

# **Proyecto Final**

# **Acceso a Datos**

The background of the cover features a large, light orange circle containing the white Ubuntu logo, which is a stylized gear or flower shape with six points.

***TIENDA ONLINE  
UBUNTU STORE***

**Santiago Hoyos Zea**

Desarrollo de aplicaciones multiplataforma  
Centro concertado Gregorio Fernández  
Valladolid- 2016

# ÍNDICE

1. [Introducción](#)
2. [Herramientas utilizada](#)
3. [Recogida de requisitos](#)
4. [Diseño](#)
  - 4.1 [Diseño de la base de datos](#)
  - 4.2 [Modelo EER de la base de datos](#)
  - 4.3 [Capturas de la aplicación](#)
5. [Conclusiones finales](#)

# 1-INTRODUCCIÓN

Este proyecto pretende realizar una tienda online hecha con las tecnologías de Java EE y Tomcat Server.

La tienda esta dedicada a la comercialización de productos de Ubuntu (Software de código abierto) los cuales están listados y almacenados en una base de datos al igual que los usuarios y compras.

La tienda puede realizar las siguientes funciones: **exhibición de productos** tanto a usuarios logueados como a los no logueados; **registrarse** a los usuarios sin identidad en la aplicación, **cambio de datos de perfil** de cliente a los usuarios registrados, **comprar** solamente a los usuarios registrados solamente; **añadir, modificar y eliminar** productos solo a usuarios con privilegio de administrador desde el panel de administración y **subir fotos** de los productos que van a parar a la carpeta “imagenesProdcutos” dentro de la carpeta “web-content”.

Para la gestión de la imágenes de los productos de este proyecto se ha optado por usar dos carpetas dentro de “web-content” una llamada “imagenesProductos” en la que están todas la imágenes de los productos, que pueden ser subidas a través del panel de administración y la carpeta “imagenesWeb” en las que están tales como el logo de la web etc.

## 2- HERRAMIENTAS UTILIZADAS

Para llevar a cabo este proyecto he usado los siguientes programas, lenguajes, frameworks o bibliotecas:

**Oracle Java 8:** esta aplicación ha sido compilada para la versión 8 de java.

**Eclipse Neón (N4):** Última versión del IDE Eclipse a febrero de 2016, requiere el uso de java 8 o superior en los proyectos.

**Apache Tomcat 6 y modulo web 2.5:** se ha compilado la aplicación con objetivo en esta versión de Tomcat, aunque lo suyo hubiera sido Tomcat 8.

**MySQL 5.6.26:** sistema gestor de base de datos utilizado.

**MySQL WorkBench:** programa para la gestión y desarrollo de bases de datos en MySQL versión 6.3 community.

**PhpMyAdmin:** aplicación web para la gestión y desarrollo de bases de datos en MySQL versión 4.4.14.

**VPS Ubuntu Linux Server:** servidor personal virtual, contiene la base de datos.

**Brackets 1.6:** IDE para el desarrollo web tanto HTML como CSS.

**HTML y CSS3:** lenguaje de marcas para el maquetado de la aplicación.

**Bootstrap:** Framework para el desarrollo web responsive con HTML 5 y CSS 3 en concreto su tema prefabricado de tienda.

**Mysql-connector-5-1-35:** biblioteca con el driver para la conexión a base de datos tipo MySql.

**Commons-fileupload-1.3.1:** biblioteca para el subir ficheros al servidor mediante protocolo http.

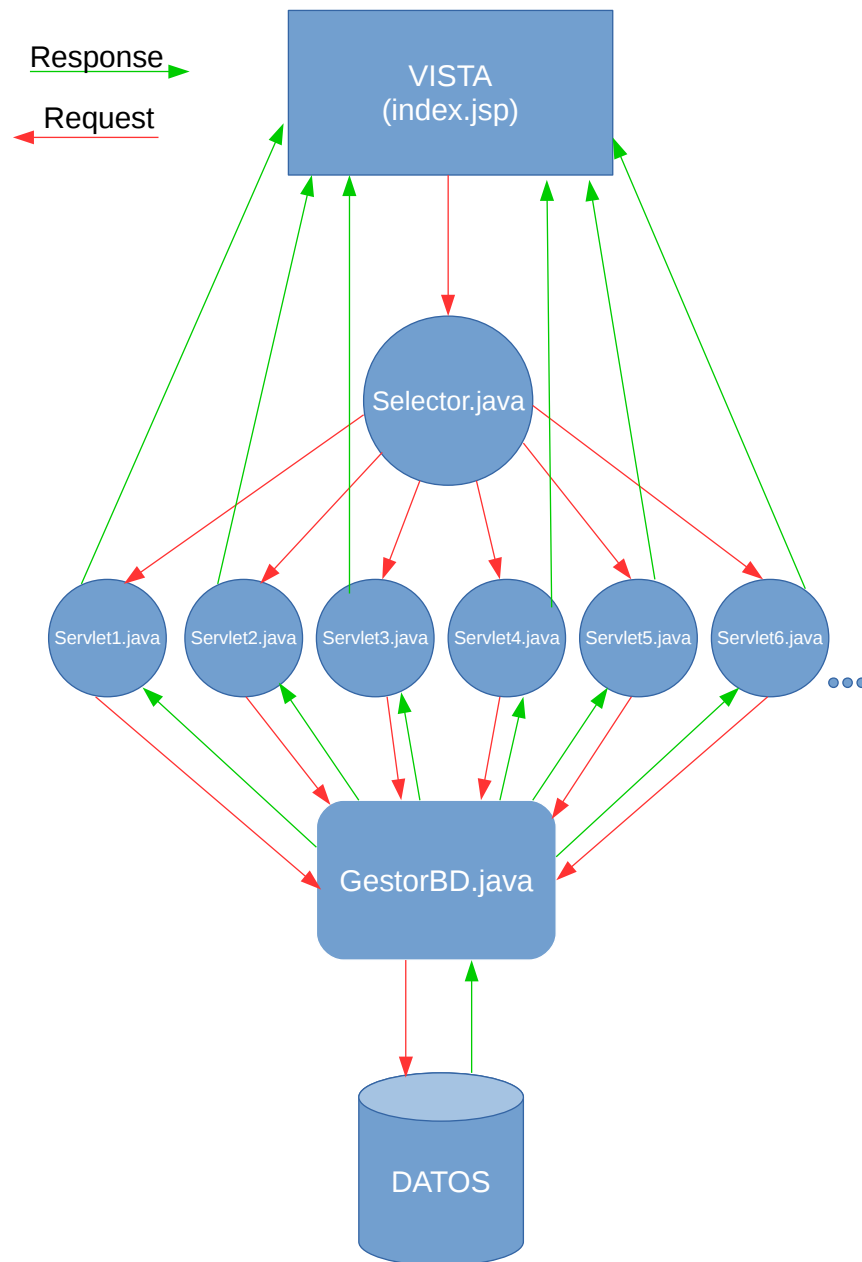
**Commons-io-2.4:** biblioteca para la gestión de operaciones de entrada-salida.

### 3- RECOGIDA DE REQUISITOS

Una vez reunido el analista programador con el cliente expone lo siguiente:

- 1) Se requiere una **aplicación web** independiente de la plataforma en la que se despliegue.
- 2) Dicha aplicación ha de ser **robusta** con un lenguaje popular y una comunidad detrás o empresas para dar buen soporte.
- 3) La aplicación **debe poderse ampliar** en el futuro sin mayores problemas.
- 4) Debe ser **independiente o al menos no dependiente 100% de la base de datos** utilizada por si se decide migrar dicha aplicación en un futuro a otro gestor de bases de datos mejor o actualizar el que este en uso.

## 4- DISEÑO



Diseño del flujo de peticiones y respuestas http

Como se puede ver en el gráfico anterior la tienda intenta cumplir el patrón de diseño vista controlador (MVC) por lo que se ha separado la vista de toda funcionalidad y dejándole solo su cometido que es representar los datos, para ello todos los *request* van a index.jsp donde luego un *include* hace la llamada al *servlet* “Selector.java” que se encarga de elegir a que destino enviar esa petición para procesarla. Luego, una vez ya la petición llega a alguno de los *servlets* del paquete “operativa” se procesa la petición, si en la lógica del servlet se requiere cargar datos o enviar datos a la base de datos

esto se hace a través de la clase “GestorBD.java”, esta es un *singleton* que contiene toda la lógica de consulta, inserción, modificación y eliminación de datos. Después, el servlet devuelve el response a la vista, y finalmente la vista representa el resultado.

#### 4.1- DISEÑO DE LA BASE DE DATOS (TIENDAONLINE)

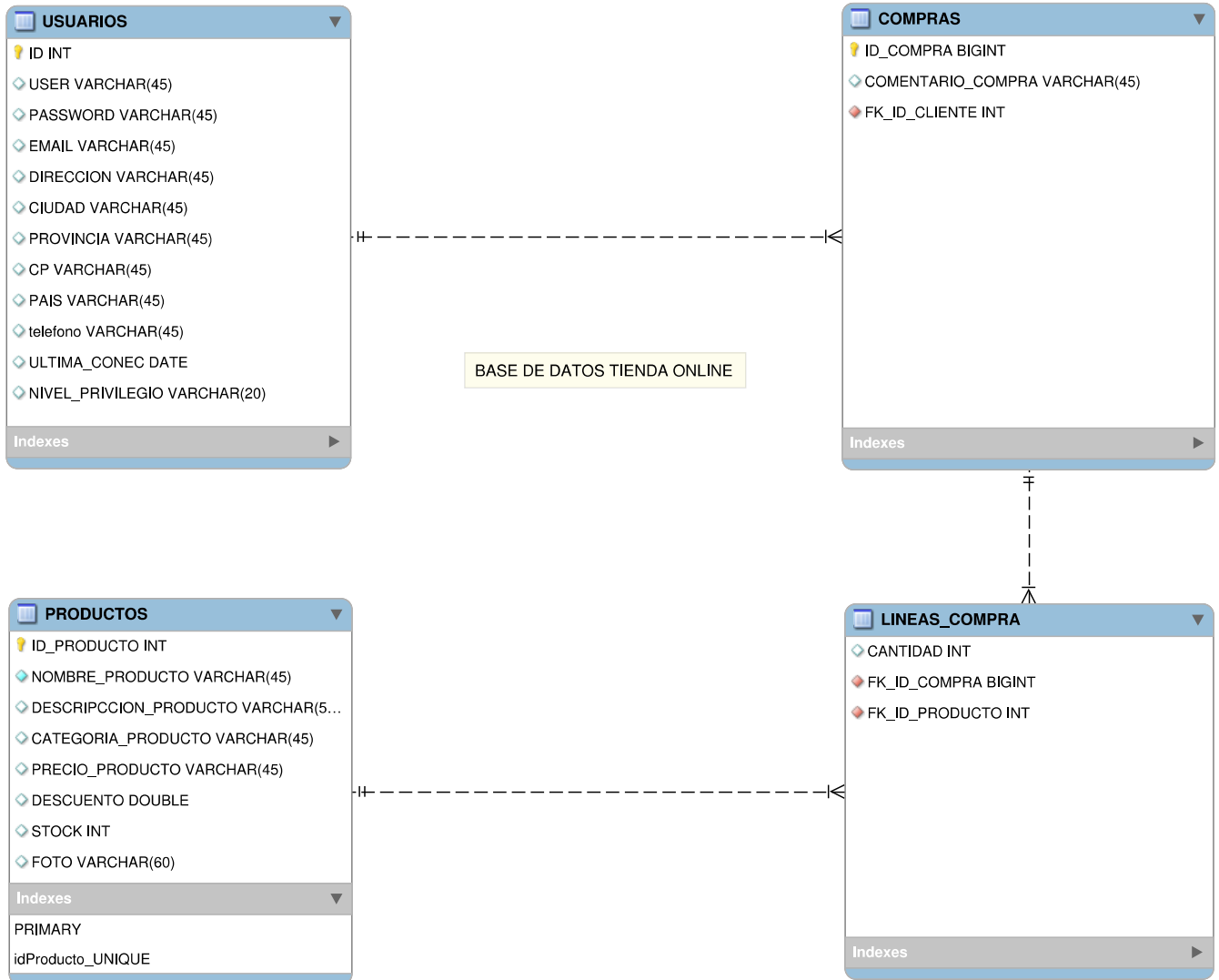
**Productos:** constan de un id de producto, nombre, descripción, precio, descuento, stock y foto. Id y stock se almacenarán como enteros, precio, y descuento como double por último nombre, descripción y foto como String (nombre de la foto en la carpeta /imagenesProductos).

**Usuarios:** constan de: id, nick, contraseña, última conexión y privilegio donde id es un entero, última conexión un Date y por último nick, contraseña y privilegio un String donde privilegio solo puede tomar como valores “admin” o “cliente”.

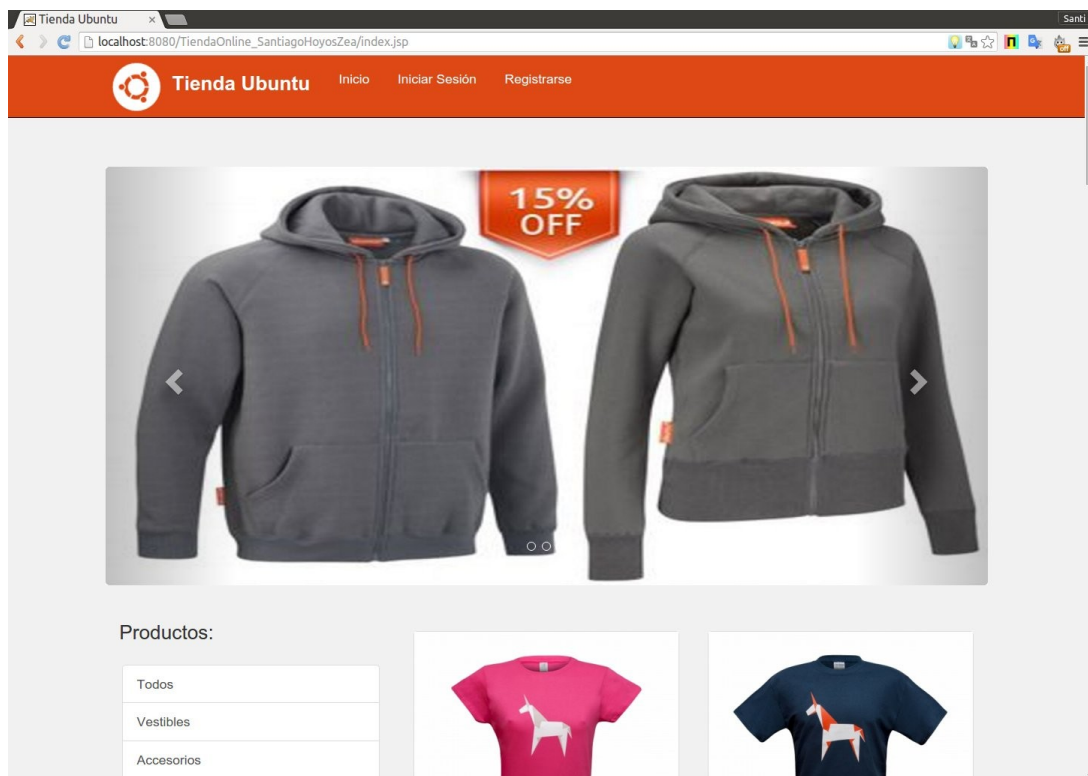
**Compras.** Para modelar las compras se ha seguido un modelado típico para estos casos que es modelar la tabla compras por un lado y la tabla línea de compra, donde compras tiene: id, fk\_id\_cliente, comentario, donde id es el id único que identifica a la compra mediante un entero y fk\_id\_cliente es clave foránea de la tabla usuarios que relaciona el cliente que ha realizado la compra con la misma. Además se permite añadir un comentario a cada compra por si se quisiera en un futuro.

**Línea de compra:** modela cada línea de las compras, están compuestas por: fk\_id\_compra, cantidad, fk\_id\_producto todos ellos almacenados como enteros donde fk\_id\_compra se corresponde con la compra dueña de esa línea en la tabla compras, cantidad se corresponde con el volumen de pedido del producto de esa línea e id que corresponde con él.

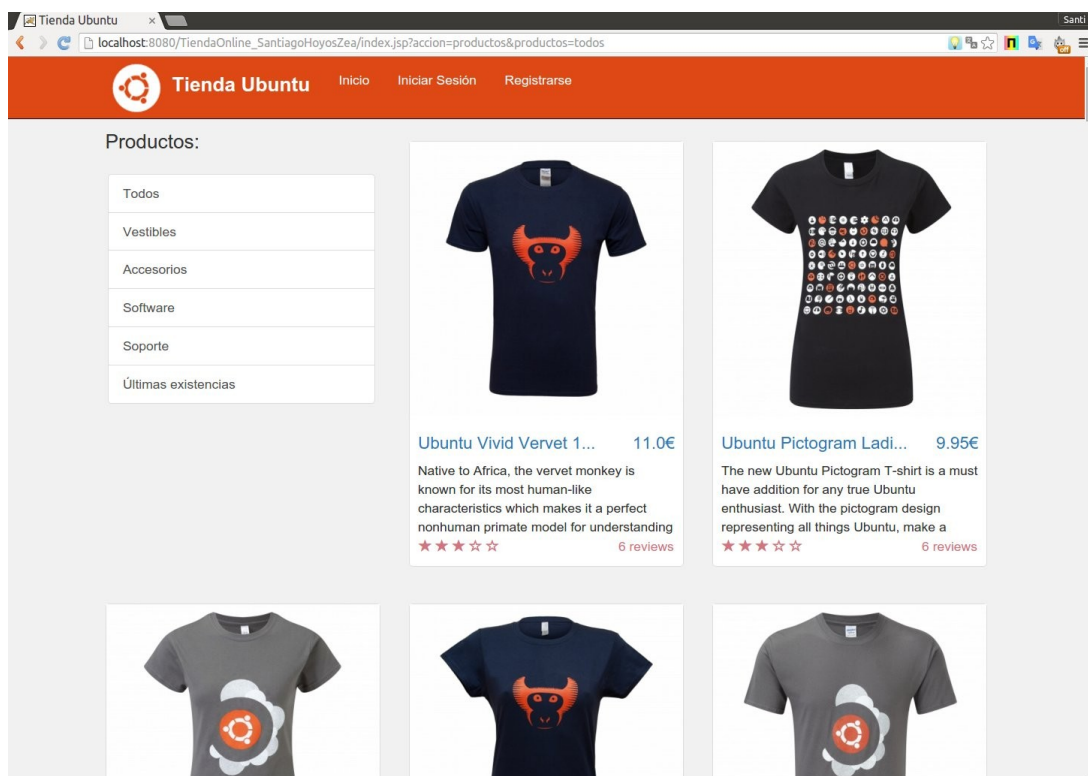
## 4-2 Modelo EER de la base de datos:



## 4-3 CAPTURAS

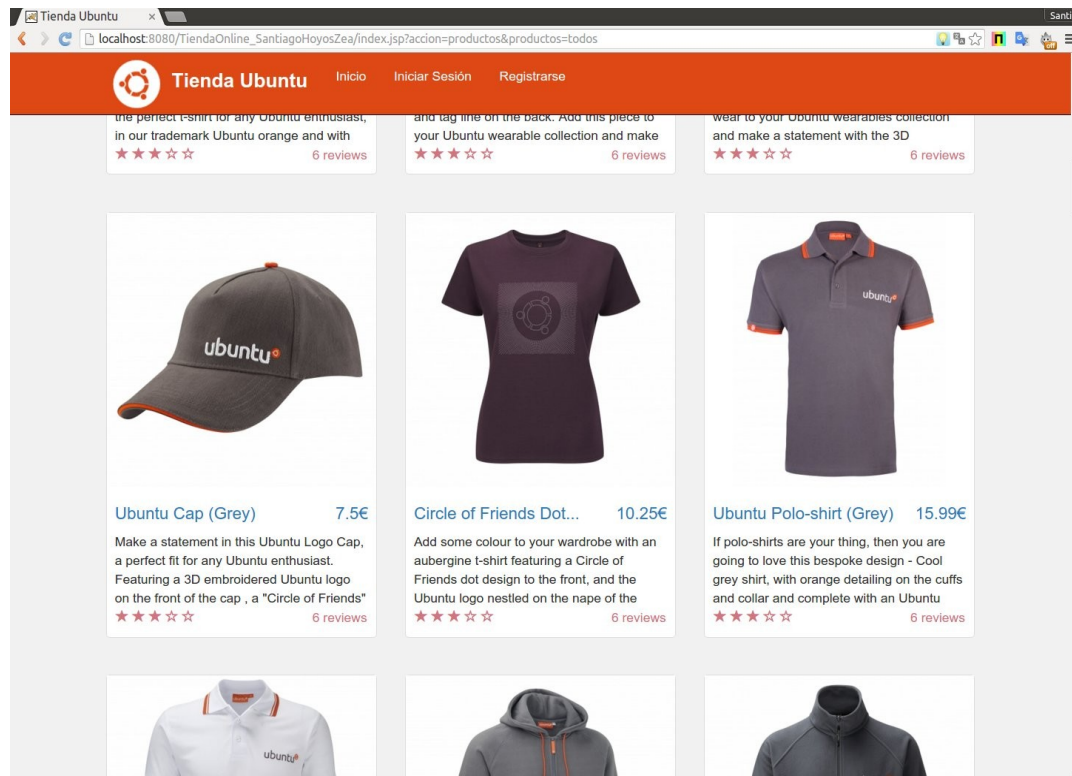


Index principal (muestra productos con descuento)

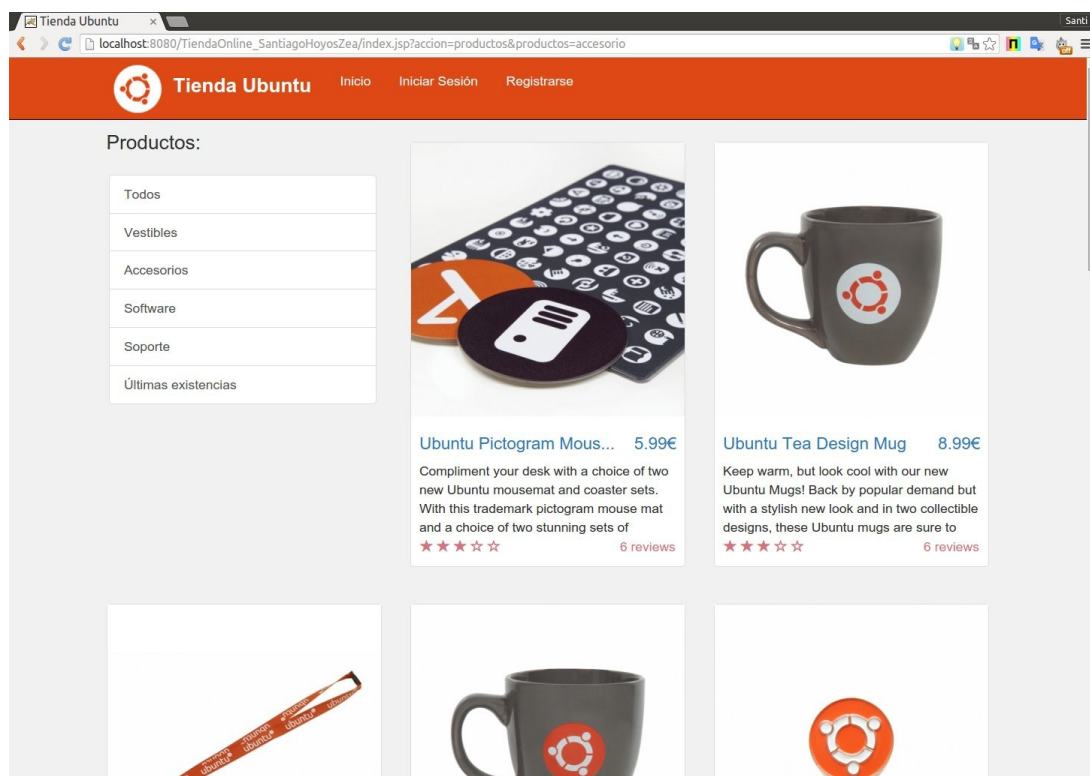


Vista al picar en Productos → Todos

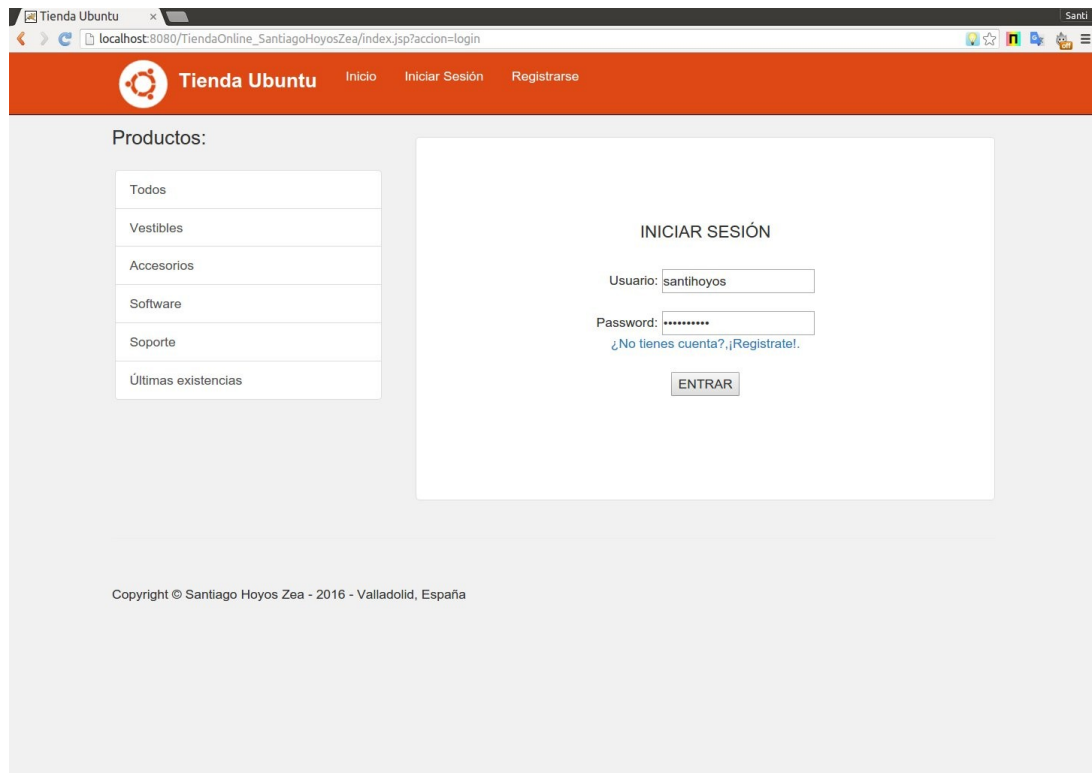




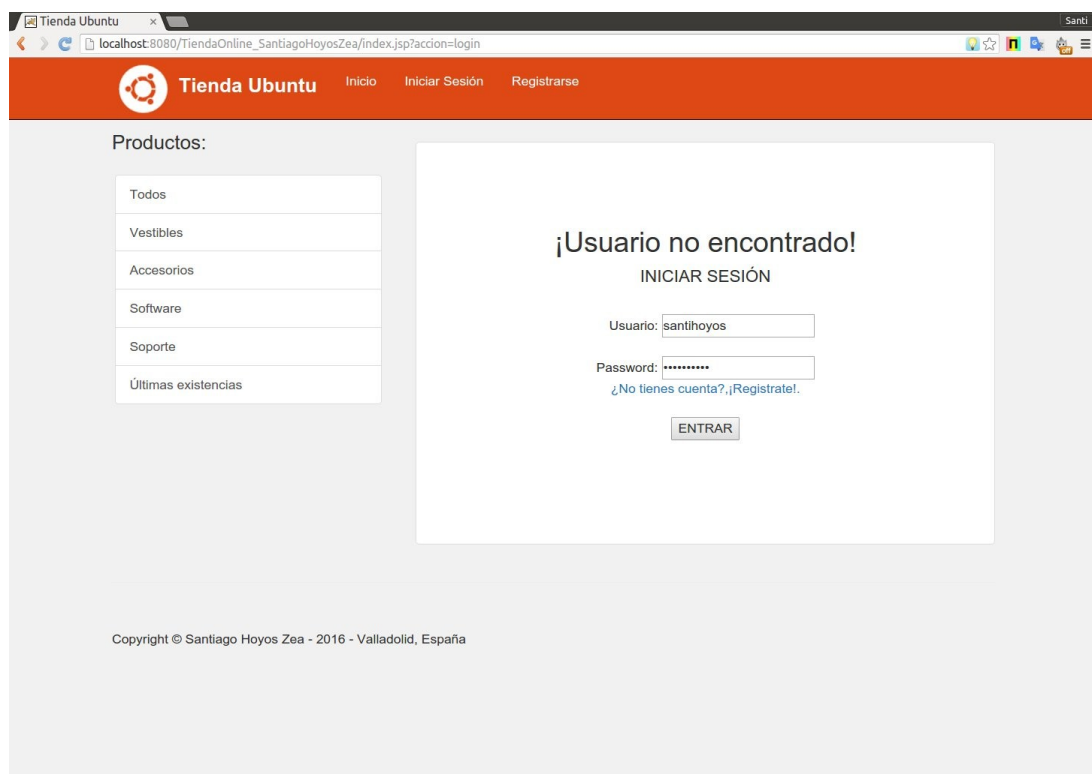
vista de todos más abajo



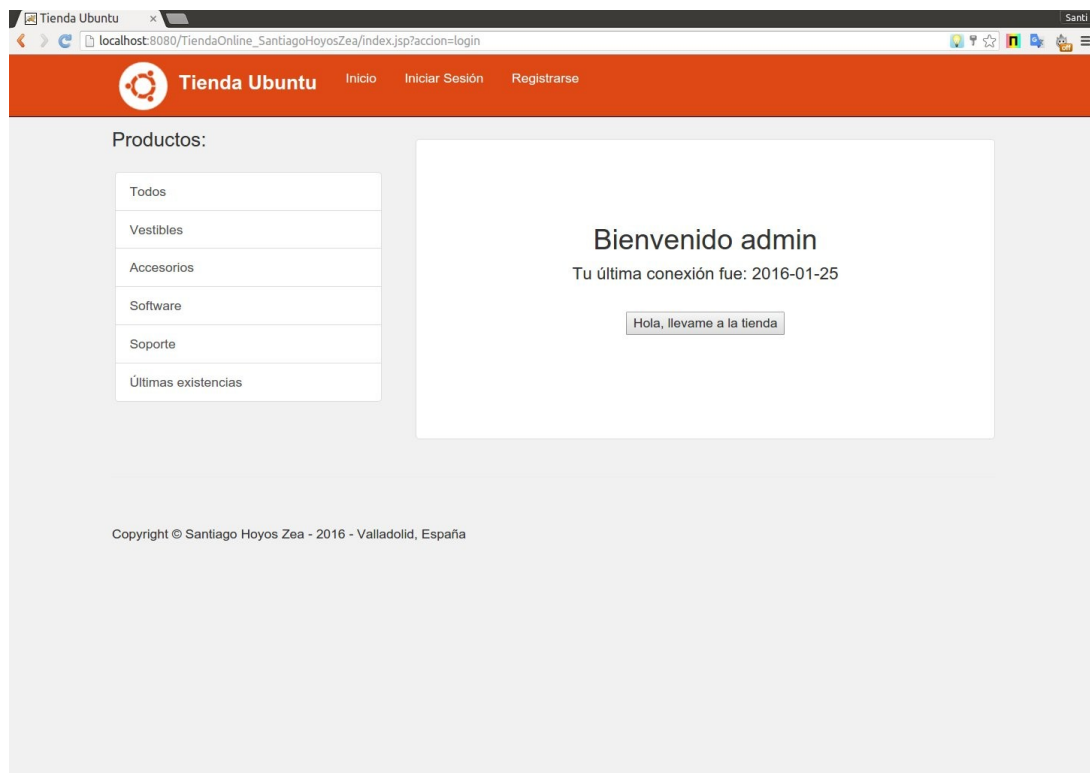
Vista Categoría "accesorios"



Vista de login



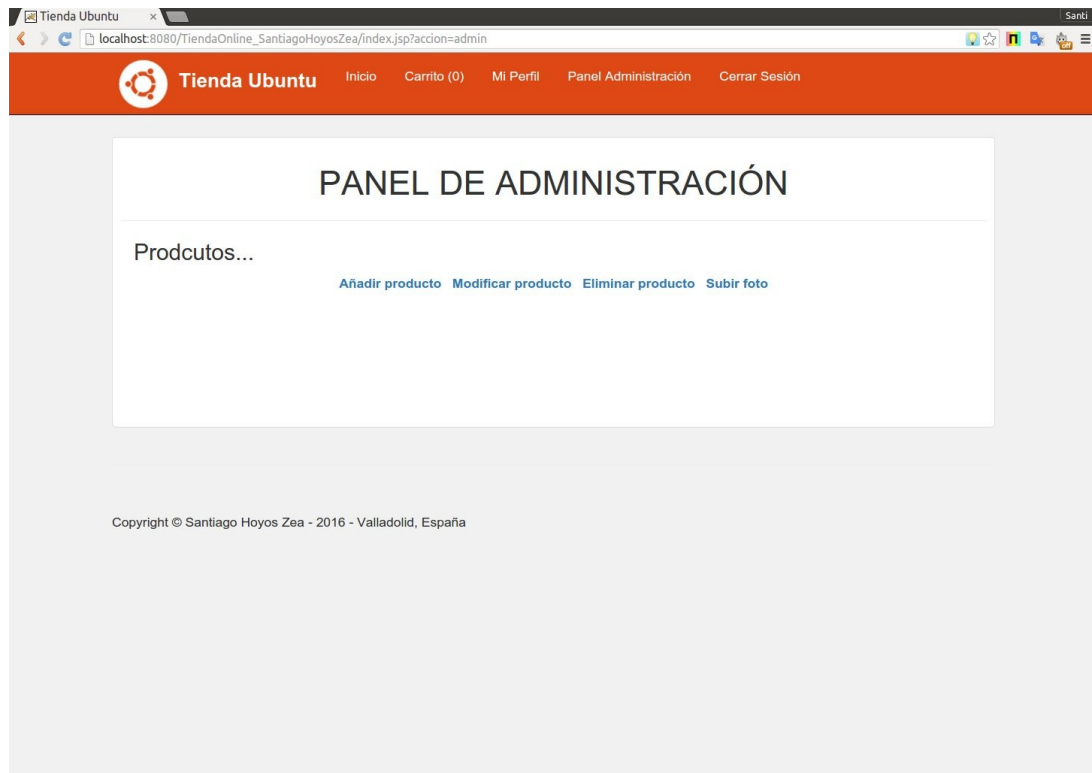
Vista login con error en usuario (con contraseña también avisa)



Vista bienvenida a usuario con su última fecha de conexión



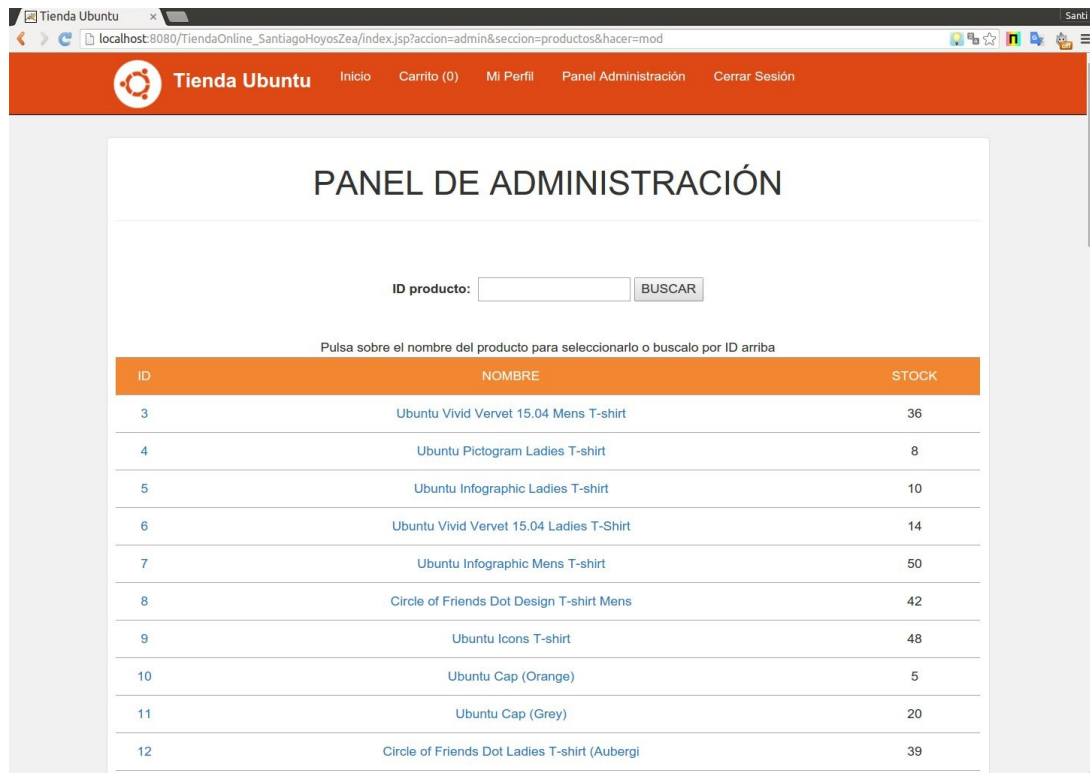
Vista de index con login admin (fijarse en el menú)



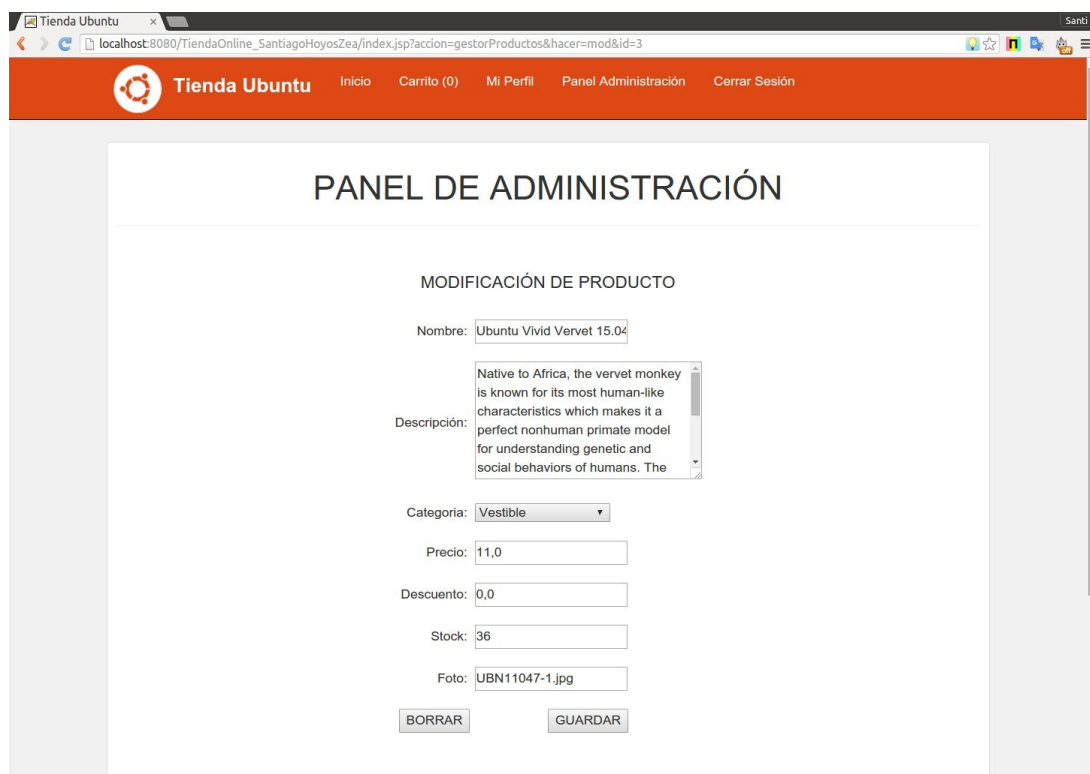
Vista panel de administración (solo admin)

The screenshot shows the same web browser window, but the URL is 'localhost:8080/TiendaOnline\_SantiagoHoyosZea/index.jsp?accion=gestorProductos&hacer=add'. The page layout is identical, but the main content area is titled 'PANEL DE ADMINISTRACIÓN' and contains a form titled 'NUEVO PRODUCTO'. The form includes the following fields: 'Nombre:' with a text input, 'Descripción:' with a larger text area, 'Categoria:' with a dropdown menu showing 'Vestible', 'Precio:' with a text input containing '0', 'Descuento:' with a text input containing '0', 'Stock:' with a text input containing '0', and 'Foto:' with a text input containing '0'. At the bottom of the form are two buttons: 'BORRAR' and 'GUARDAR'.

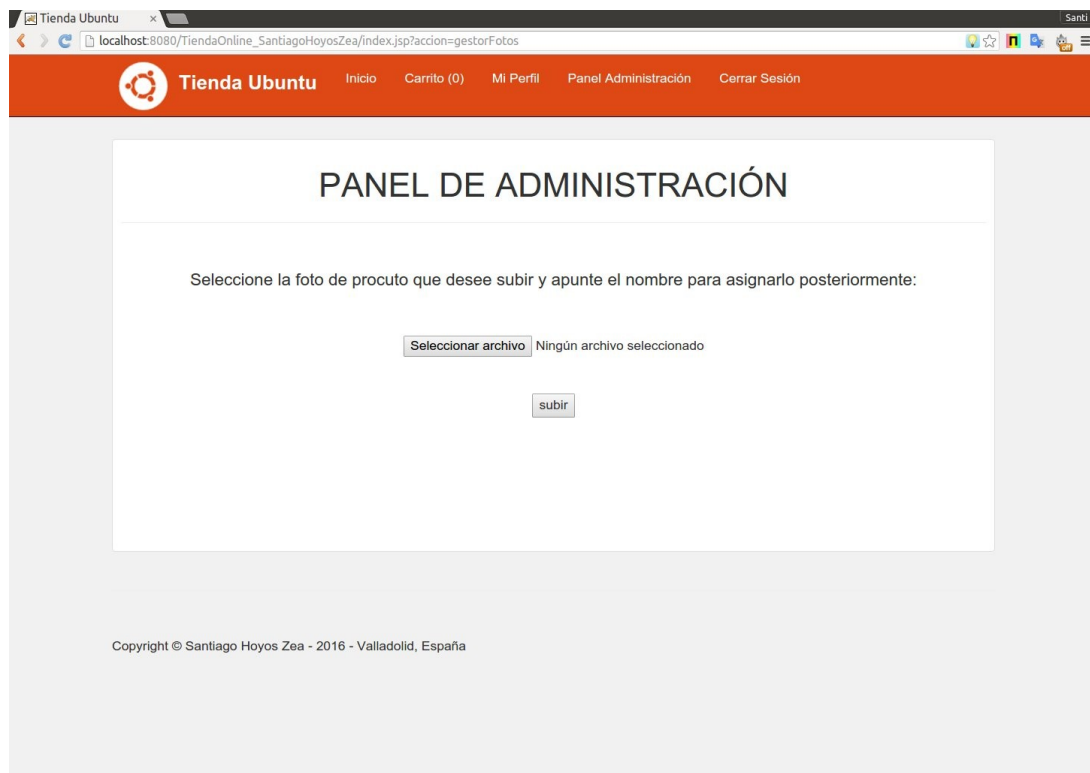
Vista de añadir nuevo producto



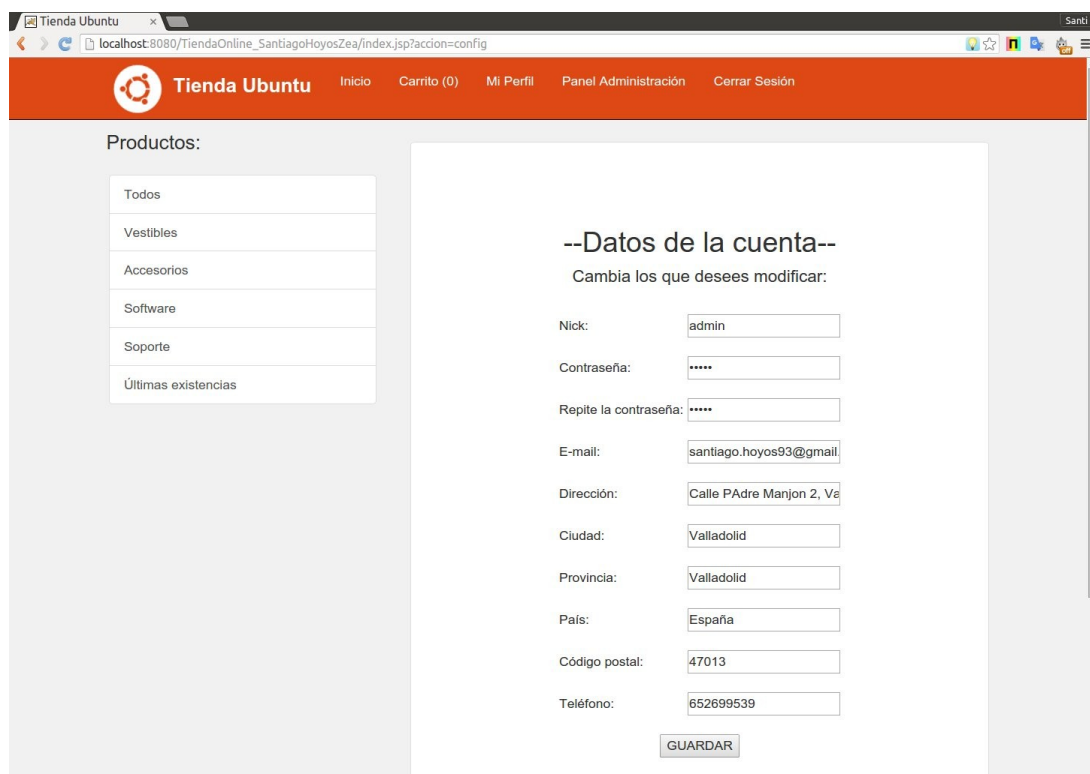
Vista de selección de producto para modificar o eliminar, se puede seleccionar tanto picando sobre el en la tabla o poniendo su id si se sabe



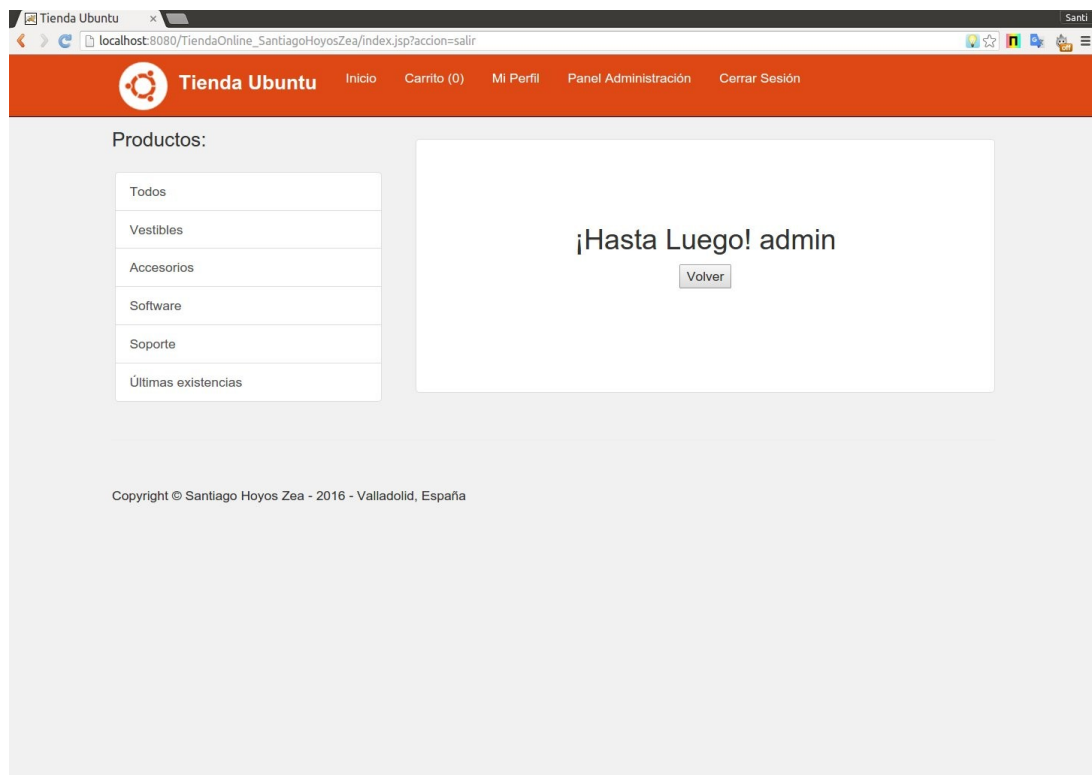
Vista de modificación de un producto previamente seleccionado en la pantalla anterior.



Vista de subida de una imagen que irá a parar a la carpeta /imagenesProdcutos



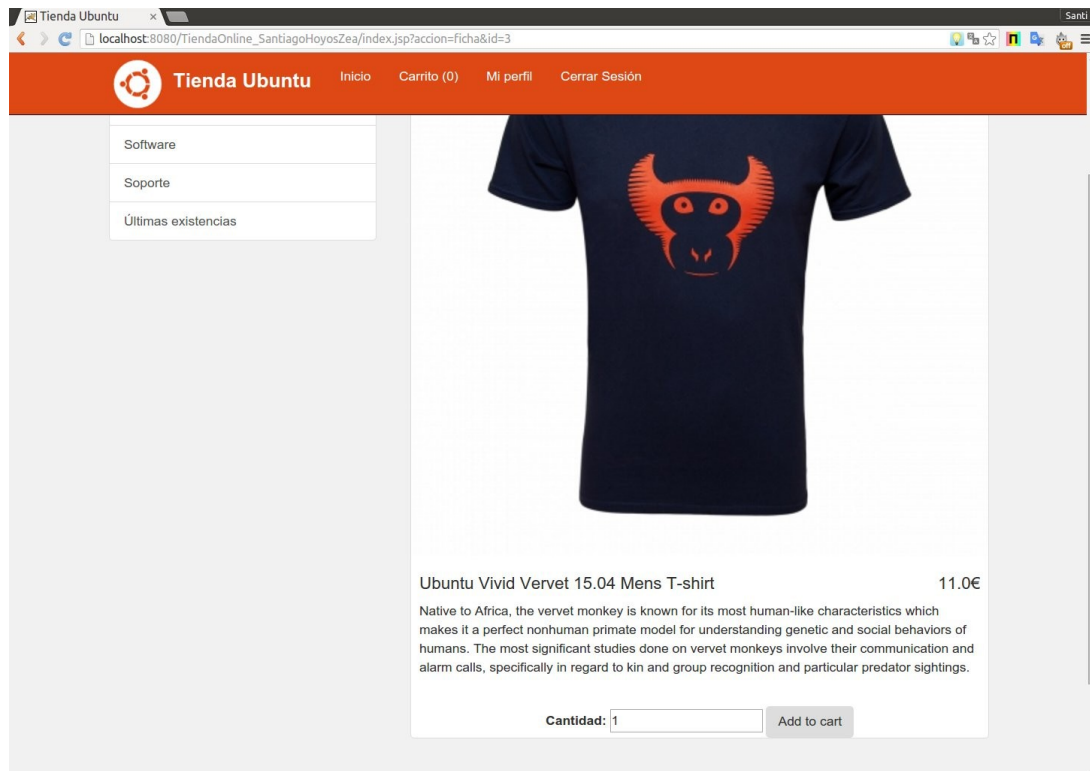
Vista de config de un usuario, en ese caso es común a un cliente y a un admin



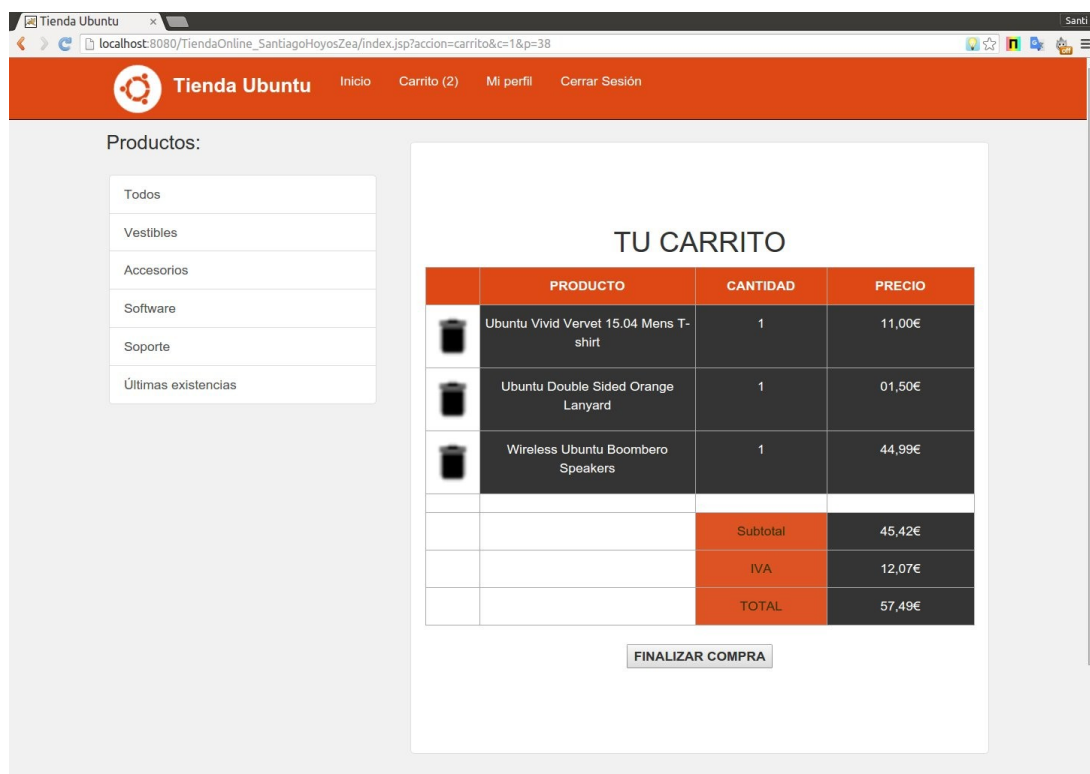
Despedida de un usuario(admin o cliente)



Index para un usuario cliente

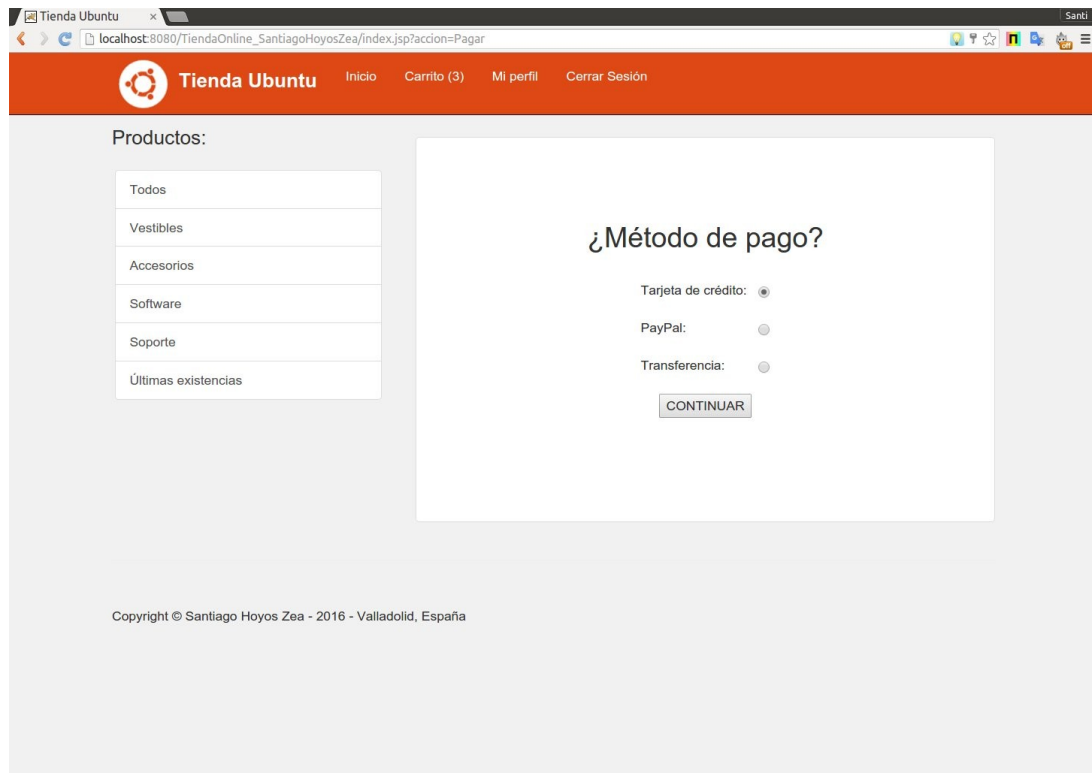


vista de la ficha de un producto, donde se puede seleccionar su cantidad para añadir al carrito.

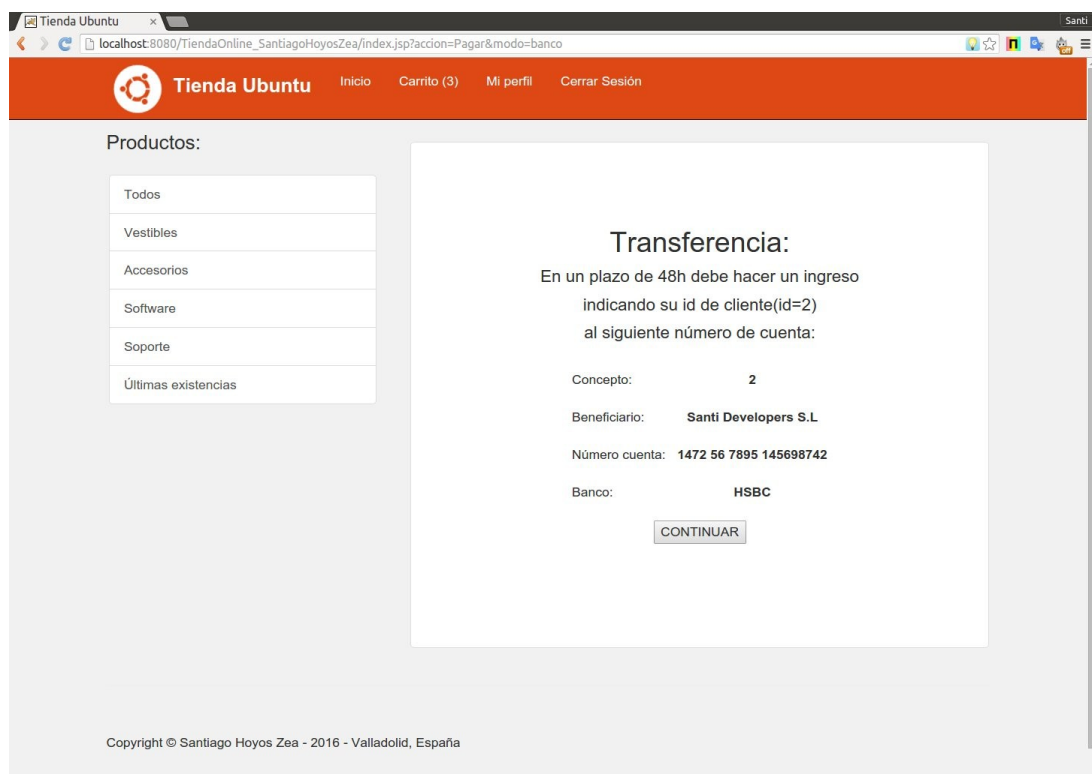


Vista de un carrito con 3 productos

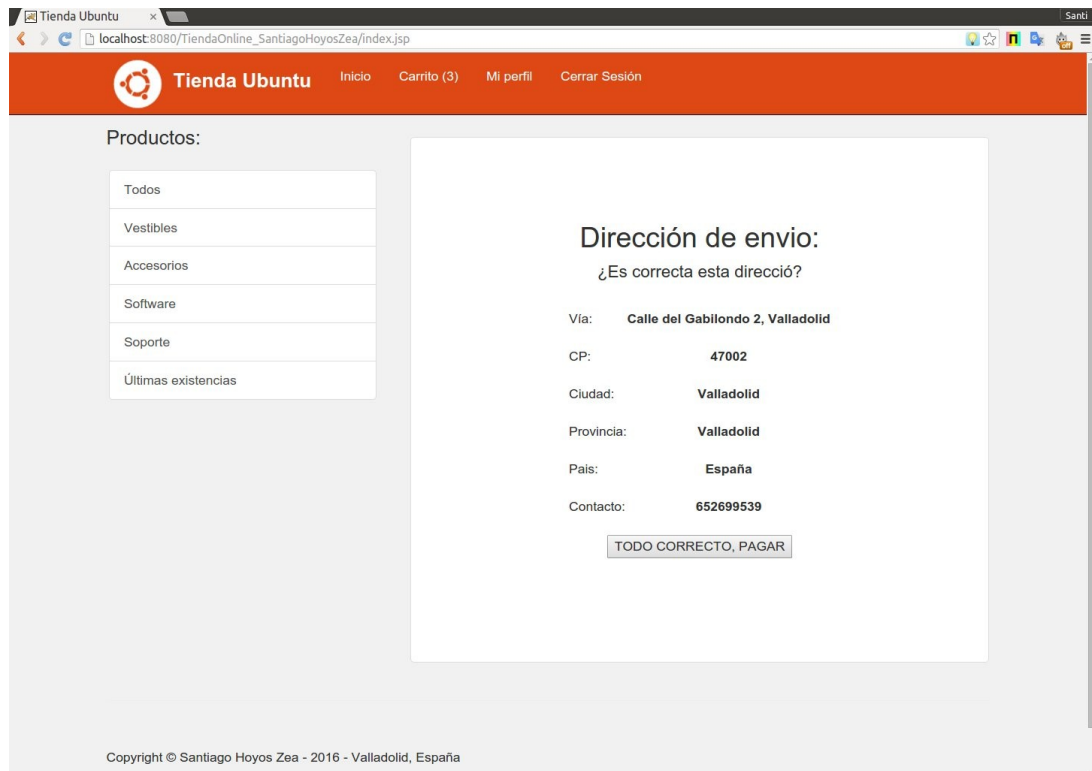




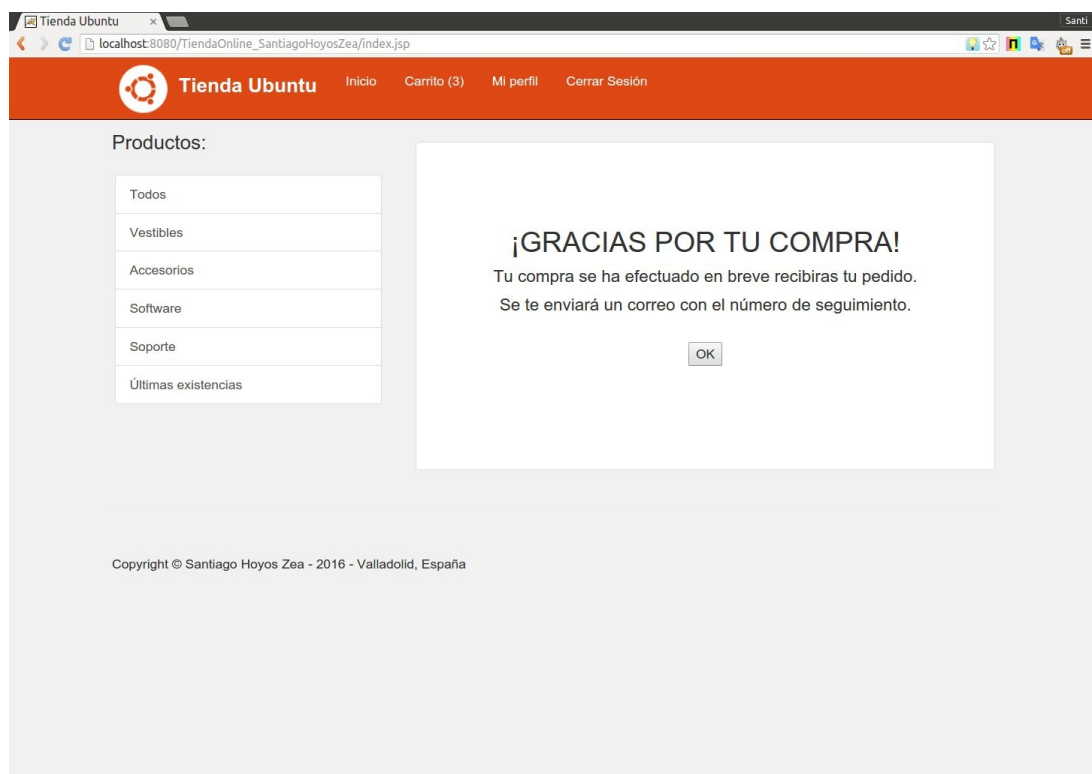
Vista de selección de pago cuando se pica en “finalizar compra” en la vista carrito.



Vista de uno de los modos de pago(el que más tiene lo otros dos solo dos textbox).



Vista de confirmación para el usuario si es su dirección a la que quiere enviarse el producto.



Vista de agradecimiento por la compra, interiormente se guarda en base de datos y se eliminan las cookies.

Tienda Ubuntu Inicio Iniciar Sesión Registrarse

Productos:

- Todos
- Vestibles
- Accesorios
- Software
- Soporte
- Últimas existencias

### --DATOS DE REGISTRO--

Nick:

Contraseña:

Repita la contraseña:

E-mail:

Dirección:

Ciudad:

Provincia:

País:

Código postal:

Teléfono:

REGISTRARSE

Vista de registro para usuario nuevo

## 5-CONCLUSIÓN:

Como conclusión es un proyecto que si se quiere hacer bien, una persona tardaría un año o más, si entendemos por hacerlo bien tanto a nivel de funcionalidad como de interfaz gráfica y no merecería la pena habiendo proyectos de la envergadura de "Prestashop". Respecto a las tecnologías usadas no me parece una opción cómoda usar Java para este tipo de apps, creo que el potencial de Java está en apps más grandes donde se requiere una robustez que Java puede satisfacer.

**Las conexiones a la base de datos...** usando el driver Mysql es algo que aunque he decidido hacerlo de forma centralizada para que sea más fácil, aún así da sus problemas y es muy desesperante de implementar de ahí que haya dicho antes que preferiría otro tipo de lenguaje para una aplicación así.

**El IDE...** no es la mejor opción actualmente para desarrollar en Java ni en web ni en escritorio.

**Líneas futuras...** la tienda aún le queda mucho por implementar sobre todo a nivel de funcionalidad en el panel de administración como son: la gestión de compras, la gestión de usuarios, modificaciones en la tienda etc. Cosas que por tiempo se dejan sin hacer.