Trabajo Final Java

Sitio web: Fevemplas

En el presente informe se describe cómo se realizó y en qué consiste el sitio web Fevemplas

2013

Integrantes: Forchino, Micaela – Laitano, Cecilia

Phren Soft

16/07/2013

Contenido

[Introducción 2](#_Toc363718865)

[Descripción del Negocio 3](#_Toc363718866)

[Funcionalidades 4](#_Toc363718867)

[Tecnologías utilizadas para el desarrollo 5](#_Toc363718868)

[Programas 5](#_Toc363718869)

[Librerías 5](#_Toc363718870)

[Lenguajes 6](#_Toc363718871)

[Diagrama de Clases 9](#_Toc363718872)

[Casos de Uso 10](#_Toc363718873)

[C.U.1: Logueo de usuarios. 10](#_Toc363718874)

[C.U.2: ABM de Productos. 12](#_Toc363718875)

[C.U.2.1: Modificar Producto. 14](#_Toc363718876)

[C.U.2.2: Eliminar Producto. 16](#_Toc363718877)

[C.U.2.3: Alta Producto. 17](#_Toc363718878)

[Código 19](#_Toc363718879)

[Conexión a la BD 19](#_Toc363718880)

[ABM 20](#_Toc363718881)

[Login 23](#_Toc363718882)

# Introducción

Como examen final de la materia Java se desarrolló un sitio web para una empresa de productos plásticos, Fevemplas, con la cual tenemos contacto directo, lo que nos facilitó la obtención completa de los requisitos del usuario.

En este informe se explicarán las funcionalidades del mismo y las distintas herramientas utilizadas para llevarlo a cabo.

A través del siguiente link usted puede visitar el sitio [www.fevemplas.com.ar](http://www.fevemplas.com.ar) .

# Descripción del Negocio

FEVEMPLAS  es una joven empresa nacida en el año 2000; se inserta dentro de la Provincia  
de Santa Fe, en la ciudad de Las Parejas, uno de los polos productivos más importantes de  
nuestra República.

Fabricamos recipientes multiformes en polietileno de baja y alta densidad, elaborados   
bajo el sistema de moldeo rotacional.

Contamos con equipos y tecnología de última generación, operarios formados y permanente  
stock de materia prima, colocándonos a la vanguardia en el campo de la industria termoplástica.

# Funcionalidades

La aplicación web tiene como objetivo informar sobre la actividad de la empresa, su ubicación y mostrar los productos que comercializa. Para esto categorizamos a los usuarios en tres niveles, cada uno con distintos permisos:

- Nivel No Loguedo: cualquier persona que visite la web podrá acceder a: la página inicial (home.jsp), a la descripción de la empresa(empresa.jsp), a la ubicación (ubicación.jsp), enviar una consulta a través de un formulario(consulta.html) y   ver todos los productos con sus características, además se incluye un buscador interno que le permite al usuario buscar aquello que desee.

-Nivel Logueado: a este nivel sólo podrán acceder los clientes de la empresa a través del formulario de login para autenticarse.

Luego de loguearse podrá configurar sus datos personales y cerrar sesión.

A todas las funcionalidades del nivel No Logueado se agrega la posibilidad de ver los precios de los productos a través de un link visible solo para ellos.

Como regla de negocio se nos solicitó que sólo puedan acceder los clientes que el administrador ha dado de alta, sin existir la posibilidad de registro.

Si un usuario desea registrarse le deberá solicitar al administrador su alta a través del formulario de contacto(contacto.html).

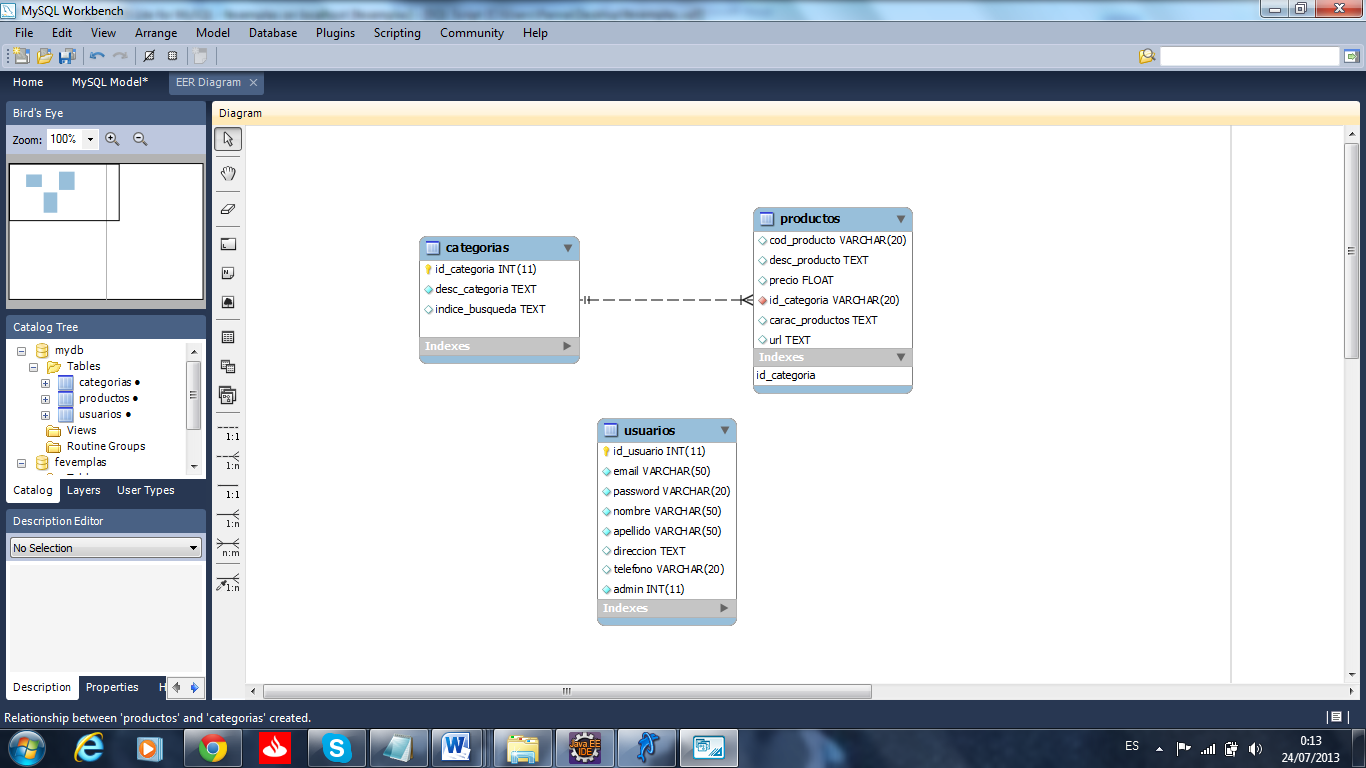
-Nivel Administrador: cuando un usuario se identifique como administrador, además de todas las funcionalidades descritas anteriormente, podrá realizar: Alta, Baja y Modificación de los clientes, productos y categorías de los mismos.

# Tecnologías utilizadas para el desarrollo

## Programas

**Base de datos:** MySql, ya que es fuertemente recomendad para la programación web, y a pesar de ser gratuita es muy estable y práctica para trabajar.

En ella se definieron tres tablas, una para cada entidad: Productos, Categorías y Usuarios. A continuación se muestra un diagrama de la misma:



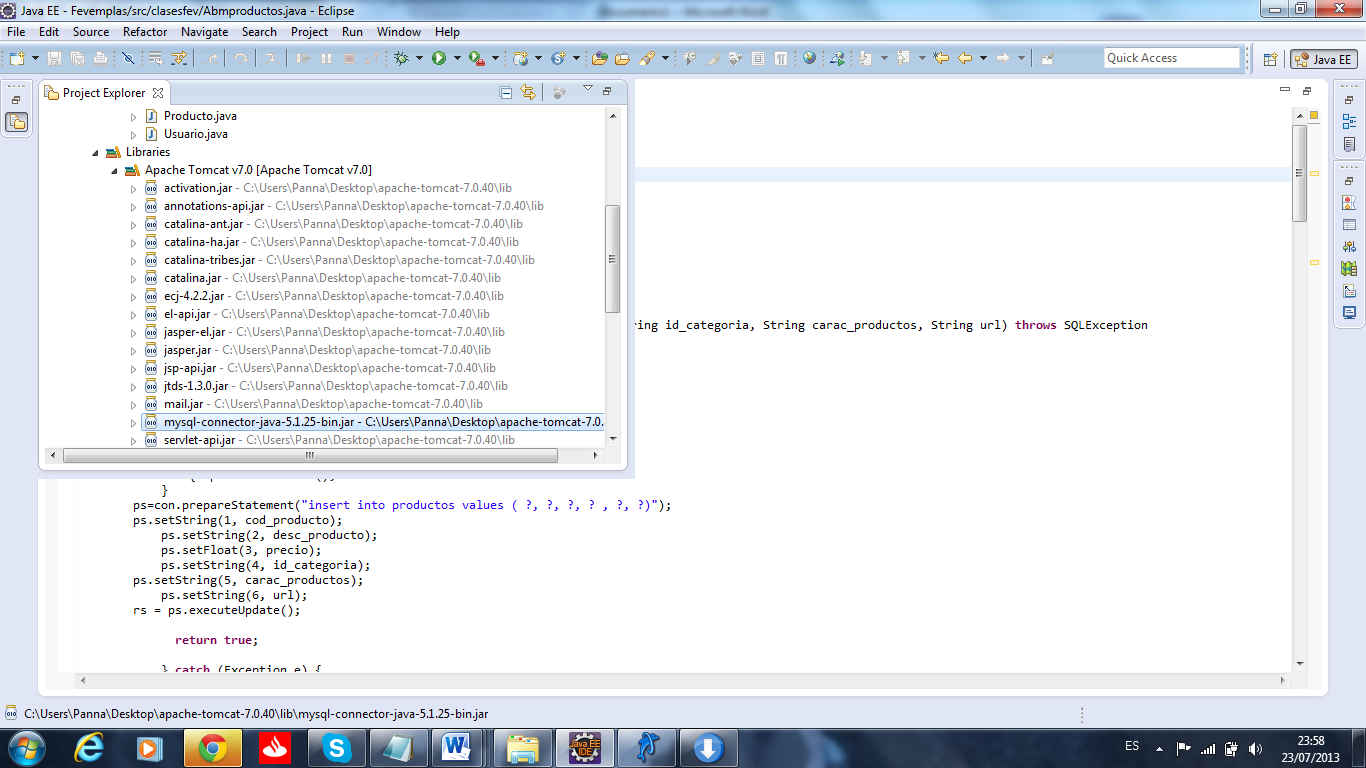
**Entorno de desarrollo:** Eclipse IDE for Java EE Developers, el cual nos permitió crear nuestro proyecto como sitio web, escribir las clases java, que luego serán interpretadas y compiladas, nos permitió conectarlo con el servidor local Apache Tomcat para ir realizando pruebas de desarrollo, y donde desarrollamos también las .jsp.

**Servidor Local**: Apache Tomcat v.07

Dreamweaver

## Librerías

El lenguaje Java nos permite establecer una conexión con MySQL con llamadas simples. Para crear la conexión entre la BD y las clases java vamos a necesitar un conector que ofrece MySql, el cual se dispuso en la carpeta de las librerías como se muestra a continuación.



## Lenguajes

1. Clases Java (.java)

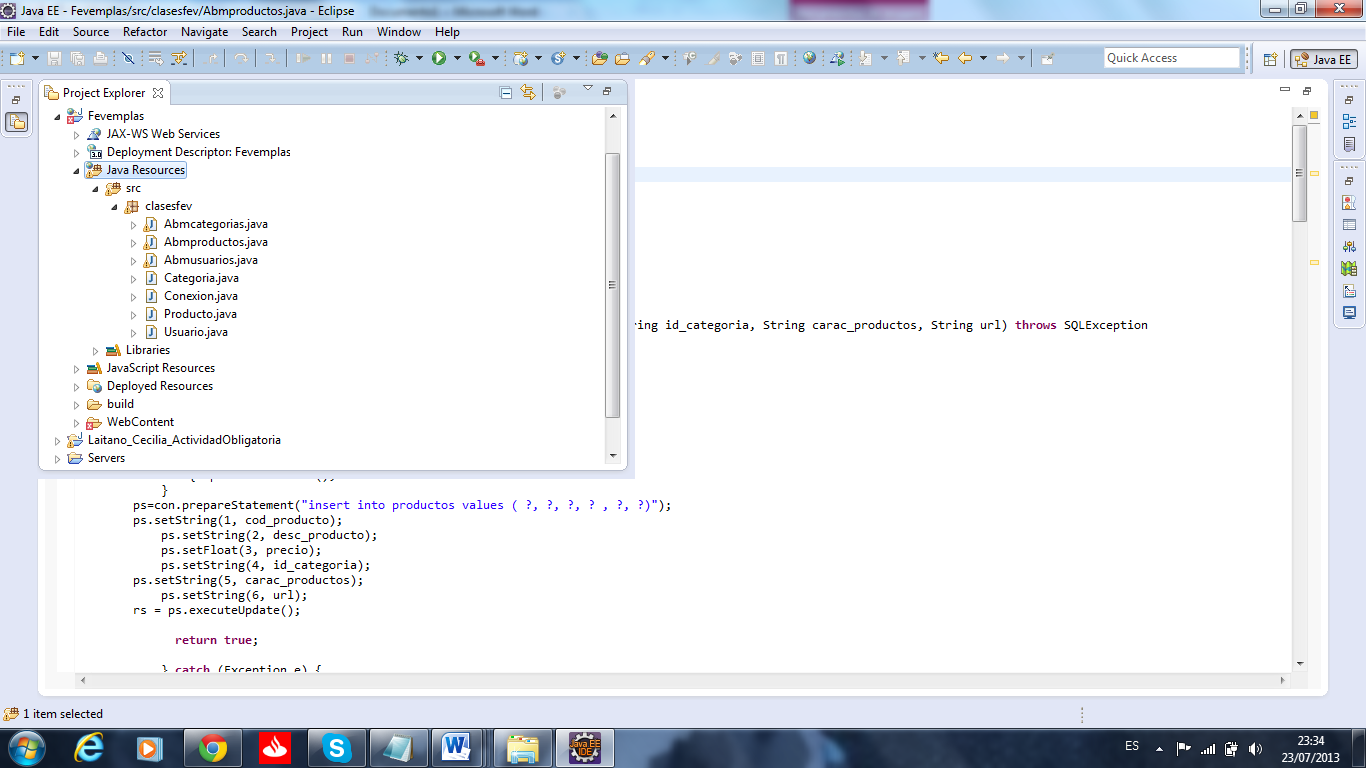
El sitio maneja tres tipos de entidades: Usuarios, Productos y Categorías.

Como permite el ABM de éstos tres (en sesión como Administrador), se utilizó una clase para cada uno, en las cuales se definen los métodos para realizar estas operaciones – Alta- Baja – Modificación. A su vez dentro del ABMUsuarios se dispuso un método doLogin para validar el inicio de sesión.

Estas clases requieren conectarse a la BD de MySQL para operar, por lo que se escribió una clase “Conexión.java” que lo realice, dándonos la ventaja de poder reutilizar código y mantenerla separada por si requiriera en algún momento una modificación.

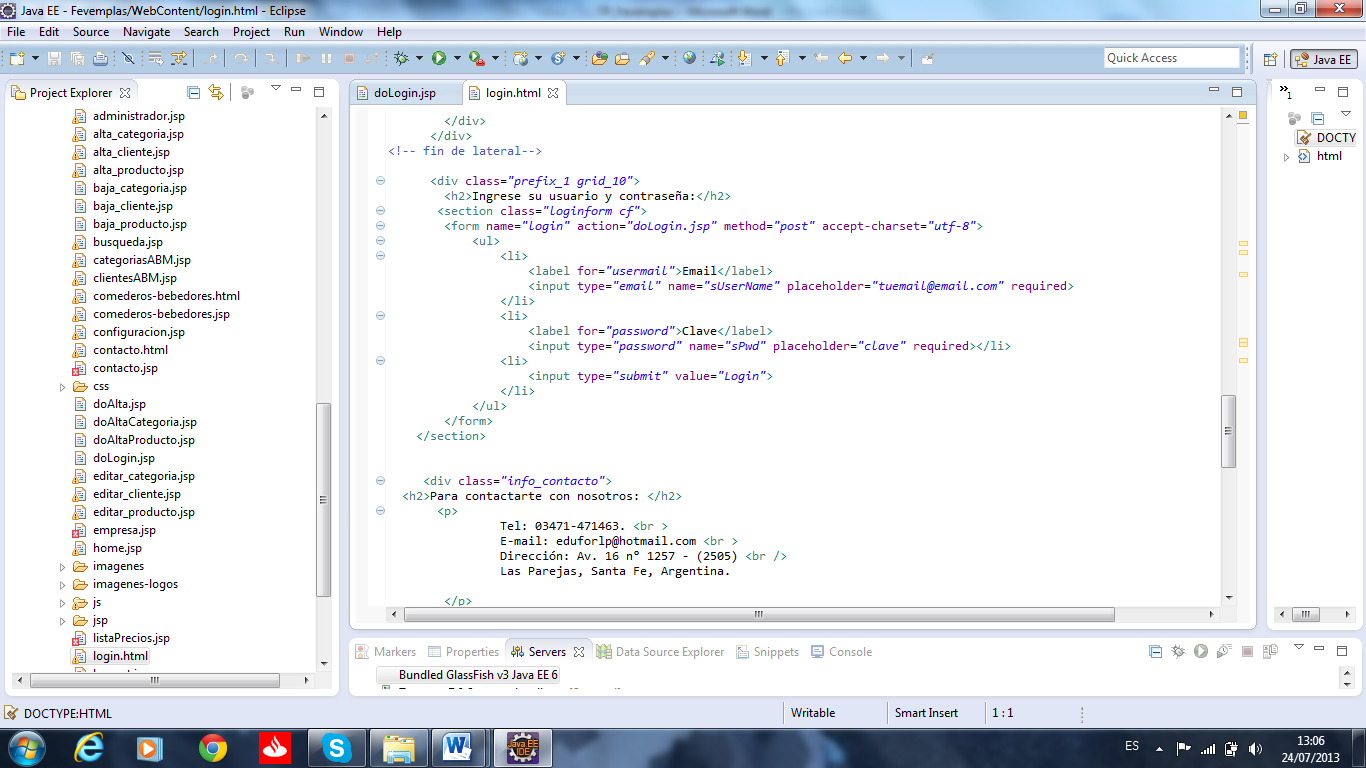
A su vez, se cuenta con una clase para cada entidad, que contiene las propiedades de los datos de cada una (utilizando métodos gets() y sets()), que serán usados por las clases ABM.

En total son 7 clases, las cuales se colocaron dentro de un package “clasesfev” el cuál será llamado para la ejecución de las .jsp que lo requieran.



1. Java Server Pages - JSP (.jsp)
2. HTML

Para realizar la parte visual de nuestro sitio web utilizamos lenguaje HTML como el fragmento de código que se muestra a continuación:



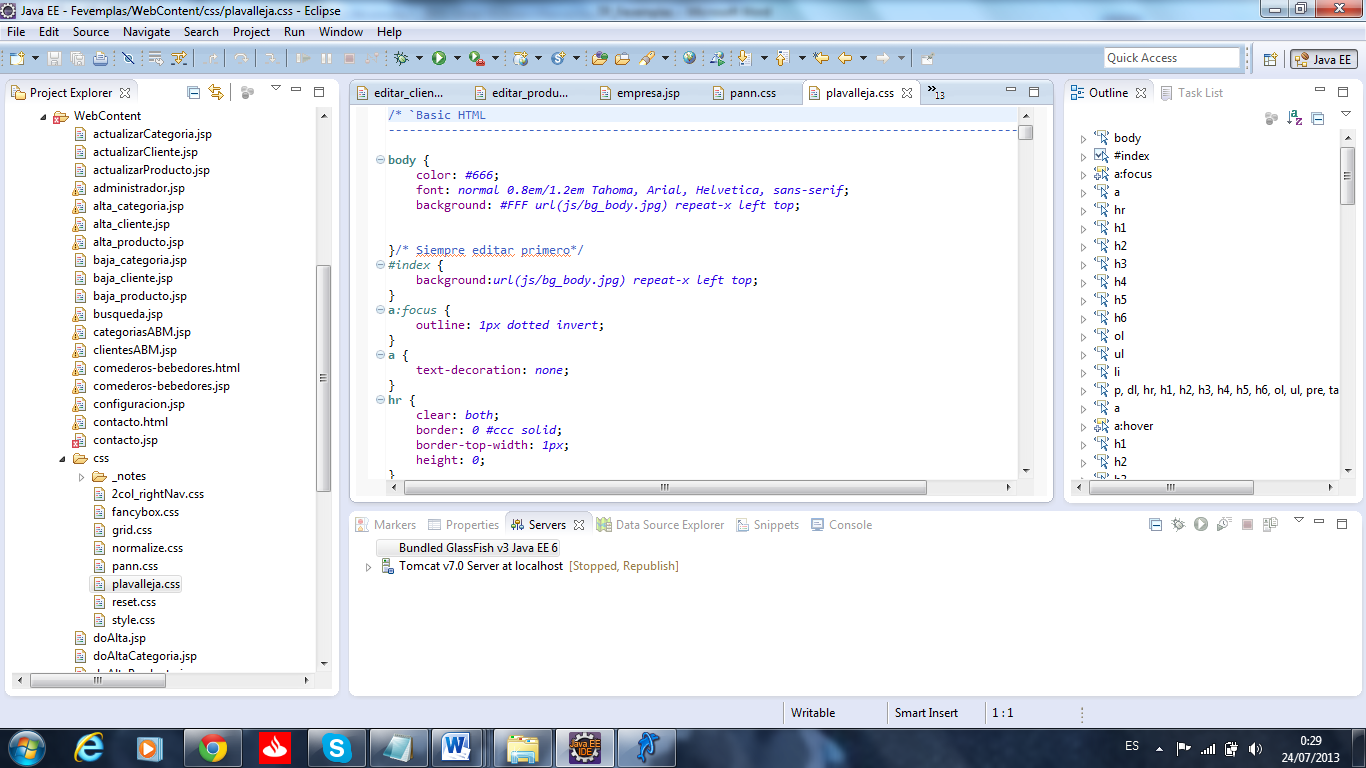
1. Scripts de JavaScripts

Para poder utilizar capas se utiliza en la masterpage el siguiente script:

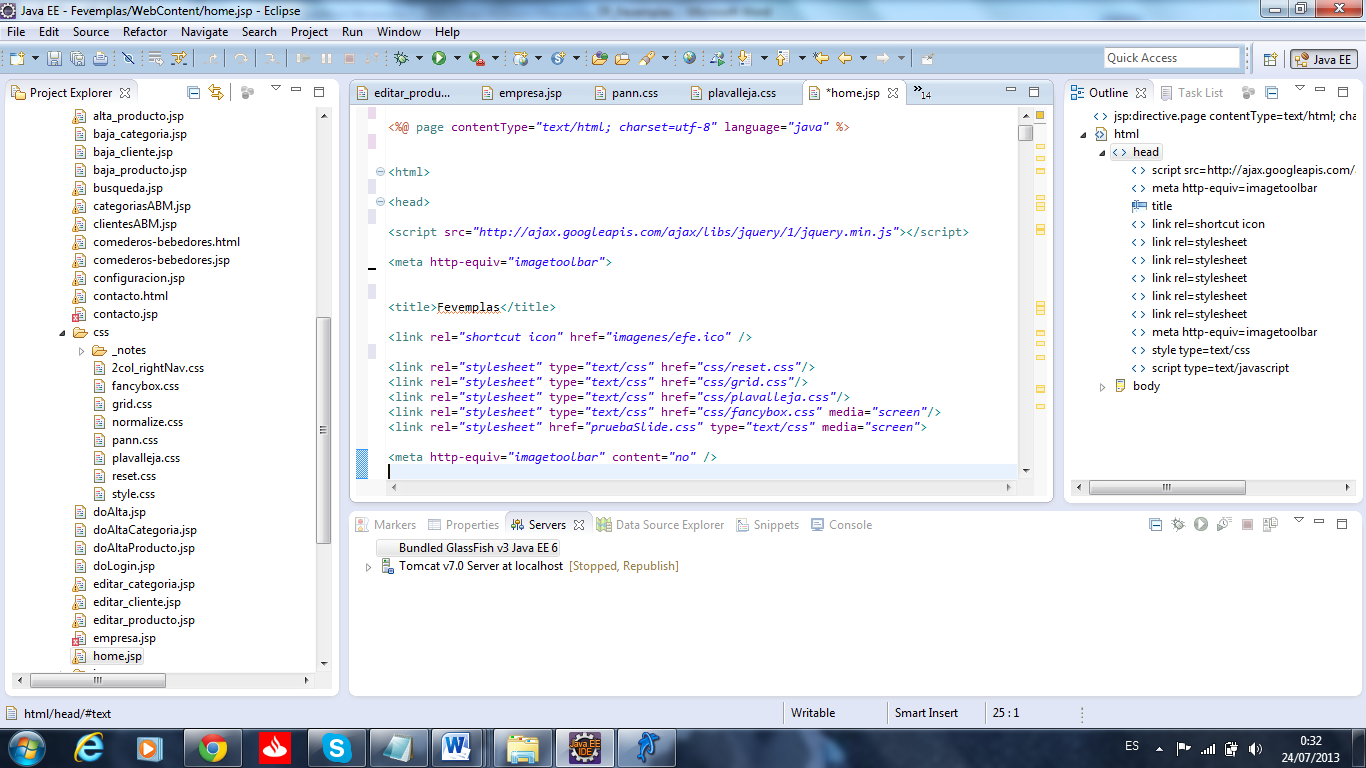
calidar email funciones en js

1. Hojas de estilo (.css)

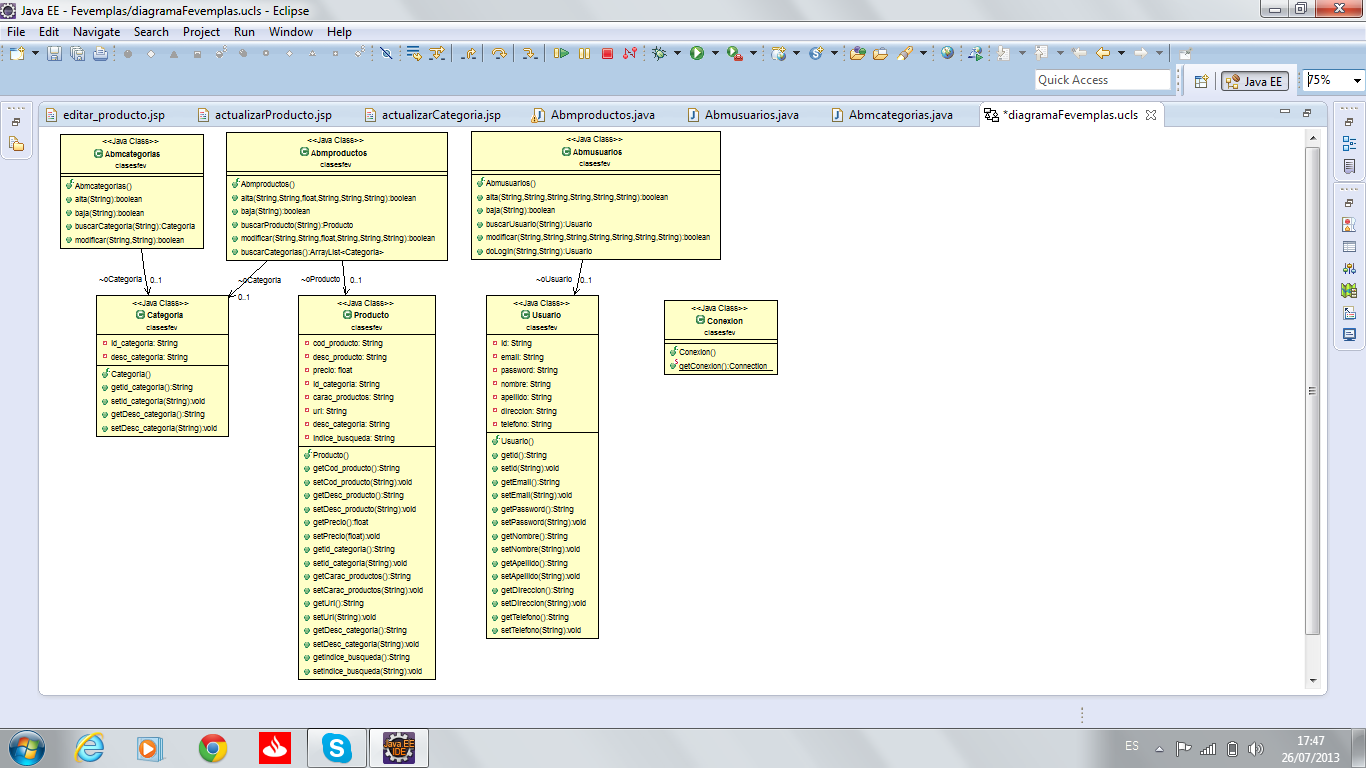
Se utilizaron 8 hojas de estilo para poder darle el aspecto logrado:



Ejemplo de llamadas a las hojas de estilo, en la home.jsp:



# Diagrama de Clases



# Casos de Uso

## C.U.1: Logueo de usuarios.

**Nivel de la meta:** Usuario **Alcance del Caso de Uso:** Sistema  **Caja:** Negra

**Instanciación:** Real **Interacción:** Dialogal **Usabilidad:** No contemplada

**ACTORES**

**Primario:** Usuario del sistema **Iniciador:** Usuario

**PRECONDICIONES:** (de sistema): el usuario tiene permisos de cliente y se encuentra registrado en el sistema.

**DISPARADOR:** El usuario ingresa al sitio web.

**FLUJO DE SUCESOS:**

**CAMINO BÁSICO:**

1. El usuario ingresa en la página de logueo e informa: nombre de usuario y contraseña.

2. El Sistema

a. Valida que exista el usuario.

b. Valida que el tipo de usuario es cliente

c. Registra al usuario en la sesión.

d. Muestra mensaje “Usuario logueado con éxito”, envía al usuario a la página principal y habilita el menú de cliente.

**CAMINOS ALTERNATIVOS:**

2. a. El usuario no existe.

2. a. 1. El sistema muestra “Usuario y/o contraseña incorrectos.” y vuelve a la página principal. Fin de CU.

2. b. El usuario es administrador.

2. b. 1. El sistema registra al usuario en sesión.

2. b. 2. El sistema muestra mensaje “Usuario logueado con éxito”, envía al

administrador a la página principal y habilita el menú de usuario administrador.

**POSTCONDICIONES: (de sistema)**

**Éxito:** El usuario accede al sitio web con permisos de cliente.

**Éxito alternativo:** El usuario accede al sitio web con permisos de administrador.

**Fracaso:** El usuario no se identificó, accede al sitio con permisos de invitado.

## C.U.2: ABM de Productos.

**Nivel de la meta:** Resumen **Alcance del Caso de Uso:** Sistema **Caja:** Negra

**Instanciación:** Real **Interacción:** Semántico **Usabilidad:** No contemplada

**ACTORES**

**Primario:** Administrador del sistema **Iniciador:** Administrador

**PRECONDICIONES:** (de sistema): el administrador se encuentra logueado en el sistema.

**DISPARADOR:** El administrador desea realizar alta, baja o modificación de los productos.

**FLUJO DE SUCESOS:**

**CAMINO BÁSICO:**

1. El administrador ingresa en el menú “Administración” y selecciona la opción ABM Productos.

2. El sistema muestra todos los productos.

3. El administrador selecciona modificar un producto, invocando al CU.2.1 – Modificar producto.

4. El sistema informa operación realizada con éxito y muestra la lista de productos actualizada.

**CAMINOS ALTERNATIVOS:**

3. a <reemplaza> El administrador selecciona eliminar un producto, invocando al CU. 2.2. Eliminar Producto.

3.a.1. Continúa en paso 4.

3. b. <reemplaza> El administrador selecciona alta producto, invocando al CU.2.3. Alta Producto.

3.b.1. Continúa en paso 4.

4. a. El sistema informa que la operación no se pudo realizar y muestra la lista de productos.

**POSTCONDICIONES:** (de sistema)

**Éxito:** El administrador pudo operar con los productos exitosamente.

**Fracaso:** Las operaciones del administrador sobre los productos, no tuvieron éxito.

### C.U.2.1: Modificar Producto.

**Nivel de la meta:** Usuario **Alcance del Caso de Uso:** Sistema **Caja:** Negra

**Instanciación:** Real **Interacción:** Dialogal **Usabilidad:** No contemplada

**ACTORES**

**Primario:** Administrador **Iniciador:** Administrador

**PRECONDICIONES:** (de sistema): el administrador se encuentra logueado en el sistema .

**DISPARADOR:** El administrador desea modificar un producto

**FLUJO DE SUCESOS:**

**CAMINO BÁSICO:**

1. El Administrador selecciona “modificar” en la fila correspondiente al producto que quiere editar.

2. El Sistema busca con el código del producto sus características en la BD y las muestra en el formulario de edición.

3. El administrador cambia los campos que requiere.

4. El sistema:

a. Valida que los datos ingresados sean correctos.

b. Actualiza el Producto en la BD.

d. Muestra mensaje de edición exitosa.

**CAMINOS ALTERNATIVOS:**

3.a. <durante> Ninguna categoría es adecuada.

3.a.1. El administrador decide crear nueva categoría invocando al CU 3.1 Alta Categoría

3.a.2. Vuelve al paso 3.

4. a. Los datos ingresados son erróneos.

4. a. 1. El sistema muestra mensaje con el error cometido.

4. a. 2. El sistema habilita al administrador a ingresar los datos nuevamente.

**POSTCONDICIONES:** (de sistema)

**Éxito:** El producto fue modificado exitosamente.

**Fracaso:** El producto no pudo modificarse.

### C.U.2.2: Eliminar Producto.

**Nivel de la meta:** Usuario **Alcance del Caso de Uso:** Sistema **Caja:** Negra

**Instanciación:** Real **Interacción:** Dialogal **Usabilidad:** No contemplada

**ACTORES**

**Primario:** Administrador **Iniciador:** Administrador

**PRECONDICIONES:** (de sistema): el administrador se encuentra logueado en el sistema .

**DISPARADOR:** El administrador desea eliminar un producto

**FLUJO DE SUCESOS:**

**CAMINO BÁSICO:**

1. El Administrador selecciona “eliminar” en la fila correspondiente al producto que quiere dar de baja.

2. El sistema requiere confirmación para dar de baja.

3. El usuario confirma eliminar el producto.

4. El sistema:

a. Busca con el código al producto en la BD y lo elimina.

b. Muestra mensaje de baja exitosa.

**CAMINOS ALTERNATIVOS:**

3.a. <reemplaza>El usuario cancela la baja del producto. Fin CU.

4. a. 1. No se pudo eliminar el producto

4. a. 2. El sistema informa situación. Fin CU.

**POSTCONDICIONES:** (de sistema)

**Éxito:** El producto fue eliminado exitosamente.

**Fracaso:** El producto no pudo eliminarse.

### C.U.2.3: Alta Producto.

**Nivel de la meta:** Usuario **Alcance del Caso de Uso:** Sistema **Caja:** Negra

**Instanciación:** Real **Interacción:** Dialogal **Usabilidad:** No contemplada

**ACTORES**

**Primario:** Administrador **Iniciador:** Administrador

**PRECONDICIONES:** (de sistema): el administrador se encuentra logueado en el sistema.

**DISPARADOR:** El administrador desea registrar un nuevo producto.

**FLUJO DE SUCESOS:**

**CAMINO BÁSICO:**

1. El Administrador selecciona categoría para el producto y completa el formulario de alta de producto.

2. El Sistema

a. Valida que los datos ingresados sean correctos.

b. Almacena el Producto en la BD.

d. Muestra mensaje de alta exitosa.

**CAMINOS ALTERNATIVOS:**

1.a. <durante> Ninguna categoría es adecuada.

1.a.1. El administrador decide crear nueva categoría invocando al CU 3.1 Alta Categoría

1.a.2. Vuelve al paso 1.

2. a. Los datos ingresados son erróneos.

2. a. 1. El sistema muestra mensaje con el error cometido.

2. a. 2. El sistema habilita al administrador a ingresar los datos nuevamente.

**POSTCONDICIONES:** (de sistema)

**Éxito:** El producto fue dado de alta.

**Fracaso:** El producto no se dio de alta.

# Código

## Conexión a la BD

**Clase Conexion.java**

**package** clasesfev;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.\*;

**public** **class** Conexion

{

**public** **static** Connection getConexion()

{

Connection con=**null**;

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306/fevemplas";

con=DriverManager.*getConnection*(url, "root", "12345");

**return** con;

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** con;

}

}

## ABM

**Clase ABMproductos.java**

**package** clasesfev;

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.\*;

**import** java.util.\*;

**public** **class** Abmproductos

{

Producto oProducto = **new** Producto();

Categoria oCategoria = **new** Categoria();

**public** **boolean** alta(String cod\_producto, String desc\_producto, **float** precio, String id\_categoria, String carac\_productos, String url) **throws** SQLException

{

PreparedStatement ps = **null**;

Connection con = **null**;

**int** rs=0;

**try**

{

**try** {

con = Conexion.*getConexion*();

} **catch** (Exception e)

{e.printStackTrace();

}

ps=con.prepareStatement("insert into productos values ( ?, ?, ?, ? , ?, ?)");

ps.setString(1, cod\_producto);

ps.setString(2, desc\_producto);

ps.setFloat(3, precio);

ps.setString(4, id\_categoria);

ps.setString(5, carac\_productos);

ps.setString(6, url);

rs = ps.executeUpdate();

**return** **true**;

} **catch** (Exception e) {

**return** **false**;

}

**finally**

{

ps.close();

con.close();

}

}

**public** **boolean** baja(String cod) **throws** SQLException

{

PreparedStatement ps = **null**;

Connection con = **null**;

**int** rs=0;

**try**

{

**try**

{

con=Conexion.*getConexion*();

}

**catch** (Exception e)

{

e.printStackTrace();

}

ps=con.prepareStatement("delete from productos where cod\_producto= ? ;");

ps.setString(1, cod);

rs = ps.executeUpdate();

**return** **true**;

}

**catch**(Exception e)

{

**return** **false**;

}

**finally**

{

ps.close();

con.close();

}

}

**public** Producto buscarProducto(String id) **throws** SQLException, ClassNotFoundException

{

PreparedStatement ps = **null**;

Connection con = **null**;

**try**

{

**try** {

con = Conexion.*getConexion*();

} **catch** (Exception e) {

e.printStackTrace();

}

ps = con.prepareStatement("SELECT \* FROM productos WHERE cod\_producto=?");

ps.setString(1, id);

ResultSet rs = ps.executeQuery();

rs.next();

**this**.oProducto.setCod\_producto(rs.getString("cod\_producto"));

**this**.oProducto.setDesc\_producto(rs.getString("desc\_producto"));

**this**.oProducto.setPrecio(rs.getFloat("precio"));

**this**.oProducto.setId\_categoria(rs.getString("id\_categoria"));

**this**.oProducto.setCarac\_productos(rs.getString("carac\_productos"));

**this**.oProducto.setUrl(rs.getString("url"));

**return** **this**.oProducto;

}

**catch** (Exception e)

{

**return** **null**;

}

**finally**

{

ps.close();

con.close();

}

}

**public** **boolean** modificar(String cod, String desc\_producto, **float** precio, String id\_categoria, String carac\_productos, String url) **throws** SQLException

{

PreparedStatement ps = **null**;

Connection con = **null**;

**int** rs=0;

**try**

{

**try**

{

con= Conexion.*getConexion*();

}

**catch** (Exception e)

{

e.printStackTrace();

}

**try**

{

ps=con.prepareStatement("update productos set desc\_producto='"+desc\_producto+"',precio='"+precio+"',id\_categoria='"+id\_categoria+"',carac\_productos='"+carac\_productos+"',url='"+url+"' where cod\_producto='"+cod+"'");

rs = ps.executeUpdate();

**if**(rs!=0)

{

**return** **true**;

}

**else**

{

**return** **false**;

}

}

**catch** (Exception e)

{

**return** **false**;

}

}

**finally**

{

ps.close();

con.close();

}

}

}

## Login

En doLogin.jsp

<%@ page language=*"java"* errorPage=*""*%>

<%@ page import=*"clasesfev.\*"* %>

<%

String sUser= request.getParameter("sUserName"); // ("sUserName") es el name del LOGINFORM

String sPass= request.getParameter("sPwd");

String message= "Usuario logueado con éxito";

String ad="admin"; // se puede modificar el NOMBRE del administrador

Abmusuarios oAbmusuarios = **new** Abmusuarios();

Usuario oUsuario = **new** Usuario();

oUsuario = oAbmusuarios.doLogin(sUser, sPass);

**if**(oUsuario==**null**)

{

message="No user or password matched" ;

String url = response.encodeRedirectURL("login.html?error="+message);

response.sendRedirect(url);

}

**else**{

String id\_us = oUsuario.getId();

String name = oUsuario.getNombre()+ " " + oUsuario.getApellido();

session.setAttribute("userID", id\_us );

session.setAttribute("userName", name );

**if**(oUsuario.getNombre().equals(ad))

{ session.setAttribute("esAdmin", ad);

String url = response.encodeRedirectURL("administrador.jsp?error="+message);

response.sendRedirect(url);

}

**else**{

String url = response.encodeRedirectURL("home.jsp?error="+message);

response.sendRedirect(url);}

}

%>

En ABMusuarios.java:

**public** Usuario doLogin(String email, String pass) **throws** SQLException, ClassNotFoundException

{

PreparedStatement ps = **null**;

Connection con = **null**;

**try**

{

**try** {

con = Conexion.*getConexion*();

} **catch** (Exception e) {

e.printStackTrace();

}

ps = con.prepareStatement("SELECT \* FROM usuarios WHERE email=? AND password=?");

ps.setString(1, email);

ps.setString(2, pass);

ResultSet rs = ps.executeQuery();

rs.next();

**this**.oUsuario.setId(rs.getString("id\_usuario"));

**this**.oUsuario.setEmail(rs.getString("email"));

**this**.oUsuario.setPassword(rs.getString("password"));

**this**.oUsuario.setNombre(rs.getString("nombre"));

**this**.oUsuario.setApellido(rs.getString("apellido"));

**this**.oUsuario.setDireccion(rs.getString("direccion"));

**this**.oUsuario.setTelefono(rs.getString("telefono"));

**return** **this**.oUsuario;

}

**catch** (Exception e)

{

**return** **null**;

}

**finally**

{

ps.close();

con.close();

}

}

}