

Actividad [3] - [Funcionamiento del carrito]

[Desarrollo de Sistemas Web II]

Ingeniería en Desarrollo de Software

Tutor: Aarón Iván Salazar Macías

Alumno: Carlos Fco Estrada Salazar

Fecha: 24/02/2025

INDICE

INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	6
Codificación	6
Pruebas del sitio	9
CONCLUSIÓN	11
REFERENCIAS	12

GitHub Link:

INTRODUCCIÓN

En el desarrollo de sistemas web, la implementación de un carrito de compras es un elemento clave dentro de una tienda en línea, ya que permite a los clientes seleccionar productos, calcular el total de su compra y gestionar su pedido antes de realizar el pago. En esta última etapa del proyecto para la tienda Sara, se enfocará en la funcionalidad del carrito de compras mediante la integración de una API REST, permitiendo la administración eficiente de los productos dentro de la base de datos y su correcta visualización en el sistema web.

El proceso de desarrollo se basará en la estructura de la base de datos creada en la primera actividad, la cual organiza la información de los productos, categorías, marcas y el carrito de compras, garantizando una estructura relacional que facilite la gestión de los datos. Posteriormente, en la segunda actividad, se diseñó el sitio web con una interfaz intuitiva, asegurando que los usuarios pudieran navegar de manera sencilla y realizar sus compras de forma eficiente. Ahora, en esta tercera actividad, se trabajará en la implementación de los servicios que permitirán realizar las operaciones fundamentales del carrito de compras: agregar, modificar, eliminar y consultar productos dentro del mismo.

Para lograrlo, se utilizará el framework Spring Boot en conjunto con Java, creando una API REST que facilite la interacción entre el sistema web y la base de datos. Mediante este desarrollo, se garantizará que el sistema web refleje en tiempo real los cambios realizados en el carrito de compras, permitiendo a los usuarios visualizar y gestionar sus productos antes de concretar la compra. Con esto, se culminará la creación del e-commerce de la tienda Sara, proporcionando un sitio funcional y optimizado para la venta en línea.

DESCRIPCIÓN

En esta tercera y última etapa del proyecto, el objetivo principal es dotar de funcionalidad al carrito de compras dentro del e-commerce de la tienda Sara. Hasta este punto, se han completado dos etapas clave: la creación de la base de datos relacional en Microsoft SQL Server Management y el diseño del sitio web en Java con Spring Boot. Ahora, se busca implementar las operaciones esenciales del carrito mediante una API REST, permitiendo a los usuarios agregar, modificar, eliminar y consultar productos en tiempo real.

El carrito de compras es un elemento central en cualquier plataforma de comercio electrónico, ya que permite a los clientes gestionar sus productos antes de realizar una compra. Para lograr esta funcionalidad, se trabajará con los datos previamente estructurados en las tablas `Carrito` y `CarritoDetalle`, estableciendo relaciones con las tablas de productos, categorías y marcas. De esta manera, se asegurará un flujo eficiente de la información dentro del sistema.

La API REST será el puente de comunicación entre la base de datos y la interfaz de usuario, permitiendo que las acciones realizadas por el cliente se reflejen en tiempo real. Esto implica la implementación de métodos para registrar nuevos productos en el carrito, modificar cantidades seleccionadas, eliminar artículos y calcular automáticamente el total de la compra.

Esta etapa culminará con la integración completa del sistema, asegurando que el carrito de compras funcione correctamente dentro del sitio web. Con esto, se garantizará que la tienda Sara cuente con una plataforma funcional y optimizada para la venta en línea, mejorando la experiencia de usuario y facilitando la gestión de compras.

JUSTIFICACIÓN

La implementación de una API REST para gestionar el funcionamiento del carrito de compras en la tienda en línea de Sara es una solución eficiente y escalable. Este enfoque permite la integración fluida entre la base de datos y la interfaz web, asegurando una experiencia de usuario óptima y un manejo estructurado de la información en tiempo real.

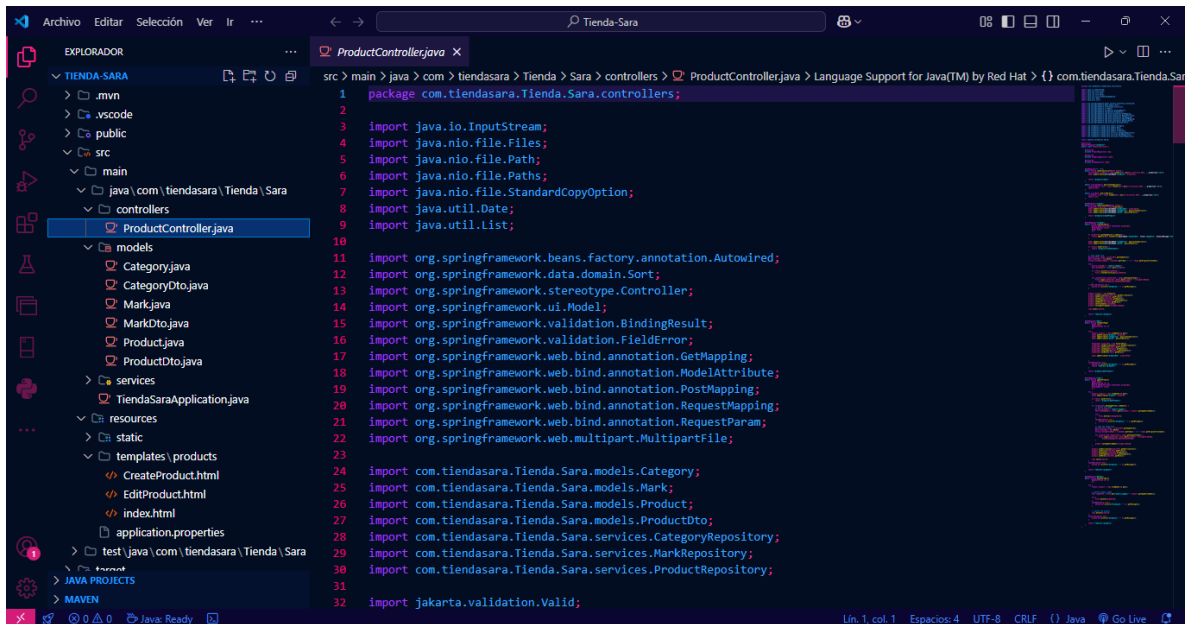
En primer lugar, el uso de una API REST facilita la comunicación entre el frontend y el backend del sistema, permitiendo que las operaciones de alta, modificación, consulta y eliminación de productos en el carrito se realicen de manera dinámica y eficiente. Esto es esencial en un entorno de comercio electrónico, donde los usuarios esperan actualizaciones inmediatas en su compra sin necesidad de recargar la página o enfrentar largos tiempos de espera.

Este modelo basado en microservicios mejora la mantenibilidad y escalabilidad del sistema. Al separar la lógica del carrito en una API independiente, se facilita la implementación de futuras mejoras o integraciones con otros sistemas, como pasarelas de pago o módulos de inventario automatizado. Asimismo, permite que diferentes clientes, como aplicaciones móviles o interfaces web, interactúen con la tienda de manera unificada.

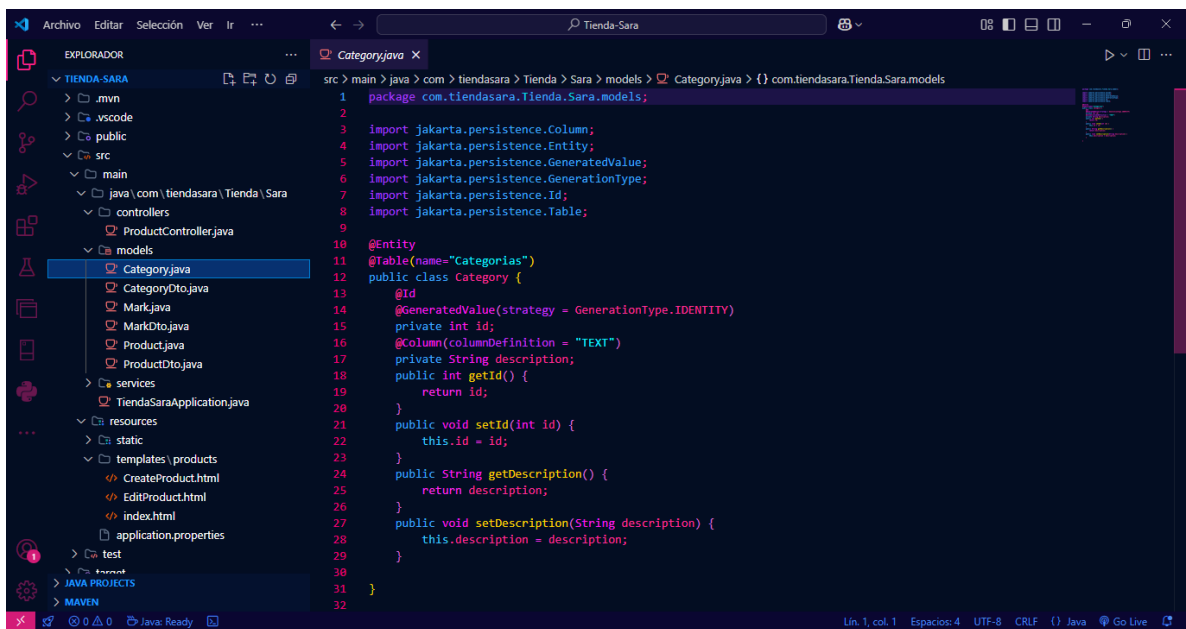
Desde un punto de vista técnico, la implementación con Java y Spring Boot garantiza robustez y seguridad en la gestión de datos. Spring Boot proporciona herramientas que optimizan el manejo de solicitudes HTTP, la autenticación de usuarios y la validación de datos, reduciendo riesgos de errores y vulnerabilidades en el sistema.

Emplear una API REST para el carrito de compras no solo mejora la experiencia del usuario, sino que también optimiza el rendimiento del sistema, permitiendo una gestión eficiente de los productos y asegurando la escalabilidad futura del e-commerce de la tienda Sara.

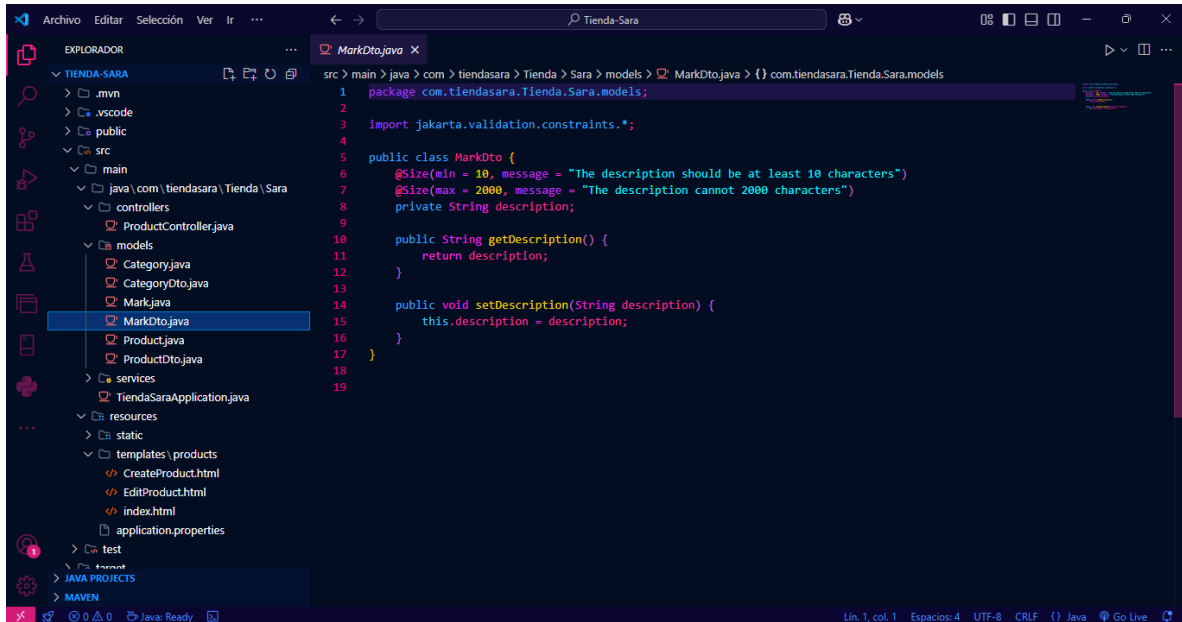
DESARROLLO Codificación



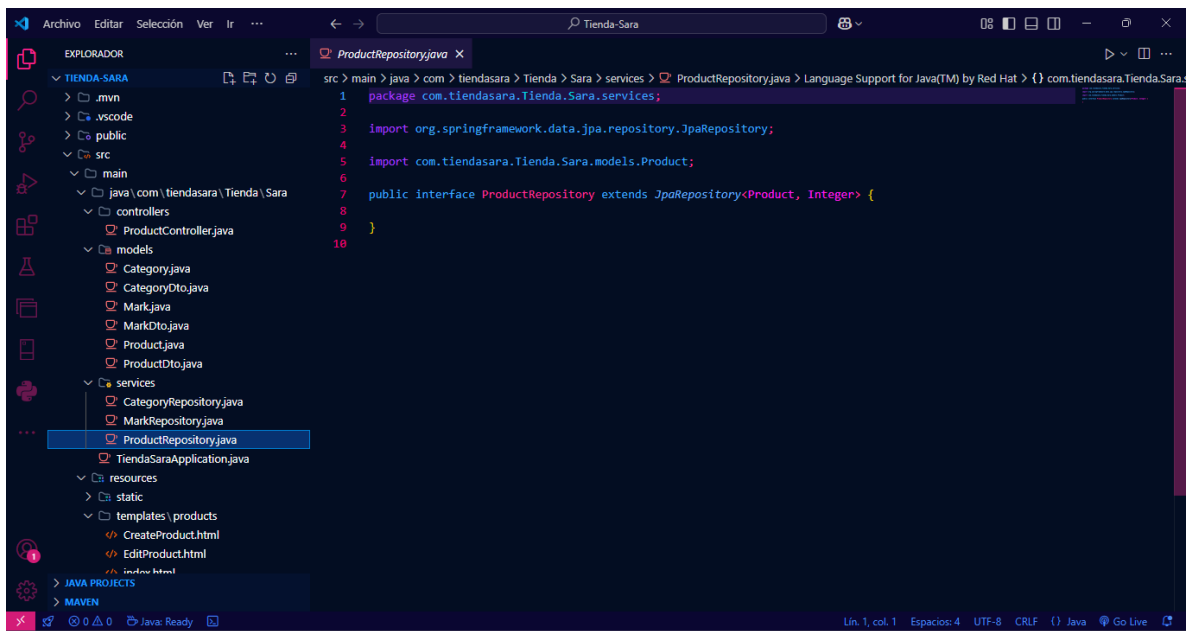
Configuración del controlador



Configuración de los modelos “Category”



Configuración de los modelos “MarkDto”



Configuración de los servicios “ProductRepository”

```

src > main > resources > templates > products > index.html
 2 <html lang="en">
 9 <body>
10 <div class="container">
15
16 <table class="table">
17 <thead>
18 <tr>
19 <th>ID</th>
20 <th>Description</th>
21 <th>Price</th>
22 <th>Amount</th>
23 <th>Category</th>
24 <th>Mark</th>
25 <th>Image</th>
26 <th>Created At</th>
27 <th>Action</th>
28 </tr>
29 </thead>
30 <tbody>
31 <tr th:each="product : ${products}">
32 <td th:text="${product.id}"></td>
33 <td th:text="${product.description}"></td>
34 <td th:text="${product.price}"></td>
35 <td th:text="${product.amount}"></td>
36 <td th:text="${product.category.description}"></td>
37 <td th:text="${product.mark.description}"></td>
38 <td>
39 
41 </td>
42 <td th:text="${product.createdAt.toString().substring(0, 10)}"></td>

```

Configuración del Template de products “index”

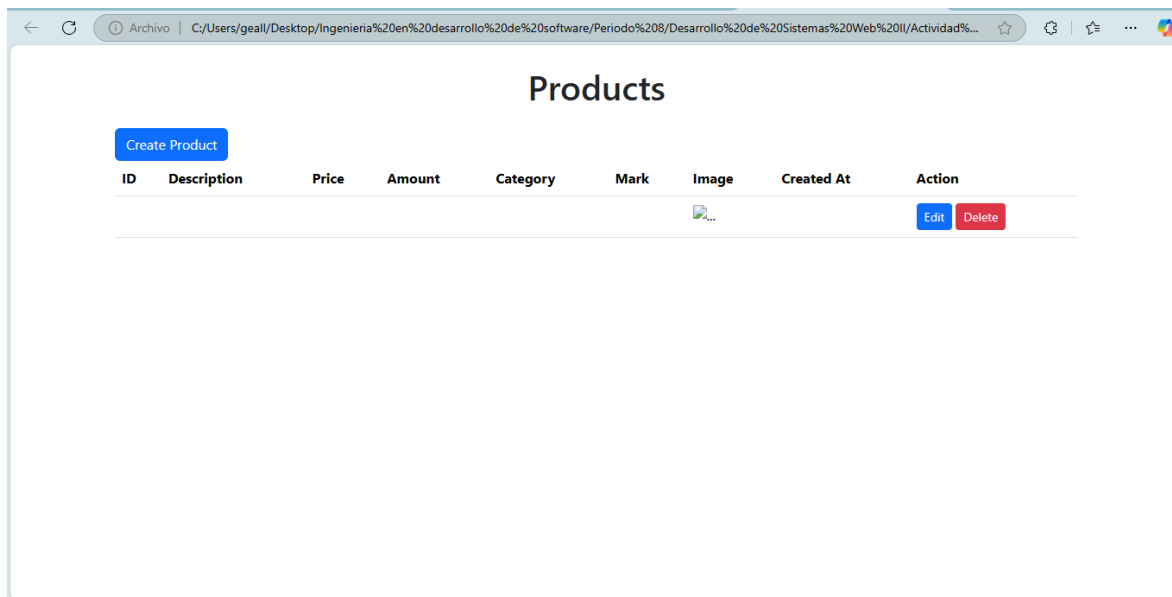
```

src > main > resources > templates > products > CreateProduct.html
 2 <html lang="en">
 9 <body>
10 <div class="container">
12 <div class="col-md-8 mx-auto rounded border p-4 m-4">
15 <form method="post" enctype="multipart/form-data" th:object="${productDto}">
17 <label class="col-sm-4 col-form-label">Description</label>
18 <div class="col-sm-8">
19 <div class="form-control" th:field="${productDto.description}"></div>
20 <p th:if="${#fields.hasErrors('description')}" th:errorclass="text-danger">
21 <th:errors="${productDto.description}"></p>
22 </div>
23 </div>
24
25 <div class="row mb-3">
26 <label class="col-sm-4 col-form-label">Price</label>
27 <div class="col-sm-8">
28 <input class="form-control" type="number" step="0.01" min="0" th:field="${productDto.price}">
29 <p th:if="${#fields.hasErrors('price')}" th:errorclass="text-danger">
30 <th:errors="${productDto.price}"></p>
31 </div>
32 </div>
33
34 <div class="row mb-3">
35 <label class="col-sm-4 col-form-label">Amount</label>
36 <div class="col-sm-8">
37 <input class="form-control" type="number" step="0.01" min="0" th:field="${productDto.amount}">
38 <p th:if="${#fields.hasErrors('amount')}" th:errorclass="text-danger">
39 <th:errors="${productDto.amount}"></p>
40 </div>
41 </div>
42

```

Configuración del Template de products “CreateProduct”

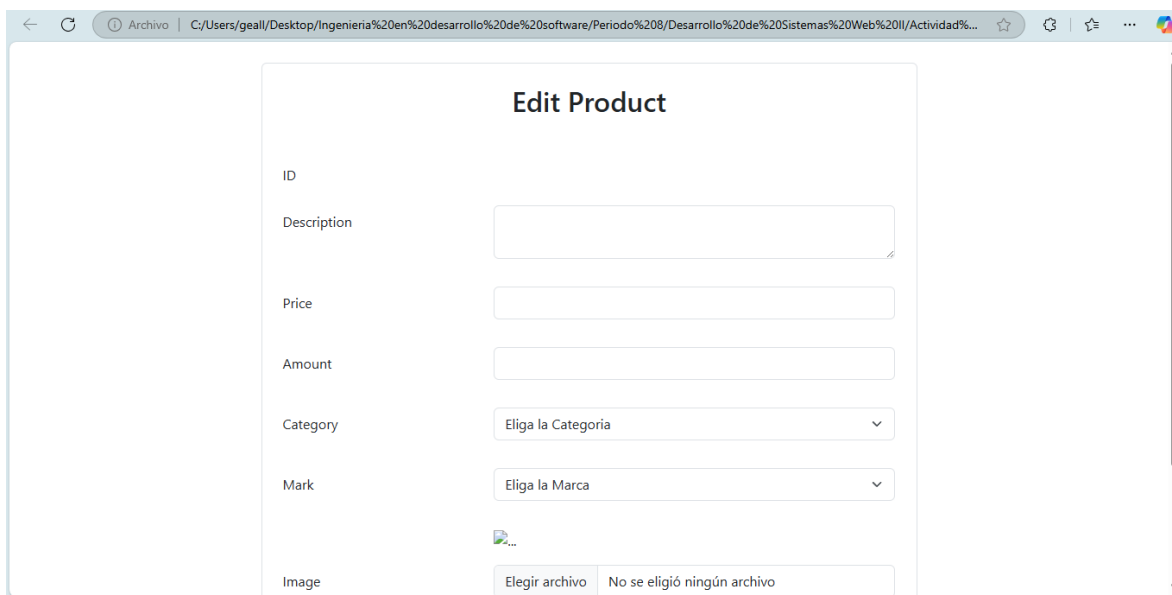
Pruebas del sitio



Pantalla Products

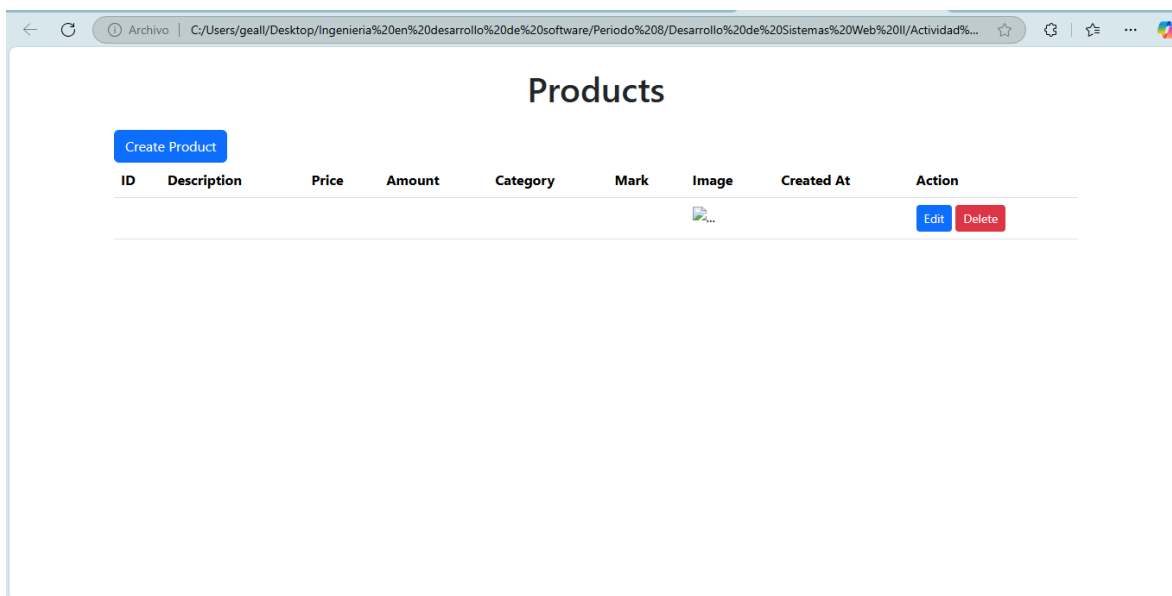
The screenshot shows a web browser window with the address bar displaying a file path. The main content area is titled "New Product". Below the title is a form with the following fields: Description (text input), Price (text input), Amount (text input), Category (dropdown menu with "Eliga la Categoria" selected), Mark (dropdown menu with "Eliga la Marca" selected), and Image (file upload field with "Elegir archivo" and "No se eligió ningún archivo" buttons). At the bottom of the form are two blue buttons: "Submit" and "Cancel".

Pantalla CreateProduct



The screenshot shows a web browser window with the address bar displaying a file path. The main content area is titled "Edit Product". It contains a form with the following fields: ID (text input), Description (text input), Price (text input), Amount (text input), Category (dropdown menu with "Eliga la Categoría" selected), Mark (dropdown menu with "Eliga la Marca" selected), and Image (file upload area with a "Elegir archivo" button and a message "No se eligió ningún archivo").

Pantalla EditProduct



The screenshot shows a web browser window with the address bar displaying a file path. The main content area is titled "Products". It features a "Create Product" button and a table with the following columns: ID, Description, Price, Amount, Category, Mark, Image, Created At, and Action. The table is currently empty, and the "Action" column contains "Edit" and "Delete" buttons.

ID	Description	Price	Amount	Category	Mark	Image	Created At	Action
								Edit Delete

No se logró realizar la conexión con la base de datos

CONCLUSIÓN

La realización de esta actividad no solo ha permitido desarrollar habilidades técnicas clave en la implementación de bases de datos, diseño web y desarrollo de API REST, sino que también ha demostrado la importancia de estos conocimientos en el ámbito laboral y la vida cotidiana. En el mundo actual, donde el comercio electrónico juega un papel fundamental en la economía, contar con plataformas eficientes y escalables es esencial para garantizar una experiencia de usuario óptima y una gestión adecuada de los recursos empresariales.

En el campo laboral, el dominio de herramientas como Microsoft SQL Server, Java y Spring Boot permite a los profesionales del desarrollo de software diseñar y optimizar sistemas complejos que requieren un alto nivel de seguridad y eficiencia. La correcta estructuración de una base de datos y su integración con una API REST facilita la manipulación de información en tiempo real, lo que es crucial en entornos de ventas en línea.

Asimismo, la capacidad de diseñar y programar sistemas de carrito de compras con funciones de alta, modificación, consulta y eliminación de productos es altamente demandada en el sector tecnológico. Esto no solo permite mejorar la operatividad de las tiendas en línea, sino que también abre oportunidades para la automatización y personalización de los procesos de compra, optimizando la conversión de ventas.

Esta actividad ha reforzado la importancia de la planificación, el diseño y la implementación de soluciones web basadas en microservicios, habilidades que son esenciales en la era digital y que pueden aplicarse en múltiples contextos, tanto en el desarrollo profesional como en el uso cotidiano de plataformas tecnológicas.

REFERENCIAS

- Caules, C. Á. (2025, February 16). *Spring Boot ¿Qué es y cómo funciona?* Arquitectura Java. <https://www.arquitecturajava.com/spring-boot-que-es/>
- Secureuser. (2023, April 18). <http://localhost:8080>. *localhost*. <https://locall.host/8080/>