

Final Laboratorio III

El objetivo de este trabajo final es la implementación de un conjunto de API's REST en la plataforma Java, utilizando el framework Spring Boot que vimos durante el cursado de la materia.

Introducción

Para este ejercicio, se deberá realizar una aplicación en capas como la del [ejemplo](#) que seguimos durante el cursado de la materia. En este caso, estaremos implementando algunos de los endpoints requeridos para el [práctico](#) de retail que hicieran durante la cursada. Se debe tener en cuenta que deben generarse caso de test para **al menos** un método de cada una de las clases de cada capa (presentación, negocio y persistencia, las clases de modelo y dto's pueden obviarse ya que deberían ser clases simples con métodos get y set unicamente).

Las entidades (clases de modelo)

Las entidades con las que vamos a estar trabajando principalmente serán 2: Categorías y Productos. Las clases que las modelan ya deberían estar desarrolladas desde el práctico anterior y por supuesto pueden ser reutilizadas. Para las categorías, queremos contar con los siguientes atributos identificador (o código) de la misma, el nombre, la descripción y una lista de los productos de la misma. En el caso de los productos, la entidad deberá tener los siguientes atributos: identificador (o código), nombre, descripción, categoría a la que pertenece, marca, precio lista, tipo de producto y una lista de especificaciones para cada tipo de producto. Por ejemplo, para las TVs, la lista de especificaciones será: tipo de pantalla, pulgadas, alto, ancho, peso, tipos y cantidad de entradas (HDMI, USB, VGA), si es Smart TV, etc. Para una Bicicleta, indicará Rodado, cantidad de cambios, tamaño del cuadro, color del cuadro, etc.

Se debe crear un diagrama UML con la especificación de las clases necesarias para el modelado de nuestro retail.

Exposición de la información con APIs REST

Queremos crear endpoints REST para las siguientes operaciones. Debajo se indica cómo deben quedar los endpoints y a qué tipo de método HTTP deben responder y un ejemplo del mismo en algunos casos. Un tip para este punto es que deberán tener en cuenta no sólo el caso feliz sino que también tener en cuenta qué ocurre por ejemplo si no existen productos a mostrar, o no existe la categoría que quiero eliminar.

Categorías:

- Crear, Modificar y Eliminar una nueva categoría
POST: /categoria
PUT: /categoria/{identificador}
DELETE: /categoria/{identificador}
- Obtener todos los productos ordenados por precio (ascendente o descendente)
GET: /categoria?order_price=asc (u order_price=desc)
- Obtener los productos filtrando por marca
GET: /categoria?marca=samsung
- Obtener productos filtrando por precio
GET: /categoria?precio_min=35000&precio_max=6000

Productos:

- Crear, Modificar y Eliminar un producto.
POST: /producto
PUT: /producto/{identificador}
DELETE: /producto/{identificador}
- Consultar un producto por su identificador
GET: /producto/{identificador}
- Consultar un producto por alguno de sus atributos: tipo, marca, categoria
GET: /producto?tipo_producto=dvd&marca=samsung&categoria=ATV
GET: /producto?tipo_producto=tv&marca=phillips&categoria=ATV