}

public TemaJoinDTO loadtopic(long id){
    Session session = _sessionFactory.getCurrentSession();
    TemaJoinDTO tema = (TemaJoinDTO)session.load(TemaJoinDTO.class, id);
    return tema;
}

public final void setSessionFactory(SessionFactory sessionFactory) {
    _sessionFactory = sessionFactory;
}
}

```

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

```
*/
```

```
package mx.uam.azc.cbi.metodos;
```

```
import java.util.List;
```

```
import mx.uam.azc.cbi.data.ServicioJoinDTO;
```

```
import mx.uam.azc.cbi.data.ServicioSaveDTO;
```

```
import mx.uam.azc.cbi.interfaces.AdminServicios;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
public class AdminServiciosImpl implements AdminServicios{
```

```
    private SessionFactory _sessionFactory;
```

```
    public long insertar(ServicioSaveDTO servicio){
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        session.save(servicio);
```

```
        return servicio.getIdServicio();
```

```
    }
```

```
public void actualizar(ServicioSaveDTO servicio){  
    Session session = _sessionFactory.getCurrentSession();  
    session.update(servicio);  
}
```

```
public ServicioSaveDTO load(Long idServicio){  
    Session session = _sessionFactory.getCurrentSession();  
    return (ServicioSaveDTO) session.load(ServicioSaveDTO.class, idServicio);  
}
```

```
public ServicioJoinDTO loadservice(Long idServicio){  
    Session session = _sessionFactory.getCurrentSession();  
    return (ServicioJoinDTO) session.load(ServicioJoinDTO.class, idServicio);  
}
```

```
public List<ServicioJoinDTO> listar(){  
    Session session = _sessionFactory.getCurrentSession();  
    Query query = session.createQuery("select s From ServicioJoinDTO s ");  
    return query.list();  
}
```

```
public List<ServicioJoinDTO> misservicios(long user){  
    Session session = _sessionFactory.getCurrentSession();  
    Query query = session.createQuery("select s From ServicioJoinDTO s Join s.idCliente u where  
        u.id_usuario = :user" );  
    query.setLong("user", user);
```

```
    return query.list();  
}
```

```
public List<ServicioJoinDTO> miserviciosp(long user){  
    Session session = _sessionFactory.getCurrentSession();  
    Query query = session.createQuery("select s From ServicioJoinDTO s where s.idProveedor =  
        :user" );  
    query.setLong("user", user);  
    return query.list();  
}
```

```
public final void setSessionFactory(SessionFactory sessionFactory) {  
    _sessionFactory = sessionFactory;  
}
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.metodos;
```

```
import java.util.List;
```

```
import mx.uam.azc.cbi.data.ProductoJoinDTO;
```

```
import mx.uam.azc.cbi.data.ProductoSaveDTO;
```

```
import mx.uam.azc.cbi.interfaces.AdminProductos;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

/**
 *
 * @author manzano
 */
public class AdminProductosImpl implements AdminProductos{

    private SessionFactory _sessionFactory;

    public long insertar(ProductoSaveDTO producto){

        Session session = _sessionFactory.getCurrentSession();

        session.save(producto);

        return producto.getIdProducto();

    }

    public ProductoSaveDTO load(Long idProducto){

        Session session = _sessionFactory.getCurrentSession();

        return (ProductoSaveDTO)session.load(ProductoSaveDTO.class, idProducto);

    }

    public ProductoJoinDTO loadproduct(Long idProducto){

        Session session = _sessionFactory.getCurrentSession();
```

```

        return (ProductoJoinDTO)session.load(ProductoJoinDTO.class, idProducto);
    }

    public void actualizar(ProductoSaveDTO producto){
        Session session = _sessionFactory.getCurrentSession();
        session.update(producto);
    }

    public List<ProductoJoinDTO> listar(){
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.createQuery("select p From ProductoJoinDTO p where
            p.isObsolete='false' ");
        return query.list();
    }

    public List<ProductoJoinDTO> misproductos(Long user){
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.createQuery("select p From ProductoJoinDTO p where
            p.isObsolete='false' ");
        return query.list();
    }

    public final void setSessionFactory(SessionFactory sessionFactory) {
        _sessionFactory = sessionFactory;
    }
}

```



```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.metodos;
```

```
import java.util.List;
```

```
import mx.uam.azc.cbi.data.OrdenCompraSaveDTO;
```

```
import mx.uam.azc.cbi.data.OrdenCompraJoinDTO;
```

```
import mx.uam.azc.cbi.interfaces.AdminOrdenes;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
/**
```

```
*
```

```
* @author gmendoza
```

```
*/
```

```
public class AdminOrdenesImpl implements AdminOrdenes{
```

```
    private SessionFactory _sessionFactory;
```

```
    public long insertar(OrdenCompraSaveDTO Orden){
```

```
        Session session = _sessionFactory.getCurrentSession();
```

```
        session.save(Orden);
```

```

        return Orden.getIdCompra();
    }

    public List<OrdenCompraJoinDTO> listar (long idUsuario){
        Session session = _sessionFactory.getCurrentSession();

        Query query = session.createQuery("select o From OrdenCompraJoinDTO o where
            o.idUsuario = :idUsuario");

        query.setLong("idUsuario", idUsuario);

        return query.list();
    }

    public List<OrdenCompraJoinDTO> listar (){
        Session session = _sessionFactory.getCurrentSession();

        Query query = session.createQuery("select o From OrdenCompraJoinDTO o ");

        return query.list();
    }

    public OrdenCompraJoinDTO load(Long idorden){
        Session session = _sessionFactory.getCurrentSession();

        return (OrdenCompraJoinDTO)session.load(OrdenCompraJoinDTO.class, idorden);
    }

    public final void setSessionFactory(SessionFactory sessionFactory) {
        _sessionFactory = sessionFactory;
    }
}

```

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import static javax.persistence.GenerationType.IDENTITY;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.OneToOne;
```

```
import javax.persistence.Table;
```

```
import org.hibernate.annotations.Proxy;
```

```
/**
```

```
*
```

```
* @author gmendoza
```

```
*/
```

```
@Entity @Table(name="orden_compra")
```

```
@Proxy (lazy=false)
```

```
public class OrdenCompraJoinDTO implements Serializable {
```

```
    private long idCompra;
```

```
    private UsuarioJoinDTO idUsuario;
```

```
    private long total;
```

```
    private long fecha;
```

```
    @Id @Column(name="id_orden")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public long getIdCompra() {
```

```
        return idCompra;
```

```
    }
```

```
    @OneToOne @JoinColumn (name="id_usuario")
```

```
    public UsuarioJoinDTO getIdUsuario() {
```

```
        return idUsuario;
```

```
    }
```

```
    @Column(name="total")
```

```
    public long getTotal() {
```

```
        return total;
```

```
    }
```

```
    @Column(name="fecha")
```

```
    public long getFecha() {
```

```
        return fecha;
```

```
}
```

```
public void setIdUsuario(UsuarioJoinDTO idUsuario) {  
    this.idUsuario = idUsuario;  
}
```

```
public void setIdCompra(long idCompra) {  
    this.idCompra = idCompra;  
}
```

```
public void setTotal(long total) {  
    this.total = total;  
}
```

```
public void setFecha(long fecha) {  
    this.fecha = fecha;  
}
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

/**
 *
 * @author manzano
 */

@Entity @Table(name="orden_compra")
@Proxy (lazy=false)

public class OrdenCompraSaveDTO implements Serializable {

    private long idCompra;
    private long idUsuario;
    private long total;
```

```
private long fecha;
```

```
@Id @Column(name="id_orden")
```

```
@GeneratedValue(strategy=IDENTITY)
```

```
public long getIdCompra() {
```

```
    return idCompra;
```

```
}
```

```
@Column(name="id_usuario")
```

```
public long getIdUsuario() {
```

```
    return idUsuario;
```

```
}
```

```
@Column(name="total")
```

```
public long getTotal() {
```

```
    return total;
```

```
}
```

```
@Column(name="fecha")
```

```
public long getFecha() {
```

```
    return fecha;
```

```
}
```

```
public void setIdCompra(long idCompra) {
```

```
    this.idCompra = idCompra;
```

```
}
```

```
public void setIdUsuario(long idUsuario) {
    this.idUsuario = idUsuario;
}

public void setTotal(long total) {
    this.total = total;
}

public void setFecha(long fecha) {
    this.fecha = fecha;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
```



```
import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.Table;

import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

import javax.persistence.Transient;

/**
 *
 * @author manzano
 */
@Entity @Table(name="productos")
@Proxy (lazy=false)

public class ProductoJoinDTO implements Serializable {

    private Long idProducto;

    private UsuarioJoinDTO idProveedor;

    private CatFabricanteDTO idFabricante;

    private String codigo;

    private String descripcion;

    private String nombre;

    private int tipo_venta = 0;

    private int descuento = 0;

    private long precio_unitario;

    private boolean isObsoleto;
```

```
private String tipoVenta;
```

```
@Id @Column(name="id_productos")  
@GeneratedValue(strategy=IDENTITY)  
public Long getIdProducto() {  
    return idProducto;  
}
```

```
@OneToOne @JoinColumn (name="id_fabricante")  
//@Column(name="id_fabricante")  
public CatFabricanteDTO getIdFabricante() {  
    return idFabricante;  
}
```

```
@OneToOne @JoinColumn (name="id_proveedor")  
//@Column(name="id_proveedor")  
public UsuarioJoinDTO getIdProveedor() {  
    return idProveedor;  
}
```

```
@Column(name="codigo")  
public String getCodigo() {  
    return codigo;  
}
```

```
@Column(name="descripcion")  
public String getDescripcion() {
```

```
    return descripcion;  
}
```

```
@Column(name="nombre")  
public String getNombre() {  
    return nombre;  
}
```

```
@Column(name="tipo_venta")  
public int getTipo_venta() {  
    return tipo_venta;  
}
```

```
@Column(name="descuento")  
public int getDescuento() {  
    return descuento;  
}
```

```
@Column(name="precio_unitario")  
public long getPrecio_unitario() {  
    return precio_unitario;  
}
```

```
@Column(name="isobsolete")  
public boolean isIsObsolete() {  
    return isObsolete;  
}
```

```
@Transient
```

```
public String getTipoVenta(){  
    return tipoVenta;  
}
```

```
public void setIdFabricante(CatFabricanteDTO idFabricante) {  
    this.idFabricante = idFabricante;  
}
```

```
public void setIdProveedor(UsuarioJoinDTO idProveedor) {  
    this.idProveedor = idProveedor;  
}
```

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

```
public void setCodigo(String codigo) {  
    this.codigo = codigo;  
}
```

```
public void setDescripcion(String descripcion) {  
    this.descripcion = descripcion;  
}
```

```
public void setNombre(String nombre) {
```

```
        this.nombre = nombre;
    }

    public void setTipo_venta(int tipo_venta) {
        this.tipo_venta = tipo_venta;
        switch(tipo_venta) {
            case 0: tipoVenta = "Pieza";
                break;
            case 1: tipoVenta = "Caja";
                break;
            case 2: tipoVenta = " Administrador ";
                break;
        }
    }

    public void setDescuento(int descuento) {
        this.descuento = descuento;
    }

    public void setPrecio_unitario(long precio_unitario) {
        this.precio_unitario = precio_unitario;
    }

    public void setIsObsolete(boolean isObsolete) {
        this.isObsolete = isObsolete;
    }

    public void setTipoVenta(String tipoVenta) {
```

```
        this.tipoVenta = tipoVenta;
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

/**
 *
 * @author manzano
 */
```

```
@Entity @Table(name="productos")
```

```
@Proxy (lazy=false)
```

```
public class ProductoSaveDTO implements Serializable {
```

```
    private Long idProducto;
```

```
    private Long idProveedor;
```

```
    private Long idFabricante;
```

```
    private String codigo;
```

```
    private String descripcion;
```

```
    private String nombre;
```

```
    private int tipo_venta;
```

```
    private int descuento;
```

```
    private long precio_unitario;
```

```
    private boolean isObsoleto;
```

```
    @Id @Column(name="id_productos")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public Long getIdProducto() {
```

```
        return idProducto;
```

```
    }
```

```
    // @OneToOne @JoinColumn (name="id_fabricante")
```

```
    @Column(name="id_fabricante")
```

```
    public Long getIdFabricante() {
```

```
        return idFabricante;
```

```
}
```

```
//@OneToOne @JoinColumn (name="id_usuario")
```

```
@Column(name="id_proveedor")
```

```
public Long getIdProveedor() {
```

```
    return idProveedor;
```

```
}
```

```
@Column(name="codigo")
```

```
public String getCodigo() {
```

```
    return codigo;
```

```
}
```

```
@Column(name="descripcion")
```

```
public String getDescripcion() {
```

```
    return descripcion;
```

```
}
```

```
@Column(name="nombre")
```

```
public String getNombre() {
```

```
    return nombre;
```

```
}
```

```
@Column(name="tipo_venta")
```

```
public int getTipo_venta() {
```

```
    return tipo_venta;
```

```
}
```



```
@Column(name="descuento")
```

```
public int getDescuento() {
```

```
    return descuento;
```

```
}
```

```
@Column(name="precio_unitario")
```

```
public long getPrecio_unitario() {
```

```
    return precio_unitario;
```

```
}
```

```
@Column(name="isobsolete")
```

```
public boolean isIsObsolete() {
```

```
    return isObsolete;
```

```
}
```

```
public void setIdFabricante(Long idFabricante) {
```

```
    this.idFabricante = idFabricante;
```

```
}
```

```
public void setIdProveedor(Long idProveedor) {
```

```
    this.idProveedor = idProveedor;
```

```
}
```

```
public void setIdProducto(Long idProducto) {
```

```
    this.idProducto = idProducto;
```

```
}
```

```
public void setCodigo(String codigo) {  
    this.codigo = codigo;  
}
```

```
public void setDescripcion(String descripcion) {  
    this.descripcion = descripcion;  
}
```

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

```
public void setTipo_venta(int tipo_venta) {  
    this.tipo_venta = tipo_venta;  
}
```

```
public void setDescuento(int descuento) {  
    this.descuento = descuento;  
}
```

```
public void setPrecio_unitario(long precio_unitario) {  
    this.precio_unitario = precio_unitario;  
}
```

```
public void setIsObsolete(boolean isObsolete) {  
    this.isObsolete = isObsolete;  
}
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.OneToOne;
```

```
import javax.persistence.Table;
```

```
import org.hibernate.annotations.Proxy;
```

```
import static javax.persistence.GenerationType.IDENTITY;
```

```
import javax.persistence.Transient;
```

```
/**
```

```
*
```

```
* @author manzano
```

```
*/
```

```
@Entity @Table(name="servicios")
@Proxy (lazy=false)
public class ServicioJoinDTO implements Serializable{

    private Long idServicio;
    private UsuarioJoinDTO idCliente;
    private UsuarioJoinDTO idProveedor;
    private UsuarioJoinDTO idUltima;
    private String descripcion;
    private CatEstadoServiciosDTO status;
    private CatServiciosDTO servicio;
    private long fecha_peticion;
    private long ultima_modificacion;
    private int prioridad = 3;
    private String priority;

    @Id @Column(name="id_servicio")
    @GeneratedValue(strategy=IDENTITY)
    public Long getIdServicio() {
        return idServicio;
    }

    @OneToOne @JoinColumn(name="id_proveedor")
    public UsuarioJoinDTO getIdProveedor() {
        return idProveedor;
    }
}
```

```
@OneToOne @JoinColumn(name="id_modificacion")  
  
public UsuarioJoinDTO getIdUltima() {  
    return idUltima;  
}
```

```
@Column(name="descripcion")  
  
public String getDescripcion() {  
    return descripcion;  
}
```

```
@Column(name="fecha_peticion")  
  
public long getFecha_peticion() {  
    return fecha_peticion;  
}
```

```
@OneToOne @JoinColumn(name="id_cliente")  
  
public UsuarioJoinDTO getIdCliente() {  
    return idCliente;  
}
```

```
@OneToOne @JoinColumn(name="status")  
  
public CatEstadoServiciosDTO getStatus() {  
    return status;  
}
```

```
@OneToOne @JoinColumn(name="servicio")  
  
public CatServiciosDTO getServicio() {  
    return servicio;  
}
```

```
}
```

```
@Column(name="ultima_modificacion")
```

```
public long getUltima_modificacion() {
```

```
    return ultima_modificacion;
```

```
}
```

```
@Column(name="prioridad")
```

```
public int getPrioridad() {
```

```
    return prioridad;
```

```
}
```

```
@Transient
```

```
public String getPriority() {
```

```
    return priority;
```

```
}
```

```
public void setDescripcion(String descripcion) {
```

```
    this.descripcion = descripcion;
```

```
}
```

```
public void setFecha_peticion(long fecha_peticion) {
```

```
    this.fecha_peticion = fecha_peticion;
```

```
}
```

```
public void setIdCliente(UsuarioJoinDTO idCliente) {
```

```
    this.idCliente = idCliente;
```

```
}
```

```
public void setIdServicio(Long idServicio) {  
    this.idServicio = idServicio;  
}
```

```
public void setStatus(CatEstadoServiciosDTO status) {  
    this.status = status;  
}
```

```
public void setIdProveedor(UsuarioJoinDTO idProveedor) {  
    this.idProveedor = idProveedor;  
}
```

```
public void setUltima_modificacion(long ultima_modificacion) {  
    this.ultima_modificacion = ultima_modificacion;  
}
```

```
public void setPrioridad(int prioridad) {  
    this.prioridad = prioridad;  
    switch(prioridad){  
        case 0:  
            priority = "Urgente";  
            break;  
        case 1:  
            priority = "Alta";  
            break;  
        case 2:  
            priority = "Media";
```

```
        break;
    case 3:
        priority = "Baja";
        break;
    }
}

public void setPriority(String priority) {
    this.priority = priority;
}

public void setIdUltima(UsuarioJoinDTO idUltima) {
    this.idUltima = idUltima;
}

public void setServicio(CatServiciosDTO servicio) {
    this.servicio = servicio;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
```



```
package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

/**
 *
 * @author manzano
 */

@Entity @Table(name="servicios")
@Proxy (lazy=false)
public class ServicioSaveDTO implements Serializable{

    private Long idServicio;
    private long idCliente;
    private long idProveedor;
    private long idUltima;
    private String descripcion;
    private int status;
```

```
private int servicio;  
  
private long fecha_peticion;  
  
private long ultima_modificacion;  
  
private int prioridad = 3;
```

```
@Id @Column(name="id_servicio")  
  
@GeneratedValue(strategy=IDENTITY)  
  
public Long getIdServicio() {  
    return idServicio;  
}
```

```
@Column(name="id_proveedor")  
  
public long getIdProveedor() {  
    return idProveedor;  
}
```

```
@Column(name="id_modificacion")  
  
public long getIdUltima() {  
    return idUltima;  
}
```

```
@Column(name="descripcion")  
  
public String getDescripcion() {  
    return descripcion;  
}
```

```
@Column(name="fecha_peticion")
```

```
public long getFecha_peticion() {  
    return fecha_peticion;  
}
```

```
@Column(name="id_cliente")  
public long getIdCliente() {  
    return idCliente;  
}
```

```
@Column(name="status")  
public int getStatus() {  
    return status;  
}
```

```
@Column(name="servicio")  
public int getServicio() {  
    return servicio;  
}
```

```
@Column(name="ultima_modificacion")  
public long getUltima_modificacion() {  
    return ultima_modificacion;  
}
```

```
@Column(name="prioridad")  
public int getPrioridad() {  
    return prioridad;  
}
```

```
public void setDescripcion(String descripcion) {  
    this.descripcion = descripcion;  
}
```

```
public void setFecha_peticion(long fecha_peticion) {  
    this.fecha_peticion = fecha_peticion;  
}
```

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

```
public void setIdServicio(Long idServicio) {  
    this.idServicio = idServicio;  
}
```

```
public void setStatus(int status) {  
    this.status = status;  
}
```

```
public void setUltima_modificacion(long ultima_modificacion) {  
    this.ultima_modificacion = ultima_modificacion;  
}
```

```
public void setPrioridad(int prioridad) {  
    this.prioridad = prioridad;  
}
```

```
}

public void setIdProveedor(long idProveedor) {
    this.idProveedor = idProveedor;
}

public void setIdUltima(long idUltima) {
    this.idUltima = idUltima;
}

public void setServicio(int servicio) {
    this.servicio = servicio;
}
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.Collection;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OneToOne;

import javax.persistence.Table;

import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

import static javax.persistence.FetchType.LAZY;

/**
 *
 * @author manzano
 */

@Entity @Table(name="temas")

@Proxy (lazy=false)

public class TemaJoinDTO implements Serializable, Comparable<MensajeSaveDTO>{

    private Collection<MensajeSaveDTO> _mensajes = new ArrayList<MensajeSaveDTO>();

    private Long idTema;

    private UsuarioJoinDTO id_usuario;

    private String nombre;

    private CatCategoriasDTO categoria;

    private String descripcion;

    private long date;
```

```
private Boolean isobsolete;

//private Calendar fecha = Calendar.getInstance();

@Id @Column(name="id_temas")
@GeneratedValue(strategy=IDENTITY)
public Long getIdTema() {
    return idTema;
}

@OneToOne @JoinColumn (name="id_usuario")
public UsuarioJoinDTO getId_usuario() {
    return id_usuario;
}

@Column(name="descripcion")
public String getDescripcion() {
    return descripcion;
}

@Column(name="fecha")
public long getDate() {
    return date;
}

//@Column(name="id_usuario")
@Column(name="nombre")
public String getNombre() {
```

```
        return nombre;
    }

    @OneToMany(fetch=LAZY)
    @JoinColumn(name="id_temas")
    public Collection<MensajeSaveDTO> getMensajes(){
        return _mensajes;
    }

    //@Column(name="categoria")
    @OneToOne @JoinColumn (name="categoria")
    public CatCategoriasDTO getCategoria(){
        return categoria;
    }

    @Column(name="isobsolete")
    public Boolean getIsobsolete() {
        return isobsolete;
    }

    public void setDate(long date) {
        this.date = date;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}
```



```
public void setIdTema(Long idTema) {  
    this.idTema = idTema;  
}
```

```
public void setId_usuario(UsuarioJoinDTO id_usuario) {  
    this.id_usuario = id_usuario;  
}
```

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

```
public void setCategoria(CatCategoriasDTO categoria){  
    this.categoria = categoria;  
}
```

```
public void setIsobsolete(Boolean isobsolete) {  
    this.isobsolete = isobsolete;  
}
```

```
public void setMensajes( Collection<MensajeSaveDTO> mensajes ){  
    _mensajes = mensajes;  
}
```

```
@Override
```

```
public int compareTo( MensajeSaveDTO mensaje ) {
```

```
        if(mensaje.getIdTema() != idTema)
            return new Integer("" + (idTema - mensaje.getIdTema())).intValue();
        return 0;
    }
```

```
    public void addMensaje( MensajeSaveDTO mensaje ){
        _mensajes.add( mensaje );
        mensaje.setIdTema( idTema );
    }
```

```
}
```

```
/*
```

```
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import java.util.ArrayList;
```

```
import java.util.Collection;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.OneToOne;
```

```
import javax.persistence.Table;

import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;
import static javax.persistence.FetchType.LAZY;

/**
 *
 * @author manzano
 */

@Entity @Table(name="temas")
@Proxy (lazy=false)

public class TemaSaveDTO implements Serializable, Comparable<MensajeSaveDTO>{

    private Collection<MensajeSaveDTO> _mensajes = new ArrayList<MensajeSaveDTO>();

    private Long idTema;

    private Long id_usuario;

    private String nombre;

    private int categoria;

    private String descripcion;

    private long date;

    private Boolean isobsolete;

    //private Calendar fecha = Calendar.getInstance();
```

```
@Id @Column(name="id_temas")
@GeneratedValue(strategy=IDENTITY)
public Long getIdTema() {
    return idTema;
}
```

```
@Column(name="descripcion")
public String getDescripcion() {
    return descripcion;
}
```

```
@Column(name="fecha")
public long getDate() {
    return date;
}
```

```
@Column(name="id_usuario")
public Long getId_usuario() {
    return id_usuario;
}
```

```
@Column(name="nombre")
public String getNombre() {
    return nombre;
}
```

```
@OneToMany(fetch=LAZY)
@JoinColumn(name="id_temas")
```

```
public Collection<MensajeSaveDTO> getMensajes(){  
    return _mensajes;  
}
```

```
@Column(name="categoria")  
public int getCategoria(){  
    return categoria;  
}
```

```
@Column(name="isobsolete")  
public Boolean getIsobsolete() {  
    return isobsolete;  
}
```

```
public void setDate(long date) {  
    this.date = date;  
}
```

```
public void setDescription(String descripcion) {  
    this.descripcion = descripcion;  
}
```

```
public void setIdTema(Long idTema) {  
    this.idTema = idTema;  
}
```

```
public void setId_usuario(Long id_usuario) {
```

```
        this.id_usuario = id_usuario;
    }

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

    public void setIsobsolete(Boolean isobsolete) {
        this.isobsolete = isobsolete;
    }

    public void setCategoria(int categoria){
        this.categoria = categoria;
    }

    public void setMensajes( Collection<MensajeSaveDTO> mensajes ){
        _mensajes = mensajes;
    }

    @Override
    public int compareTo( MensajeSaveDTO mensaje ) {
        if(mensaje.getIdTema() != idTema)
            return new Integer("" + (idTema - mensaje.getIdTema())).intValue();
        return 0;
    }

    public void addMensaje( MensajeSaveDTO mensaje ){
```

```
_mensajes.add( mensaje );
mensaje.setIdTema( idTema );
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Transient;
import org.hibernate.annotations.Proxy;

/**
 *
 * @author Manzano
 */
```

```
*/
```

```
@Entity @Table(name="tiempos")
```

```
@Proxy (lazy=false)
```

```
public class TiemposJoinDTO implements Serializable{
```

```
    private long idTiempo;
```

```
    private ServicioJoinDTO idServicio;
```

```
    private UsuarioJoinDTO idProveedor;
```

```
    private UsuarioJoinDTO idUltima;
```

```
    private CatEstadoServiciosDTO estado;
```

```
    private long fecha_modificacion;
```

```
    private String resumen;
```

```
    private int prioridad;
```

```
    private String priority;
```

```
    @Id @Column(name="id_tiempo")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public long getIdTiempo() {
```

```
        return idTiempo;
```

```
    }
```

```
    @OneToOne @JoinColumn(name="id_servicio")
```

```
    public ServicioJoinDTO getIdServicio() {
```

```
        return idServicio;
```

```
    }
```



```
@OneToOne @JoinColumn(name="id_proveedor")  
  
public UsuarioJoinDTO getIdProveedor() {  
    return idProveedor;  
}
```

```
@OneToOne @JoinColumn(name="id_modificacion")  
  
public UsuarioJoinDTO getIdUltima() {  
    return idUltima;  
}
```

```
//@Column(name="status")  
  
@OneToOne @JoinColumn(name="status")  
  
public CatEstadoServiciosDTO getEstado() {  
    return estado;  
}
```

```
@Column(name="ultima_modificacion")  
  
public long getFecha_modificacion() {  
    return fecha_modificacion;  
}
```

```
@Column(name="descripcion")  
  
public String getResumen() {  
    return resumen;  
}
```

```
@Column(name="prioridad")  
  
public int getPrioridad() {
```

```
    return prioridad;
}
```

```
@Transient
```

```
public String getPriority() {
    return priority;
}
```

```
public void setIdTiempo(long idTiempo) {
    this.idTiempo = idTiempo;
}
```

```
public void setIdServicio(ServicioJoinDTO idServicio) {
    this.idServicio = idServicio;
}
```

```
public void setIdProveedor(UsuarioJoinDTO idProveedor) {
    this.idProveedor = idProveedor;
}
```

```
public void setIdUltima(UsuarioJoinDTO idUltima) {
    this.idUltima = idUltima;
}
```

```
public void setEstado(CatEstadoServiciosDTO estado) {
    this.estado = estado;
}
```

```
public void setFecha_modificacion(long fecha_modificacion) {  
    this.fecha_modificacion = fecha_modificacion;  
}
```

```
public void setResumen(String resumen) {  
    this.resumen = resumen;  
}
```

```
public void setPrioridad(int prioridad) {  
    this.prioridad = prioridad;  
    switch(prioridad){  
        case 0:  
            priority = "Urgente";  
            break;  
        case 1:  
            priority = "Alto";  
            break;  
        case 2:  
            priority = "Medio";  
            break;  
        case 3:  
            priority = "Bajo";  
            break;  
    }  
}
```

```
public void setPriority(String priority) {  
    this.priority = priority;
```

```
}
```

```
}
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import static javax.persistence.GenerationType.IDENTITY;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
import org.hibernate.annotations.Proxy;
```

```
/**
```

```
*
```

```
* @author Manzano
```

```
*/
```

```
@Entity @Table(name="tiempos")
```

```
@Proxy (lazy=false)
```

```
public class TiemposSaveDTO implements Serializable{
```

```
    private long idTiempo;
```

```
    private long idServicio;
```

```
    private long idProveedor;
```

```
    private long idUltima;
```

```
    private int estado;
```

```
    private long fecha_modificacion;
```

```
    private String resumen;
```

```
    private int prioridad;
```

```
    @Id @Column(name="id_tiempo")
```

```
    @GeneratedValue(strategy=IDENTITY)
```

```
    public long getIdTiempo() {
```

```
        return idTiempo;
```

```
    }
```

```
    @Column(name="id_servicio")
```

```
    public long getIdServicio() {
```

```
        return idServicio;
```

```
    }
```

```
    @Column(name="id_proveedor")
```

```
    public long getIdProveedor() {
```

```
    return idProveedor;  
}
```

```
@Column(name="id_modificacion")  
public long getIdUltima() {  
    return idUltima;  
}
```

```
@Column(name="status")  
public int getEstado() {  
    return estado;  
}
```

```
@Column(name="ultima_modificacion")  
public long getFecha_modificacion() {  
    return fecha_modificacion;  
}
```

```
@Column(name="descripcion")  
public String getResumen() {  
    return resumen;  
}
```

```
@Column(name="prioridad")  
public int getPrioridad() {  
    return prioridad;  
}
```

```
public void setIdTiempo(long idTiempo) {  
    this.idTiempo = idTiempo;  
}
```

```
public void setIdServicio(long idServicio) {  
    this.idServicio = idServicio;  
}
```

```
public void setIdProveedor(long idProveedor) {  
    this.idProveedor = idProveedor;  
}
```

```
public void setIdUltima(long idUltima) {  
    this.idUltima = idUltima;  
}
```

```
public void setEstado(int estado) {  
    this.estado = estado;  
}
```

```
public void setFecha_modificacion(long fecha_modificacion) {  
    this.fecha_modificacion = fecha_modificacion;  
}
```

```
public void setResumen(String resumen) {  
    this.resumen = resumen;  
}
```

```
public void setPrioridad(int prioridad) {
    this.prioridad = prioridad;
}

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Version;
import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;
```



```
/**
```

```
*
```

```
* @author Manzano
```

```
*/
```

```
@Entity @Table(name="usuarios")
```

```
@Proxy (lazy=false)
```

```
public class UsuarioJoinDTO implements Serializable{
```

```
    private Long id_usuario;
```

```
    private CredencialDTO id_credencial;
```

```
    private String nombre;
```

```
    private String apelliop;
```

```
    private String apellidom;
```

```
    private long fecha_nacimiento;
```

```
    private String rfc;
```

```
    private String telefono;
```

```
    private String celular;
```

```
    private String email;
```

```
    private CatTipoUsuarioDTO tipo_usuario;
```

```
    private String calle;
```

```
    private String colonia;
```

```
    private String delegacion;
```

```
    private CatEstadosDTO entidad;
```

```
    private int cp;
```

```
private long fecha_ingreso;
```

```
private int version;
```

```
@Id @Column(name="id_usuario")
```

```
@GeneratedValue(strategy=IDENTITY)
```

```
public Long getId_usuario() {
```

```
    return id_usuario;
```

```
}
```

```
//@Column(name="id_credencial")
```

```
@OneToOne @JoinColumn(name="id_credencial")
```

```
public CredencialDTO getId_credencial() {
```

```
    return id_credencial;
```

```
}
```

```
@Column(name="apellidom")
```

```
public String getApellidom() {
```

```
    return apellidom;
```

```
}
```

```
@Column(name="apelliodp")
```

```
public String getApelliop() {
```

```
    return apelliop;
```

```
}
```

```
@Column(name="calle")
```

```
public String getCalle() {
```

```
    return calle;  
}
```

```
@Column(name="celular")  
public String getCelular() {  
    return celular;  
}
```

```
@Column(name="colonia")  
public String getColonia() {  
    return colonia;  
}
```

```
@Column(name="cp")  
public int getCp() {  
    return cp;  
}
```

```
@Column(name="delegacion")  
public String getDelegacion() {  
    return delegacion;  
}
```

```
@Column(name="email")  
public String getEmail() {  
    return email;  
}
```

```
//@Column(name="entidad")
@OneToOne @JoinColumn(name="entidad")
public CatEstadosDTO getEntidad() {
    return entidad;
}

@Column(name="fecha_ingreso") // Fecha de ingreso
public long getFecha_ingreso() {
    return fecha_ingreso;
}

@Column(name="fecha_nacimiento") // Fecha de nacimiento
public long getFecha_nacimiento() {
    return fecha_nacimiento;
}

@Column(name="nombre")
public String getNombre() {
    return nombre;
}

@Column(name="rfc")
public String getRfc() {
    return rfc;
}

@Column(name="particular")
public String getTelefono() {
```

```
        return telefono;
    }

    //@Column(name="tipo_usuario") // Tipo Usuario
    @OneToOne @JoinColumn(name="tipo_usuario")
    public CatTipoUsuarioDTO getTipo_usuario() {
        return tipo_usuario;
    }
}
```

```
@Version
@Column(name="version")
public int getVersion() {
    return version;
}
}
```

```
// Setters
```

```
public void setApellidoom(String apellidoom) {
    this.apellidoom = apellidoom;
}
}
```

```
public void setApelliop(String apelliop) {
    this.apelliop = apelliop;
}
}
```

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

```
}
```

```
public void setCelular(String celular) {
```

```
    this.celular = celular;
```

```
}
```

```
public void setColonia(String colonia) {
```

```
    this.colonia = colonia;
```

```
}
```

```
public void setCp(int cp) {
```

```
    this.cp = cp;
```

```
}
```

```
public void setDelegacion(String delegacion) {
```

```
    this.delegacion = delegacion;
```

```
}
```

```
public void setEmail(String email) {
```

```
    this.email = email;
```

```
}
```

```
public void setEntidad(CatEstadosDTO entidad) {
```

```
    this.entidad = entidad;
```

```
}
```

```
public void setFecha_ingreso(long fecha_ingreso) {
```

```
    this.fecha_ingreso = fecha_ingreso;
```

```
}
```

```
public void setFecha_nacimiento(long fecha_nacimiento) {
```

```
    this.fecha_nacimiento = fecha_nacimiento;
```

```
}
```

```
public void setId_credencial(CredencialDTO id_credencial) {
```

```
    this.id_credencial = id_credencial;
```

```
}
```

```
public void setId_usuario(Long id_usuario) {
```

```
    this.id_usuario = id_usuario;
```

```
}
```

```
public void setNombre(String nombre) {
```

```
    this.nombre = nombre;
```

```
}
```

```
public void setRfc(String rfc) {
```

```
    this.rfc = rfc;
```

```
}
```

```
public void setTelefono(String telefono) {
```

```
    this.telefono = telefono;
```

```
}
```

```
public void setTipo_usuario(CatTipoUsuarioDTO tipo_usuario) {
```

```
    this.tipo_usuario = tipo_usuario;
```

```
    }

    public void setVersion(int version) {
        this.version = version;
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Version;
import org.hibernate.annotations.Proxy;

import static javax.persistence.GenerationType.IDENTITY;

/**
 *
```



```
* @author Manzano
```

```
*/
```

```
@Entity @Table(name="usuarios")
```

```
@Proxy (lazy=false)
```

```
public class UsuarioSaveDTO implements Serializable{
```

```
    private Long id_usuario;
```

```
    private Long id_credencial;
```

```
    private String nombre;
```

```
    private String apelliop;
```

```
    private String apellidom;
```

```
    private long fecha_nacimiento;
```

```
    private String rfc;
```

```
    private String telefono;
```

```
    private String celular;
```

```
    private String email;
```

```
    private int tipo_usuario;
```

```
    private String calle;
```

```
    private String colonia;
```

```
    private String delegacion;
```

```
    private int entidad;
```

```
    private int cp;
```

```
    private long fecha_ingreso;
```

```
    private int version;
```

```
@Id @Column(name="id_usuario")
@GeneratedValue(strategy=IDENTITY)
public Long getId_usuario() {
    return id_usuario;
}
```

```
@Column(name="id_credencial")
//@OneToOne @JoinColumn(name="id_credencial")
public Long getId_credencial() {
    return id_credencial;
}
```

```
@Column(name="apellidom")
public String getApellidom() {
    return apellidom;
}
```

```
@Column(name="apelliop")
public String getApelliop() {
    return apelliop;
}
```

```
@Column(name="calle")
public String getCalle() {
    return calle;
}
```

```
}
```

```
@Column(name="celular")
```

```
public String getCelular() {
```

```
    return celular;
```

```
}
```

```
@Column(name="colonia")
```

```
public String getColonia() {
```

```
    return colonia;
```

```
}
```

```
@Column(name="cp")
```

```
public int getCp() {
```

```
    return cp;
```

```
}
```

```
@Column(name="delegacion")
```

```
public String getDelegacion() {
```

```
    return delegacion;
```

```
}
```

```
@Column(name="email")
```

```
public String getEmail() {
```

```
    return email;
```

```
}
```

```
@Column(name="entidad")
```

```
public int getEntidad() {  
    return entidad;  
}
```

```
@Column(name="fecha_ingreso") // Fecha de ingreso  
public long getFecha_ingreso() {  
    return fecha_ingreso;  
}
```

```
@Column(name="fecha_nacimiento") // Fecha de nacimiento  
public long getFecha_nacimiento() {  
    return fecha_nacimiento;  
}
```

```
@Column(name="nombre")  
public String getNombre() {  
    return nombre;  
}
```

```
@Column(name="rfc")  
public String getRfc() {  
    return rfc;  
}
```

```
@Column(name="particular")  
public String getTelefono() {  
    return telefono;  
}
```

```
@Column(name="tipo_usuario") // Tipo Usuario
public int getTipo_usuario() {
    return tipo_usuario;
}
```

```
@Version
@Column(name="version")
public int getVersion() {
    return version;
}
```

```
// Setters
```

```
public void setApellidom(String apellidom) {
    this.apellidom = apellidom;
}
```

```
public void setApelliop(String apelliop) {
    this.apelliop = apelliop;
}
```

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

```
public void setCelular(String celular) {
```

```
    this.celular = celular;
}

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

public void setCp(int cp) {
    this.cp = cp;
}

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

public void setEmail(String email) {
    this.email = email;
}

public void setEntidad(int entidad) {
    this.entidad = entidad;
}

public void setFecha_ingreso(long fecha_ingreso) {
    this.fecha_ingreso = fecha_ingreso;
}

public void setFecha_nacimiento(long fecha_nacimiento) {
```

```
    this.fecha_nacimiento = fecha_nacimiento;
}

public void setId_credencial(Long id_credencial) {
    this.id_credencial = id_credencial;
}

public void setId_usuario(Long id_usuario) {
    this.id_usuario = id_usuario;
}

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

public void setRfc(String rfc) {
    this.rfc = rfc;
}

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

public void setTipo_usuario(int tipo_usuario) {
    this.tipo_usuario = tipo_usuario;
}

public void setVersion(int version) {
```

```
        this.version = version;
    }

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mx.uam.azc.cbi.data;

import java.io.Serializable;

/**
 *
 * @author gmendoza
 */

public class CarritoComprasDTO implements Serializable {

    private ProductoJoinDTO idProducto;

    private int cantidad = 1;

    public ProductoJoinDTO getIdProducto() {
        return idProducto;
    }
}
```



```
}
```

```
public void setIdProducto(ProductoJoinDTO idProducto) {
```

```
    this.idProducto = idProducto;
```

```
}
```

```
public int getCantidad() {
```

```
    return cantidad;
```

```
}
```

```
public void setCantidad(int cantidad) {
```

```
    this.cantidad = cantidad;
```

```
}
```

```
}
```

```
/*
```

```
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package mx.uam.azc.cbi.data;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;

import org.hibernate.annotations.Proxy;

/**
 *
 * @author gmendoza
 */

@Entity @Table (name="cat_categorias")
@Proxy(lazy=false)

public class CatCategoriasDTO implements Serializable{

    private int _id;

    private String _categoria;

    @Id @Column(name="id_categoria")

    public int getId() {

        return _id;

    }

    @Column(name="categoria")

    public String getCategoria() {

        return _categoria;

    }

}
```

```
public void setId(int id) {  
    _id = id;  
}  
  
public void setCategoria(String categoria) {  
    _categoria = categoria;  
}  
  
}  
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package mx.uam.azc.cbi.data;  
  
import org.hibernate.annotations.Proxy;  
  
import java.io.Serializable;  
import javax.persistence.*;  
  
/**  
 *  
 * @author manzano  
 */  
@Entity @Table (name="cat_estados")
```

```
@Proxy(lazy=false)
```

```
public class CatEstadosDTO implements Serializable{
```

```
    private int id;
```

```
    private String estado;
```

```
    @Id @Column(name="id_estado")
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    @Column(name="estado")
```

```
    public String getEstado() {
```

```
        return estado;
```

```
    }
```

```
    public void setId(int id) {
```

```
        this.id = id;
```

```
    }
```

```
    public void setEstado(String estado) {
```

```
        this.estado = estado;
```

```
    }
```

```
}
```

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

```
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports  
  http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="ServicioAdmin"  
  language="groovy" pageWidth="842" pageHeight="595" orientation="Landscape"  
  columnWidth="802" leftMargin="20" rightMargin="20" topMargin="20" bottomMargin="20"  
  uuid="ef4df39c-4c62-47ec-9bd2-4382cd0ebf6b">
```

```
  <property name="ireport.zoom" value="1.0"/>
```

```
  <property name="ireport.x" value="0"/>
```

```
  <property name="ireport.y" value="0"/>
```

```
  <queryString language="SQL">
```

```
    <![CDATA[SELECT usuario.nombre || ' ' || usuario.apellido || ' ' || usuario.apellidom  
    as cliente,
```

```
    proveedor.nombre||' '||proveedor.apellido||' '||proveedor.apellidom AS proveedor,
```

```
    servicio.descripcion as descripcion,
```

```
    ( SELECT estado FROM cat_estadoservicio WHERE id_estado = servicio.status ) as status,
```

```
    case prioridad
```

```
      when 0 then 'Urgente'
```

```
      when 1 then 'Alto'
```

```
      when 2 then 'Medio'
```

```
      when 3 then 'Bajo'
```

```
    end prioridad,
```

```
    TO_CHAR(TO_TIMESTAMP(fecha_peticion),'DD/MM/YYYY') as fecha
```

```
  FROM
```

```
    usuarios usuario INNER JOIN servicios servicio ON usuario.id_usuario = servicio.id_cliente
```

```
    INNER JOIN usuarios proveedor ON servicio.id_proveedor = proveedor.id_usuario]]>
```

```
  </queryString>
```

```
  <field name="cliente" class="java.lang.String"/>
```

```
  <field name="proveedor" class="java.lang.String"/>
```

```
  <field name="descripcion" class="java.lang.String"/>
```

```

<field name="status" class="java.lang.String"/>
<field name="prioridad" class="java.lang.String"/>
<field name="fecha" class="java.lang.String"/>
<variable name="proveedor_1" class="java.lang.Integer" resetType="Group"
resetGroup="proveedor" calculation="Count">
    <variableExpression><![CDATA[{$F{proveedor}}]></variableExpression>
</variable>
<group name="proveedor" isStartNewPage="true">
    <groupExpression><![CDATA[{$F{proveedor}}]></groupExpression>
    <groupHeader>
        <band height="53">
            <textField>
                <reportElement uuid="9183526d-f123-4e58-ab32-
7acf8222db7b" x="100" y="0" width="342" height="20"/>
                <textElement>
                    <font size="14" isBold="true"/>
                </textElement>
            </textField>
            <textFieldExpression><![CDATA[{$F{proveedor}}]></textFieldExpression>
            </textField>
            <staticText>
                <reportElement uuid="f24867f6-dd32-4d97-8f3e-
dd734ae85131" x="0" y="0" width="100" height="20" bgcolor="#800000"/>
                <textElement>
                    <font size="14" isBold="true"/>
                </textElement>
                <text><![CDATA[Proveedor:]]></text>
            </staticText>
            <staticText>

```

<reportElement uuid="398658bb-8388-4a40-ab48-072e43f6c411" mode="Opaque" x="0" y="19" width="301" height="34" forecolor="#FFFFFF" bgcolor="#0080C0"/>

<textElement textAlignment="Center">

<font isBold="true"/>

</textElement>

<text><![CDATA[Descripción]]></text>

</staticText>

<staticText>

<reportElement uuid="398658bb-8388-4a40-ab48-072e43f6c411" mode="Opaque" x="486" y="19" width="165" height="34" forecolor="#FFFFFF" bgcolor="#0080C0"/>

<textElement textAlignment="Center">

<font isBold="true"/>

</textElement>

<text><![CDATA[Estatus]]></text>

</staticText>

<staticText>

<reportElement uuid="398658bb-8388-4a40-ab48-072e43f6c411" mode="Opaque" x="623" y="19" width="179" height="34" forecolor="#FFFFFF" bgcolor="#0080C0"/>

<textElement textAlignment="Center">

<font isBold="true"/>

</textElement>

<text><![CDATA[Fecha]]></text>

</staticText>

<staticText>

<reportElement uuid="398658bb-8388-4a40-ab48-072e43f6c411" mode="Opaque" x="301" y="19" width="185" height="34" forecolor="#FFFFFF" bgcolor="#0080C0"/>

<textElement textAlignment="Center">

```
<font isBold="true"/>
</textElement>
<text><![CDATA[Cliente]]></text>
</staticText>
</band>
</groupHeader>
<groupFooter>
<band height="26">
<textField>
<reportElement uuid="92aada59-81f6-4d39-92a6-
5efd14a25571" x="640" y="0" width="162" height="26"/>
<textElement/>
<textFieldExpression><![CDATA["Total Ordenes: " +
${proveedor_1}]]></textFieldExpression>
</textField>
</band>
</groupFooter>
</group>
<group name="cliente">
<groupExpression><![CDATA[${cliente}]]></groupExpression>
</group>
<group name="prioridad">
<groupExpression><![CDATA[${prioridad}]]></groupExpression>
</group>
<group name="status">
<groupExpression><![CDATA[${status}]]></groupExpression>
</group>
<background>
<band splitType="Stretch"/>
```



```
</background>

<title>
  <band height="34" splitType="Stretch">
    <staticText>
      <reportElement uuid="bc4b30ab-c67e-4620-b678-4c3a1a204dba"
x="0" y="0" width="284" height="34"/>
      <textElement>
        <font fontName="Times New Roman" size="20"
isBold="true" isItalic="true" pdfFontName="Helvetica-BoldOblique"/>
      </textElement>
      <text><![CDATA[Reporte de ordenes de servicio ]]></text>
    </staticText>
  </band>
</title>

<pageHeader>
  <band height="35" splitType="Stretch"/>
</pageHeader>

<detail>
  <band height="30" splitType="Stretch">
    <textField evaluationTime="Group" evaluationGroup="proveedor">
      <reportElement uuid="1c81c4f0-8a6a-4744-850a-ff378eb9770b"
x="284" y="0" width="174" height="30"/>
      <textElement/>
      <textFieldExpression><![CDATA[{$F{cliente}}]></textFieldExpression>
    </textField>
    <textField>
      <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="0" y="0" width="284" height="30"/>
      <textElement/>
    </textField>
  </band>
</detail>
```

```

<textFieldExpression><![CDATA[{$F{descripcion}}]></textFieldExpression>
    </textField>
    <textField>
        <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="458" y="0" width="182" height="30"/>
        <textElement textAlignment="Center"/>
    </textField>
    <textFieldExpression><![CDATA[{$F{status}}]></textFieldExpression>
    </textField>
    <textField>
        <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="640" y="0" width="162" height="30"/>
        <textElement textAlignment="Center"/>
    </textField>
    <textFieldExpression><![CDATA[{$F{fecha}}]></textFieldExpression>
    </textField>
</band>
</detail>
<columnFooter>
    <band height="45" splitType="Stretch"/>
</columnFooter>
<pageFooter>
    <band height="54" splitType="Stretch"/>
</pageFooter>
<summary>
    <band height="42" splitType="Stretch"/>
</summary>
</jasperReport>

```

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

<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
  http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="reporte2"
  pageWidth="595" pageHeight="842" columnWidth="555" leftMargin="20" rightMargin="20"
  topMargin="30" bottomMargin="30" uuid="b2e96685-ee00-4428-8733-eb054d6fb7d6">

  <property name="com.jasperassistant.designer.Grid" value="false"/>

  <property name="com.jasperassistant.designer.SnapToGrid" value="false"/>

  <property name="com.jasperassistant.designer.GridWidth" value="12"/>

  <property name="com.jasperassistant.designer.GridHeight" value="12"/>

  <property name="ireport.zoom" value="1.0"/>

  <property name="ireport.x" value="0"/>

  <property name="ireport.y" value="0"/>

  <parameter name="tipo" class="java.math.BigDecimal"/>

  <queryString>

    <![CDATA[select * from usuarios where tipo_usuario = $P{tipo}]]>

  </queryString>

  <field name="id_usuario" class="java.lang.Long"/>

  <field name="id_credencial" class="java.lang.Long"/>

  <field name="nombre" class="java.lang.String"/>

  <field name="apellidop" class="java.lang.String"/>

  <field name="apellidom" class="java.lang.String"/>

  <field name="rfc" class="java.lang.String"/>

  <field name="particular" class="java.lang.String"/>

  <field name="celular" class="java.lang.String"/>

  <field name="email" class="java.lang.String"/>

  <field name="tipo_usuario" class="java.lang.Integer"/>

  <field name="calle" class="java.lang.String"/>

  <field name="colonia" class="java.lang.String"/>
```

```

<field name="delegacion" class="java.lang.String"/>
<field name="cp" class="java.lang.Integer"/>
<field name="entidad" class="java.lang.Integer"/>
<field name="version" class="java.lang.Integer"/>
<field name="fecha_ingreso" class="java.lang.Long"/>
<field name="fecha_nacimiento" class="java.lang.Long"/>
<variable name="creditoTotal" class="java.math.BigDecimal" resetType="Group"
resetGroup="grupoPais" calculation="Sum">
    <variableExpression><![CDATA[]]></variableExpression>
</variable>
<group name="grupoPais" isReprintHeaderOnEachPage="true">
    <groupExpression><![CDATA[]]></groupExpression>
    <groupHeader>
        <band height="50" splitType="Stretch">
            <staticText>
                <reportElement mode="Opaque" x="0" y="30" width="30"
height="15" forecolor="#FFFFFF" backcolor="#0080C0" uuid="991d175b-0e98-41e9-86a3-
063c9b314683"/>
                <textElement textAlignment="Center">
                    <font isBold="true"/>
                </textElement>
            </staticText>
            <staticText>
                <reportElement mode="Opaque" x="30" y="30" width="200"
height="15" forecolor="#FFFFFF" backcolor="#0080C0" uuid="398658bb-8388-4a40-ab48-
072e43f6c411"/>
                <textElement textAlignment="Center">
                    <font isBold="true"/>
                </textElement>
                <text><![CDATA[Nombre del Cliente]]></text>

```

</staticText>

<staticText>

<reportElement mode="Opaque" x="230" y="30" width="100" height="15" forecolor="#FFFFFF" backcolor="#0080C0" uuid="50a7f980-1018-4a5b-a312-8a49b13bb354"/>

<textElement textAlignment="Center">

<font isBold="true"/>

</textElement>

<text><![CDATA[Localidad]]></text>

</staticText>

<staticText>

<reportElement mode="Opaque" x="330" y="30" width="100" height="15" forecolor="#FFFFFF" backcolor="#0080C0" uuid="9315aa03-1942-4952-b07f-7e2c3851560d"/>

<textElement textAlignment="Center">

<font isBold="true"/>

</textElement>

<text><![CDATA[Credito]]></text>

</staticText>

<staticText>

<reportElement x="10" y="10" width="100" height="20" backcolor="#800000" uuid="f24867f6-dd32-4d97-8f3e-dd734ae85131"/>

<textElement>

<font size="14" isBold="true"/>

</textElement>

<text><![CDATA[Ordenes:]]></text>

</staticText>

<line>

<reportElement x="0" y="48" width="430" height="1" backcolor="#000080" uuid="63f59f41-2d62-493d-bfba-045e28d0b86a"/>

</line>

```
</band>
</groupHeader>
<groupFooter>
  <band height="50" splitType="Stretch">
    <textField>
      <reportElement x="30" y="10" width="150" height="20"
        uuid="49bf48e5-1a60-49ec-b7d4-53cd9434f05d"/>
      <textElement>
        <font fontName="Times New Roman"
          isBold="true"/>
      </textElement>
      <textFieldExpression><![CDATA["Total Ordenes: " +
        $V{grupoPais_COUNT}]]></textFieldExpression>
    </textField>
  </band>
</groupFooter>
</group>
<title>
  <band height="50" splitType="Stretch">
    <staticText>
      <reportElement x="0" y="10" width="250" height="30"
        uuid="bc4b30ab-c67e-4620-b678-4c3a1a204dba"/>
      <textElement>
        <font fontName="Times New Roman" size="20"
          isBold="true" isItalic="true" pdfFontName="Helvetica-BoldOblique"/>
      </textElement>
      <text><![CDATA[Reporte de ordenes de compra]]></text>
    </staticText>
    <line>
      <reportElement x="2" y="45" width="550" height="1"
        forecolor="#008000" uuid="6c8a7c00-1a76-48e3-a7c4-7f2a0a9b117d"/>
    </line>
  </band>
</title>
```

```

        </line>

        <textField>

            <reportElement x="468" y="9" width="79" height="23"
            uuid="e5d00935-05cd-4b01-a373-ecab90eafc02"/>

            <textFieldExpression><![CDATA["Año "]]></textFieldExpression>

        </textField>

    </band>

</title>

<pageHeader>

    <band height="30" splitType="Stretch">

        <staticText>

            <reportElement x="0" y="5" width="200" height="20"
            forecolor="#800080" uuid="a100fb1b-e48a-4df1-a3f8-405fd4fb0d2c"/>

            <textElement>

                <font fontName="Lucida Sans" size="14" isBold="true"
                isUnderline="true" pdfFontName="Helvetica-BoldOblique"/>

            </textElement>

            <text><![CDATA[Informacion General]]></text>

        </staticText>

    </band>

</pageHeader>

<detail>

    <band height="15" splitType="Stretch">

        <textField>

            <reportElement x="30" y="0" width="162" height="15"
            uuid="668cc568-bf26-4edb-928d-c37ac98c0051"/>

            <textFieldExpression><![CDATA[{$F{id_usuario} + ", " +
            $F{nombre}.trim()}]]></textFieldExpression>

        </textField>

        <textField>

```

```
<reportElement x="230" y="0" width="100" height="15"
uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"/>
```

```
<textFieldExpression><![CDATA[{$F{rfc}.trim()}]></textFieldExpression>
```

```
</textField>
```

```
</band>
```

```
</detail>
```

```
<pageFooter>
```

```
<band height="40" splitType="Stretch"/>
```

```
</pageFooter>
```

```
<summary>
```

```
<band height="60" splitType="Stretch"/>
```

```
</summary>
```

```
</jasperReport>
```

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

```
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="reporte2"
pageWidth="595" pageHeight="842" columnWidth="555" leftMargin="20" rightMargin="20"
topMargin="30" bottomMargin="30" uuid="b2e96685-ee00-4428-8733-eb054d6fb7d6">
```

```
<property name="com.jasperassistant.designer.Grid" value="false"/>
```

```
<property name="com.jasperassistant.designer.SnapToGrid" value="false"/>
```

```
<property name="com.jasperassistant.designer.GridWidth" value="12"/>
```

```
<property name="com.jasperassistant.designer.GridHeight" value="12"/>
```

```
<property name="ireport.zoom" value="2.0"/>
```

```
<property name="ireport.x" value="531"/>
```

```
<property name="ireport.y" value="0"/>
```

```
<queryString>
```



```

                <![CDATA[select Servicios.id_servicio,
Usuarios.apellidop,
Usuarios.apellidom,
Usuarios.nombre,
Servicios.Status,
Servicios.fecha_peticion,
Servicios.descripcion

from
servicios Servicios
inner Join usuarios Usuarios on Servicios.id_cliente = Usuarios.id_usuario]]>

        </queryString>

        <field name="id_servicio" class="java.lang.Long"/>
        <field name="apellidop" class="java.lang.String"/>
        <field name="apellidom" class="java.lang.String"/>
        <field name="nombre" class="java.lang.String"/>
        <field name="status" class="java.lang.Integer"/>
        <field name="fecha_peticion" class="java.lang.Long"/>
        <field name="descripcion" class="java.lang.String"/>

        <variable name="creditoTotal" class="java.math.BigDecimal" resetType="Group"
        resetGroup="grupoPais" calculation="Sum">

                <variableExpression><![CDATA[]]></variableExpression>

        </variable>

        <group name="grupoPais" isReprintHeaderOnEachPage="true">

                <groupExpression><![CDATA[]]></groupExpression>

                <groupHeader>

                        <band height="49" splitType="Stretch">

```

```
<staticText>
    <reportElement uuid="991d175b-0e98-41e9-86a3-
063c9b314683" mode="Opaque" x="0" y="30" width="30" height="15" forecolor="#FFFFFF"
backcolor="#0080C0"/>
    <textElement textAlignment="Center">
        <font isBold="true"/>
    </textElement>
</staticText>
<staticText>
    <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="30" y="30" width="200" height="15"
forecolor="#FFFFFF" backcolor="#0080C0"/>
    <textElement textAlignment="Center">
        <font isBold="true"/>
    </textElement>
    <text><![CDATA[Nombre del Cliente]]></text>
</staticText>
<staticText>
    <reportElement uuid="50a7f980-1018-4a5b-a312-
8a49b13bb354" mode="Opaque" x="230" y="30" width="100" height="15"
forecolor="#FFFFFF" backcolor="#0080C0"/>
    <textElement textAlignment="Center">
        <font isBold="true"/>
    </textElement>
    <text><![CDATA[Fecha]]></text>
</staticText>
<staticText>
    <reportElement uuid="9315aa03-1942-4952-b07f-
7e2c3851560d" mode="Opaque" x="330" y="30" width="225" height="15"
forecolor="#FFFFFF" backcolor="#0080C0"/>
    <textElement textAlignment="Center">
```

```
<font isBold="true"/>
</textElement>
<text><![CDATA[Descripción]]></text>
</staticText>
<staticText>
    <reportElement uuid="f24867f6-dd32-4d97-8f3e-
dd734ae85131" x="10" y="10" width="152" height="20" bgcolor="#800000"/>
    <textElement>
        <font size="14" isBold="true"/>
    </textElement>
    <text><![CDATA[Ordenes servicio:]]></text>
</staticText>
<line>
    <reportElement uuid="63f59f41-2d62-493d-bfba-
045e28d0b86a" x="0" y="48" width="555" height="1" bgcolor="#000080"/>
</line>
</band>
</groupHeader>
<groupFooter>
    <band height="50" splitType="Stretch">
        <textField>
            <reportElement uuid="49bf48e5-1a60-49ec-b7d4-
53cd9434f05d" x="30" y="10" width="150" height="20"/>
            <textElement>
                <font fontName="Times New Roman"
isBold="true"/>
            </textElement>
            <textFieldExpression><![CDATA["Total Ordenes: " +
$V{grupoPais_COUNT}]]></textFieldExpression>
        </textField>
```

```
</band>
</groupFooter>
</group>
<title>
  <band height="50" splitType="Stretch">
    <staticText>
      <reportElement uuid="bc4b30ab-c67e-4620-b678-4c3a1a204dba"
x="0" y="10" width="250" height="30"/>
      <textElement>
        <font fontName="Times New Roman" size="20"
isBold="true" isItalic="true" pdfFontName="Helvetica-BoldOblique"/>
      </textElement>
      <text><![CDATA[Reporte de ordenes de servicio]]></text>
    </staticText>
    <line>
      <reportElement uuid="6c8a7c00-1a76-48e3-a7c4-7f2a0a9b117d"
x="2" y="45" width="550" height="1" forecolor="#008000"/>
    </line>
  </band>
</title>
<pageHeader>
  <band height="30" splitType="Stretch">
    <staticText>
      <reportElement uuid="a100fb1b-e48a-4df1-a3f8-405fd4fb0d2c"
x="0" y="5" width="200" height="20" forecolor="#800080"/>
      <textElement>
        <font fontName="Lucida Sans" size="14" isBold="true"
isUnderline="true" pdfFontName="Helvetica-BoldOblique"/>
      </textElement>
      <text><![CDATA[Informacion General]]></text>
```

```
        </staticText>
    </band>
</pageHeader>
<detail>
    <band height="16" splitType="Stretch">
        <textField>
            <reportElement uuid="668cc568-bf26-4edb-928d-c37ac98c0051"
x="10" y="0" width="240" height="15"/>
            <textElement/>
            <textFieldExpression><![CDATA[{$F{apellidop} + " " + $F{apellidom}
+ " " + $F{nombre}}]></textFieldExpression>
        </textField>
        <textField>
            <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="239" y="0" width="124" height="15"/>
            <textElement/>
            <textFieldExpression><![CDATA[{$F{fecha_peticion}}]></textFieldExpression>
        </textField>
        <textField>
            <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="407" y="0" width="124" height="15"/>
            <textElement/>
            <textFieldExpression><![CDATA[{$F{descripcion}}]></textFieldExpression>
        </textField>
    </band>
</detail>
<pageFooter>
    <band height="40" splitType="Stretch"/>
</pageFooter>
```

```

        <summary>
            <band height="60" splitType="Stretch"/>
        </summary>
</jasperReport>

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

<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
  http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="null" language="groovy"
  pageWidth="842" pageHeight="595" orientation="Landscape" columnWidth="802"
  leftMargin="20" rightMargin="20" topMargin="20" bottomMargin="20" uuid="10077997-
  be64-4bb0-9fd8-23eabb5aa7bb">

  <property name="ireport.zoom" value="1.0000000000000007"/>

  <property name="ireport.x" value="0"/>

  <property name="ireport.y" value="0"/>

  <parameter name="id_servicio" class="java.math.BigDecimal">
    <defaultValueExpression><![CDATA[]]></defaultValueExpression>
  </parameter>

  <queryString>
    <![CDATA[select usuario.nombre || ' ' || usuario.apellidoop || ' ' || usuario.apellidom as
    Proveedor,

(select estado from cat_estadoservicio where id_estado = tiempo.status) as Status,
tiempo.descripcion as descripcion,
case prioridad
when 0 then 'Urgente'
when 1 then 'Alto'
when 2 then 'Medio'
when 3 then 'Bajo'
end prioridad,
TO_CHAR(TO_TIMESTAMP(ultima_modificacion), 'DD/MM/YYYY') as fecha
]]>
  </queryString>

```

```
from tiempos tiempo inner join usuarios usuario on usuario.id_usuario = tiempo.id_proveedor
where tiempo.id_servicio = ${P{id_servicio}}]]>
```

```
</queryString>
```

```
<field name="proveedor" class="java.lang.String"/>
```

```
<field name="status" class="java.lang.String"/>
```

```
<field name="descripcion" class="java.lang.String"/>
```

```
<field name="prioridad" class="java.lang.String"/>
```

```
<field name="fecha" class="java.lang.String"/>
```

```
<group name="proveedor">
```

```
  <groupExpression><![CDATA[]]></groupExpression>
```

```
  <groupHeader>
```

```
    <band height="50">
```

```
      <line>
```

```
        <reportElement uuid="63f59f41-2d62-493d-bfba-045e28d0b86a" x="0" y="49" width="802" height="1" bgcolor="#000080"/>
```

```
      </line>
```

```
      <staticText>
```

```
        <reportElement uuid="398658bb-8388-4a40-ab48-072e43f6c411" mode="Opaque" x="0" y="15" width="301" height="34" forecolor="#FFFFFF" bgcolor="#0080C0"/>
```

```
          <textElement textAlignment="Center">
```

```
            <font isBold="true"/>
```

```
          </textElement>
```

```
          <text><![CDATA[Descripción]]></text>
```

```
        </staticText>
```

```
      <staticText>
```

```
        <reportElement uuid="398658bb-8388-4a40-ab48-072e43f6c411" mode="Opaque" x="301" y="15" width="185" height="34" forecolor="#FFFFFF" bgcolor="#0080C0"/>
```

```
          <textElement textAlignment="Center">
```

```
<font isBold="true"/>
</textElement>
<text><![CDATA[Proveedor]]></text>
</staticText>
<staticText>
  <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="486" y="15" width="165" height="34"
forecolor="#FFFFFF" bgcolor="#0080C0"/>
  <textElement textAlignment="Center">
    <font isBold="true"/>
  </textElement>
  <text><![CDATA[Estatus]]></text>
</staticText>
<staticText>
  <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="623" y="15" width="179" height="34"
forecolor="#FFFFFF" bgcolor="#0080C0"/>
  <textElement textAlignment="Center">
    <font isBold="true"/>
  </textElement>
  <text><![CDATA[Fecha]]></text>
</staticText>
</band>
</groupHeader>
<groupFooter>
  <band height="50"/>
</groupFooter>
</group>
<background>
  <band splitType="Stretch"/>
```



```
</background>

<title>

  <band height="41" splitType="Stretch">

    <staticText>

      <reportElement uuid="bc4b30ab-c67e-4620-b678-4c3a1a204dba"
x="0" y="0" width="274" height="41"/>

      <textElement>

        <font fontName="Times New Roman" size="20"
isBold="true" isItalic="true" pdfFontName="Helvetica-BoldOblique"/>

      </textElement>

      <text><![CDATA[Detalle servicio]]></text>

    </staticText>

  </band>

</title>

<detail>

  <band height="30" splitType="Stretch">

    <textField>

      <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="0" y="0" width="301" height="30"/>

      <textElement/>

    <textFieldExpression><![CDATA[{$F{descripcion}}]></textFieldExpression>

    </textField>

    <textField>

      <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="476" y="0" width="182" height="30"/>

      <textElement textAlignment="Center"/>

    <textFieldExpression><![CDATA[{$F{status}}]></textFieldExpression>

    </textField>

    <textField>
```

```
<reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="658" y="0" width="144" height="30"/>
```

```
<textElement textAlignment="Center"/>
```

```
<textFieldExpression><![CDATA[{$F{fecha}}]></textFieldExpression>
```

```
</textField>
```

```
<textField>
```

```
<reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="301" y="0" width="175" height="30"/>
```

```
<textElement>
```

```
<font size="10" isBold="false"/>
```

```
</textElement>
```

```
<textFieldExpression><![CDATA[{$F{proveedor}}]></textFieldExpression>
```

```
</textField>
```

```
</band>
```

```
</detail>
```

```
<columnFooter>
```

```
<band height="45" splitType="Stretch"/>
```

```
</columnFooter>
```

```
<pageFooter>
```

```
<band height="54" splitType="Stretch"/>
```

```
</pageFooter>
```

```
<summary>
```

```
<band height="42" splitType="Stretch"/>
```

```
</summary>
```

```
</jasperReport>
```

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

```

<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
  http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="null" language="groovy"
  pageWidth="842" pageHeight="595" orientation="Landscape" columnWidth="802"
  leftMargin="20" rightMargin="20" topMargin="20" bottomMargin="20" uuid="10077997-
  be64-4bb0-9fd8-23eabb5aa7bb">

  <property name="ireport.zoom" value="1.0000000000000007"/>

  <property name="ireport.x" value="0"/>

  <property name="ireport.y" value="0"/>

  <parameter name="id_cliente" class="java.math.BigDecimal">
    <defaultValueExpression><![CDATA[]]></defaultValueExpression>
  </parameter>

  <queryString>
    <![CDATA[SELECT usuario.nombre || ' ' || usuario.apellidoop || ' ' || usuario.apellidom
    as cliente,
    proveedor.nombre||' '||proveedor.apellidoop||' '||proveedor.apellidom AS proveedor,
    servicio.descripcion as descripcion,
    ( SELECT estado FROM cat_estadoservicio WHERE id_estado = servicio.status ) as status,
    case prioridad
      when 0 then 'Urgente'
      when 1 then 'Alto'
      when 2 then 'Medio'
      when 3 then 'Bajo'
    end prioridad,
    TO_CHAR(TO_TIMESTAMP(fecha_peticion),'DD/MM/YYYY') as fecha
  FROM
    usuarios usuario INNER JOIN servicios servicio ON usuario.id_usuario = servicio.id_cliente
    INNER JOIN usuarios proveedor ON servicio.id_proveedor = proveedor.id_usuario
  where usuario.id_usuario = ${id_cliente}]]>
  </queryString>

```

```

<field name="cliente" class="java.lang.String"/>
<field name="proveedor" class="java.lang.String"/>
<field name="descripcion" class="java.lang.String"/>
<field name="status" class="java.lang.String"/>
<field name="prioridad" class="java.lang.String"/>
<field name="fecha" class="java.lang.String"/>
<group name="proveedor">
    <groupExpression><![CDATA[]]></groupExpression>
    <groupHeader>
        <band height="50">
            <line>
                <reportElement uuid="63f59f41-2d62-493d-bfba-
045e28d0b86a" x="0" y="49" width="802" height="1" bgcolor="#000080"/>
            </line>
            <staticText>
                <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="0" y="15" width="301" height="34"
forecolor="#FFFFFF" bgcolor="#0080C0"/>
                <textElement textAlignment="Center">
                    <font isBold="true"/>
                </textElement>
                <text><![CDATA[Descripción]]></text>
            </staticText>
            <staticText>
                <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="301" y="15" width="185" height="34"
forecolor="#FFFFFF" bgcolor="#0080C0"/>
                <textElement textAlignment="Center">
                    <font isBold="true"/>
                </textElement>
            </staticText>
        </band>
    </groupHeader>

```

```
<text><![CDATA[Proveedor]]></text>
</staticText>
<staticText>
    <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="486" y="15" width="165" height="34"
forecolor="#FFFFFF" backcolor="#0080C0"/>
    <textElement textAlignment="Center">
        <font isBold="true"/>
    </textElement>
    <text><![CDATA[Estatus]]></text>
</staticText>
<staticText>
    <reportElement uuid="398658bb-8388-4a40-ab48-
072e43f6c411" mode="Opaque" x="623" y="15" width="179" height="34"
forecolor="#FFFFFF" backcolor="#0080C0"/>
    <textElement textAlignment="Center">
        <font isBold="true"/>
    </textElement>
    <text><![CDATA[Fecha]]></text>
</staticText>
</band>
</groupHeader>
<groupFooter>
    <band height="50"/>
</groupFooter>
</group>
<background>
    <band splitType="Stretch"/>
</background>
<title>
```

```
<band height="41" splitType="Stretch">
  <staticText>
    <reportElement uuid="bc4b30ab-c67e-4620-b678-4c3a1a204dba"
x="0" y="0" width="274" height="41"/>
    <textElement>
      <font fontName="Times New Roman" size="20"
isBold="true" isItalic="true" pdfFontName="Helvetica-BoldOblique"/>
    </textElement>
    <text><![CDATA[Reporte de ordenes de servicio: ]]></text>
  </staticText>
  <textField>
    <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="274" y="0" width="528" height="41"/>
    <textElement>
      <font fontName="Times New Roman" size="20"
isBold="true"/>
    </textElement>
    <textFieldExpression><![CDATA[{$F{cliente}}]></textFieldExpression>
  </textField>
</band>
</title>
<detail>
  <band height="30" splitType="Stretch">
    <textField>
      <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="0" y="0" width="301" height="30"/>
      <textElement/>
      <textFieldExpression><![CDATA[{$F{descripcion}}]></textFieldExpression>
    </textField>
```

```
<textField>
    <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="476" y="0" width="182" height="30"/>
    <textElement textAlignment="Center"/>
</textField>
<textFieldExpression><![CDATA[{$F{status}}]></textFieldExpression>
</textField>
<textField>
    <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="658" y="0" width="144" height="30"/>
    <textElement textAlignment="Center"/>
</textField>
<textFieldExpression><![CDATA[{$F{fecha}}]></textFieldExpression>
</textField>
<textField>
    <reportElement uuid="b5328463-2b08-46ba-adb8-cb613ad1bd1e"
x="301" y="0" width="175" height="30"/>
    <textElement>
        <font size="10" isBold="false"/>
    </textElement>
</textField>
<textFieldExpression><![CDATA[{$F{proveedor}}]></textFieldExpression>
</textField>
</band>
</detail>
<columnFooter>
    <band height="45" splitType="Stretch"/>
</columnFooter>
<pageFooter>
    <band height="54" splitType="Stretch"/>
</pageFooter>
```

```
<summary>
    <band height="42" splitType="Stretch"/>
</summary>
</jasperReport>

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

    <constant name="struts.devMode" value="true"/> <!-- True se encuentra en forma de desarrollo -
        ->

    <constant name="struts.objectFactory"
        value="org.apache.struts2.spring.StrutsSpringObjectFactory"/>

    <package name="crm" extends="struts-default" >

    <!-- Direccionamiento Paginas Principales -->

    <action name="GestionCRM_*" method="{1}"
        class="mx.uam.azc.cbi.action.GestionCRMAction">

        <result name="valida">/jsp/valida.jsp</result>

        <result name="indexAdmin">/jsp/indexAdmin.jsp</result>

        <result name="indexUser">/jsp/indexUser.jsp</result>

    <!-- Paginas de respuestas -->
```



```
</action>
```

```
<!-- Credenciales -->
```

```
<action name="Credencial_*" method="{1}" class="mx.uam.azc.cbi.action.CredencialAction">
```

```
  <result name="respuesta">/jsp/credencial/respuesta.jsp</result>
```

```
  <result name="cambiarpass">/jsp/credencial/actualizar.jsp</result>
```

```
</action>
```

```
<!-- Usuarios -->
```

```
<action name="Usuarios_*" method="{1}" class="mx.uam.azc.cbi.action.UsuariosAction">
```

```
  <result name="listaradmin">/jsp/usuarios/listaradmin.jsp</result>
```

```
  <result name="listaruser">/jsp/usuarios/listaruser.jsp</result>
```

```
  <result name="insertar">/jsp/usuarios/insertar.jsp</result>
```

```
  <result name="form_actualizar">/jsp/usuarios/actualizar.jsp</result>
```

```
  <result name="form_perfil">/jsp/usuarios/perfil.jsp</result>
```

```
  <result name="listarusuarios">/jsp/usuarios/listarusuarios.jsp</result>
```

```
  <result name="loaduser">/jsp/usuarios/loaduser.jsp</result>
```

```
  <result name="respuesta">/jsp/usuarios/respuesta.jsp</result>
```

```
</action>
```

```
<!-- Productos -->
```

```
<action name="Productos_*" method="{1}" class="mx.uam.azc.cbi.action.ProductosAction">
```

```
  <result name="admin">/jsp/productos/listar.jsp</result>
```

```
  <result name="user">/jsp/productos/catalogo.jsp</result>
```

```
  <result name="insertar">/jsp/productos/insertar.jsp</result>
```

```
  <result name="form_actualizar">/jsp/productos/actualizar.jsp</result>
```

```
<result name="respuesta">/jsp/productos/respuesta.jsp</result>

<result name="listarproductos">/jsp/productos/listarproductos.jsp</result>

<result name="loadproducto">/jsp/productos/loadproduct.jsp</result>

</action>

<action name="Ordenes_*" method="{1}" class="mx.uam.azc.cbi.action.OrdenesAction">

  <result name="listar">/jsp/ordenes/listar.jsp</result>

  <result name="listarOrdenes">/jsp/ordenes/listarordenes.jsp</result>

  <result name="listarCompra">/jsp/ordenes/listarCompra.jsp</result>

  <result name="respuesta">/jsp/ordenes/respuesta.jsp</result>

  <result name="detalles">/jsp/ordenes/detalles.jsp</result>

</action>

<!-- Documentos -->

<action name="Documentos_*" method="{1}" class="mx.uam.azc.cbi.action.DocumentosAction">

  <result name="excel" type="stream">

    <param name="contentType">application/vnd.ms-excel</param>

    <param name="inputName">inputStream</param>

    <param name="contentDisposition">attachment;filename="OrdenesCompra.xls"</param>

    <param name="bufferSize">1024</param>

  </result>

  <result name="excelOrden" type="stream">

    <param name="contentType">application/vnd.ms-excel</param>

    <param name="inputName">inputStream</param>

    <param name="contentDisposition">attachment;filename="OrdenesCompra.xls"</param>

    <param name="bufferSize">1024</param>
```

</result>

<result name="excelordendetalle" type="stream">

<param name="contentType">application/vnd.ms-excel</param>

<param name="inputName">inputStream</param>

<param name="contentDisposition">attachment;filename="OrdenesCompraDetalle.xls"</param>

<param name="bufferSize">1024</param>

</result>

<result name="excelservice" type="stream">

<param name="contentType">application/vnd.ms-excel</param>

<param name="inputName">inputStream</param>

<param name="contentDisposition">attachment;filename="Servicios.xls"</param>

<param name="bufferSize">1024</param>

</result>

<result name="excelestados" type="stream">

<param name="contentType">application/vnd.ms-excel</param>

<param name="inputName">inputStream</param>

<param name="contentDisposition">attachment;filename="DetallesServicios.xls"</param>

<param name="bufferSize">1024</param>

</result>

<result name="pdf" type="stream">

<param name="contentType">application/pdf</param>

<param name="inputName">inputStream</param>

<param name="contentDisposition">attachment;filename="OrdenesCompra.pdf"</param>

```
<param name="bufferSize">1024</param>
</result>

<result name="pdfOrden" type="stream">
  <param name="contentType">application/pdf</param>
  <param name="inputName">inputStream</param>
  <param name="contentDisposition">attachment;filename="OrdenesCompra.pdf"</param>
  <param name="bufferSize">1024</param>
</result>

<result name="pdfOrdendetalle" type="stream">
  <param name="contentType">application/pdf</param>
  <param name="inputName">inputStream</param>
  <param
    name="contentDisposition">attachment;filename="OrdenesCompraDetalle.pdf"</param>
  <param name="bufferSize">1024</param>
</result>

<result name="pdfservice" type="stream">
  <param name="contentType">application/pdf</param>
  <param name="inputName">inputStream</param>
  <param name="contentDisposition">attachment;filename="Servicios.pdf"</param>
  <param name="bufferSize">1024</param>
</result>

<result name="pdfestados" type="stream">
  <param name="contentType">application/pdf</param>
  <param name="inputName">inputStream</param>
```

```
<param name="contentDisposition">attachment;filename="DetallesServicios.pdf"</param>
```

```
<param name="bufferSize">1024</param>
```

```
</result>
```

```
</action>
```

```
<!-- Servicios -->
```

```
<action name="Servicios_*" method="{1}" class="mx.uam.azc.cbi.action.ServiciosAction">
```

```
<result name="listar">/jsp/servicios/listar.jsp</result>
```

```
<result name="listaruser">/jsp/servicios/listaruser.jsp</result>
```

```
<result name="insertar_admin">/jsp/servicios/insertar_admin.jsp</result>
```

```
<result name="insertar_user">/jsp/servicios/insertar_user.jsp</result>
```

```
<result name="insertar_proveedor">/jsp/servicios/insertar_proveedor.jsp</result>
```

```
<result name="form_actualizar">/jsp/servicios/actualizar.jsp</result>
```

```
<result name="form_actualizar_proveedor">/jsp/servicios/actualizar_proveedor.jsp</result>
```

```
<result name="respuesta">/jsp/servicios/respuesta.jsp</result>
```

```
<result name="listarservicios">/jsp/servicios/listarservicios.jsp</result>
```

```
<result name="loadservice">/jsp/servicios/loadservice.jsp</result>
```

```
</action>
```

```
<action name="Tiempos_*" method="{1}" class="mx.uam.azc.cbi.action.TiemposAction">
```

```
<result name="listar">/jsp/tempos/listar.jsp</result>
```

```
<result name="listardetalles">/jsp/tempos/listardetalles.jsp</result>
```

```
</action>
```

```
<!-- Temas -->
```

```
<action name="Temas_*" method="{1}" class="mx.uam.azc.cbi.action.TemasAction">
```

```
  <result name="listar">/jsp/temas/listar.jsp</result>
```

```
  <result name="insertar">/jsp/temas/insertar.jsp</result>
```

```
  <result name="form_actualizar">/jsp/temas/actualizar.jsp</result>
```

```
  <result name="listartemas">/jsp/temas/listartemas.jsp</result>
```

```
  <result name="loadtopic">/jsp/temas/loadtopic.jsp</result>
```

```
  <result name="respuesta">/jsp/temas/respuesta.jsp</result>
```

```
</action>
```

```
<action name="Mensajes_*" method="{1}" class="mx.uam.azc.cbi.action.MensajesAction">
```

```
  <result name="listar">/jsp/mensajes/listar.jsp</result>
```

```
  <result name="insertar">/jsp/mensajes/insertar.jsp</result>
```

```
  <result name="form_actualizar">/jsp/mensajes/actualizar.jsp</result>
```

```
  <result name="listarmensajes">/jsp/mensajes/listarmensajes.jsp</result>
```

```
  <result name="loadmensaje">/jsp/mensajes/loadmensaje.jsp</result>
```

```
  <result name="respuesta">/jsp/mensajes/respuesta.jsp</result>
```

```
</action>
```

```
<action name="Carrito_*" method="{1}" class="mx.uam.azc.cbi.action.CarritoAction">
```

```
  <result name="respuesta">/jsp/carrito/respuesta.jsp</result>
```

```
  <result name="listar">/jsp/carrito/listar.jsp</result>
```

```
  <result name="listarcarrito">/jsp/carrito/listarcarrito.jsp</result>
```

```
</action>
```

```
<action name="Combo_*" method="{1}" class="mx.uam.azc.cbi.action.CombosAction">
  <result name="estados">/jsp/combos/estados.jsp</result>
  <result name="tipousuario">/jsp/combos/tipousuario.jsp</result>
  <result name="estadoservicio">/jsp/combos/estadoservicio.jsp</result>
  <result name="categorias">/jsp/combos/categorias.jsp</result>
  <result name="fabricantes">/jsp/combos/fabricantes.jsp</result>
  <result name="proveedores">/jsp/combos/proveedores.jsp</result>
  <result name="clientes">/jsp/combos/clientes.jsp</result>
  <result name="servicios">/jsp/combos/servicios.jsp</result>
</action>
</package>
```

```
</struts>
```

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

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:tx="http://www.springframework.org/schema/tx"
  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-2.5.xsd
  http://www.springframework.org/schema/tx
  http://www.springframework.org/schema/tx/spring-tx-2.5.xsd
  http://www.springframework.org/schema/context
  http://www.springframework.org/schema/context/spring-context-2.5.xsd"
  default-autowire="byName" default-lazy-init="true">
```

```
<bean
  class="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor"/>
```

```
<!-- Acceso a las Bases de Datos -->
```

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-
  method="close">
```

```
<property name="driverClassName">
```

```
<value>org.postgresql.Driver</value>
```

```
</property>
```

```
<property name="url">
```

```
<value>jdbc:postgresql://localhost:5432/GestionCRM?autoReconnect=true</value>
```

```
</property>
```

```
<property name="maxWait" value="10000"/>
```

```
<property name="username" value="postgres"/>
```

```
<property name="password" value="g3r4rd0m4nz4n0!"/>
```

```
</bean>
```

```
<bean id="transactionManager"
  class="org.springframework.orm.hibernate3.HibernateTransactionManager"/>
```

```
<tx:annotation-driven></tx:annotation-driven>
```

```
<bean id="sessionFactory"
  class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
```

```
<property name="annotatedClasses">
```

```
<list>
```

```
<!-- ***** Catalogos ***** -->
```



<value>mx.uam.azc.cbi.data.CatEstadosDTO</value>  
<value>mx.uam.azc.cbi.data.CatTipoUsuarioDTO</value>  
<value>mx.uam.azc.cbi.data.CatEstadoServiciosDTO</value>  
<value>mx.uam.azc.cbi.data.CatCategoriasDTO</value>  
<value>mx.uam.azc.cbi.data.CatFabricanteDTO</value>  
<value>mx.uam.azc.cbi.data.CatServiciosDTO</value>

<!-- \*\*\*\*\* Credenciales \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.CredencialDTO</value>

<!-- \*\*\*\*\* Usuarios \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.UsuarioSaveDTO</value>

<value>mx.uam.azc.cbi.data.UsuarioJoinDTO</value>

<!-- \*\*\*\*\* Servicios \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.ServicioJoinDTO</value>

<value>mx.uam.azc.cbi.data.ServicioSaveDTO</value>

<!-- \*\*\*\*\* Tiempos \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.TiemposJoinDTO</value>

<value>mx.uam.azc.cbi.data.TiemposSaveDTO</value>

<!-- \*\*\*\*\* Temas \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.TemaSaveDTO</value>

<value>mx.uam.azc.cbi.data.TemaJoinDTO</value>

<!-- \*\*\*\*\* Mensajes \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.MensajeSaveDTO</value>

<value>mx.uam.azc.cbi.data.MensajeJoinDTO</value>

<!-- \*\*\*\*\* Productos \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.ProductoSaveDTO</value>

<value>mx.uam.azc.cbi.data.ProductoJoinDTO</value>

<!-- \*\*\*\*\* Ordenes de Compra \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.OrdenCompraSaveDTO</value>

<value>mx.uam.azc.cbi.data.OrdenCompraJoinDTO</value>

<!-- \*\*\*\*\* Compras \*\*\*\*\* -->

<value>mx.uam.azc.cbi.data.CompraSaveDTO</value>

<value>mx.uam.azc.cbi.data.CompraJoinDTO</value>

</list>

</property>

<property name="hibernateProperties">

<props>

<prop key="hibernate.dialect">

org.hibernate.dialect.PostgreSQLDialect

</prop>

</props>

</property>

</bean>

```
<bean id="administradorCredencial" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminCredencialesImpl"/>

<bean id="administradorUsuario" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminUsuariosImpl"/>

<bean id="administradorTema" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminTemasImpl"/>

<bean id="administradorMensaje" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminMensajesImpl"/>

<bean id="administradorCombos" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminCombosImpl"/>

<bean id="administradorTiempos" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminTiemposImpl"/>

<bean id="administradorServicios" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminServiciosImpl"/>

<bean id="administradorProductos" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminProductosImpl"/>

<bean id="administradorOrdenes" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminOrdenesImpl"/>

<bean id="administradorCompras" lazy-init="true"
      class="mx.uam.azc.cbi.metodos.AdminComprasImpl"/>

</beans>

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

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
         http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

<display-name>GestionCRM</display-name>

<!-- ***** LISTENERS ***** -->
```

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/applicationContext.xml</param-value>
</context-param>
```

```
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

```
<!--***** FILTERS *****-->
```

```
<filter>
  <filter-name>struts2</filter-name>
  <filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
</filter>
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
<!--***** ADDITIONAL CONFIGURATION *****-->
```

```
<welcome-file-list>
  <welcome-file>login.jsp</welcome-file>
</welcome-file-list>
```

```
<!--<context-param>
```

```
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/applicationContext.xml</param-value>
</context-param>-->
</web-app>
```

```
Ext.onReady(function(){
if(Ext.getCmp('user-form')!=null){
    Ext.getCmp('user-form').destroy();
}

Ext.QuickTips.init();
Ext.form.Field.prototype.msgTarget = 'side';
```

```
var storeGrid = new Ext.data.JsonStore({
    url:'Tiempos_listardetalles.action',
    totalProperty: 'totalCount',
    root:'data',
    fields:['proveedor','estado','descripcion', 'modificado', 'ultima', 'prioridad']
});
```

```
storeGrid.load({params:{start:0, limit:3}});
```

```
var pager = new Ext.PagingToolbar({
    store: storeGrid,
    displayInfo:true,
    displayMsg:'{0} - {1} de {2} Peticiones',
    empty:'No hay Peticiones',
```

```

        pageSize: 5,
        plugins: new Ext.ux.ProgressBarPager()
    });

var xg = Ext.grid;

var expander = new Ext.ux.grid.RowExpander({
    tpl : new Ext.Template(
        '<p><b>Descripción:</b> {descripcion}</p>'
    )
});

var servicios = new xg.GridPanel ({
    store: storeGrid,
    bbar: pager,
    cm: new xg.ColumnModel({
        defaults: {
            width: 60,
            sortable: true
        },
        columns: [
            expander,{
                header: "Proveedor",
                width: 300,
                sortable: true,
                locked:false,
                dataIndex: 'proveedor'
            },{

```

```
    header: "Estatus",
    width: 100,
    sortable: true,
    dataIndex: 'estado'
  },{
    header: "Prioridad",
    width: 100,
    sortable: true,
    dataIndex: 'prioridad'
  },{
    header: "Modificado por",
    width: 300,
    sortable: true,
    locked:false,
    dataIndex: 'modificado'
  },{
    header: "Fecha modificación",
    width: 120,
    sortable: true,
    dataIndex: 'ultima'
  }
]
}),
viewConfig: {
  forceFit:true,
  stripeRows: true
},
width: 1200,
```

height: 500,

tbar:{{

text:'Regresar',

tooltip:'Regresar',

iconCls:'back',

listeners: {

'click':function(){

Ext.get(tablas).load({

url:'Servicios\_listar.action',

scripts:true

});

}

}

},{

text:'Exportar Excel',

tooltip:'Exportar Excel',

iconCls:'excel',

listeners:{

'click':function(){

parent.location='/GestionCRM/Documentos\_excelestados.action';

}

}

},{

text:'Exportar PDF',

tooltip:'Exportar PDF',

iconCls:'pdf',

listeners:{



```
        'click': function(){
            parent.location='/GestionCRM/Documentos_pdfestados.action';
        }
    }
}],
plugins: expander,
title: '::: Detalles Servicio :::',
renderTo: 'tablas'

});
});
```

```
Ext.onReady(function(){

    Ext.QuickTips.init();

    var formulario = new Ext.FormPanel({
        labelAlign: 'top',
        renderTo: 'tablas',
        title: 'Agregar Tema',
        bodyStyle:'padding:5px 5px 0',
        width: 550,
        items:[{
            xtype: 'fieldset',
            title:'Detalle Tema',
            items:[{
                layout: 'column',
                border: false,
```

```
items:{{
  columnWidth:.5,
  layout: 'form',
  border:false,
  items: {{
    xtype: 'textfield',
    fieldLabel: 'Nombre',
    name: 'nombre',
    maxLength: 30,
    anchor:'95%',
    allowBlank:false
  }}
},{columnWidth:.5,
  layout: 'form',
  border:false,
  items:[combocategoria]
}}
},{
  xtype: 'textarea',
  fieldLabel:'Descripción',
  name: 'descripcion',
  height:200,
  anchor:'95%',
  allowBlank:false
}}
}}
,
buttons: {{
```

```
text: 'Aceptar',
handler: function(){
    if ( formulario.form.isValid() ) {
        formulario.form.submit({
            url:'Temas_agregar.action',
            waitMsg:'Guardando...',
            reset: false,
            failure: function(formulario, action) {
                Ext.MessageBox.alert('Acceso denegado', action.result.result);
            },
            success: function(formulario, action) {
                Ext.MessageBox.show({
                    title: 'Tema Agregado',
                    msg: action.result.result,
                    closable: false,
                    resizable: false,
                    buttons: Ext.MessageBox.OK
                });
                Ext.get(tablas).load({
                    url:'Temas_listar.action',
                    scripts:true
                });
            }
        });
    } else {
        Ext.Msg.show({
            title:'Mensaje',
            msg: 'No debe haber Campos vacios',
```

```
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text:'Cancelar/Salir',
    width:'80',
    listeners:{
        'click': function(){
            Ext.get(tablas).load({
                url:'Temas_listar.action',
                scripts:true
            });
        }
    }
}
});
});
```

```
Ext.onReady(function(){
    Ext.QuickTips.init();
```

```
var formulario = new Ext.FormPanel({
    labelAlign: 'top',
    renderTo: 'tablas',
    title: 'Actualizar Tema',
    bodyStyle:'padding:5px 5px 0',
    width: 550,
    items:[{
        xtype: 'fieldset',
        title:'Detalle Tema',
        items:[{
            layout: 'column',
            border: false,
            items:[{
                columnWidth:.5,
                layout: 'form',
                border:false,
                items: [{
                    xtype: 'textfield',
                    fieldLabel: 'Nombre',
                    name: 'nombre',
                    //ref: idTema,
                    maxLength: 30,
                    anchor:'95%',
                    allowBlank:false
                }
            ]
        }],{columnWidth:.5,
```

```
        layout: 'form',
        border:false,
        items:[combocategoria]
    }}
    },{
    xtype: 'textarea',
    fieldLabel:'Descripción',
    name: 'descripcion',
    height:200,
    anchor:'95%',
    allowBlank:false,
    autoLoad: true
    }}
    }}
    ,
    buttons: [{
        text: 'Aceptar',
        handler: function(){
            if ( formulario.form.isValid() ) {
                formulario.form.submit({
                    url:'Temas_actualizar.action',
                    waitMsg:'Guardando...',
                    reset: false,
                    failure: function(formulario, action) {
                        Ext.MessageBox.alert('Acceso denegado', action.result.result);
                    },
                    success: function(formulario, action) {
                        Ext.MessageBox.show({
```

```
        title: 'Tema Agregado',
        msg: action.result.result,
        closable: false,
        resizable: false,
        buttons: Ext.MessageBox.OK
    });

    Ext.get(tablas).load({
        url: 'Temas_listar.action',
        scripts: true
    });
    }
    });
} else {
    Ext.Msg.show({
        title: 'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text: 'Cancelar/Salir',
    width: '80',
    listeners: {
        'click': function() {
            Ext.get(tablas).load({
```

```
        url:'Temas_listar.action',
        scripts:true
    });
    }
}
}}

});
```

formulario.getForm().load({// Aqui cargo los datos del form

```
    url:'Temas_cargar.action',
    success: function() {
        Ext.MessageBox.alert('Success', action.result.result);
    },
```

```
    failure: function() {
        Ext.Msg.show({
            title : 'ERROR',
            msg : 'Carga de datos fallida!',
            icon : Ext.MessageBox.ERROR,
            buttons: Ext.Msg.OK
        });
    }
}
```

```
});
```

```
});
```



```
Ext.onReady(function(){
    if(Ext.getCmp('user-form')!=null){
        Ext.getCmp('user-form').destroy();
    }

    Ext.QuickTips.init();
    Ext.form.Field.prototype.msgTarget = 'side';

    var storeGrid = new Ext.data.JsonStore({
        url:'Temas_listartemas.action',
        totalProperty: 'totalCount',
        root:'data',
        fields:['idTema', 'nombre','categoria','descripcion', 'fecha', 'usuario']
    });

    storeGrid.load({params:{start:0,limit:3}});

    var pager = new Ext.PagingToolbar({
        store: storeGrid,
        displayInfo:true,
        displayMsg:'{0} - {1} de {2} Peticiones',
        empty:'No hay Peticiones',
        pageSize: 5,
        plugins: new Ext.ux.ProgressBarPager()
    });
```

```
var xg = Ext.grid;
```

```
var expander = new Ext.ux.grid.RowExpander({  
    tpl : new Ext.Template(  
        '<p><b>Descripción:</b> {descripcion}</p>'  
    )  
});
```

```
var sm2 = new xg.CheckboxSelectionModel({  
    hidden: true,  
    listeners: {  
        selectionchange: function(sm) {  
            if (sm.getCount() == 1) {  
                temas.comentarios.enable();  
                temas.update.enable();  
            } else {  
                temas.comentarios.disable();  
                temas.update.disable();  
            }  
            if (sm.getCount()>=1) {  
                temas.elimina.enable();  
            } else {  
                temas.elimina.disable();  
            }  
        }  
    }  
});
```

```
var temas = new xg.GridPanel({
  id: 'topic-form',
  store: storeGrid,
  bbar: pager,
  cm: new xg.ColumnModel({
    defaults: {
      width: 70,
      sortable: true
    },
    columns: [
      expander,
      {
        id: 'idTema',
        header: "Tema",
        width: 450,
        sortable: true,
        locked: false,
        dataIndex: 'nombre'
      }, {
        header: "Categoría",
        width: 250,
        sortable: true,
        dataIndex: 'categoria'
      }, {
        header: "Fecha",
        width: 200,
        sortable: true,
        dataIndex: 'fecha'
      }
    ]
  })
});
```

```
},{
  header: "Usuario",
  width: 300,
  sortable: true,
  dataIndex: 'usuario'
},
sm2
]
}),
viewConfig: {
  stripeRows: true
},
sm: sm2,
width: 1200,
height: 500,
tbar: [{
  text: 'Agregar',
  tooltip: 'Agregar',
  iconCls: 'topicadd',
  listeners: {
    'click': function () {
      Ext.get(tablas).load({
        url: 'Temas_insertar.action',
        scripts: true
      });
    }
  }
}],
}, {
```

```
text:'Actualizar',
tooltip:'Actualizar',
iconCls:'actualizar',
ref: '../update',
disabled: true,
listeners:{
  'click':function(){
    var chk= temas.getSelectionModel().getSelections();
    if(chk.length==1){
      Ext.Ajax.request({
        url: 'Temas_load.action',
        params: {
          idTema: chk[0].get('idTema')
        },
        success: function() {
          Ext.get(tablas).load({
            url:'Temas_form_actualizar.action',
            scripts:true
          });
        },
        failure: function(action) {
          Ext.MessageBox.alert('Acceso denegado', action.result.result);
        }
      });
    }
    else{
      Ext.Msg.alert('Alerta','Elige al menos un tema');
    }
  }
}
```

```

    }
  }
},{
  text:'Ver Comentarios',
  tooltip:'Ver Comentarios',
  iconCls:'comentarios',
  ref: '../comentarios',
  disabled: true,
  listeners:{
    'click':function(){
      var chk= temas.getSelectionModel().getSelections();
      if(chk.length==1){
        Ext.Ajax.request({
          url: 'Temas_load.action',
          params: {
            idTema: chk[0].get('idTema')
          },
          success: function() {
            Ext.get(tablas).load({
              url:'Mensajes_listar.action',
              scripts:true
            });
          },
          failure: function(action) {
            Ext.MessageBox.alert('Acceso denegado', action.result.result);
          }
        });
      }
    }
  }
}

```



```

    },
    success: function() {
        Ext.MessageBox.alert('Tema eliminado');

        Ext.get(tablas).load({
            url:'Temas_listar.action',
            scripts:true
        });
    },
    failure: function(action) {
        Ext.MessageBox.alert('Acceso denegado', action.result.result);
    }
});
}
}
}
}
}
},
plugins: expander,
title: '::: Temas :::',
iconCls: 'bitacora',
renderTo: 'tablas'
});
});

Ext.onReady(function(){
    if(Ext.getCmp('user-form')!=null){
        Ext.getCmp('user-form').destroy();
    }
}

```



```
}
```

```
Ext.QuickTips.init();
```

```
Ext.form.Field.prototype.msgTarget = 'side';
```

```
var storeGrid = new Ext.data.JsonStore({  
    url:'Servicios_listarservicios.action',  
    totalProperty: 'totalCount',  
    root:'data',  
    fields:['idServicio', 'cliente', 'proveedor', 'estado', 'prioridad', 'descripcion', 'servicio', 'Fecha' ,  
    'ultima', 'modificado']  
});
```

```
storeGrid.load({  
    params:{  
        start:0,  
        limit:3  
    }  
});
```

```
var pager = new Ext.PagingToolbar({  
    store: storeGrid,  
    displayInfo:true,  
    displayMsg:'{0} - {1} de {2} Peticiones',  
    empty:'No hay Peticiones',  
    pageSize: 5,  
    plugins: new Ext.ux.ProgressBarPager()  
});
```

```
var xg = Ext.grid;
```

```
var expander = new Ext.ux.grid.RowExpander({  
    tpl : new Ext.Template(  
        '<p><b>Descripción:</b> {descripcion}</p>'  
    )  
});
```

```
var sm2 = new xg.CheckboxSelectionModel({  
    hidden: true,  
    listeners: {  
        selectionchange: function(sm) {  
            if (sm.getCount()==1) {  
                servicios.detalle.enable();  
                servicios.update.enable();  
            } else {  
                servicios.detalle.disable();  
                servicios.update.disable();  
            }  
        }  
    }  
});
```

```
var servicios = new xg.GridPanel ({  
    store: storeGrid,  
    bbar: pager,  
    cm: new xg.ColumnModel({
```

```
defaults: {
  width: 60,
  sortable: true
},
columns: [
  expander,{
    id:'idServicio',
    header: "Cliente",
    width: 200,
    sortable: true,
    locked:false,
    dataIndex: 'cliente'
  },{
    header: "Proveedor",
    width: 200,
    sortable: true,
    locked:false,
    dataIndex: 'proveedor'
  },{
    header: "Estatus",
    width: 100,
    sortable: true,
    dataIndex: 'estado'
  },{
    header: "Prioridad",
    width: 100,
    sortable: true,
    dataIndex: 'prioridad'
```

```
},{
  header: "Fecha",
  width: 100,
  sortable: true,
  dataIndex: 'Fecha'
},{
  header: "Servicio",
  width: 100,
  sortable: true,
  dataIndex: 'servicio'
},{header: "Modificado por",
  width: 200,
  sortable: true,
  locked:false,
  dataIndex: 'modificado'
},{
  header: "Ultima Modificación",
  width: 100,
  sortable: true,
  dataIndex: 'ultima'
},
sm2
]
}),
viewConfig: {
  forceFit:true,
  stripeRows: true
},
```

```
sm: sm2,
width: 1200,
height: 500,
tbar:{{
    text: 'Agregar',
    tooltip:'Agregar',
    iconCls:'serviceadd',
    listeners: {
        'click':function(){
            Ext.get(tablas).load({
                url:'Servicios_insertar.action',
                scripts:true
            });
        }
    },{
text:'Ver Detalle',
tooltip:'Ver detalle',
iconCls:'detalle',
ref: '../detalle',
disabled: true,
listeners: {
    'click':function(){
        var chk= servicios.getSelectionModel().getSelections();
        if(chk.length==1){
            Ext.Ajax.request({
                url: 'Servicios_load.action',
                params: {
```

```
        idServicio: chk[0].get('idServicio')
    },
    success: function() {
        Ext.get(tablas).load({
            url:'Tiempos_listar.action',
            scripts:true
        });
    },
    failure: function(action) {
        Ext.MessageBox.alert('Acceso denegado', action.result.result);
    }
});

    }
}
}

},{
    text:'Exportar Excel',
    tooltip:'Exportar Excel',
    iconCls:'excel',
    listeners:{
        'click':function(){
            parent.location='/GestionCRM/Documentos_excelservice.action';
        }
    }
}
},{
    text:'Exportar PDF',
```

```
        tooltip:'Exportar PDF',
        iconCls:'pdf',
        listeners:{
            'click': function(){
                parent.location='/GestionCRM/Documentos_pdfservice.action';
            }
        }
    }],
    plugins: expander,
    title: ':: Listado Servicios ::',
    iconCls: 'bitacora',
    renderTo: 'tablas'

});

});
```

```
Ext.onReady(function(){

    Ext.QuickTips.init();

    var formulario = new Ext.FormPanel({
        labelAlign: 'top',
        renderTo: 'tablas',
        title: 'Nuevo Servicio',
        bodyStyle:'padding:5px 5px 0',
        width: 1200,
        items:[{
            xtype: 'fieldset',
```

```
title:'Detalle Usuario',
items:{{
  layout: 'column',
  border: false,
  items:{{
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[comboservicios]
  },{columnWidth:.33,
    layout: 'form',
    border:false,
    items:[comboprioridad]
  }}
},{
  xtype: 'textarea',
  fieldLabel: 'Descripción',
  maxLength: 100,
  name: 'descripcion',
  anchor:'95%',
  allowBlank:false
}}
}],
buttons: [{
  text: 'Aceptar',
  handler: function(){
    if ( formulario.form.isValid() ) {
      formulario.form.submit({
```



```
url:'Servicios_agregar.action',
waitMsg:'Guardando...',
reset: false,
failure: function(formulario, action) {
    Ext.MessageBox.alert('Acceso denegado', action.result.result);
},
success: function(formulario, action) {
    Ext.MessageBox.show({
        title: 'Servicio Agregado',
        msg: action.result.result,
        closable: false,
        resizable: false,
        buttons: Ext.MessageBox.OK
    });
    Ext.get(tablas).load({
        url:'Servicios_listar.action',
        scripts:true
    });
}
});
} else {
    Ext.Msg.show({
        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
```

```
    }  
  }  
},{  
  text:'Cancelar/Salir',  
  width:'80',  
  listeners:{  
    'click': function(){  
      Ext.get(tablas).load({  
        url:'Servicios_listar.action',  
        scripts:true  
      });  
    }  
  }  
}  
});  
});
```

```
Ext.onReady(function(){  
  
  Ext.QuickTips.init();  
  
  var formulario = new Ext.FormPanel({  
    labelAlign: 'top',  
    renderTo: 'tablas',  
    title: 'Nuevo Servicio',  
    bodyStyle:'padding:5px 5px 0',  
    width: 1200,  
    items:{{
```

```
xtype: 'fieldset',
title:'Detalle Usuario',
items:[{
  layout: 'column',
  border: false,
  items:[{
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[comboclientes]
  },{columnWidth:.33,
    layout: 'form',
    border:false,
    items:[comboservicios]
  },{columnWidth:.33,
    layout: 'form',
    border:false,
    items:[comboprioridad]
  ]
},{
  xtype: 'textarea',
  fieldLabel: 'Descripción',
  maxLength: 100,
  name: 'descripcion',
  anchor:'95%',
  allowBlank:false
}]
}],
```

```
buttons: [{
  text: 'Aceptar',
  handler: function(){
    if ( formulario.form.isValid() ) {
      formulario.form.submit({
        url:'Servicios_agregar.action',
        waitMsg:'Guardando...',
        reset: false,
        failure: function(formulario, action) {
          Ext.MessageBox.alert('Acceso denegado', action.result.result);
        },
        success: function(formulario, action) {
          Ext.MessageBox.show({
            title: 'Servicio Agregado',
            msg: action.result.result,
            closable: false,
            resizable: false,
            buttons: Ext.MessageBox.OK
          });
          Ext.get(tablas).load({
            url:'Servicios_listar.action',
            scripts:true
          });
        }
      });
    } else {
      Ext.Msg.show({
        title:'Mensaje',
```

```
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text:'Cancelar/Salir',
    width:'80',
    listeners:{
        'click': function(){
            Ext.get(tablas).load({
                url:'Servicios_listar.action',
                scripts:true
            });
        }
    }
}
});
});
```

```
Ext.onReady(function(){
```

```
    Ext.QuickTips.init();
```

```
    var formulario = new Ext.FormPanel({
```

```
        labelAlign: 'top',
```

```
renderTo: 'tablas',
title: 'Nuevo Servicio',
bodyStyle: 'padding: 5px 5px 0',
width: 1000,
items: [{
    xtype: 'fieldset',
    title: 'Detalle Usuario',
    items: [{
        layout: 'column',
        border: false,
        items: [{
            columnWidth: .5,
            layout: 'form',
            border: false,
            items: [comboclientes, comboproveedores]
        }, {columnWidth: .5,
            layout: 'form',
            border: false,
            items: [comboservicios, comboprioridad ]
        }
    ]
}, {
    xtype: 'textarea',
    fieldLabel: 'Descripción',
    maxLength: 100,
    name: 'descripcion',
    anchor: '95%',
    allowBlank: false
}]
```

```
    }},
    buttons: [{
        text: 'Aceptar',
        handler: function(){
            if ( formulario.form.isValid() ) {
                formulario.form.submit({
                    url:'Servicios_agregar.action',
                    waitMsg:'Guardando...',
                    reset: false,
                    failure: function(formulario, action) {
                        Ext.MessageBox.alert('Acceso denegado', action.result.result);
                    },
                    success: function(formulario, action) {
                        Ext.MessageBox.show({
                            title: 'Servicio Agregado',
                            msg: action.result.result,
                            closable: false,
                            resizable: false,
                            buttons: Ext.MessageBox.OK
                        });
                        Ext.get(tablas).load({
                            url:'Servicios_listar.action',
                            scripts:true
                        });
                    }
                });
            } else {
                Ext.Msg.show({
```

```

        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text:'Cancelar/Salir',
    width:'80',
    listeners:{
        'click': function(){
            Ext.get(tablas).load({
                url:'Servicios_listar.action',
                scripts:true
            });
        }
    }
}
});
});

```

```

Ext.onReady(function(){

```

```

    Ext.QuickTips.init();

```

```

    var formulario = new Ext.FormPanel({

```



```
labelAlign: 'top',
renderTo: 'tablas',
title: 'Actualizar Servicio',
bodyStyle:'padding:5px 5px 0',
width: 1200,
items:[{
  xtype: 'fieldset',
  title:'Detalle Usuario',
  items:[{
    layout: 'column',
    border: false,
    items:[{
      columnWidth:.33,
      layout: 'form',
      border:false,
      items:[{
        xtype: 'textfield',
        fieldLabel: 'Cliente',
        emptyText: 'Cliente',
        name: 'cliente',
        anchor:'95%',
        allowBlank:true,
        disabled:true
      },comboprioridad]
    },{columnWidth:.33,
      layout: 'form',
      border:false,
      items:[{
```

```
        xtype: 'textfield',
        fieldLabel: 'Proveedor',
        emptyText: 'Proveedor',
        name: 'proveedor',
        anchor:'95%',
        allowBlank:true,
        disabled:true
    },
    comboservicios]
}, {columnWidth:.33,
    layout: 'form',
    border:false,
    items:[comboestadoservicio]
}]
}, {
    xtype: 'textarea',
    fieldLabel: 'Descripción',
    maxLength: 50,
    name: 'descripcion',
    anchor:'95%',
    allowBlank:false
}]
}],
buttons: [{
    text: 'Aceptar',
    handler: function(){
        if ( formulario.form.isValid() ) {
            formulario.form.submit({
```

```
url:'Servicios_actualizar.action',
waitMsg:'Guardando...',
reset: false,
failure: function(formulario, action) {
    Ext.MessageBox.alert('Acceso denegado', action.result.result);
},
success: function(formulario, action) {
    Ext.MessageBox.show({
        title: 'Servicio Actualizado',
        msg: action.result.result,
        closable: false,
        resizable: false,
        buttons: Ext.MessageBox.OK
    });
    Ext.get(tablas).load({
        url:'Servicios_listar.action',
        scripts:true
    });
}
});
} else {
    Ext.Msg.show({
        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
```

```
    }  
  }  
},{  
  text:'Cancelar/Salir',  
  width:'80',  
  listeners:{  
    'click': function(){  
      Ext.get(tablas).load({  
        url:'Servicios_listar.action',  
        scripts:true  
      });  
    }  
  }  
}]  
});
```

```
formulario.getForm().load({// Aqui cargo los datos del form  
  url:'Servicios_cargar.action',  
  success: function() {  
    Ext.MessageBox.alert('Success', action.result.result);  
  },
```

```
  failure: function() {  
    Ext.Msg.show({  
      title : 'ERROR',  
      msg : 'Carga de datos fallida!',  
      icon : Ext.MessageBox.ERROR,
```

```
        buttons: Ext.Msg.OK
    });
}

});

});

Ext.onReady(function(){
    if(Ext.getCmp('user-form')!=null){
        Ext.getCmp('user-form').destroy();
    }

    Ext.QuickTips.init();

    Ext.form.Field.prototype.msgTarget = 'side';

    var storeGrid = new Ext.data.JsonStore({
        url:'Servicios_listarservicios.action',
        totalProperty: 'totalCount',
        root:'data',
        fields:['idServicio', 'cliente', 'proveedor', 'estado', 'prioridad', 'descripcion', 'servicio', 'Fecha' ,
        'ultima', 'modificado']
    });

    storeGrid.load({
        params:{
            start:0,
            limit:3
```

```

    }
});

var pager = new Ext.PagingToolbar({
    store: storeGrid,
    displayInfo:true,
    displayMsg:'{0} - {1} de {2} Peticiones',
    empty:'No hay Peticiones',
    pageSize: 5,
    plugins: new Ext.ux.ProgressBarPager()
});

var xg = Ext.grid;

var expander = new Ext.ux.grid.RowExpander({
    tpl : new Ext.Template(
        '<p><b>Descripción:</b> {descripcion}</p>'
    )
});

var sm2 = new xg.CheckboxSelectionModel({
    hidden: true,
    listeners: {
        selectionchange: function(sm) {
            if (sm.getCount()>=1) {
                servicios.detalle.enable();
                servicios.update.enable();
            } else {

```

```
        servicios.detalle.disable();  
        servicios.update.disable();  
    }  
}  
}  
});
```

```
var servicios = new xg.GridPanel ({  
    store: storeGrid,  
    bbar: pager,  
    cm: new xg.ColumnModel({  
        defaults: {  
            width: 60,  
            sortable: true  
        },  
        columns: [  
            expander,{  
                id:'idServicio',  
                header: "Cliente",  
                width: 200,  
                sortable: true,  
                locked:false,  
                dataIndex: 'cliente'  
            },{  
                header: "Proveedor",  
                width: 200,  
                sortable: true,  
                locked:false,
```

```
    dataIndex: 'proveedor'
  },{
    header: "Estatus",
    width: 100,
    sortable: true,
    dataIndex: 'estado'
  },{
    header: "Prioridad",
    width: 100,
    sortable: true,
    dataIndex: 'prioridad'
  },{
    header: "Fecha",
    width: 100,
    sortable: true,
    dataIndex: 'Fecha'
  },{
    header: "Servicio",
    width: 100,
    sortable: true,
    dataIndex: 'servicio'
  },{header: "Modificado por",
    width: 200,
    sortable: true,
    locked:false,
    dataIndex: 'modificado'
  },{
    header: "Ultima Modificación",
```



```
        width: 100,
        sortable: true,
        dataIndex: 'ultima'
    },
    sm2
]
}),
viewConfig: {
    forceFit:true,
    stripeRows: true
},
sm: sm2,
width: 1200,
height: 500,
tbar:[{
    text: 'Agregar',
    tooltip:'Agregar',
    iconCls:'serviceadd',
    listeners: {
        'click':function(){
            Ext.get(tablas).load({
                url:'Servicios_insertar.action',
                scripts:true
            });
        }
    }
},
{
    text:'Actualizar',
```

```
tooltip:'Actualizar',
iconCls:'update',
ref: '../update',
disabled: true,
listeners: {
    'click':function(){
        var chk= servicios.getSelectionModel().getSelections();
        if(chk.length==1){
            Ext.Ajax.request({
                url: 'Servicios_load.action',
                params: {
                    idServicio: chk[0].get('idServicio')
                },
                success: function() {
                    Ext.get(tablas).load({
                        url:'Servicios_form_actualizar.action',
                        scripts:true
                    });
                },
                failure: function(action) {
                    Ext.MessageBox.alert('Acceso denegado', action.result.result);
                }
            });
        }
    }
},
text:'Ver Detalle',
```

```
tooltip:'Ver detalle',
iconCls:'detalle',
ref: '../detalle',
disabled: true,
listeners: {
  'click':function(){
    var chk= servicios.getSelectionModel().getSelections();
    if(chk.length==1){
      Ext.Ajax.request({
        url: 'Servicios_load.action',
        params: {
          idServicio: chk[0].get('idServicio')
        },
        success: function() {
          Ext.get(tablas).load({
            url:'Tiempos_listar.action',
            scripts:true
          });
        },
        failure: function(action) {
          Ext.MessageBox.alert('Acceso denegado', action.result.result);
        }
      });
    }
  }
}
```

```
},{
  text:'Exportar Excel',
  tooltip:'Exportar Excel',
  iconCls:'excel',
  listeners:{
    'click':function(){
      parent.location='/GestionCRM/Documentos_excelservice.action';
    }
  }
},{
  text:'Exportar PDF',
  tooltip:'Exportar PDF',
  iconCls:'pdf',
  listeners:{
    'click': function(){
      parent.location='/GestionCRM/Documentos_pdfservice.action';
    }
  }
}],
plugins: expander,
title: '::: Listado Servicios :::',
iconCls: 'bitacora',
renderTo: 'tablas'

});

});

Ext.onReady(function(){
```

```
Ext.QuickTips.init();
```

```
var formulario = new Ext.FormPanel({  
    renderTo: 'tablas',  
    labelAlign: 'top',  
    id:'form-user',  
    title: 'Agregar producto',  
    bodyStyle:'padding:5px 5px 0',  
    width: 950,  
    items:[{  
        xtype: 'fieldset',  
        title: 'Detalle Producto',  
        items:[{  
            layout: 'column',  
            border: false,  
            items:[{  
                columnWidth:.33,  
                layout: 'form',  
                border:false,  
                items:[{  
                    xtype: 'textfield',  
                    fieldLabel: 'Nombre',  
                    maxLength: 50,  
                    name: 'nombre',  
                    anchor:'90%',  
                    allowBlank:false  
                },comboventa
```

```
,comboproveedores
]
},{
  columnWidth:.33,
  layout: 'form',
  border:false,
  items:[{
    xtype: 'textfield',
    fieldLabel: 'Código',
    maxLength: 50,
    anchor:'90%',
    name: 'codigo',
    allowBlank:false
  },{
    xtype: 'numberfield',
    fieldLabel: 'Descuento',
    maxLength: 50,
    name: 'descuento',
    anchor:'90%',
    allowBlank:false
  }]
},{
  columnWidth:.33,
  layout: 'form',
  border:false,
  items:[{
    xtype: 'numberfield',
    fieldLabel: 'Precio Unitario',
```

```

        maxLength: 50,
        anchor:'90%',
        name: 'precio',
        allowBlank:false
    },
    combofabricantes
]
}
]
},{
    xtype: 'textarea',
    fieldLabel: 'Descripción',
    maxLength: 100,
    anchor:'95%',
    name: 'descripcion',
    allowBlank:false
}]

}],
buttons: [{
    text: 'Aceptar',
    handler: function(){
        if ( formulario.form.isValid() ) {
            formulario.form.submit({
                url:'Productos_agregar.action',
                waitMsg:'Guardando...',
                reset: false,
                failure: function(formulario, action) {

```

```
        Ext.MessageBox.alert('Acceso denegado', action.result.result);
    },
    success: function(formulario, action) {
        Ext.MessageBox.show({
            title: 'Producto Agregado',
            msg: action.result.result,
            closable: false,
            resizable: false,
            buttons: Ext.MessageBox.OK
        });
        Ext.get(tablas).load({
            url:'Productos_listar.action',
            scripts:true
        });
    }
});
} else {
    Ext.Msg.show({
        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
text:'Cancelar/Salir',
```



```
width:'80',  
listeners:{  
  'click': function(){  
    Ext.get(tablas).load({  
      url:'Productos_listar.action',  
      scripts:true  
    });  
  }  
}  
}]  
});  
});
```

```
Ext.onReady(function(){  
  
  Ext.QuickTips.init();  
  Ext.form.Field.prototype.msgTarget = 'side';  
  
  var storeGrid = new Ext.data.JsonStore({  
    url:'Usuarios_listarusuarios.action',  
    totalProperty: 'totalCount',  
    root:'data',  
    fields:['idCredencial', 'idUsuario', 'nombre','apellidop','apellidom', 'rfc', 'telefono',  
      'tipo', 'mail', 'celular', 'colonia', 'calle', 'delegacion', 'cp', 'estado',  
      'nacimiento', 'ingreso', 'username']  
  });
```

```
storeGrid.load();

var xg = Ext.grid;

var pager = new Ext.PagingToolbar({
    store: storeGrid,
    displayInfo:true,
    displayMsg:'{0} - {1} de {2} Peticiones',
    empty:'No hay Peticiones',
    pageSize: 5,
    plugins: new Ext.ux.ProgressBarPager()
});

var sm2 = new xg.CheckboxSelectionModel({
    hidden: true,
    listeners: {
        selectionchange: function(sm) {
            if (sm.getCount()==1) {
                usuarios.update.enable();
                usuarios.pass.enable();
            } else {
                usuarios.update.disable();
                usuarios.pass.disable();
            }
            if (sm.getCount()>=1) {
                usuarios.eliminar.enable();
            } else {
                usuarios.eliminar.disable();
            }
        }
    }
});
```

```
    }  
  }  
}  
});
```

```
var usuarios = new xg.GridPanel ({
```

```
  store: storeGrid,  
  bbar: pager,  
  cm: new xg.ColumnModel({  
    defaults: {  
      width: 60,  
      sortable: true  
    },  
    columns: [  
      {  
        header: "Nombre",  
        width: 100,  
        sortable: true,  
        dataIndex: 'nombre'  
      },  
      {  
        header: "Apellido Paterno",  
        width: 130,  
        sortable: true,  
        dataIndex: 'apellidop'  
      },  
      {  
        header: "Apellido Materno",
```

```
width: 130,  
sortable: true,  
dataIndex: 'apellidom'  
},  
{  
  header: "Tipo Usuario",  
  width: 100,  
  sortable: true,  
  dataIndex: 'tipo'  
},  
{  
  header: "Mail",  
  width: 140,  
  sortable: true,  
  dataIndex: 'mail'  
},  
{  
  header: "Username",  
  width: 100,  
  sortable: true,  
  dataIndex: 'username'  
},{  
  header: "RFC",  
  width: 100,  
  sortable: true,  
  dataIndex: 'rfc'  
},{  
  header: "Celular",
```

```
width: 100,
sortable: true,
dataIndex: 'celular'
},{
  hidden: true,
  name: 'idUsuario'
},
sm2]
}),
viewConfig: {
  forceFit:true,
  stripeRows: true
},
sm:sm2,
width: 1200,
height: 500,
tbar:[{
  text: 'Agregar',
  tooltip:'Agregar',
  iconCls:'useradd',
  listeners: {
    'click':function(){
      Ext.get(tablas).load({
        url:'Usuarios_insertar.action',
        scripts:true
      });
    }
  }
}]
}
```

```
    },{
text:'Actualizar',
tooltip:'Actualizar',
iconCls:'update',
ref: '../update',
disabled: true,
listeners: {
    'click': function(){
        var chk= usuarios.getSelectionModel().getSelections();
        if(chk.length==1){
            Ext.Ajax.request({
                url: 'Usuarios_load.action',
                params: {
                    idUsuario: chk[0].get('idUsuario')
                },
                success: function() {
                    Ext.get(tablas).load({
                        url:'Usuarios_form_actualizar.action',
                        scripts:true
                    });
                },
                failure: function(action) {
                    Ext.MessageBox.alert('Acceso denegado', action.result.result);
                }
            });
        }
    }
}
}
```

```

},{
  text:'Cambiar contraseña',
  tooltip:'Cambiar contraseña',
  iconCls:'user',
  ref: '../pass',
  listeners:{
    'click':function(){
      Ext.get(tablas).load({
        url:'Credencial_cambiarpass.action',
        scripts:true
      })
    }
  }
},
},{
  text:'Resetear contraseña',
  tooltip:'Resetear contraseña',
  iconCls:'user',
  ref: '../pass',
  disabled: true,
  listeners:{
    'click':function(){
      var chk= usuarios.getSelectionModel().getSelections();
      if(chk.length==1){
        Ext.MessageBox.confirm('Confirm', '¿Esta seguro de resetear la contraseña al usuario seleccionado?',validate);
      }
      else{
        Ext.Msg.alert('Alerta','Elige al menos un producto');
      }
    }
  }
}

```

```

    }
    function validate(btn){
        if(btn=='yes'){
            Ext.Ajax.request({
                url: 'Credencial_resetpass.action',
                params: {
                    idCredencial: chk[0].get('idCredencial')
                },
                success: function() {
                    Ext.MessageBox.alert('Se ha enviado un correo al usuario con su nueva
contraseña');
                    Ext.get(tablas).load({
                        url:'Usuarios_listar.action',
                        scripts:true
                    });
                },
                failure: function(action) {
                    Ext.MessageBox.alert('Acceso denegado', action.result.result);
                }
            });
        }
    }
}
},{
    text:'Eliminar',
    tooltip:'Eliminar',

```



```
iconCls:'eliminar',
ref: '../eliminar',
disabled: true,
listeners:{
    'click': function(){
        var chk= usuarios.getSelectionModel().getSelections();
        if(chk.length==1){
            Ext.MessageBox.confirm('Confirm', '¿Esta seguro de eliminar a los usuarios
seleccionadas?',validate);
        }
        else{
            Ext.Msg.alert('Alerta','Elige al menos un producto');
        }
        function validate(btn){
            if(btn=='yes'){
                Ext.Ajax.request({
                    url: 'Credencial_eliminar.action',
                    params: {
                        idCredencial: chk[0].get('idCredencial')
                    },
                    success: function() {
                        Ext.MessageBox.alert('Usuario eliminado');
                        Ext.get(tablas).load({
                            url:'Usuarios_listar.action',
                            scripts:true
                        });
                    },
                },
```

```
        failure: function(action) {
            Ext.MessageBox.alert('Acceso denegado', action.result.result);
        }
    });
}
}
}
}
}],
title: '::: Usuarios :::',
renderTo: 'tablas'

});

});
```

```
Ext.onReady(function(){
    Ext.QuickTips.init();

    var formulario = new Ext.FormPanel({
        labelAlign: 'top',
        id:'form-user',
        renderTo: 'tablas',
        title: 'Actualizar ',
        bodyStyle:'padding:5px 5px 0',
        width: 1200,
        items: [{
```

```
xtype: 'fieldset',
title:'Datos Personales',
items: [{
  layout: 'column',
  border: false,
  items:[{
    columnWidth:.33,
    layout: 'form',
    border:false,
    items :[{
      xtype: 'textfield',
      fieldLabel: 'Nombre',
      emptyText: 'Nombre',
      name: 'nombre',
      anchor:'95%',
      allowBlank:false
    },{
      xtype: 'textfield',
      fieldLabel: 'RFC',
      name: 'rfc',
      anchor:'95%',
      emptyText: 'RFC'
    }
  ]
}, {
  columnWidth:.33,
  layout: 'form',
  border:false,
  items:[{
```

```
xtype: 'textfield',
fieldLabel: 'Apellido Paterno',
emptyText: 'Apellido Paterno',
anchor:'95%',
name: 'apellidop'
},{
fieldLabel: 'Fecha nacimiento',
name: 'nacimiento',
xtype: 'datefield',
emptyText: 'mm/dd/yyyy',
anchor:'50%',
renderer : Ext.util.Format.dateRenderer('m/d/Y')
}]
},{
columnWidth:.33,
layout: 'form',
border:false,
items:[{
xtype: 'textfield',
fieldLabel: 'Apellido Materno',
emptyText: 'Apellido Materno',
anchor:'95%',
name: 'apellidom'
},combotipo
]
}]
}]
```

```
},{
  xtype: 'fieldset',
  title: 'Contacto',
  items: [{
    layout: 'column',
    border: false,
    items: [{
      columnWidth: .33,
      layout: 'form',
      border: false,
      items: [{
        xtype: 'textfield',
        fieldLabel: 'Telefono Casa',
        emptyText: 'Telefono Casa',
        maxLength: 30,
        anchor: '95%',
        name: 'telefono'
      }]
    }]
  }],
  columnWidth: .33,
  layout: 'form',
  border: false,
  items: [{
    xtype: 'textfield',
    fieldLabel: 'Telefono Celular',
    emptyText: 'Celular',
    maxLength: 30,
    anchor: '95%',
```

```
        name: 'celular'
    }}
}, {
    columnWidth: .33,
    layout: 'form',
    border: false,
    items: [{
        xtype: 'textfield',
        fieldLabel: 'Email',
        emptyText: 'Email',
        vtype: 'email',
        anchor: '95%',
        name: 'mail',
        allowBlank: false

    }}
    }}
}, {
    xtype: 'fieldset',
    title: 'Domicilio',

    items: [{
        layout: 'column',
        border: false,
        items: [{
            columnWidth: .33,
            layout: 'form',
```

```
border:false,
items:[{
  xtype: 'textfield',
  fieldLabel: 'Calle',
  emptyText: 'Calle',
  anchor:'95%',
  name: 'calle'
},{
  xtype: 'textfield',
  fieldLabel: 'Codigo Postal',
  emptyText: 'Codigo Postal',
  anchor:'95%',
  name: 'cp'
}]
},{
columnWidth:.33,
layout: 'form',
border:false,
items:[{
  xtype: 'textfield',
  fieldLabel: 'Colonia',
  emptyText: 'Colonia',
  anchor:'95%',
  name: 'colonia'
},comboestado
]
},{
columnWidth:.33,
```

```
layout: 'form',
border:false,
items:[{
    xtype: 'textfield',
    fieldLabel: 'Delegación',
    emptyText: 'Delegación',
    anchor:'95%',
    name: 'delegacion'
}]
}]
}]

}],
buttons: [{
    text: 'Aceptar',
    handler: function(){
        if ( formulario.form.isValid() ) {
            formulario.form.submit({
                url:'Usuarios_actualizar.action',
                waitMsg:'Guardando...',
                reset: false,
                failure: function(formulario, action) {
                    Ext.MessageBox.alert('Acceso denegado', action.result.result);
                },
                success: function(formulario, action) {
                    Ext.MessageBox.show({
                        title: 'Usuario Actualizado',
                        msg: action.result.result,
```



```
        closable: false,

        resizable: false,

        buttons: Ext.MessageBox.OK

    });

    Ext.get(tablas).load({

        url:'Usuarios_listar.action',

        scripts:true

    });

}

});

} else {

    Ext.Msg.show({

        title:'Mensaje',

        msg: 'No debe haber Campos vacios',

        buttons: Ext.Msg.OK,

        animEl: document.body,

        icon: Ext.MessageBox.INFO

    });

}

}

};{

    text:'Cancelar/Salir',

    width:'80',

    listeners:{

        'click': function(){

            Ext.get(tablas).load({

                url:'Usuarios_listar.action',

                scripts:true
```

```
        });  
    }  
}  
}]  
});
```

formulario.getForm().load({// Aqui cargo los datos del form

```
    url:'Usuarios_cargar.action',  
    success: function() {  
        Ext.MessageBox.alert('Success', action.result.result);  
    },
```

```
    failure: function() {  
        Ext.Msg.show({  
            title : 'ERROR',  
            msg : 'Carga de datos fallida!',  
            icon : Ext.MessageBox.ERROR,  
            buttons: Ext.Msg.OK  
        });  
    }  
}
```

```
});
```

```
});
```

```
Ext.apply(Ext.form.VTypes, {  
    password : function(val, field) {  
        if (field.initialPassField) {
```

```
        var pwd = Ext.getCmp(field.initialPassField);
        return (val == pwd.getValue());
    }
    return true;
},
passwordText : 'El Password no coinciden'
});
```

```
Ext.onReady(function(){
    Ext.QuickTips.init();

    var formulario = new Ext.FormPanel({
        labelAlign: 'top',
        id:'form-user',
        title: 'Agregar Usuario',
        bodyStyle:'padding:5px 5px 0',
        width: 1200,
        items: [{
            xtype: 'fieldset',
            title:'Detalle Usuario',
            items:[{
                layout: 'column',
                border: false,
                items:[{
                    columnWidth:.33,
                    layout: 'form',
                    border:false,
```

```
items :[{
    xtype: 'textfield',
    fieldLabel: 'Nombre Usuario',
    emptyText: 'Nombre de usuario',
    name: 'username',
    anchor:'95%',
    id: 'username'
}]
},{
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[{
        xtype: 'textfield',
        fieldLabel: 'Contraseña',
        inputType: 'password',
        name: 'pass',
        id: 'pass',
        anchor:'95%',
        allowBlank:true
    }]
},{
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[{
        xtype: 'textfield',
        fieldLabel: 'Confirmar Contraseña',
```

```
        inputType: 'password',
        name: 'passcfrm',
        vtype: 'password',
        allowBlank:true,
        anchor:'95%',
        initialPassField: 'pass' // id of the initial password field
    }}
}}
}}
},{
xtype: 'fieldset',
title:'Datos Personales',
items: [{
    layout: 'column',
    border: false,
    items:[{
        columnWidth:.33,
        layout: 'form',
        border:false,
        items :[{
            xtype: 'textfield',
            fieldLabel: 'Nombre',
            emptyText: 'Nombre',
            name: 'nombre',
            anchor:'95%',
            allowBlank:false
        }],{
            xtype: 'textfield',
```

```
        fieldLabel: 'RFC',
        name: 'rfc',
        anchor:'95%',
        emptyText: 'RFC'
    ]
}, {
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[{
        xtype: 'textfield',
        fieldLabel: 'Apellido Paterno',
        emptyText: 'Apellido Paterno',
        anchor:'95%',
        name: 'apellidop'
    }, {
        fieldLabel: 'Fecha nacimiento',
        name: 'nacimiento',
        xtype: 'datefield',
        emptyText: 'mm/dd/yyyy',
        anchor:'50%',
        renderer : Ext.util.Format.dateRenderer('m/d/Y')
    }
    ]
}, {
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[{
```

```
        xtype: 'textfield',
        fieldLabel: 'Apellido Materno',
        emptyText: 'Apellido Materno',
        anchor:'95%',
        name: 'apellidom'
    },combotipo
    ]
}
}]
```

```
}, {
    xtype: 'fieldset',
    title:'Contacto',
    items:[{
        layout: 'column',
        border: false,
        items:[{
            columnWidth:.33,
            layout: 'form',
            border:false,
            items :[{
                xtype: 'textfield',
                fieldLabel: 'Telefono Casa',
                emptyText: 'Telefono Casa',
                maxLength: 30,
                anchor:'95%',
                name: 'telefono'
            }
        ]
    }
    ]
}
```

```
},{
  columnWidth:.33,
  layout: 'form',
  border:false,
  items:[{
    xtype: 'textfield',
    fieldLabel: 'Telefono Celular',
    emptyText: 'Celular',
    maxLength: 30,
    anchor:'95%',
    name: 'celular'
  ]
},{
  columnWidth:.33,
  layout: 'form',
  border:false,
  items:[{
    xtype: 'textfield',
    fieldLabel: 'Email',
    emptyText: 'Email',
    vtype:'email',
    anchor:'95%',
    name: 'mail',
    allowBlank:false
  ]
}
}
```



xtype: 'fieldset',

title:'Domicilio',

items:{{

  layout: 'column',

  border: false,

  items:{{

    columnWidth:.33,

    layout: 'form',

    border:false,

    items:{{

      xtype: 'textfield',

      fieldLabel: 'Calle',

      emptyText: 'Calle',

      anchor:'95%',

      name: 'calle'

    },{

      xtype: 'textfield',

      fieldLabel: 'Codigo Postal',

      emptyText: 'Codigo Postal',

      anchor:'95%',

      name: 'cp'

    }}  
  }}

  },{

    columnWidth:.33,

    layout: 'form',

    border:false,

    items:{{

```
        xtype: 'textfield',
        fieldLabel: 'Colonia',
        emptyText: 'Colonia',
        anchor:'95%',
        name: 'colonia'
    },comboestado
]
},{
    columnWidth:.33,
    layout: 'form',
    border:false,
    items:[{
        xtype: 'textfield',
        fieldLabel: 'Delegación',
        emptyText: 'Delegación',
        anchor:'95%',
        name: 'delegacion'
    }
    ]
}
}],
buttons: [{
    text: 'Aceptar',
    handler: function(){
        if ( formulario.form.isValid() ) {
            formulario.form.submit({
                url:'Usuarios_agregar.action',
```

```
waitMsg:'Guardando...',

reset: false,

failure: function(formulario, action) {

    Ext.MessageBox.alert('Acceso denegado', action.result.result);

},

success: function(formulario, action) {

    Ext.MessageBox.show({

        title: 'Usuario Agregado',

        msg: action.result.result,

        closable: false,

        resizable: false,

        buttons: Ext.MessageBox.OK

    });

    Ext.get(tablas).load({

        url:'Usuarios_listar.action',

        scripts:true

    });

}

});

} else {

    Ext.Msg.show({

        title:'Mensaje',

        msg: 'No debe haber Campos vacios',

        buttons: Ext.Msg.OK,

        animEl: document.body,

        icon: Ext.MessageBox.INFO

    });

}
```

```

    }
  },{
    text:'Cancelar/Salir',
    width:'80',
    listeners:{
      'click': function(){
        Ext.get(tablas).load({
          url:'Usuarios_listar.action',
          scripts:true
        });
      }
    }
  ]
});

formulario.render(tablas);
});

Ext.onReady(function(){

  Ext.QuickTips.init();

  var formulario = new Ext.FormPanel({
    labelAlign: 'top',
    renderTo: 'tablas',
    title: 'Actualizar Mensaje',
    bodyStyle:'padding:5px 5px 0',
    width: 550,

```

```
items: [{
    xtype: 'textarea',
    fieldLabel: 'Comentario',
    maxLength: 100,
    name: 'comentario',
    anchor:'95%',
    allowBlank:false
}],
buttons: [{
    text: 'Aceptar',
    handler: function(){
        if ( formulario.form.isValid() ) {
            formulario.form.submit({
                url:'Mensajes_actualizar.action',
                waitMsg:'Guardando...',
                reset: false,
                failure: function(formulario, action) {
                    Ext.MessageBox.alert('Acceso denegado', action.result.result);
                },
                success: function(formulario, action) {
                    Ext.MessageBox.show({
                        title: 'Comentario Agregado',
                        msg: action.result.result,
                        closable: false,
                        resizable: false,
                        buttons: Ext.MessageBox.OK
                    });
                    Ext.get(tablas).load({
```

```
        url:'Mensajes_listar.action',
        scripts:true
    });
    }
});
} else {
    Ext.Msg.show({
        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text:'Cancelar/Salir',
    width:'80',
    listeners:{
        'click': function(){
            Ext.get(tablas).load({
                url:'Mensajes_listar.action',
                scripts:true
            });
        }
    }
}
});
});
```

```
formulario.getForm().load({// Aqui cargo los datos del form

    url:'Mensajes_cargar.action',

    success: function() {

        Ext.MessageBox.alert('Success', action.result.result);

    },

    failure: function() {

        Ext.Msg.show({

            title : 'ERROR',

            msg : 'Carga de datos fallida!',

            icon : Ext.MessageBox.ERROR,

            buttons: Ext.Msg.OK

        });

    }

});

});

Ext.onReady(function(){

if(Ext.getCmp('user-form')!=null){

    Ext.getCmp('user-form').destroy();

}

Ext.QuickTips.init();
```

```
Ext.form.Field.prototype.msgTarget = 'side';
```

```
var storeGrid = new Ext.data.JsonStore({  
    id:'datos',  
    url:'Mensajes_listarmensajes.action',  
    totalProperty: 'totalCount',  
    root:'data',  
    fields:['idMensaje', 'idTema', 'Tema','usuario','comentario', 'fecha']  
});
```

```
storeGrid.load({params:{start:0,limit:3}});
```

```
var pager = new Ext.PagingToolbar({  
    store: storeGrid,  
    displayInfo:true,  
    displayMsg:'{0} - {1} de {2} Peticiones',  
    empty:'No hay Peticiones',  
    pageSize: 5,  
    plugins: new Ext.ux.ProgressBarPager()  
});
```

```
var xg = Ext.grid;
```

```
var expander = new Ext.ux.grid.RowExpander({  
    tpl : new Ext.Template(  
        '<p><b>Descripción:</b> {comentario}</p>'  
    )  
});
```



```
var sm2 = new xg.CheckboxSelectionModel({
  hidden: true,
  listeners: {
    selectionchange: function(sm) {
      if (sm.getCount()==1) {
        temas.comentarios.enable();
      } else {
        temas.comentarios.disable();
      }
      if (sm.getCount()>=1) {
        temas.elimina.enable();
      } else {
        temas.elimina.disable();
      }
    }
  }
});
```

```
var temas = new xg.GridPanel({
  id: 'message-form',
  store: storeGrid,
  bbar: pager,
  cm: new xg.ColumnModel({
    defaults: {
      width: 70,
      sortable: true
    },
  },
```

```
columns: [  
  expander,{  
    id:'idMensaje',  
    header: "Mensaje",  
    width: 800,  
    sortable: true,  
    locked:false,  
    dataIndex: 'usuario'  
  },{  
    header: "Fecha",  
    width: 400,  
    sortable: true,  
    dataIndex: 'fecha'  
  },  
  sm2  
],  
)),  
viewConfig: {  
  stripeRows: true  
},  
sm: sm2,  
width: 1200,  
height: 500,  
tbar:[{  
  text:'Agregar comentario',  
  tooltip:'Agregar comentario',  
  iconCls:'nota',  
  listeners:{
```

```

'click':function(){
    Ext.get(tablas).load({
        url:'Mensajes_insertar.action'
    });
}
}
},{
text:'Actualizar',
tooltip:'Actualizar',
iconCls:'update',
ref: '../comentarios',
disabled: true,
listeners: {
'click':function(){
    var chk= temas.getSelectionModel().getSelections();
    if(chk.length==1){
        Ext.Ajax.request({
            url: 'Mensajes_load.action',
            params: {
                idMensaje: chk[0].get('idMensaje')
            },
            success: function() {
                Ext.get(tablas).load({
                    url:'Mensajes_form_actualizar.action',
                    scripts:true
                });
            },
            failure: function(action) {

```

```

        Ext.MessageBox.alert('Acceso denegado', action.result.result);
    }
    });
}
}
}
},{
    text:'Eliminar',
    tooltip:'Eliminar',
    iconCls:'eliminar',
    ref: './elimina',
    disabled: true,
    listeners:{
        'click':function(){
            var chk= temas.getSelectionModel().getSelections();
            if(chk.length>=1){
                Ext.MessageBox.confirm('Confirm', '¿Esta seguro de eliminar los mensajes
seleccionados?',validate);
            }
            else{
                Ext.Msg.alert('Alerta','Elige al menos un mensaje');
            }
            function validate(btn){
                if(btn=='yes'){
                    Ext.Ajax.request({
                        url: 'Mensajes_eliminar.action',
                        params: {

```

```
        idMensaje: chk[0].get('idMensaje')
    },
    success: function() {
        Ext.MessageBox.alert('Mensaje eliminado');
        Ext.get(tablas).load({
            url:'Mensajes_listar.action',
            scripts:true
        });
    },
    failure: function() {
        Ext.MessageBox.alert('Acceso denegado', action.result.result);
    }
});
}
}
}
}
}
},{
    text:'Regresar',
    tooltip:'Regresar',
    iconCls:'back',
    listeners:{
        'click':function(){
            Ext.get(tablas).load({
                url:'Temas_listar.action',
                scripts:true
            });
        }
    }
}
```

```
    }  
  }  
],  
  plugins: expander,  
  title: ':: Mensajes ::',  
  iconCls: 'bitacora',  
  renderTo: 'tablas'  
});  
});
```

```
function createForm(div,title){
```

```
  var gridForm = new Ext.FormPanel({  
    id: 'mensajes-form',  
    renderTo: div,  
    frame: true,  
    labelAlign: 'left',  
    collapsible: true,  
    animCollapse: false,  
  
    title: 'Listar Mensajes' + title,  
    bodyStyle:'padding:5px',  
    width: 1000,  
    layout: 'column', // Specifies that the items will now be arranged in columns  
    items: [{
```

```
columnWidth: 0.55,

layout: 'fit',

items: {

    xtype: 'grid',

    //store: storeGrid,

    //bbar: pager,

    //plugins: expander,

    viewConfig: {

        forceFit: true

    },

    //cm: colModel,

//    sm: new Ext.grid.RowSelectionModel({

//        singleSelect: true,

//        listeners: {

//            rowselect: function(sm, row, rec) {

//                Ext.getCmp("topic-form").getForm().loadRecord(rec);

//            }

//        }

//    }),

width: 800,

height: 400,

title: 'Mensajes',

border: true,

listeners: {

    viewready: function(g) {

        g.getSelectionModel().selectRow(0);

    } // Allow rows to be rendered.

}

}
```

```

    }
  },{
    columnWidth: 0.45,
    xtype: 'fieldset',
    //checkboxToggle:true,
    //collapsible:true,
    labelWidth: 90,
    title:'Detalle Tema',
    defaults: {width: 140, border:false}, // Default config options for child items
    defaultType: 'textfield',
    autoHeight: true,
    bodyStyle: Ext.isIE ? 'padding:0 0 5px 15px;' : 'padding:10px 15px;',
    border: true,
    style: {
      "margin-left": "10px", // when you add custom margin in IE 6...
      "margin-right": Ext.isIE6 ? (Ext.isStrict ? "-10px" : "-13px") : "0" // you have to
adjust for it somewhere else
    },
    items: [{
      xtype:'textfield',
      fieldLabel: 'Nombre',
      maxLength: 30,
      anchor: '81%',
      allowBlank:false,
      name: 'nombre'
    },{
      fieldLabel: 'Categoria',
      maxLength: 30,

```



```

        anchor: '81%',
        allowBlank:false,
        name: 'categoria'
    },{
        xtype:'textarea',
        fieldLabel: 'Descripción',
        maxLength: 50,
        anchor: '81%',
        allowBlank:false,
        name: 'descripcion'
    },{
        xtype: 'hidden',
        name: 'idTema'
    }
}]{
    columnWidth: 0.4,
    xtype: 'fieldset',
    collapsible:false,
    labelWidth: 90,
    defaults: {width: 140, border:false}, // Default config options for child items
    defaultType: 'textfield',
    autoHeight: true,
    bodyStyle: Ext.isIE ? 'padding:0 0 5px 15px;' : 'padding:10px 15px;',
    border: false,
    style: {
        "margin-left": "10px", // when you add custom margin in IE 6...
        "margin-right": Ext.isIE6 ? (Ext.isStrict ? "-10px" : "-13px") : "0" // you have to
adjust for it somewhere else
    }
}

```

```

    },
    buttons: [{
        text: 'Actualizar',
        handler: function(){
            if ( gridForm.form.isValid() ) {
                gridForm.form.submit({
                    url: 'Temas_actualizar.action',
                    waitMsg:'Guardando...',
                    icon:'ext-mb-download',
                    reset: false,
                    failure: function(gridForm, action) {
                        Ext.MessageBox.alert('Acceso denegado', action.result.result);
                    },
                    success: function(gridForm, action) {
                        Ext.MessageBox.show({
                            title: 'Datos Actualizados',
                            msg: action.result.result,
                            closable: false,
                            resizable: false,
                            buttons: Ext.MessageBox.OK
                        });
                        Ext.get(tablas).load({
                            url:'Temas_listar.action',
                            scripts:true
                        });
                    }
                });
            }
        }
    }
    ]
} else {

```

```
Ext.Msg.show({
    title:'Mensaje',
    msg: 'Debe Elegir a un Usuario',
    buttons: Ext.Msg.OK,
    animEl: document.body,
    icon: Ext.MessageBox.INFO
});
}
}
},{
    text: 'Ver comentarios',
    handler: function(){
        gridForm.getForm().reset();
    }
},{
    text: 'Eliminar',
    handler: function(){
        if ( gridForm.form.isValid() ){
            gridForm.form.submit({
                url: 'Temas_eliminar.action',
                waitMsg:'Eliminando...',
                reset: false,
                failure: function(gridForm, action) {
                    Ext.MessageBox.alert('Acceso denegado', action.result.result);
                },
                success: function(gridForm, action) {
                    Ext.MessageBox.show({
                        title: 'Datos Eliminados',
```

```
        msg: action.result.result,
        closable: false,
        resizable: false,
        buttons: Ext.MessageBox.OK
    });
    Ext.get(tablas).load({
        url:'Temas_listar.action',
        scripts:true
    });
}
});
} else {
    Ext.Msg.show({
        title:'Mensaje',
        msg: 'Debe de Elejir a un Usuario para eliminarlo',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text: 'Cancelar',
    handler: function(){
        gridForm.getForm().reset();
    }
}
}
}
```

```
});  
  
}
```

```
function ajaxDivUpdate(url,div){  
    Ext.get(div).load({  
        url:url,  
        scripts:true  
    });  
}
```

```
function mascara(activa){  
    var mask=Ext.getCmp('mascara');  
    if(mask!=null){  
        if(activa==true){  
            if(mask.isVisible()==false){  
                mask.show();  
            }  
        }  
        else{  
            if(mask.isVisible()==true){
```

```
        mask.hide();
    }
}
}
```

```
function divColor(val){
    return '<div id="color" style="background:'+val+';"></div>';
}
```

```
function paint(val){
    return '<span style="color:#000000;text-decoration: underline;">' + val + '</span>';
}
```

```
Ext.BLANK_IMAGE_URL='ext/resources/images/default/s.gif';
```

```
Ext.Updater.defaults.loadScripts=true;
```

```
Ext.Updater.defaults.disableCaching=true;
```

```
Ext.onReady(function() {
    new Ext.Toolbar({
        renderTo: document.getElementById('contenedor_menu'),
        height: 50,
        items: [
            {
                xtype: 'tbseparator'
            }
        ]
    });
});
```

```
height:45,
width:70,
xtype: 'tbutton',
text: 'Inicio',
handler:function(){
    Ext.get(tablas).load({
        url:'GestionCRM_contenido.action',
        scripts:true
    });
}
},{
    xtype: 'tbseparator'
},{
    xtype:'splitbutton',
    height:45,
    width:70,
    text: 'Perfil',
    iconCls: 'add16',
    menu: [
        {text: 'Perfil',
            handler:function(){
                Ext.get(tablas).load({
                    url:'Usuarios_listar.action',
                    scripts:true
                });
            }
        }
    ]
},{
```

```
xtype: 'tbseparator',
height:45
},{
xtype:'splitbutton',
height:45,
width:70,
text: 'Productos',
iconCls: 'add16',
menu: [
    {text: 'Listar',
        handler:function(){
            Ext.get(tablas).load({
                url:'Productos_listar.action',
                scripts:true
            });
        }
    }
],{
text: 'Ordenes de Compra',
handler:function(){
    Ext.get(tablas).load({
        url:'Ordenes_listar.action',
        scripts:true
    });
}
}
],{
xtype: 'tbseparator'
},{
```



```
xtype:'splitbutton',
height:45,
width:70,
text: 'Servicios',
iconCls: 'add16',
menu: [
    {text: 'Listar',
        handler:function(){
            Ext.get(tablas).load({
                url:'Servicios_listar.action',
                scripts:true
            });
        }
    }
]
},{
xtype: 'tbseparator'
},{
height:45,
width:70,
xtype: 'tbbutton',
text: 'Carrito de compras',
handler:function(){
    Ext.get(tablas).load({
        url:'Carrito_listar.action',
        scripts:true
    });
}
},{
```

```
xtype: 'tbseparator'
},{
xtype:'splitbutton',
height:45,
width:60,
text: 'Foro',
iconCls: 'add16',
menu: [
    {text: 'Listar',
        handler:function(){
            Ext.get(tablas).load({
                url:'Temas_listar.action',
                scripts:true
            });
        }
    }
]
},{
xtype: 'tbseparator'
},{
height:45,
width:70,
xtype: 'tbbutton',
text: 'Cerrar Sesión',
handler: function(){
    window.location='http://localhost:8080/GestionCRM';
}
}
});
```

```
});  
  
Ext.onReady(function(){  
  
    Ext.QuickTips.init();  
  
    var formulario = new Ext.FormPanel({  
        labelAlign: 'top',  
        renderTo: 'tablas',  
        title: 'Agregar Mensaje',  
        bodyStyle:'padding:5px 5px 0',  
        width: 550,  
        items: [{  
            xtype: 'textarea',  
            fieldLabel: 'Comentario',  
            maxLength: 100,  
            name: 'comentario',  
            anchor:'95%',  
            allowBlank:false  
        }],  
        buttons: [{  
            text: 'Aceptar',  
            handler: function(){  
                if ( formulario.form.isValid() ) {  
                    formulario.form.submit({  
                        url:'Mensajes_agregar.action',  
                        waitMsg:'Guardando...',  
                        reset: false,  
                        failure: function(formulario, action) {
```

```
        Ext.MessageBox.alert('Acceso denegado', action.result.result);
    },
    success: function(formulario, action) {
        Ext.MessageBox.show({
            title: 'Comentario Agregado',
            msg: action.result.result,
            closable: false,
            resizable: false,
            buttons: Ext.MessageBox.OK
        });
        Ext.get(tablas).load({
            url:'Mensajes_listar.action',
            scripts:true
        });
    }
});
} else {
    Ext.Msg.show({
        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
text:'Cancelar/Salir',
```

```
width:'80',  
listeners:{  
    'click': function(){  
        Ext.get(tablas).load({  
            url:'Mensajes_listar.action',  
            scripts:true  
        });  
    }  
}  
});  
});
```

```
Ext.apply(Ext.form.VTypes, {  
    password : function(val, field) {  
        if (field.initialPassField) {  
            var pwd = Ext.getCmp(field.initialPassField);  
            return (val == pwd.getValue());  
        }  
        return true;  
    },  
    passwordText : 'El Password no coinciden'  
});
```

```
Ext.onReady(function(){  
  
    Ext.QuickTips.init();
```

```
var formulario = new Ext.FormPanel({  
    frame:true,  
    renderTo: 'tablas',  
    title: 'Cambiar contraseña',  
    bodyStyle:'padding:5px 5px 0',  
    width: 450,  
    fieldDefaults: {  
        msgTarget: 'side',  
        labelWidth: 75  
    },  
    defaultType: 'textfield',  
    defaults: {  
        anchor: '90%'  
    },  
  
    items: [{  
        fieldLabel: 'Contraseña',  
        inputType: 'password',  
        name: 'password',  
        id: 'password',  
        allowBlank:false  
    },{  
        fieldLabel: 'Confirmar contraseña',  
        inputType: 'password',  
        name: 'passcfrm',  
        vtype: 'password',  
        allowBlank:false,  
        initialPassField: 'password' // id of the initial password field
```

```
    }},
  buttons: [{
    text: 'Aceptar',
    handler: function(){
      if ( formulario.form.isValid() ) {
        formulario.form.submit({
          url:'Credencial_actualizar.action',
          waitMsg:'Actualizando...',
          reset: false,
          failure: function(formulario, action) {
            Ext.MessageBox.alert('Acceso denegado', action.result.result);
          },
          success: function(formulario, action) {
            Ext.MessageBox.show({
              title: 'Contraseña Actualizada',
              msg: action.result.result,
              closable: false,
              resizable: false,
              buttons: Ext.MessageBox.OK
            });
            Ext.get(tablas).load({
              url:'Usuarios_listar.action',
              scripts:true
            });
          }
        });
      } else {
        Ext.Msg.show({
```

```
        title:'Mensaje',
        msg: 'No debe haber Campos vacios',
        buttons: Ext.Msg.OK,
        animEl: document.body,
        icon: Ext.MessageBox.INFO
    });
}
}
},{
    text:'Cancelar/Salir',
    width:'80',
    listeners:{
        'click': function(){
            Ext.get(tablas).load({
                url:'Usuarios_listar.action',
                scripts:true
            });
        }
    }
}
});
});
```

```
Ext.onReady(function(){
```

```
Ext.QuickTips.init();
```

```
Ext.form.Field.prototype.msgTarget = 'side';
```



```
var storeGrid = new Ext.data.JsonStore({
    url:'Carrito_listarcarrito.action',
    totalProperty: 'totalCount',
    root:'data',
    fields:['idProducto', 'nombre','codigo','descripcion', 'fabricante' , 'cantidad', 'total']
});
```

```
storeGrid.load({params:{start:0, limit:3}});
```

```
var pager = new Ext.PagingToolbar({
    store: storeGrid,
    displayInfo:true,
    displayMsg:'{0} - {1} de {2} Peticiones',
    empty:'No hay Peticiones',
    pageSize: 5,
    plugins: new Ext.ux.ProgressBarPager()
});
```

```
var xg = Ext.grid;
```

```
var expander = new Ext.ux.grid.RowExpander({
    tpl : new Ext.Template(
        '<p><b>Descripción:</b> {descripcion}</p>'
    )
});
```

```
var sm2 = new xg.CheckboxSelectionModel({
    singleSelect: true,
    hidden: true,
```

```
listeners: {  
  selectionchange: function(sm) {  
    if (sm.getCount()) {  
      tareas.removeButton.enable();  
    } else {  
      tareas.removeButton.disable();  
    }  
  }  
}  
});
```

```
var tareas = new xg.EditorGridPanel ({  
  store: storeGrid,  
  bbar: pager,  
  cm: new xg.ColumnModel({  
    defaults: {  
      width: 60,  
      sortable: true  
    },  
    columns: [  
      sm2,  
      expander,  
      {  
        header: "Producto",  
        width: 150,  
        dataIndex: 'nombre',  
        hideable: false,  
        editor: new Ext.form.TextField({
```

```

        allowBlank: false
    })),
    {
        header: "Código",
        width: 50,
        dataIndex: 'codigo'
    },
    {header: "Fabricante", width: 300, dataIndex: 'fabricante'},
    //{header: "Proveedor", width: 100, dataIndex: 'proveedor'},
    {header: "Cantidad", width: 100, dataIndex: 'cantidad'},
    {header: "Total", width: 100, dataIndex: 'total'}

]
}),
viewConfig: {
    forceFit:true
},
sm: sm2,
width: 1200,
height: 500,
tbar:[{
    text:'Comprar',
    tooltip:'Comprar',
    iconCls:'comprar',
    listeners:{
        'click': function(){
            Ext.MessageBox.confirm('Confirm', '¿Esta seguro de comprar los
productos?', validate);

```

```

function validate(btn){
    if(btn=='yes'){
        Ext.Ajax.request({
            url: 'Carrito_comprar.action',
            success: function() {
                Ext.MessageBox.alert('Producto(s) Comprado(s)');
                Ext.get(tablas).load({
                    url:'Ordenes_listar.action',
                    scripts:true
                });
            },
            failure: function(action) {
                Ext.MessageBox.alert('Acceso denegado', action.result.result);
            }
        });
    }
}

},{
    text:'Eliminar Producto',
    tooltip:'Eliminar',
    iconCls:'eliminar',
    ref: './removeButton',
    disabled: true,
    listeners:{
        'click':function(){
            var chk= tareas.getSelectionModel().getSelections();

```

```
if(chk.length==1){
    Ext.MessageBox.confirm('Confirm', '¿Esta seguro de eliminar los productos
seleccionados?',validate);
}
else{
    Ext.Msg.alert('Alerta','Elige al menos un producto');
}
function validate(btn){
    var chk= tareas.getSelectionModel().getSelections();
    if(btn=='yes'){
        Ext.Ajax.request({
            url: 'Carrito_eliminar.action',
            params: {
                idProducto: chk[0].get('idProducto')
            },
            success: function() {
                Ext.MessageBox.alert('Producto Eliminado');
                Ext.get(tablas).load({
                    url:'Carrito_listar.action',
                    scripts:true
                });
            },
            failure: function(action) {
                Ext.MessageBox.alert('Acceso denegado', action.result.result);
            }
        });
    }
}
```

```
        }
    }
}
}],
plugins: expander,
title: ':: Carrito de compras ::',
iconCls: 'bitacora',
renderTo: 'tablas'
});
});
```

```
var estados = new Ext.data.JsonStore({
    url: 'Combo_estados.action',
    root: 'estados' ,
    fields: ['id' , 'estado']

});
```

```
var comboestado = new Ext.form.ComboBox({
    typeAhead: true,
    dataIndex: 'estado',
    fieldLabel: 'Estado',
    emptyText: 'Seleccione un Estado...' ,
    triggerAction: 'all',
    store: estados,
    lazyRender:true,
    model:'local',
```

```
loadingText: 'Cargando lista...',  
hiddenName: 'estado',  
displayField: 'estado',  
valueField: 'id',  
name: 'estado',  
anchor: '90%',  
forceSelection: true,  
selectOnFocus: true  
});
```

```
var tipousuario = new Ext.data.JsonStore({  
    url: 'Combo_tipousuario.action',  
    root: 'tipousuario',  
    fields: ['id', 'tipo']  
});
```

```
var combotipo = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'tipo',  
    fieldLabel: 'Tipo Usuario',  
    emptyText: 'Seleccione un Tipo...' ,  
    triggerAction: 'all',  
    store: tipousuario,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'tipo',
```

```
displayField: 'tipo',  
valueField: 'id',  
name: 'tipo',  
anchor: '90%',  
forceSelection: true,  
selectOnFocus: true  
});
```

```
var categoria = new Ext.data.JsonStore({  
    url: 'Combo_categorias.action',  
    root: 'categorias' ,  
    fields: [ 'id' , 'categoria']  
  
});
```

```
var combocategoria = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'usuarios',  
    fieldLabel: 'Categoria',  
    emptyText: 'Seleccione una Categoria...' ,  
    triggerAction: 'all',  
    store: categoria,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'categoria',  
    displayField: 'categoria',  
    valueField: 'id',
```



```
name: 'categoria',  
anchor: '85%',  
forceSelection: true,  
selectOnFocus: true  
});
```

```
var fabricantes = new Ext.data.JsonStore({  
    url: 'Combo_fabricantes.action',  
    root: 'fabricantes' ,  
    fields: ['id' , 'fabricante']  
  
});
```

```
var combofabricantes = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'fabricante',  
    fieldLabel: 'Fabricante',  
    emptyText: 'Seleccione un Fabricante...' ,  
    triggerAction: 'all',  
    store: fabricantes,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'fabricante',  
    displayField: 'fabricante',  
    valueField: 'id',  
    name: 'tipo',  
    anchor: '90%',
```

```
forceSelection: true,  
selectOnFocus: true  
});
```

```
var proveedores = new Ext.data.JsonStore({  
    url: 'Combo_proveedores.action',  
    root: 'proveedores',  
    fields: ['id', 'proveedor']  
  
});
```

```
var comboproveedores = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'proveedor',  
    fieldLabel: 'Proveedor',  
    emptyText: 'Seleccione un Proveedor...' ,  
    triggerAction: 'all',  
    store: proveedores,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'proveedor',  
    displayField: 'proveedor',  
    valueField: 'id',  
    name: 'proveedor',  
    anchor: '90%',  
    forceSelection: true,  
    selectOnFocus: true
```

```
});
```

```
var clientes = new Ext.data.JsonStore({  
    url: 'Combo_clientes.action',  
    root: 'clientes' ,  
    fields: [ 'id' , 'cliente']
```

```
});
```

```
var comboclientes = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'cliente',  
    fieldLabel: 'Cliente',  
    emptyText: 'Seleccione un Cliente...' ,  
    triggerAction: 'all',  
    store: clientes,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'cliente',  
    displayField: 'cliente',  
    valueField: 'id',  
    name: 'cliente',  
    anchor: '85%',  
    forceSelection: true,  
    selectOnFocus: true  
});
```

```
var estadoservicio = new Ext.data.JsonStore({  
    url: 'Combo_estadoservicio.action',  
    root: 'estados' ,  
    fields: [ 'id' , 'estado']  
  
});
```

```
var comboestadoservicio = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'estado',  
    fieldLabel: 'Estado Servicio',  
    emptyText: 'Seleccione un Estado...' ,  
    triggerAction: 'all',  
    store: estadoservicio,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'estado',  
    displayField: 'estado',  
    valueField: 'id',  
    name: 'estado',  
    anchor: '85%',  
    forceSelection: true,  
    selectOnFocus: true  
});
```

```
var servicios = new Ext.data.JsonStore({  
    url: 'Combo_servicios.action',  
    root: 'servicios' ,  
    fields: [ 'id' , 'services']  
  
});
```

```
var comboservicios = new Ext.form.ComboBox({  
    typeAhead: true ,  
    dataIndex: 'services',  
    fieldLabel: 'Servicio',  
    emptyText: 'Seleccione un Servicio...' ,  
    triggerAction: 'all',  
    store: servicios,  
    lazyRender:true,  
    model:'remote',  
    loadingText: 'Cargando lista...',  
    hiddenName: 'services',  
    displayField: 'services',  
    valueField: 'id',  
    name: 'services',  
    anchor: '85%',  
    forceSelection: true,  
    selectOnFocus: true  
  
});
```

```
var comboprioridad = new Ext.form.ComboBox({
```

```
typeAhead: true,
dataIndex: 'prioridad',
fieldLabel: 'Prioridad',
emptyText: 'Seleccione una Prioridad...' ,
triggerAction: 'all',
lazyRender:true,
mode: 'local',
loadingText: 'Cargando lista...',
hiddenName: 'prioridad',
displayField: 'prioridad',
valueField: 'id',
name: 'prioridad',
anchor: '85%',
forceSelection: true,
selectOnFocus: true,
store: new Ext.data.ArrayStore({
    fields: [
        'id',
        'prioridad'
    ],
    data: [[0, 'Urgente'], [1, 'Alta'], [2, 'Media'], [3, 'Baja']]
})
});
```

```
var comboventa = new Ext.form.ComboBox({
    typeAhead: true,
```

```
dataIndex: 'tipo',
fieldLabel: 'Tipo de venta',
emptyText: 'Seleccione un Tipo de Venta...' ,
triggerAction: 'all',
lazyRender:true,
mode: 'local',
hiddenName: 'tipo',
displayField: 'tipo',
valueField: 'tipo',
loadingText: 'Cargando lista...',
anchor: '85%',
store: new Ext.data.ArrayStore({
    fields: [
        'id',
        'tipo'
    ],
    data: [[0, 'Pieza'], [1, 'Caja']]
})
});
```

```
Ext.onReady(function(){
if(Ext.getCmp('user-form')!=null){
    Ext.getCmp('user-form').destroy();
}
}
```

```
Ext.QuickTips.init();
```

```
Ext.form.Field.prototype.msgTarget = 'side';
```

```
var storeGrid = new Ext.data.JsonStore({
    url:'Ordenes_detalle.action',
    totalProperty: 'totalCount',
    root:'data',
    fields:[ 'producto','catidad','fabricante', 'proveedor', 'total']
});
```

```
storeGrid.load({params:{start:0, limit:3}});
```

```
var pager = new Ext.PagingToolbar({
    store: storeGrid,
    displayInfo:true,
    displayMsg:'{0} - {1} de {2} Peticiones',
    empty:'No hay Peticiones',
    pageSize: 5,
    plugins: new Ext.ux.ProgressBarPager()
});
```

```
var xg = Ext.grid;
```

```
var expander = new Ext.ux.grid.RowExpander({
    tpl : new Ext.Template(
        '<p><b>Descripción:</b> {descripcion}</p>'
    )
});
```

```
new xg.GridPanel ({
```



```
store: storeGrid,  
bbar: pager,  
cm: new xg.ColumnModel({  
  defaults: {  
    width: 60,  
    sortable: true  
  },  
  columns: [  
    expander, {  
      header: "Producto",  
      width: 300,  
      sortable: true,  
      locked: false,  
      dataIndex: 'producto'  
    }, {  
      header: "Fabricante",  
      width: 100,  
      sortable: true,  
      dataIndex: 'fabricante'  
    }, {  
      header: "Proveedor",  
      width: 300,  
      sortable: true,  
      dataIndex: 'proveedor'  
    }, {  
      header: "Cantidad",  
      width: 100,  
      sortable: true,  
      dataIndex: 'cantidad'    }  
  ]  
})
```

```
        dataIndex: 'cantidad'
    },{
        header: "Total",
        width: 100,
        sortable: true,
        locked:false,
        dataIndex: 'total'
    }
]
}),
viewConfig: {
    forceFit:true,
    stripeRows: true
},
width: 1200,
height: 500,
tbar:[{
    text:'Regresar',
    tooltip:'Regresar',
    iconCls:'back',
    listeners: {
        'click':function(){
            Ext.get(tablas).load({
                url:'Ordenes_listar.action',
                scripts:true
            });
        }
    }
}
]
```

```
    },{
      text:'Exportar Excel',
      tooltip:'Exportar Excel',
      iconCls:'excel',
      listeners:{
        'click':function(){
          parent.location='/GestionCRM/Documentos_excelordendetalle.action';
        }
      }
    },{
      text:'Exportar PDF',
      tooltip:'Exportar PDF',
      iconCls:'pdf',
      listeners:{
        'click': function(){
          parent.location='/GestionCRM/Documentos_pdfOrdendetalle.action';
        }
      }
    }
  ],
  plugins: expander,
  title: '::: Detalles Servicio :::',
  renderTo: 'tablas'

});

});

/*
```

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

```
Ext.BLANK_IMAGE_URL='ext/resources/images/default/s.gif';
```

```
Ext.Updater.defaults.loadScripts=true;
```

```
Ext.Updater.defaults.disableCaching=true;
```

```
Ext.onReady(function(){
```

```
    Ext.QuickTips.init();
```

```
    var formulario= new Ext.FormPanel({
```

```
        id:'formulario',
```

```
        renderTo:'login',
```

```
        labelWidth: 75,
```

```
        title: 'Inicio de Sesión',
```

```
        bodyStyle:'padding:5px 5px 0',
```

```
        width: 350,
```

```
        defaultType: 'textfield',
```

```
        defaults: {
```

```
            listeners: {
```

```
                specialkey: function(field,e){
```

```
                    if (e.getKey() == e.ENTER) {
```

```
                        if ( formulario.form.isValid() ) {
```

```
                            formulario.form.submit({
```

```
                                url:'GestionCRM_valida.action',
```

```
                                waitMsg:'Validando...',
```

```
                                success: function(formulario, action) {
```

```
                                    parent.location='!/GestionCRM/GestionCRM_index.action';
```

```
    },
    failure: function(formulario, action) {

        Ext.MessageBox.show({
            title: 'Datos Incorrectos',
            msg: 'Favor de verificar los datos.',
            iconCls: 'x-status-error',// icon: Ext.MessageBox.ERROR ,
            buttons: Ext.MessageBox.OK

        });
    }
});
}
}
}
}
}
},
items: [{
    fieldLabel: 'Usuario',
    name: 'user',
    vtype: 'alpha',
    allowBlank:false

},{
    fieldLabel: 'Password',
    inputType: 'password',
    name: 'password',
    allowBlank:false
```

```
    }},
    buttons: [{
        text: 'Aceptar',
        handler: function(){
            if ( formulario.form.isValid() ) {
                formulario.form.submit({
                    url:'GestionCRM_valida.action',
                    waitMsg:'Validando...',
                    success: function(formulario, action) {
                        parent.location='/GestionCRM/GestionCRM_index.action';
                    },
                    failure: function(formulario, action) {

                        Ext.MessageBox.show({
                            title: 'Datos Incorrectos',
                            msg: 'Favor de verificar los datos.',
                            clear: true,
                            icon: Ext.MessageBox.ERROR ,
                            buttons: Ext.MessageBox.OK

                        });

                    }

                });
            }
        }
    }
    ],{
        text: 'Reset',
```

```
        handler: function(){
            formulario.getForm().reset();
        }
    }
]
});
});
```

```
Ext.BLANK_IMAGE_URL='ext/resources/images/default/s.gif';
Ext.Updater.defaults.loadScripts=true;
Ext.Updater.defaults.disableCaching=true;
```

```
Ext.onReady(function() {
    new Ext.Toolbar({
        renderTo: document.getElementById('contenedor_menu'),
        height: 50,
        items: [
            {
                xtype: 'tbseparator'
            },{
                height:45,
                width:70,
                xtype: 'tbbutton',
                text: 'Inicio',
                handler:function(){
                    Ext.get(tablas).load({
```

```
        url:'GestionCRM_contenido.action',
        scripts:true
    });
}
},
{
    xtype: 'tbseparator'
},{
    xtype:'splitbutton',
    height:45,
    width:70,
    text: 'Usuarios',
    iconCls: 'add16',
    menu: [
        {text: 'Listar',
            handler:function(){
                Ext.get(tablas).load({
                    url:'Usuarios_listar.action',
                    scripts:true
                });
            }
        }
    ]
},
{
    xtype: 'tbseparator',
    height:45
},{
```



```
xtype:'splitbutton',
height:45,
width:70,
text: 'Productos',
iconCls: 'add16',
menu: [
    {text: 'Listar',
        handler:function(){
            Ext.get(tablas).load({
                url:'Productos_listar.action',
                scripts:true
            });
        }
    },{text: 'Ordenes de Compra',
        handler:function(){
            Ext.get(tablas).load({
                url:'Ordenes_listar.action',
                scripts:true
            });
        }
    }
]
},
{
    xtype: 'tbseparator'
},{
    xtype:'splitbutton',
    height:45,
```

```
width:70,
text: 'Servicios',
iconCls: 'add16',
menu: [
    {text: 'Listar',
        handler:function(){
            Ext.get(tablas).load({
                url:'Servicios_listar.action',
                scripts:true
            });
        }
    ]
},
{
    xtype: 'tbseparator'
},{
    xtype:'splitbutton',
    height:45,
    width:70,
    text: 'Foro',
    iconCls: 'add16',
    menu: [
        {text: 'Listar',
            handler:function(){
                Ext.get(tablas).load({
                    url:'Temas_listar.action',
                    scripts:true
                });
            }
        ]
    }
}
```

```
        }
    }
]
},
{
    xtype: 'tbseparator'
},
{
    height:45,
    width:70,
    xtype: 'tbutton',
    text: 'Cerrar Sesión',
    handler: function(){
        window.location='http://localhost:8080/GestionCRM';
    }
}
]
});
});
```

## Resumen

El sistema de manejo de clientes, se planteó como un sistema de seguimiento que se ejecuta sobre ambiente Web, desarrollado a base de diferentes tecnologías con la finalidad de hacer posible la intercomunicación con clientes que puede atender una empresa.

De acuerdo a los objetivos planteados en el proyecto, de los cuales se desprende que es de suma importancia tener una base de datos la cual contenga la información detallada de los clientes así como un buen seguimiento de los productos y servicios que ofrece la empresa, lo anterior con la finalidad de facilitar el acceso y uso de la aplicación a los clientes que utilicen el servicio, ya sea una pequeña o mediana empresa.

## Bibliografía

<http://www.oracle.com/global/lad/applications/peoplesoft-enterprise.htm>  
[http://www.es.tectura.com/Page/cm389/Microsoft\\_dynamics\\_nav\\_navision\\_389.asp?d=1](http://www.es.tectura.com/Page/cm389/Microsoft_dynamics_nav_navision_389.asp?d=1)  
<http://www.gcg.com.mx/solomonpyme/solomonpyme1.html>  
[http://www.sap.com/argentina/solutions/businesssuite/crm/pdf/mySAP\\_CRM\\_Analytics.pdf](http://www.sap.com/argentina/solutions/businesssuite/crm/pdf/mySAP_CRM_Analytics.pdf)  
<http://www.microsoft.com/spain/businesssolutions/dynamics/productos/crm/demos/default.msp>  
[http://iteso.mx/~carlosc/pagina/cursoUAHI/UAHI\\_99\\_1/proyecto\\_inf.htm](http://iteso.mx/~carlosc/pagina/cursoUAHI/UAHI_99_1/proyecto_inf.htm)  
<http://www.gestiopolis.com/canales5/emp/imposiste.htm>  
<http://www.monografias.com/trabajos14/estrategiacrm/estrategiacrm.shtml>  
[http://jasperforge.org/website/jasperreportswebsite/trunk/index.html?group\\_id=252](http://jasperforge.org/website/jasperreportswebsite/trunk/index.html?group_id=252)  
<http://dev.sencha.com/deploy/dev/examples/layout-browser/layout-browser.html>

## Referencias

Hibernate

<http://www.onjava.com/pub/a/onjava/2005/09/21/what-is-hibernate.html>

Spring

<http://www.springsource.org/spring-framework>

Struts

<http://www.onjava.com/pub/a/onjava/2005/11/02/what-is-struts.html>

Apache Tomcat

<http://www.omii.ac.uk/docs/2.3.1/reference/tomcat/what-is-tomcat.htm>

PostgreSQL

<http://en.wikipedia.org/wiki/PostgreSQL>

ExtJS

<http://en.wikipedia.org/wiki/ExtJS>

JavaScript

<http://javascript.about.com/od/reference/p/javascript.htm>

JSP

<http://en.wikipedia.org/wiki/JavaServerPages>