Núcleo 3 Evaluación de una Aplicación WEB con Spring Boot Caso de estudio

Presentado por: Diego Fernando Riveros Vanegas

Angie Lizeth Moreno Celis

Mario Alejandro Rueda Arcos

Docente: German Andrés Cheque Molina

Bogotá, 27 de Abril de 2025

FACULTAD DE TÉCNICAS DE INGENIERÍA

Programa de Desarrollo de Software

Descripción e instrucciones para realizar la actividad

En grupos de al menos tres (3) personas resolveremos la siguiente situación:

Doña Rosa tiene una tienda en un barrio de la ciudad. En este barrio los datos de celular llegan continuamente y por la competencia entre operadores últimamente se puede evidenciar una disminución en el costo de los planes. Por lo cual, el acceso a internet no es un problema, sin embargo, la instalación de aplicaciones al dispositivo de Doña Rosa no es una opción, por la gama del dispositivo y por el espacio del mismo.

Doña Rosa vende diferentes productos en su tienda entre los cuales destacan los granos, las frutas y las legumbres, así como algunos dulces. Ella lleva sus datos de movimientos en un cuaderno, sin embargo, ha pensado en una aplicación que le permita llenar esos mismos datos sin necesidad de instalar nada y solo accediendo al navegador del dispositivo.

La tabla siguiente representa los productos que vende Doña Rosa en su tienda:

Código	Producto	Precio	Cantidad
1	Peras	4000	65
2	Limones	1500	25
3	Moras	2000	30
4	Piñas	3000	15
5	Tomates	1000	30
6	Fresas	3000	12
7	Frunas	300	50
8	Galletas	500	400
9	Chocolates	1200	500
10	Arroz	1200	60

Versiones

Apache-Neatbens 23

Spring-boot 3.4.5

MySQL conector 8.0.17

XAMPP V3.3.0

Postman

URLS

http://localhost:8080/productos

Delete
http://localhost:8080/productos/6

Put

http://localhost:8080/productos/6

Post

http://localhost:8080/productos/

Get

http://localhost:8080/productos

En este trabajo desarrollamos una API REST utilizando **Spring Boot** y **Spring Data JPA** para la gestión de productos dentro de una tienda virtual. Se implementaron operaciones básicas de **CRUD (Create, Read, Update, Delete)** y se verificó el correcto funcionamiento de cada endpoint mediante **Postman**.

Estructura del Proyecto

Seguimos el modelo **MVC (Model-View-Controller)**, organizando el código en las siguientes capas:

- **Modelo (Producto.java)** → Representación de los datos almacenados en la base de datos.
- **Repositorio (ProductoRepository.java)** → Manejo de las consultas a la base de datos con **Spring Data JPA**.
- **Controlador (ProductoController.java)** → Gestión de las solicitudes HTTP para insertar, leer, actualizar y eliminar productos.

Endpoints Implementados

Ver todos los productos (GET) → http://localhost:8080/productos
Ver un producto por ID (GET) → http://localhost:8080/productos/{id}
Insertar un nuevo producto (POST) → http://localhost:8080/productos
Actualizar un producto (PUT) → http://localhost:8080/productos/{id}
Eliminar un producto (DELETE) → http://localhost:8080/productos/{id}

Pruebas Realizadas

Se verificó el listado de productos mediante GET /productos.
Se probó la inserción de nuevos productos con POST.
Se actualizaron productos con PUT y se verificaron los cambios.
Se eliminaron productos con DELETE, asegurando que ya no aparecían en la base de datos.

Conclusión

Este desarrollo permitió establecer una API funcional para la gestión de productos, cumpliendo con los principios de **REST** y **Spring Boot**. Las pruebas confirmaron su correcto funcionamiento, asegurando que los datos se manejan de manera eficiente en la base de datos.

Este resumen es ideal para agregar al documento Word y completar con capturas de pantalla de las pruebas realizadas.

¿Quieres incluir algún otro detalle?