

# CURSO JAVA

## PRÁCTICA FINAL: Tienda Online



### ENUNCIADO

La práctica final consistirá en el desarrollo de una aplicación web de una Tienda Online utilizando las tecnologías vistas a lo largo del curso. De manera general:

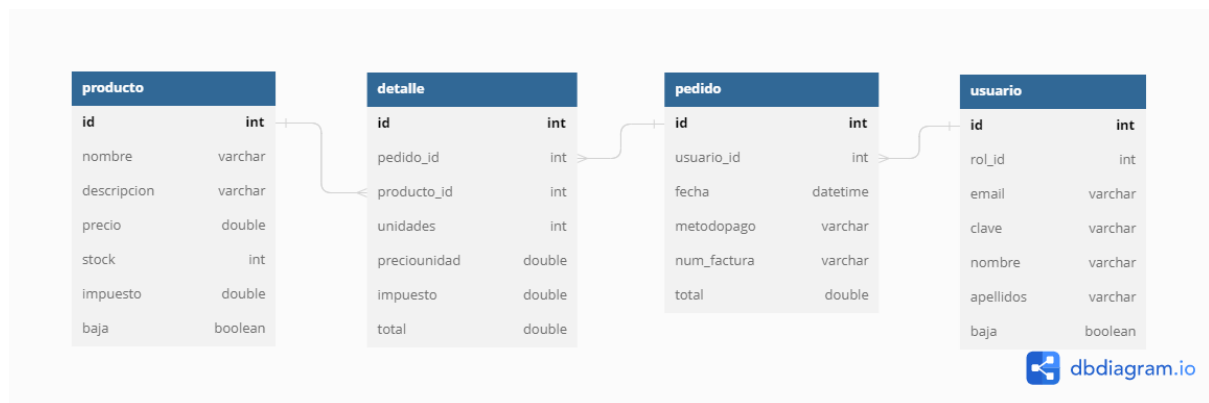
Front:

- HTML.
- CSS.
- JavaScript (y/o JQuery).
- Bootstrap

Back:

- Java EE (JSP y Servlet).
- Gestor de proyectos (Maven).
- Base de datos MySQL/MariaDB (JDBC o Hibernate).
- Uso de patrones (DAO, Singleton, etc.).
- Uso de librerías de terceros (excel, pdf, encriptación, etc.).

A continuación se detallan las entidades principales a manejar dentro del proyecto con la información inicial almacenada de cada una de ellas:



El diagrama anterior está disponible en la siguiente url, desde dónde se debe realizar una copia y ampliar con los datos utilizados dentro del proyecto propio:

<https://dbdiagram.io/d/643c257e6b31947051ad96b6>



### REQUISITOS DE FORMATO Y CONFIGURACIÓN

El proyecto se deberá desarrollar con el IDE Eclipse y tendrá el siguiente nombre: **TIENDA\_NOMBRE\_APELLIDO1\_APELLIDO2**, siendo NOMBRE tu nombre propio y APELLIDO1 y APELLIDO2 tus apellidos respectivamente separados por guiones bajos, en mayúsculas y todo sin tildes, ni caracteres especiales, ni eñes, ni espacios (si tu nombre y/o apellidos son compuestos, dejarlos juntos, es decir, sólo deben aparecer 2 guiones bajos).

El paquete base de las clases que formen el proyecto será el siguiente: **curso.java.tienda** (a partir de ahí puedes montar la estructura que necesites).

La versión del JDK con la que se compilará y entregará el proyecto será la **8u212**.

La entrada a la aplicación se hará directamente a través de la url base del proyecto (**http://localhost:8080/TIENDA\_NOMBRE\_APELLIDO1\_APELLIDO2**), no a otra subruta ni directamente al fichero index.jsp.

Los datos de conexión a utilizar para la base de datos deberán ser:

- servidor: **localhost**
- base de datos: **tienda\_nombre\_apellido1\_apellido2**
- usuario: **root** (sin contraseña)



### REQUISITOS 1

1. La web contará con una cabecera y un pie común a todas las páginas que componen la aplicación (uso de *includes*), así como de un menú para la navegación entre las diferentes páginas de la web.
2. Si un usuario de la aplicación no está dado de alta (usuario 'anónimo'), podrá registrarse en la aplicación a través del formulario correspondiente (tabla *Usuario*), tomando el rol 'cliente' por defecto. El nombre de usuario dentro de la aplicación será el propio email (campo *email*), debiendo de ser único dentro del sistema.
3. La tienda contará con un formulario de login, permitiendo al usuario 'cliente' logarse (uso de *sesiones*) en todo momento.
4. Al acceder a la página inicial se mostrará el catálogo completo de productos (tabla *Producto*), desde la cual se podrán comprar los productos ofertados.
5. Tanto el usuario 'anónimo' (sin logarse) como el usuario 'cliente' (ya logado), podrán ver el catálogo de productos de la tienda y añadir productos al carrito de la compra desde la vista del catálogo.
6. El carrito estará accesible en todo momento desde cualquier punto de la aplicación, y desde allí se podrán incrementar o decrementar las unidades de los productos almacenados en el propio carrito.
7. Se debe controlar el stock de los productos del catálogo, no permitiendo comprar más productos de los disponibles.
8. Se simulará el proceso de compra (no es necesario enlazar con una pasarela de pago real), obligando al usuario a logarse antes de realizar el pedido (solo usuarios con el rol 'cliente'). Si el usuario no está dado de alta en el sistema, se le obligará a realizar dicha alta. Inicialmente se dispondrá de dos métodos de pago: 'tarjeta' o 'paypal' (campo *metodopago*).
9. Al realizar la compra correctamente, el pedido realizado (tabla *Pedido*) quedará registrado para el usuario 'cliente' logado, vaciando el carrito y volviendo a la vista principal de la tienda.



10. Cada uno de los pedidos estará formado por varias líneas de pedido (tabla *DetallePedido*), asociadas cada una de estas a un producto y al número de unidades compradas de dicho producto.



### REQUISITOS 2

11. Se podrá navegar a través del catálogo, pudiendo ver los detalles de los productos pinchando en cada uno de ellos desde la vista del catálogo.
12. Se podrán añadir productos al carrito de la compra, tanto desde la vista del catálogo como desde la del detalle del producto.
13. La aplicación contará con un formulario de contacto desde el cuál el usuario podrá realizar consultas.
14. Los productos estarán agrupados por categorías (tabla *Categoria*).
15. En la página principal existirá un filtro de búsqueda de productos del catálogo por diferentes características: precio, categoría, mejor valorados, más comprados, etc.
16. Validaciones de datos: se realizarán comprobaciones de los datos introducidos por los usuarios, tanto en el lado del cliente como en el lado del servidor (en un solo formulario).
17. Doble validación de contraseña: tanto en el proceso de alta de los usuarios como en el cambio de contraseña.
18. La aplicación contará con un sistema de log para el registro de las trazas (uso de *log4j*).
19. Encriptación de la contraseña: se realizará una encriptación de la clave de acceso en el proceso de alta de los usuarios.
20. Internacionalización de la aplicación (*i18n*): la aplicación estará disponible en español e inglés y preparada para la ampliación a otros idiomas.
21. Los datos de la tienda (nombre, cif, dirección, ...), que aparecerán tanto en la propia web como en las facturas generadas, estarán almacenados en una tabla *Configuracion*, formada por dos campos, clave y valor. Será una especie de mapa dónde se guardarán datos explícitos de la aplicación.
22. El número que aparecerá en la factura será incremental. Para llevar un contador del número de factura se recomienda hacer uso de la tabla *Configuracion*.

# CURSO JAVA

## PRÁCTICA FINAL: Tienda Online



Funcionalidad dependiendo del rol del usuario

### **Cliente**

- 23. Podrá ver y actualizar los datos de su perfil de usuario, incluida su contraseña de acceso a la aplicación.
- 24. Podrá ver su historial de pedidos realizados, pudiendo filtrarlos por fecha.
- 25. Podrá ver el detalle de cada uno de los pedidos realizados.



### REQUISITOS 3

26. Existirán 3 roles (tabla *Rol*) de usuario (cliente, empleado, administrador), además del rol por defecto 'anónimo' cuando el usuario no está logado.
27. Una vez que el usuario está logado, la funcionalidad disponible, a través de las opciones del menú de la aplicación (tabla *OpcionMenu*), vendrá determinada por el rol de dicho usuario.
28. Se hará uso de una plantilla Bootstrap *responsive* para la maquetación de las vistas de la aplicación. La tienda contará con una interfaz diferenciada para el manejo de la aplicación desde los roles 'empleado' y 'administrador'.
29. Los pedidos realizados podrán encontrarse en alguno de los siguientes estados (campo *estado* de la tabla *Pedido*): 'pendiente envío' (PE), 'pendiente cancelación' (PC), 'enviado' (E) o 'cancelado' (C).
30. El número de factura únicamente se poblará cuando el pedido pase a estado 'enviado', de la misma forma que el total de la factura (tabla *Pedido*). Hay que tener en cuenta que el número de la factura no se corresponde con el id de los pedidos ya que éstos pueden quedarse en estado 'cancelado', por lo que nunca tendrán un número de factura.

### Funcionalidad dependiendo del rol del usuario

#### Cliente

31. El usuario con el rol 'cliente' podrá ver el estado de los pedidos realizados desde su historial de pedidos.
32. Cancelar pedido: podrá solicitar la cancelación total del pedido realizado (a través del historial de pedidos).

#### Empleado

33. Gestión de productos (altas, actualizaciones).
34. Gestión de clientes (altas, actualizaciones).
35. Gestión de categorías (altas, actualizaciones).
36. Procesar pedidos: validará los pedidos de los clientes (cambio de estado del pedido de 'pendiente envío' a 'enviado').

# CURSO JAVA

## PRÁCTICA FINAL: Tienda Online



### **Administrador**

Además de las funciones descritas para el usuario con rol 'empleado':

- 37. Gestión de productos (bajas lógicas).
- 38. Gestión de clientes (bajas lógicas).
- 39. Gestión de empleados (altas, actualizaciones, bajas lógicas).
- 40. Procesar cancelaciones: validará las cancelaciones de los pedidos solicitadas por los clientes (cambio de estado del pedido de 'pendiente cancelación' a 'cancelado').





### REQUISITOS 4

41. Al iniciarse la aplicación se registrará automáticamente, si no existiera ya en la base de datos, un usuario con el rol 'administrador' con el nombre *admin* y cuya clave de acceso sea *admin*, obligando a cambiar la contraseña en el primer acceso de dicho usuario a la aplicación. Será el único usuario que podrá gestionar el resto de usuarios con el rol 'administrador'.
42. Procesado automático de pedidos (hilos): cada cierto tiempo el sistema procesará los pedidos en estado 'pendiente envío' (cambio de estado a 'enviado'). La opción disponible por la cual un usuario 'empleado'/'administrador' puede procesar los pedidos manualmente, seguirá estando activa.
43. Mensajería dinámica (hilos): en el caso de que varios clientes estén conectados a la tienda y añadan el mismo artículo a sus carritos, el sistema deberá comprobar si la suma de esas unidades supera la cantidad de stock, mostrando un mensaje modal en cada cliente avisando de que varios clientes tienen el mismo artículo y que es posible que si no se cierra su compra no pueda conseguir dicho artículo.

### Funcionalidad dependiendo del rol del usuario

#### Cliente

44. Cancelación parcial: podrá cancelar una o varias líneas de un pedido mientras dicho pedido se encuentre en estado 'pendiente envío'.
45. Notificaciones automáticas: se enviará un email al usuario cuando un pedido realizado pase al estado 'enviado'.
46. Descarga de facturas: se podrá descargar la factura de un pedido en formato pdf únicamente cuando se encuentre en estado 'enviado'.
47. Valoración de productos: podrá valorar sólo aquellos productos que haya comprado, accediendo a través del historial de pedidos y una vez que el producto esté en estado 'enviado' (tabla *Valoracion*).

#### Empleado

48. Exportación de productos: se podrá generar un listado, en formato excel, del catálogo de productos que tiene la tienda actualmente.

# CURSO JAVA

## PRÁCTICA FINAL: Tienda Online



- 49. Podrá ver y actualizar los datos de su perfil de usuario, incluida su contraseña de acceso a la aplicación.
- 50. Gestión de proveedores (altas, actualizaciones, bajas lógicas). Se almacenará la información de los proveedores de los productos (tabla *Proveedor*).

### **Administrador**

Además de las funciones descritas para el usuario con rol 'empleado':

- 51. Importación de productos: a partir de una excel, se podrán importar masivamente un listado de nuevos productos.
- 52. Modificación de datos de configuración de la aplicación (tabla *Configuracion*).
- 53. Visualización de estadísticas: el administrador dispondrá de estadísticas de ventas, promociones y valoraciones.
- 54. Gestión de descuentos: podrá establecer códigos de descuento que el cliente podrá utilizar a la hora de realizar el pago (tabla *Descuento*).
- 55. Gestión de administradores (altas, actualizaciones, bajas lógicas): solo el usuario *admin* (superadministrador).



### REQUISITOS DE DOCUMENTACIÓN Y ENTREGA

Manual técnico: Documentación del código generado con *Javadoc*. Se deberá almacenar en una carpeta **/javadoc** en la raíz del proyecto.

Documentación de la aplicación: El proyecto deberá ir acompañado de la correspondiente documentación, que se almacenará en una carpeta **/documentacion** en la raíz del proyecto.

Dicha documentación deberá contener lo siguiente:

- Requisitos: Se deberá realizar una descripción funcional de la aplicación así como documentar aquellos requisitos extra desarrollados de manera personal. Se creará un documento de texto con dicha información y se incluirá el pdf generado.
- Manual de usuario: Se deberá realizar un manual con capturas de pantalla y una breve descripción del uso de la aplicación. Se incluirá el pdf generado.
- Diagrama de bases de datos: Se incluirá el enlace público al esquema del diagrama desarrollado en *dbdiagram.io*.
- Registro de tareas: Se incluirá el enlace público al tablero de *Trello* utilizado para registrar las tareas del proyecto.
- Informe del proyecto: Se incluirá el pdf generado con el informe completo de las tareas del proyecto registradas con *Toggl*.
- Script de generación de la base de datos, tablas y datos iniciales necesarios para su funcionamiento (**tienda\_nombre\_apellido1\_apellido2.sql**).
- Todo aquello que se considere relevante en la elaboración del proyecto.

Una vez se vaya a realizar la entrega, **se comprimirá el proyecto completo** en un .rar o .zip y se subirá a la plataforma en la tarea de entrega correspondiente, o bien se adjuntará el enlace público al repositorio del proyecto en GitHub.