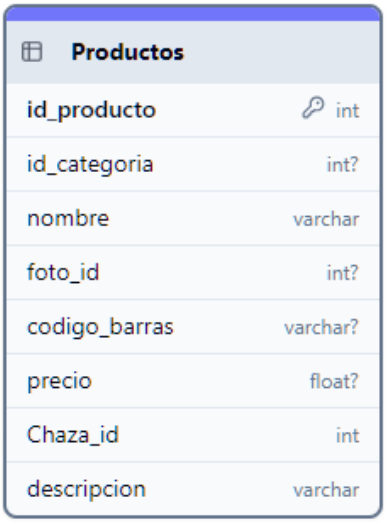
Universidad Nacional de Colombia - sede Bogotá
Facultad de Ingeniería
Departamento de Sistemas e Industrial
Ingeniería de Software 1 (2016701)

Historia de Usuario #01: Adición de productos

Anexo de Documentos Relacionados:

Base de datos

Para implementar esta funcionalidad, se utiliza la tabla Productos para contener la información de cada producto.



Descripción conceptual

Módulo	*Adición de productos*
Descripción de la(s) funcionalidad(es) requerida(s):	*Los usuarios podrán agregar nuevos productos al catálogo de una chaza, ingresando información como el nombre, la categoría y el precio.*

Backend

<table>
<tr><th>URL</th><th>Método</th><th>Código html</th></tr>
<tr><td>/api/chazas/:idchaza/productos</td><td>POST</td><td>200: Éxito, 400: Solicitud incorrecta, 500: Error interno del servidor</td></tr>
<tr><td colspan="3">Caso de uso técnico
Este método permite que el vendedor cree productos en su chaza. Se devuelve un 200 cuando el producto es creado con éxito, un 400 si hay datos inválidos o faltantes y 500 si hay un error interno en el servidor.</td></tr>
<tr><td colspan="2">Datos de entrada
<pre>Unset
{
   "nombre":"Coca-Cola",
   "categoria":"Bebidas",
   "precio":3500,

"foto":"https://ej.com/foto.jpg",
   "codigo_barras":123456789
}</pre></td><td>Datos de salida
200:
<pre>Unset
{
   "status":"success",
   "message":"Producto creado correctamente"
}</pre></td></tr>
<tr><td colspan="2"><pre>Unset
{}</pre></td><td>400:
<pre>Unset
{
   "status":"error",
   "message":"Error:Hay datos faltantes"
}</pre></td></tr>
<tr><td colspan="2"><pre>Unset
{
   "nombre":"Coca-Cola",
   "categoria":"Bebidas",
   "precio":3500,

"foto":"https://ej.com/foto.jpg",</pre></td><td>500:
<pre>Unset
{
   "status":"error",
   "message":"No se pudo crear el producto debido a un error interno"
}</pre></td></tr>
</table>

<pre>"codigo_barras":123456789
}</pre>	

URL	Método	Código html
/api/chazas/:idchaza/productos/:idProducto	PUT	200: Éxito, 400: Solicitud incorrecta, 500: Error interno del servidor

Caso de uso técnico

Este método permite que el vendedor actualice la información de productos en su chaza. Se devuelve un 200 cuando el producto es actualizado con éxito, un 400 si hay datos invalidos o faltantes y 500 si hay un error interno en el servidor.

Datos de entrada

```
Unset
{
   "nombre":"Coca-Cola",
   "categoria":"Bebidas",
   "precio":4500,

"foto":"https://ej.com/foto.jpg",
   "codigo_barras":123456789
}
```

Datos de salida

200:

```
Unset
{
   "status":"success",
   "message":"Producto
actualizado correctamente"
}
```

```
Unset
{}
```

400:

```
Unset
{
   "status":"error",
   "message":"Error:Hay datos
faltantes"
}
```

Unset
{
"nombre":"Coca-Cola",
"categoria":"Bebidas",
"precio":4500,
 	
"foto":"https://ej.com/foto.jpg",
"codigo_barras":123456789
}
	500:
	Unset
	{
	"status":"error",
	"message":"No se pudo actualizar el producto debido a un error interno"
	}

URL	**Método**	**Código html**
/api/chazas/:idchaza/productos/:idProducto	DELETE	200: Éxito, 404: No encontrado, 500: Error interno del servidor

Caso de uso técnico

Este método permite que el vendedor elimine productos de su chaza. Se devuelve un 200 cuando el producto es eliminado con éxito, un 404 si no es encontrado y 500 si hay un error interno en el servidor.

Datos de entrada	**Datos de salida**

Unset
N/A el ID se pasa como parte del URL
	200:
	Unset
	{
	"status":"success",
	"message":"Producto eliminado exitosamente"
	}

Unset
N/A el ID se pasa como parte del URL
	404:
	Unset
	{
	"status":"error",
	"message":"El producto con el ID proporcionado no existe"
	}

Unset
N/A el ID se pasa como parte del URL
	500:
	Unset
	{
	"status":"error",
	"message":"No se pudo eliminar el producto debido a un error interno"
	}

URL	**Método**	**Código html**
/api/chazas/:idchaza/productos/:idProducto	GET	200: Éxito, 404: No encontrado, 500: Error interno del servidor

Caso de uso técnico

Este método permite consultar la información de un producto existente. Se devuelve un código 200 cuando se encuentra el producto con éxito, un 404 si el producto no es encontrado y 500 si hay un error interno en el servidor.

Datos de entrada	**Datos de salida**

Unset
N/A el ID se pasa como parte del URL
	200:
	Unset
	{
	"status":"success",
	"data":{
	"nombre":"Coca-Cola",
	"categoria":"Bebidas",
	"precio":4500,
	"foto":"https://ejemplo.com/foto_producto.jpg",
	"codigo_barras":12345678
	}
	}

	404:

Unset
N/A el ID se pasa como parte del URL
	Unset
	{
	"status":"error",
	"message":"El producto con el ID proporcionado no existe"
	}

Unset
N/A el ID se pasa como parte del URL
	500:
	Unset
	{
	"status":"error",
	"message":"No se pudo encontrar el producto debido a un error interno"
	}

Frontend

Frontend

1. Crear un producto:

Interacción esperada:
El vendedor interactúa con un formulario en la sección de "Mis Productos" para registrar un nuevo producto.

Flujo visual y eventos:

- **Abrir formulario de creación:**
  - El vendedor hace clic en el botón "Agregar Producto".
  - Aparece un formulario vacío con campos para nombre, categoría, foto, código de barras y precio.
- **Registrar los datos:**
  - El vendedor completa los campos obligatorios (nombre, precio,categoría).
  - Sube una foto del producto (opcional).
  - Sube un código de barras del producto (opcional).
  - Hace clic en el botón "Guardar".
- **Solicitud al backend:**
  - Se realiza una solicitud POST al backend con los datos del formulario.
  - Si la operación es exitosa, se agrega el nuevo producto a la lista de productos del vendedor y la vista se actualiza.
- **Mensajes:**
  - Éxito: "Producto creado correctamente".
  - Error: "No se pudo crear el producto. Intente nuevamente más tarde".

---

2. Actualizar un producto:

Interacción esperada:
El vendedor interactúa con un formulario lleno con la información almacenada de un producto.

Flujo visual y eventos:

- **Abrir formulario de edición:**
  - El vendedor hace clic en el botón "Editar".
  - Se despliega un formulario con los campos actuales del producto.
- **Actualizar los datos:**
  - El vendedor edita los campos deseados (por ejemplo, nombre, precio, categoría, etc.).
  - Hace clic en el botón "Guardar cambios".
- **Solicitud al backend:**
  - Se realiza una solicitud PUT al backend con los datos editados.
  - Si la operación es exitosa, se actualiza la vista con los nuevos datos del producto.
- **Mensajes:**
  - Éxito: "Producto actualizado exitosamente".
  - Error: "No se pudo actualizar el producto. Intente nuevamente más tarde".

---

3. Eliminar un producto:

Interacción esperada:
El vendedor interactúa con un botón para eliminar en el apartado del producto.

Flujo visual y eventos:

- **Iniciar eliminación:**
  - El vendedor hace clic en el botón rojo "Eliminar".
  - Aparece un cuadro de diálogo de confirmación: "¿Estás seguro de que desea eliminar este producto?".
- **Confirmar acción:**
  - Si el vendedor confirma, el cuadro de diálogo se cierra y se realiza una solicitud DELETE al backend.
- **Solicitud al backend:**
  - Si la operación es exitosa, se redirige al vendedor a la lista de sus productos con un mensaje de confirmación.
- **Mensajes:**
  - Éxito: "Producto eliminado exitosamente".
  - Error: "No se pudo eliminar el producto. Intente nuevamente más tarde".

---

4. Consultar un producto:

Interacción esperada:
Un usuario interactúa con una tarjeta para ver la información específica de un producto.

Flujo visual y eventos:

- **Abrir los detalles de la chaza:**
  - El usuario hace clic en una tarjeta de un producto.
  - Se carga una nueva vista con los detalles completos del producto.
- **Visualización de datos:**
  - Se muestran los datos obtenidos del backend: nombre,categoría, precio, foto y código de barras.
- **Mensajes:**
  - Si ocurre un error en la carga: "No se pudo cargar la información del producto. Intenta nuevamente más tarde".

---

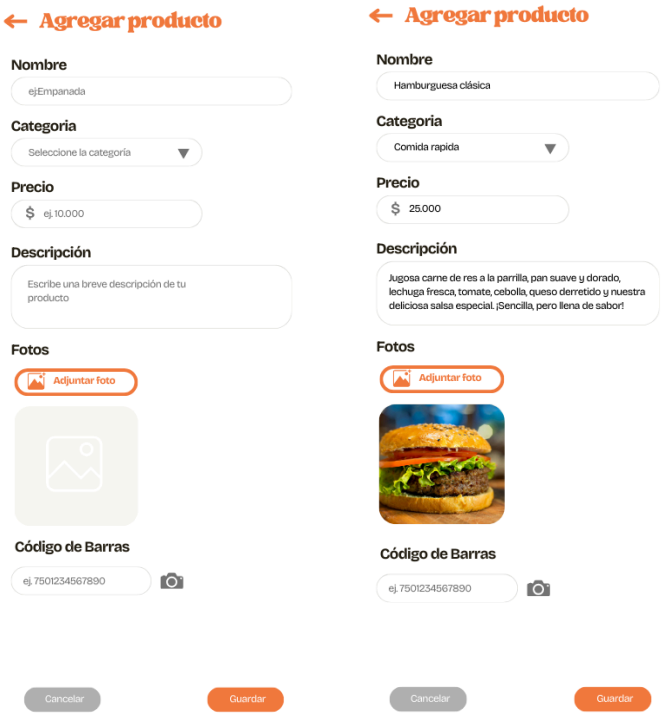
5. Error general en las operaciones:

Flujo visual y eventos:

- Si ocurre un error durante cualquier operación (POST, PUT, DELETE o GET):
  - Mostrar un mensaje de error en la parte inferior de la pantalla o en un modal:
    - "Ocurrió un error al procesar tu solicitud. Intenta nuevamente más tarde".

Mockups/Prototipos:

- **Crear un producto**



- **Actualizar un producto**

- **Consulta de un producto**

- **Eliminar un producto**