

**Desarrollo de aplicaciones empresariales**

**Ing. Sistemas computacionales**

**Documentación por módulos**

**PRESENTA:**

**Isaac Perez Segura.**

**Cynthia Aideé Álvarez Castro.**

**Juan de Dios Gutiérrez Ramírez.**

**NOMBRE DEL MAESTRA:**

**Lic. Angélica María Ortiz Gaucín**

**LEÓN, GUANAJUATO Periodo: Enero – junio 2019**

# Introducción

MercaTec es un sistema web encargado de permitir vender productos a la comunidad ITL (en un futuro a cualquier universidad). Los módulos que lo integran son los siguientes:

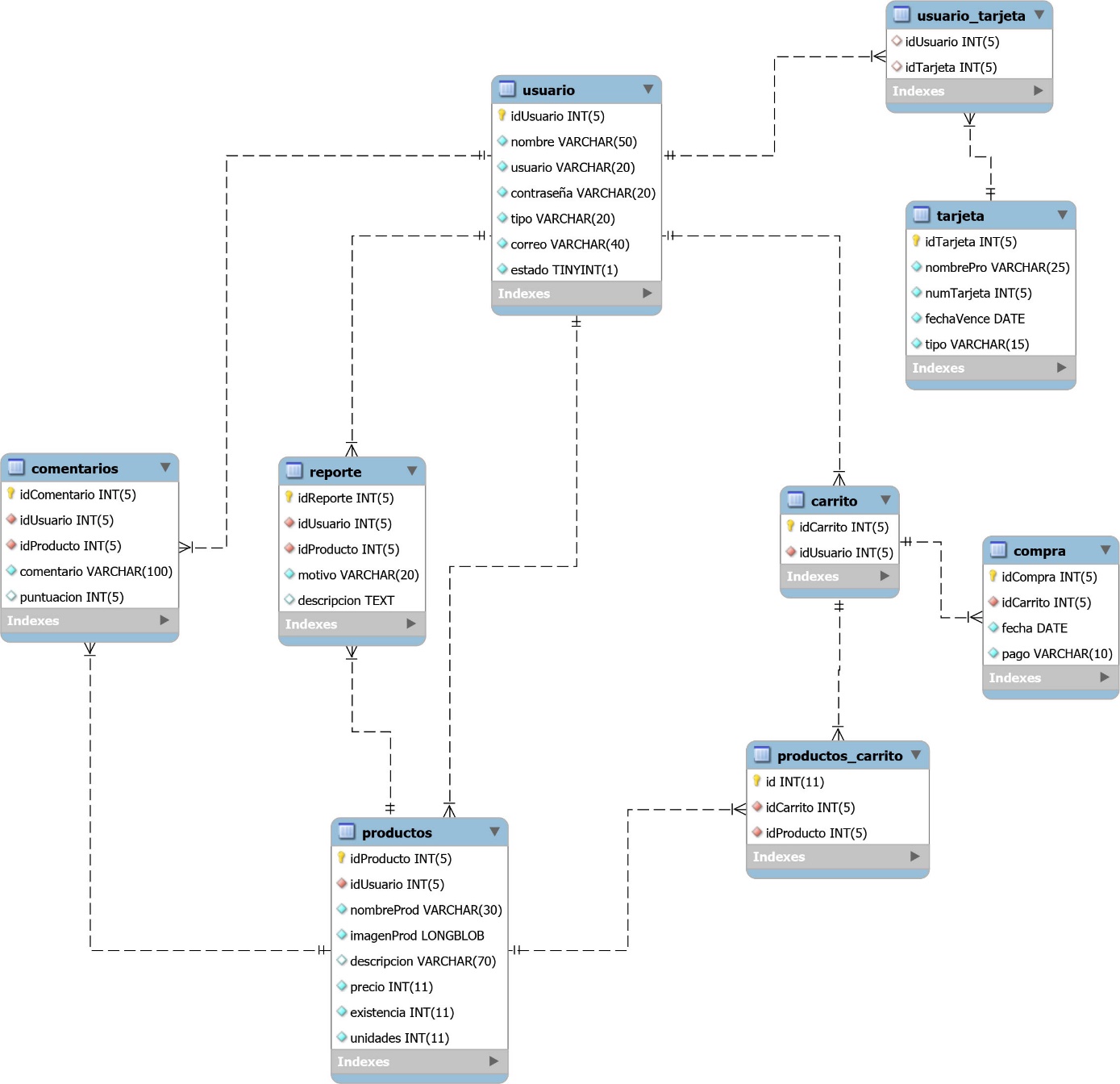
* Login
* Página de inicio
* Carrito
* Barra de búsqueda
* Publicar artículos
* Modificar y eliminar artículos
* Comentarios
* Reportes
* Alta y baja de usuarios
* Ver reportes
* Eliminar artículos

## Roles

|  |  |  |
| --- | --- | --- |
| **Modulo** | **Usuario** | **Descripción** |
| Publicación y eliminación de artículos | Administrador**,** usuario | Permitirá al usuario publicar el articulo deseado a vender y el usuario puede eliminar dicha publicación, el administrador solo podrá eliminarlo. |
| Alta/baja usuarios | Administrador | Permite registrar en el sistema a los usuarios, otorgando un usuario y contraseña. |
| Reportes | Usuario, administrador | Permite realizar un reporte sobre artículos inadecuados y falsos, que a la vez notifican al administrador. |
| Carrito de compras | Usuario | Permite visualizar los productos deseados a comprar hasta el momento, así como la eliminación de los mismos, mostrando un monto total. |

# Base de datos

El siguiente diagrama entidad relación muestra como esta estructura la base de datos:



## Script

**create** database mercatec**;**

**use** mercatec**;**

**CREATE** **TABLE** Usuario **(**

idUsuario **INT(**5**)** **PRIMARY** **KEY** **NOT** **NULL** AUTO\_INCREMENT**,**

nombre **VARCHAR(**50**)** **NOT** **NULL,**

usuario **VARCHAR(**20**)** **NOT** **NULL,**

contraseña **VARCHAR(**20**)** **NOT** **NULL,**

tipo **VARCHAR(**20**)** **NOT** **NULL,** -- Si es profesor, alumno...

correo **VARCHAR(**40**)** **NOT** **NULL,**

estado **TINYINT(**1**)** **NOT** **NULL** -- Indica si está activa o inactiva la cuenta.

**);**

**INSERT** **INTO** Usuario**(**idUsuario**,** nombre**,** usuario**,** contraseña**,** tipo**,** correo**,** estado**)** **VALUES**

**(NULL,** 'Aideé Alvarez'**,** 'Aidee'**,** '1234'**,** 'Alumno'**,** '15240528@itleon.edu.mx'**,** 1**);**

**INSERT** **INTO** Usuario**(**idUsuario**,** nombre**,** usuario**,** contraseña**,** tipo**,** correo**,** estado**)** **VALUES**

**(NULL,** 'Yahir Saldivar'**,** 'Admin'**,** '1234'**,** 'Administrador'**,** '15240528@itleon.edu.mx'**,** 1**);**

**INSERT** **INTO** Carrito**(**idCarrito**,**idUsuario**)** **VALUES** **(NULL,** 1**);**

**INSERT** **INTO** Carrito**(**idCarrito**,**idUsuario**)** **VALUES** **(NULL,** 2**);**

**CREATE** **TABLE** Tarjeta**(**

idTarjeta **INT(**5**)** **PRIMARY** **KEY** **NOT** **NULL** AUTO\_INCREMENT**,**

nombrePro **VARCHAR(**25**)** **NOT** **NULL,**

numTarjeta **INT(**5**)** **NOT** **NULL,**

fechaVence **DATE** **NOT** **NULL,**

tipo **VARCHAR(**15**)** **NOT** **NULL** -- Crédito o débito.

**);**

**CREATE** **TABLE** Productos**(**

idProducto **INT(**5**)** **PRIMARY** **KEY** **NOT** **NULL** AUTO\_INCREMENT**,**

idUsuario **INT(**5**)** **NOT** **NULL,** -- Id del usuario que desea vender un producto.

nombreProd **VARCHAR(**30**)** **NOT** **NULL,**

imagenProd LONGBLOB **NOT** **NULL,** -- imagen del producto guardado como un blob.

descripcion **VARCHAR(**70**),**

precio **INT** **NOT** **NULL,**

existencia **INT** **NOT** **NULL,** -- Existencia de las unidades actuales.

unidades **INT** **NOT** **NULL** -- Unidades iniciales del producto.

**);**

**CREATE** **TABLE** Reporte**(**

idReporte **INT(**5**)** **PRIMARY** **KEY** **NOT** **NULL** AUTO\_INCREMENT**,**

idUsuario **INT(**5**)** **NOT** **NULL,**

idProducto **INT(**5**)** **NOT** **NULL,**

motivo **VARCHAR(**20**)** **NOT** **NULL,**

descripcion TEXT**(**500**)**

**);**

**CREATE** **TABLE** Comentarios**(**

idComentario **INT(**5**)** **PRIMARY** **KEY** **NOT** **NULL** AUTO\_INCREMENT**,**

idUsuario **INT(**5**)** **NOT** **NULL,**

idProducto **INT(**5**)** **NOT** **NULL,**

comentario **VARCHAR(**100**)** **NOT** **NULL,**

puntuacion **INT(**5**)**

**);**

**CREATE** **TABLE** Carrito**(**

idCarrito **INT(**5**)** **PRIMARY** **KEY** **NOT** **NULL** AUTO\_INCREMENT**,**

idUsuario **INT(**5**)** **NOT** **NULL**

**);**

**CREATE** **TABLE** Productos\_Carrito**(**

id **int** **primary** **key** **not** **null** auto\_increment**,**

idCarrito **INT(**5**)** **NOT** **NULL,**

idProducto **INT(**5**)** **NOT** **NULL**

**);**

**CREATE** **TABLE** Compra**(**

idCompra **INT(**5**)** **NOT** **NULL** **PRIMARY** **KEY** AUTO\_INCREMENT**,**

idCarrito **INT(**5**)** **NOT** **NULL,**

fecha **DATE** **NOT** **NULL,**

pago **VARCHAR(**10**)** **NOT** **NULL**

**);**

**CREATE** **TABLE** usuario\_tarjeta **(**

idUsuario **INT(**5**),**

idTarjeta **INT(**5**)**

**);**

/\* CLAVES FORÁNEAS USUARIO\_TARJETA\*/

**ALTER** **TABLE** usuario\_tarjeta **add** **foreign** **key** **(**idUsuario**)** **references** Usuario**(**idUsuario**);**

**ALTER** **TABLE** usuario\_tarjeta **add** **foreign** **key** **(**idTarjeta**)** **references** Tarjeta**(**idTarjeta**);**

/\* CLAVES FORÁNEAS REPORTE\*/

**ALTER** **TABLE** Reporte **add** **foreign** **key** **(**idUsuario**)** **references** Usuario**(**idUsuario**);**

**ALTER** **TABLE** Reporte **add** **foreign** **key** **(**idProducto**)** **references** Productos**(**idProducto**);**

/\* CLAVES FORÁNEAS COMENTARIOS\*/

**ALTER** **TABLE** Comentarios **add** **foreign** **key** **(**idUsuario**)** **references** Usuario**(**idUsuario**);**

**ALTER** **TABLE** Comentarios **add** **foreign** **key** **(**idProducto**)** **references** Productos**(**idProducto**);**

/\* CLAVE FORÁNEA DE CARRITO y PRODUCTOS\_CARRITO\*/

**ALTER** **TABLE** Carrito **add** **foreign** **key** **(**idUsuario**)** **references** Usuario**(**idUsuario**);**

**ALTER** **TABLE** Productos\_Carrito **add** **foreign** **key** **(**idCarrito**)** **references** Carrito**(**idCarrito**);**

**ALTER** **TABLE** Productos\_Carrito **add** **foreign** **key** **(**idProducto**)** **references** Productos**(**idProducto**);**

/\* CLAVES FORÁNEAS DE COMPRA \*/

**ALTER** **TABLE** Compra **add** **foreign** **key** **(**idCarrito**)** **references** Carrito**(**idCarrito**);**

/\* CLAVE FORÁNEA A PRODUCTOS\*/

**ALTER** **TABLE** Productos **add** **foreign** **key** **(**idUsuario**)** **references** Usuario**(**idUsuario**);**

**CREATE** **PROCEDURE** getNextIDUser**()**

**SELECT** AUTO\_INCREMENT

**FROM** information\_schema**.**TABLES

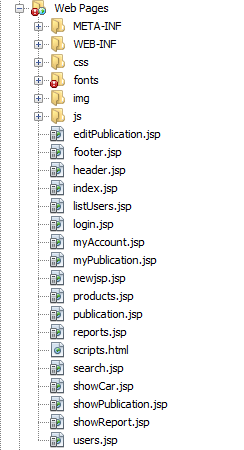
**WHERE** TABLE\_SCHEMA **=** "Mercatec"

**AND** TABLE\_NAME **=** "Usuarios"**;**

# Estructura del sistema

El sistema cuenta con el patrón de diseño MVC, su estructura general es la siguiente:

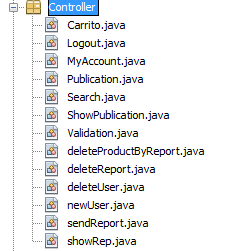
### Vistas



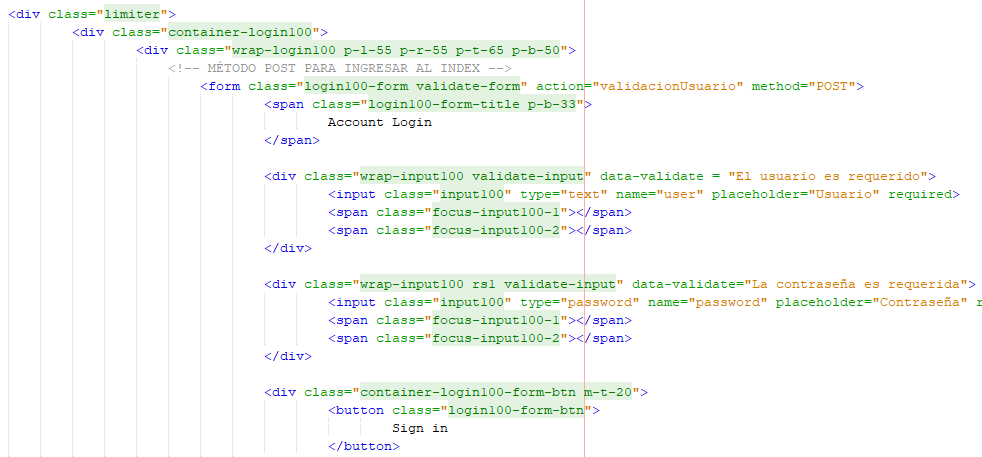
### Modelos

### 

### Controladores



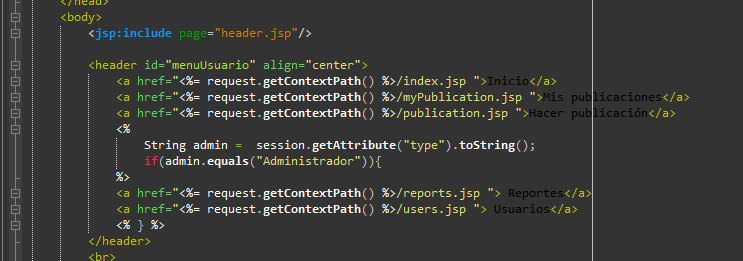
**Funcionalidad usuario.**



En la primea vista funcional del proyecto tenemos el login, es el formulario de validación para la verificación de los tipos de usuarios, utiliza consultas a la base de datos para la validación de los tipos de usuario (administrador y usuario regular) como también hace la confirmación de si el nickname y la contraseña están en la base de datos

**Funcionalidad Administrador.**

A pesar de que el administrador también es un usuario, este tiene una diferencia en la parte de la cabecera de “my account”, debido a que el tiene más prioridades de modificar y visualizar páginas que otros usuarios no pueden.

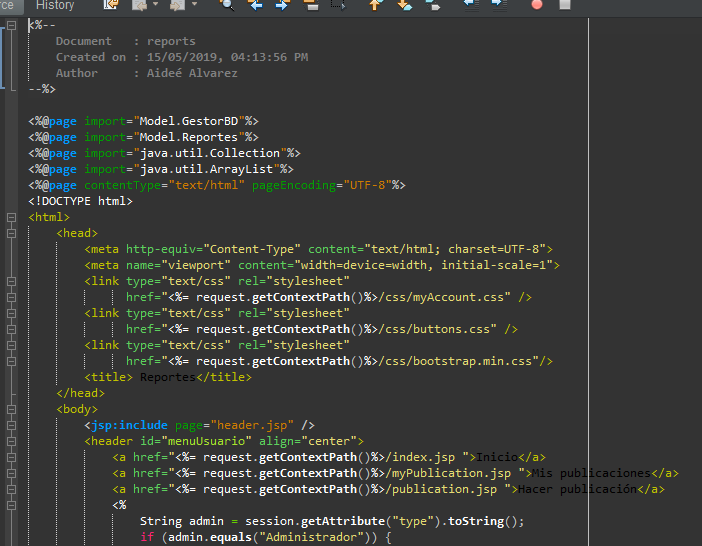


*Cabecera del header, sección My Account.*

Como podemos observar, tenemos un “if”, que nos ayudará a saber qué tipo de usuario es el que está entrando.

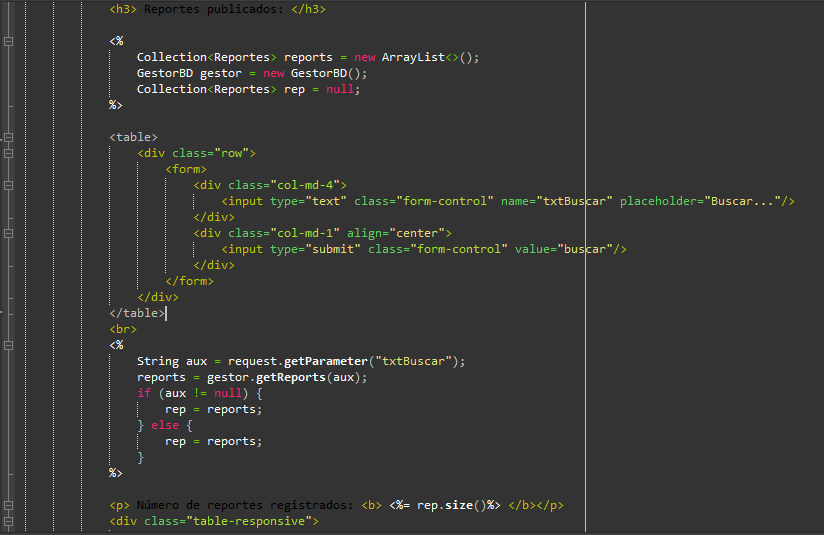
El administrador puede ver otras dos páginas: Reportes y usuarios.

**Reportes.**



*Cuerpo, sección Reportes.*

Tenemos la primera sección, que es el cuerpo del documento de usuarios. El administrador es el único que puede ingresar y eliminar a cada usuario del MercaTec.



*Cuerpo, sección Reportes, parte 2.*

En esta segunda sección se hace una colección de reportes, ya que, con nuestro gestor de la base de datos, podemos traer la información de todos los reportes registrados por los usuarios. También, se crea un buscador dinámico a la tabla de reportes.



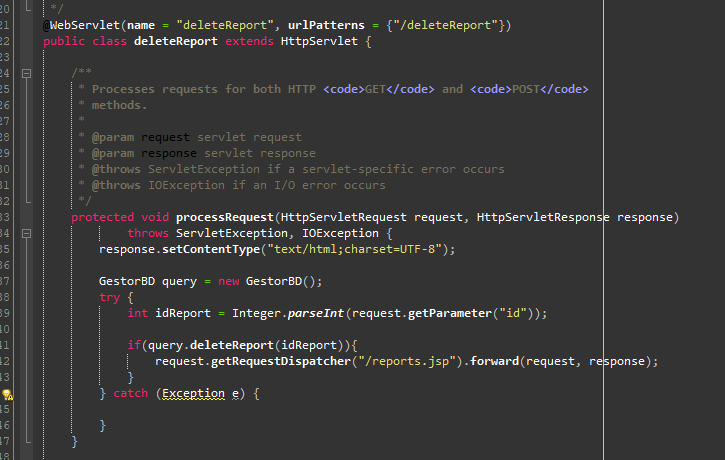
*Cuerpo, sección reportes, parte 3.*

En la sección 3, junto a las demás secciones, mostramos los datos de los reportes como tablas. Esta incluye un botón de ver los reportes, que nos mandaría a una página llamada “ShowPublication.jsp”, y como tal, eliminar reportes individuales.

Debido a que queríamos evitar la parte de duplicar páginas, reutilizamos la que se mencionó anteriormente, para no sobrecargar al servidor.

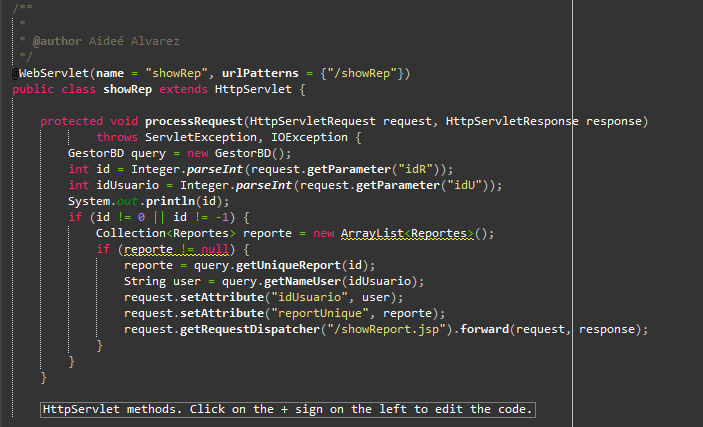
SERVLETS UTILIZADOS EN REPORTES.

Para darle funcionamiento a esto, nuestros controladores son los siguientes:



*Servlet - borrar reportes.*

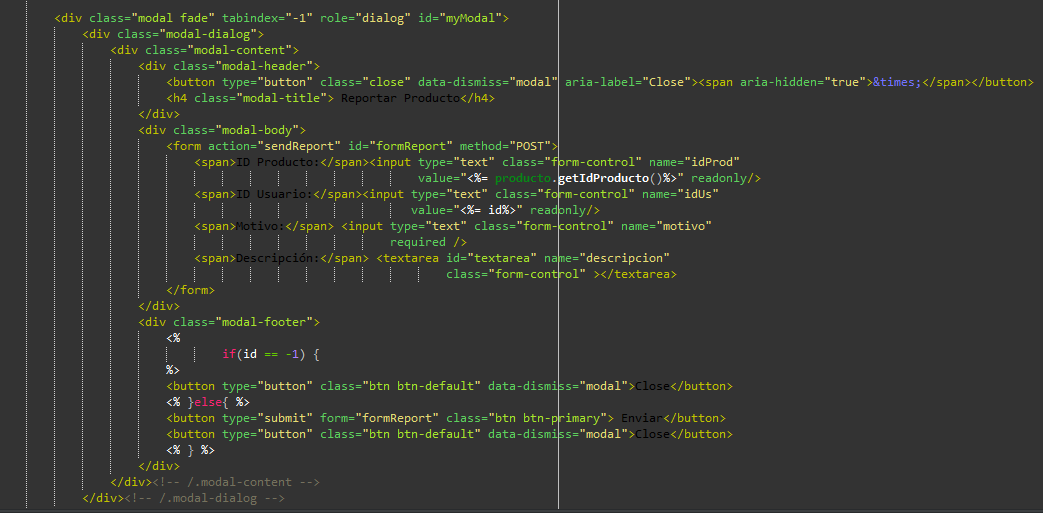
De acuerdo a la sección 3 de reports.jsp, nos manda un hipervínculo que sería el controlador que se muestra arriba. Solamente obtenemos el ID correspondiente y se borra,



*Servlet – Mostrar reporte.*

También el administrador puede ver los reportes de una manera detallada (ver sección 3 de reportes) gracias a otro hipervínculo que está conectado al servlet de arriba. Aquí consta de traer dos ID: Reporte (para saber qué reporte es el que se verá) y el de usuario (Saber qué usuario es el que reporta tal producto).

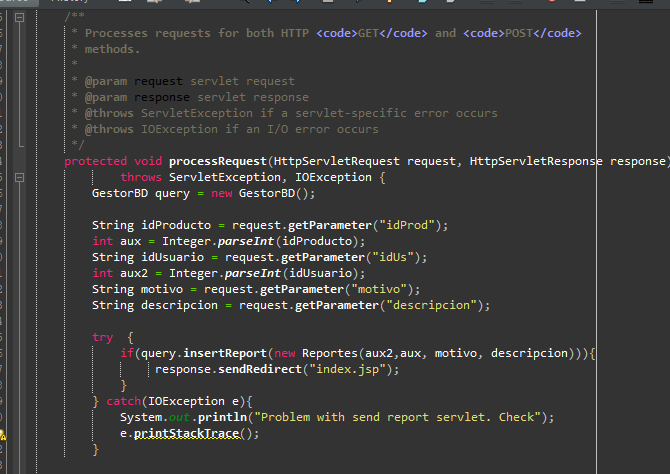
Dejando un lado el administrador, el usuario puede llegar a ser reportes, ya sea por cualquier motivo que no haya sido de su agrado. Para ello, el usuario puede ingresar reportes gracias a esta página:



*Reportar un producto, sección de Show publication.*

Cuando el usuario da clic al botón, se le abrirá un modal, que es una especie de div flotante en la misma página (controlada por post) donde tendrá que registrar ciertos datos y poder enviarlos al administrador.

Su servlet es el siguiente:

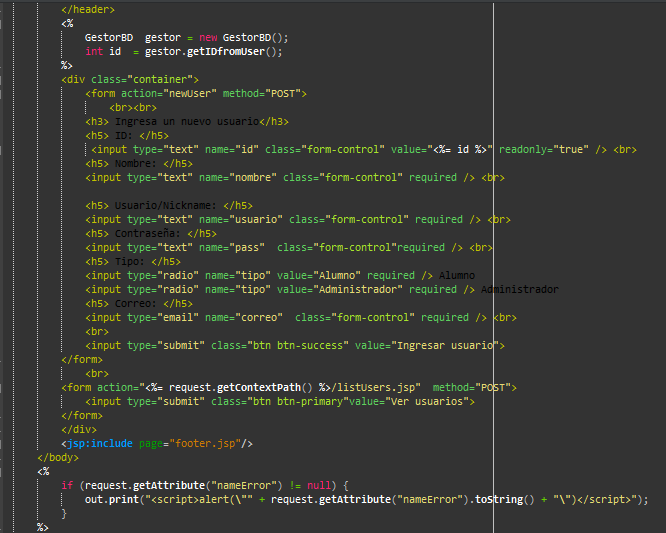


*Servlet – sendReport.*

Como tal, todos los parámetros se obtienen gracias al request y se mandan directamente al gestor de la base de datos. Si este se completa, se redirecciona al index, aunque el usuario libremente puede regresar al producto si lo desea.

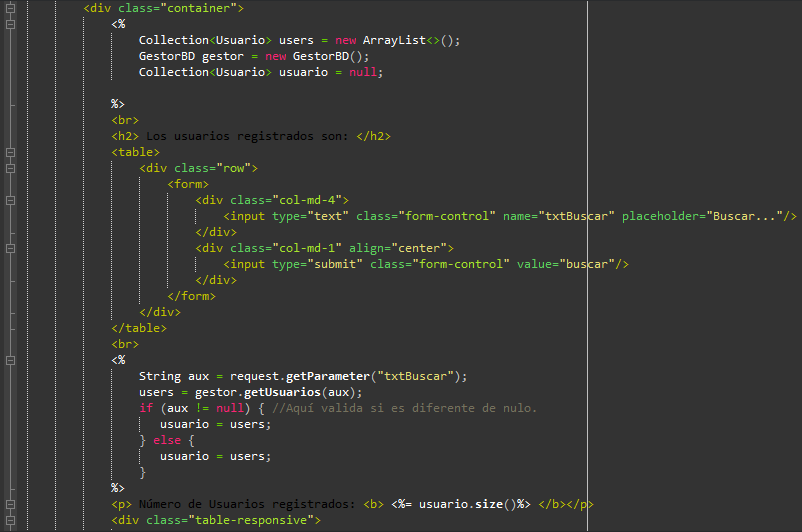
**Usuarios.**

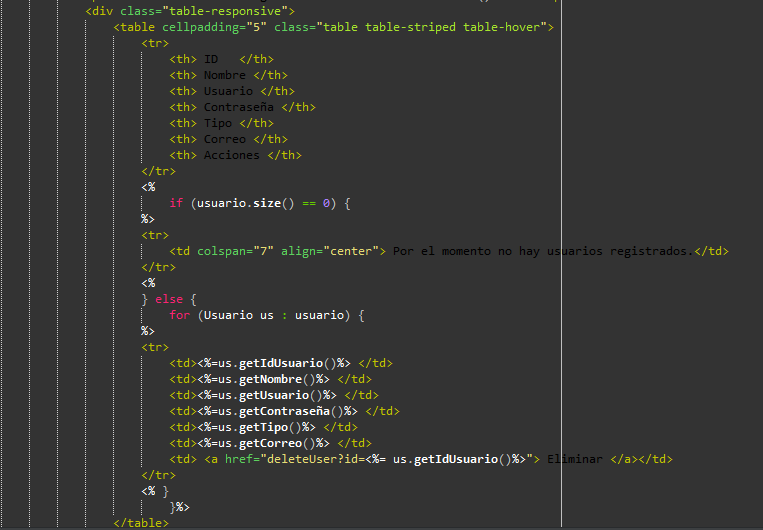
Como tal, el administrador también puede ver y agregar usuarios. Sus páginas son las siguientes:



*Página - users.jsp*

Como podemos observar, se ingresan algunos inputs text para poder hacer el registro de un usuario (formulario) y este nos redirecciona por medio de un form action a un servlet llamado “newUser”.

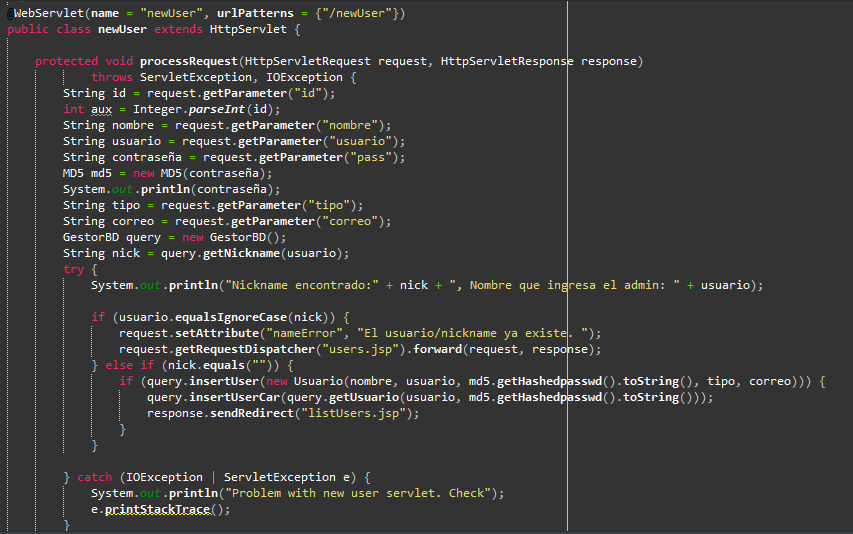




*Página – listUsers.jsp*

Esta página muestra, así como en reportes, todos los usuarios registrados en la página de MercaTec donde tenemos un hipervínculo para poder eliminar a un usuario en particular.

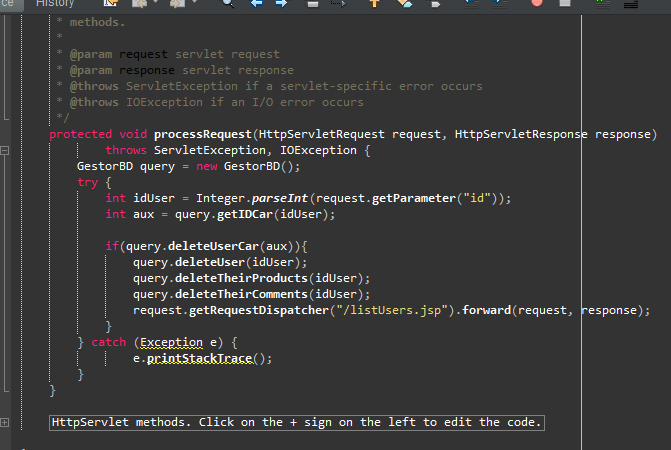
Los servlets correspondientes a las dos páginas anteriores, son los siguientes:



*Servlet – new user.*

Como tal, este servlet nos ayudará a ingresar a un usuario. Su funcionamiento es sencillo. Primero vamos a buscar si el nickname no se repite, debido a que el administrador por las veces que ha metido a usuarios no recuerde todos. Ese nickname es con el que una persona ingresará al sistema, y si el administrador no se acuerda, se le manda una alerta de que ya existe ese nickname. Lo demás es obtener todo por medio de parámetros get e insertarlos a la base de datos del sistema.

Como todo usuario puede comprar, en esta sección se le añade desde un principio a un usuario su propio carrito.



*Servlet – delete user.*

Y para finalizar la parte del administrador, este servlet nos ayudará a borrar a un usuario.

Debido a que la mayoría de las tablas tienen relación con usuario y es un campo requerido, primero se debe borrar todo lo que puede hacer, como: sus productos, comentarios hechos a los productos y su carrito.

**Métodos utilizados por el administrador en el gestor de la base de datos.**

A continuación, se muestran todos los métodos que implementa el administrador y que en si son sentencias SQL para poder hacer inserciones, modificaciones y eliminaciones en la base de datos.

