

1. Escribir una página JSP para “contactar” con nuestra organización. La página permitirá al usuario enviar un mensaje de texto a nuestra organización mediante un formulario que además, contendrá los siguientes campos obligatorios: Nombre, primer apellido, segundo apellido, ocupación, e-mail y empleado al que va dirigido el mensaje. Los mensajes recibidos se almacenarán en una base de datos junto con los datos del usuario, su dirección IP y la fecha y hora en que se realizó.
2. Realizar una pequeña aplicación que permita a los empleados de nuestra organización consultar los mensajes recibidos mediante la página del ejercicio anterior. El empleado deberá identificarse, y a continuación se mostrará un listado con los mensajes recibidos, ordenados por fecha. Cuando el empleado seleccione un mensaje, podrá consultar el texto de dicho mensaje y tendrá la opción de borrarlo. Si existen más mensajes del mismo remitente, se mostrarán las opciones “ver anterior..” y/o “Ver siguiente...”, según corresponda, que permitirán navegar entre los mensajes del mismo remitente.
3. Desarrollar en JSP una página de “login” para los empleados de una empresa. Para ello, ampliaremos la clase Empleado realizada en ejercicios anteriores (y la tabla de la BD) , con los campos login y password. En dicha página se pedirá al usuario que introduzca el login y el password de un empleado. Si éstos son correctos (coinciden con los de un empleado de la BD), se creará un objeto de tipo Empleado, se rellenarán sus campos a partir de los datos almacenados y se guardará dicho objeto en la sesión de usuario, para usarlo posteriormente en las demás páginas.
4. Incluir en todas las páginas de la sesión, una cabecera con información sobre el empleado conectado, y una opción de cerrar sesión. Comprobar previamente que la sesión ha sido validada en la página de “login”.
5. Añadir a la tabla y a la clase de empleados un campo “tipo” que podrá contener los valores “normal” o “administrador”. De esta forma sólo los empleados de tipo “administrador” podrán gestionar empleados.
6. Realizar un módulo para gestionar empleados (accesible sólo para los administradores), que permitirá consultar los empleados, crear, editar y borrar empleados.
7. Desarrollar en JSP una aplicación de chat sencilla. Los usuarios se identificarán con un nickname y accederán a la página de chat, donde dispondrán de un área para introducir información y otro para mostrar toda la conversación. Cuando el usuario introduzca una línea de texto y pulse “enviar”, se enviará dicha información al servidor, que la almacenará para reenviarla a todos los clientes conectados, incluyendo el que la envió. La página de chat se actualizará periódicamente, de forma que al hacer una nueva petición, el servidor le enviará la información actualizada. Como se trata de una aplicación sencilla no se controlan distintas conversaciones ni grupos. Todos los usuarios conectados al servidor estarán en la misma conversación.
8. Realizar en JSP una aplicación sencilla de comercio electrónico, en la que se ofrecerá a los clientes un catálogo de productos que se encuentran registrados en la Base de Datos de la aplicación. El cliente, que puede ser un usuario cualquiera, sin necesidad de registrar, entrará

en una página de búsqueda de productos, y a partir de ese momento dispondrá de un carrito de la compra al que podrá añadir productos con sus cantidades. En dicha página podrá realizar diferentes búsquedas que le mostrarán la lista de productos coincidentes. Pulsando sobre un producto se mostrarán los detalles del producto y se permitirá añadirlo al carrito. Cuando el cliente finalice la compra, deberá registrarse, indicando sus datos personales y la dirección de envío. En ese momento se almacenarán tanto los datos del usuario como los artículos comprados para su tramitación. En ésta aplicación no realizaremos el cobro del importe. A continuación se describen las tablas de la base de datos:

**Productos** (la tabla que contiene el catálogo de productos, y sólo se usará para consultar):

idproducto : int autonumérico  
descripcion : varchar(40)  
precio : decimal(6,2)  
urlFoto : varchar(40)

**Ventas** (almacena los datos del cliente para una venta):

idVenta : int autonumerico  
nombreCliente : varchar(40)  
direccionCliente : varchar(40)

**VentasProductos** (almacena cada detalle de una venta, es decir, un par producto- cantidad)

idVenta : int (clave foránea relacionada con Ventas)  
lineaVenta : int (número de línea dentro de la venta)  
idProducto : int (clave foránea relacionada con Productos)  
cantidad: int  
subtotal : decimal(6,2) (se calcula como cantidad \* precio del producto)

Para manejar la lógica de negocio vamos a disponer de los siguientes beans:

#### **Almacen**

*productos* : Lista de Producto (sólo lectura). Se obtienen los productos de la Base de Datos.

#### **Producto**

*id* : int (lectura/escritura)  
*descripcion* : String (lectura/escritura)  
*precio* : double (lectura/escritura)  
*urlFoto* : String (lectura/escritura)

#### **Carrito**

*nombreCliente* : String (lectura/escritura)  
*direccionCliente* : String (lectura/escritura)  
*lineas* : Lista de LineaVenta (sólo lectura)  
*nuevaLinea* : LineaVenta (sólo escritura) Añade una nueva línea a la compra.  
*total* : double (sólo lectura) devuelve el total de la compra.

#### **LineaVenta**

*idProducto* : int (lectura/escritura)  
*cantidad* : int (lectura/escritura)

*precio* : double (lectura)  
*descripcion* : String (lectura)  
*subtotal* : double (lectura)