### Documentación pruebas de caja negra

### Índice

- 1. Introducción a las Pruebas de Caja Negra
- 2. Objetivos de las Pruebas de Caja Negra
- 3. Casos de Prueba Propuestos
  - a. 3.1 Pruebas para Crear Productos (POST)
  - b. 3.2 Pruebas para Consultar Productos (GET)
  - c. 3.3 Pruebas para Actualizar Productos (PUT)
  - d. 3.4 Pruebas para Eliminar Productos (DELETE)
- 4. Criterios de Aceptación
- 5. Conclusiones

### 1. Introducción a las Pruebas de Caja Negra

Para este proyecto, realizaremos pruebas de caja negra sobre las **operaciones CRUD** (Crear, Leer, Actualizar y Eliminar) que forman parte del sistema.

### 2. Objetivos de las Pruebas de Caja Negra

- Validar las operaciones CRUD mediante la interacción con los endpoints del backend y la interfaz gráfica en Vue.js.
- **Identificar errores en las respuestas** del servidor, validaciones incorrectas, y flujos de usuario fallidos.
- Comprobar la correcta integración entre frontend (Vue.js) y backend (Flask).
- Verificar que el manejo de errores sea adecuado en casos inesperados (como datos faltantes o solicitudes no permitidas).

# 3. Casos de Prueba Propuestos

Aquí se presentan los **casos de prueba de caja negra** categorizados por cada operación CRUD (POST, GET, PUT, DELETE).

#### 3.1 Pruebas para Crear Productos (POST)

ID	Caso de Prueba	Entrada	Salida Esperada	Estado
TC-001	Crear un producto	Nombre:	Respuesta: 201, Mensaje:	<b>✓</b>
	válido	"Laptop",	"Producto creado"	
		Precio:		
		1200		
TC-002	Crear un producto	Nombre: "",	Respuesta: 400, "El nombre	X
	sin nombre	Precio:	es obligatorio"	
		1200		
TC-003	Crear un producto	Nombre:	Respuesta: 400, "El precio	X
	sin precio	"Laptop",	es obligatorio"	
		Precio: ""		
TC-004	Enviar datos no	Nombre:	Respuesta: 400, "Formato	X
	válidos (string en vez	"Laptop",	inválido"	
	de número)	Precio: "mil		
		pesos"		

Adjunto algunas capturas de pantalla respecto a las pruebas realizadas

#### Captura TC-001:

# Gestión de Productos

## **Agregar Producto**

Obj 1	120	<b>‡</b>	Guardar
ID Nombre Precio Accion	05		

# Gestión de Productos

# **Agregar Producto**

Nombre		0		Guardar		
ID Nombre Precio Acciones						
1 Obj 1	120	Editar	Eliminar			

127.0.0.1 - - [15/Oct/2024 14:02:57] "POST /products HTTP/1.1" 201 - 127.0.0.1 - - [15/Oct/2024 14:02:57] "GET /products HTTP/1.1" 200 -

#### Captura TC-002:

# Gestión de Productos

### **Agregar Producto**



#### Captura TC-003:

### Gestión de Productos

### **Agregar Producto**



#### Captura TC-004:

# Gestión de Productos

# **Agregar Producto**



El campo esta realiza para que reciba exclusivamente tipos de datos "Int"

#### 3.2 Pruebas para Consultar Productos (GET)

ID	Caso de Prueba	Entr	Salida Esperada	Estado
		ada		

TC-005	Obtener la lista de productos	N/A	Lista de productos con	<b>✓</b>
			código 200	

Adjunto capturas de pantalla de las pruebas realizadas:

# Captura TC-005:

ID Nombre Precio Acciones						
1 Obj 1	120	Editar	Eliminar			
2 Obj 2	240	Editar	Eliminar			
3 Obj 3	360	Editar	Eliminar			
4 Obj 4	480	Editar	Eliminar			
5 Obj 5	670	Editar	Eliminar			

### 3.3 Pruebas para Actualizar Productos (PUT)

ID	Caso de Prueba	Entrada	Salida Esperada	Estado
TC-006	Actualizar un	ID: 1, Nombre:	Respuesta: 200,	<b>✓</b>
	producto con datos	"Tablet", Precio: 800	"Producto	
	válidos		actualizado"	
TC-007	Actualizar un	ID: 1, Precio: 800	Respuesta: 400, "El	X
	producto sin enviar		nombre es	
	el nombre		obligatorio"	

Adjunto capturas de pantalla de las pruebas realizadas:

#### Captura TC-006:

# Gestión de Productos

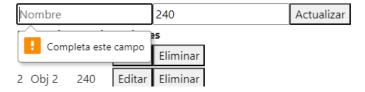
# **Editar Producto**



#### Captura TC-007:

# Gestión de Productos

### **Editar Producto**



#### 3.4 Pruebas para Eliminar Productos (DELETE)

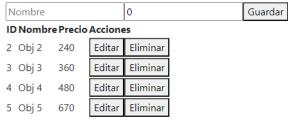
ID	Caso de Prueba	Entrada	Salida Esperada	Estado
TC-008	Eliminar un producto	Eliminar	Respuesta: 200,	<b>✓</b>
			"Producto eliminado"	

Adjunto capturas de pantalla de las pruebas realizadas:

#### Captura TC-008:

## Gestión de Productos

# **Agregar Producto**



# 4. Criterios de Aceptación

Los criterios de aceptación del proyecto son los siguientes:

- 1. Todas las operaciones CRUD deben funcionar correctamente.
  - a. Crear, leer, actualizar y eliminar productos con las respuestas y códigos HTTP esperados.

#### 2. Validaciones en el backend correctamente implementadas.

- a. El backend debe retornar errores **400 (Bad Request)** cuando se envíen datos incompletos o mal formados.
- 3. Sin errores en la consola del navegador o en la terminal de backend.
  - a. Al realizar las pruebas desde la interfaz gráfica en Vue.js, no deben aparecer errores en la consola.
- 4. Manejo adecuado de CORS (Cross-Origin Resource Sharing).
  - a. Las solicitudes entre el frontend (Vue.js) y el backend (Flask) deben funcionar sin bloqueos por CORS.

#### 5. Conclusiones

Estas pruebas de caja negra validan que la integración entre el frontend y el backend funcione correctamente y que las operaciones CRUD cumplan con las expectativas. A través de los casos de prueba detallados, se pueden identificar y corregir errores en los flujos de datos, validaciones y manejo de errores.

Se recomienda documentar los **resultados de cada caso de prueba** en un informe para tener un historial claro del comportamiento del sistema. Además, es útil repetir estas pruebas cada vez que se realice una actualización importante en el sistema.