

PRÁCTICA 3

Persistencia de datos con JPA en Pizzeria App

Curso 2024 - 2025

UNIDAD 3. PERSISTENCIA DE DATOS CON JPA E HIBERNATE

ACCESO A DATOS – 2º DAM

I.E.S THIAR

Implementa en Java los siguientes ejercicios, utilizando como base la práctica anterior. Al finalizar, adjunta el proyecto comprimido en un zip en la tarea con esta nomenclatura: Apellido1-Nombre-Practica3.zip. Para la realización de esta práctica hay que seguir las buenas prácticas de programación:

- Seguir las “naming conventions” de Java.
- Gestionar de forma eficiente los recursos externos.
- Control de posibles excepciones.
- Encapsulamiento.
- Principio de Responsabilidad Única.
- Baja complejidad ciclomática. Mantener simplicidad de las funciones.
- Se deben testear las funcionalidades de la aplicación.

El esquema de la BD ya estará creado previamente con el nombre “pizzeria-jpa”, ofreciéndose el servicio en el puerto 3306 con el motor de MySQL. El esquema estará vacío, por lo que la responsabilidad de crear las tablas, insertar registros en ellas, leerlos, actualizarlos y borrarlos recaerá totalmente en la tecnología JPA.

Requisitos de la aplicación:

- La actualización y borrado de registros será en Cascada.
- La recuperación de relaciones en las OneToMany y ManyToMany será carga “Lazy” o “Eager” según los requisitos establecidos.
- La arquitectura de la aplicación seguirá los patrones de diseño MVC y DAO.
- Se aplicará la estrategia de mapeo para la herencia de tabla única (“single table”).
- Se deben controlar los casos de error.
- Las fechas deben contener únicamente la fecha

Funcionalidades de la aplicación:

1. **Registro de cliente.** Se registra en la base de datos con la información del modelo.
2. **Login.** El cliente se loguea y recibimos ese objeto cliente de la base de datos.
3. **Catálogo de Productos.** El cliente puede añadir los productos uno a uno a la base de datos y puede visualizarlos todos ellos (“Catálogo de productos”).
4. **Añadir al carrito.** El cliente visualiza el Catálogo de Productos y genera una línea de pedido al seleccionar un producto, cantidad y tamaño. En caso de existir un pedido en estado pendiente se añade la línea a este pedido y en caso contrario se crea un pedido nuevo donde se añade la línea.
5. **Finalizar pedido.** Se recibe cliente y el método de pago, se busca el único pedido pendiente que debería haber y se finaliza.
6. **Cancelar pedido.** Se recibe cliente se busca el único pedido pendiente que debería haber y se cancela.
7. **Entregar pedido.** Se recibe id del pedido, se busca en la base de datos el pedido y se entrega.

(2 puntos). Persistencia de clientes. Debe implementarse un CRUD, teniendo en cuenta que el cliente NO debe contener los pedidos. Además, hay que implementar las siguientes operaciones:

- Obtener por email. Dado el email del cliente, debe devolver el cliente del sistema asociado a ese email.
- Obtener todos. Debe devolver todos los clientes del sistema.

(4 puntos). Persistencia de productos. Debe implementarse un CRUD en el que se utilice la estrategia “single table” y cuando se persista el producto, se deben persistir sus ingredientes y alérgenos, y utilizando tablas intermedias EFICIENTES. Ten en cuenta que el producto debe contener los ingredientes y sus alérgenos con estrategia Eager. Además del CRUD se deben implementar las siguientes operaciones:

- Obtener todos. Debe devolver todos los productos del sistema.
- Obtener ingredientes por producto. Dado un id de producto debe devolver todos sus ingredientes.
- Obtener alérgenos por ingrediente. Dado un id de ingrediente debe devolver sus alérgenos.

(4 puntos). Persistencia de pedidos. Debe implementar un CRUD, que cuando persista el pedido persista sus líneas de pedido. El pedido debe contener las líneas de pedido sin los productos asociados. Además del CRUD se deben implementar las siguientes operaciones:

- Obtener pedidos por cliente. Dado el id de un cliente, debe devolver todos los pedidos del sistema asociados a ese cliente
- Obtener pedidos por estado. Dado un estado de pedido, debe devolver una lista con los pedidos en dicho estado.
- Obtener líneas pedido por pedido. Dado el id de un pedido, devuelve su lista de líneas de pedido.
- Agregar línea pedido. Persiste la línea de pedido y el pedido asociado en caso de no existir.