Proyecto final integrador: Tienda de Productos Argentinos

**Enlace Repositorio**: <https://github.com/MariaBelenHaddad/PWA_pagina_tienda_productos_argentinos.git>

*Nota: ya tenía armado en un solo repositorio la API y la tienda, antes de escuchar en clase que debían ser dos repositorios.*

Funcionamiento de la página

**Home Page:** "localhost:3000" (WEB)



Con el botón “Comprar ahora” vamos a la tienda donde se pueden ver todos los productos de la base de datos.

Para traer todos los productos desde la api usé el endpoint: "localhost:4000/products" (MÉTODO GET API)



Para traer un producto con su id desde la api usé el endpoint: "localhost:4000/products/search/:productId" (MÉTODO GET API)

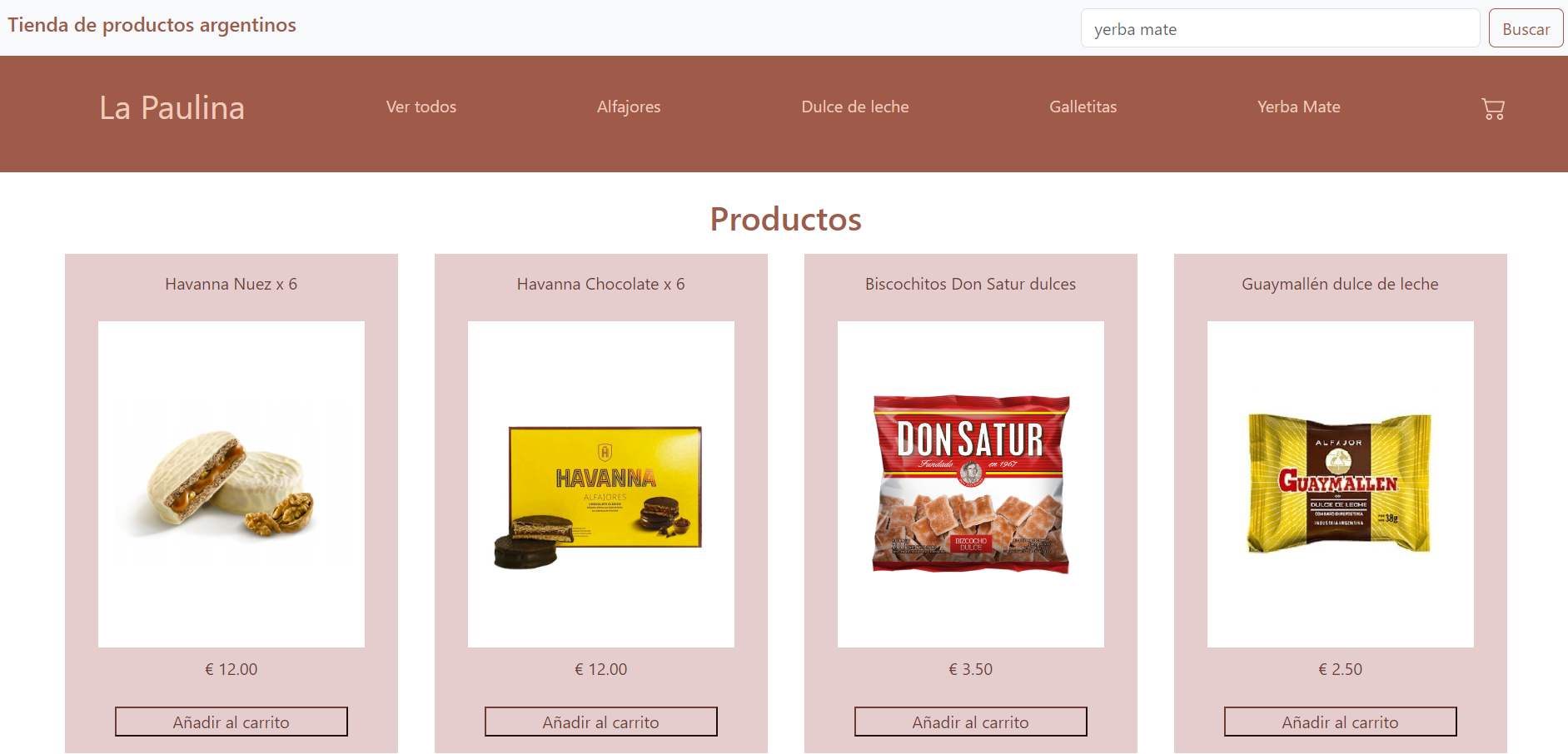
Con el botón “Ver todos los productos” también podemos ir a la tienda.

En el footer tenemos un formulario para suscribirse al newsletter.

Desde la api usé el endpoint: "localhost:4000/sendEmail" (MÉTODO POST API)



**Tienda:** "localhost:3000/products" (WEB)



En la barra de navegación podemos ver botones (Ver todos, Alfajores, Dulce de leche, Galletitas, Yerba Mate), al hacer clic en cualquiera de ellos la API me trae los productos por categoría. Para esto usé el endpoint:

"localhost:4000/products/search" + query (category=alfajores) (MÉTODO GET API)

Nota: me faltó lograr la funcionalidad del carrito y el buscador.

**Panel del administrador:** "localhost:3000/admin" (WEB)

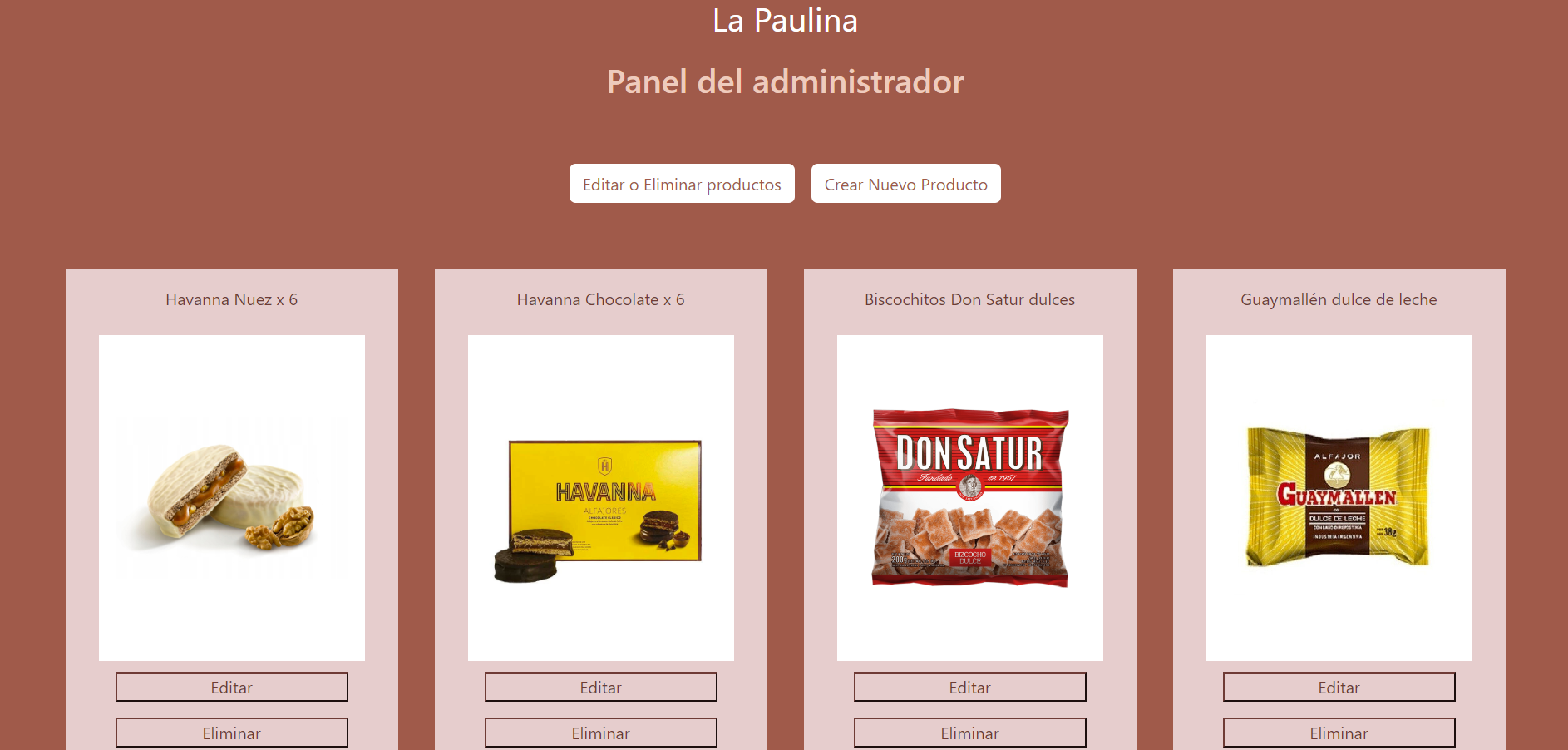
Al ingresar tenemos dos botones “Editar o Eliminar productos” y “Crear Nuevo Producto”.

Para crear un nuevo producto hacemos clic en “Crear Nuevo Producto” y la página nos redirige a otra ventana: "localhost:3000/newProduct" (WEB)



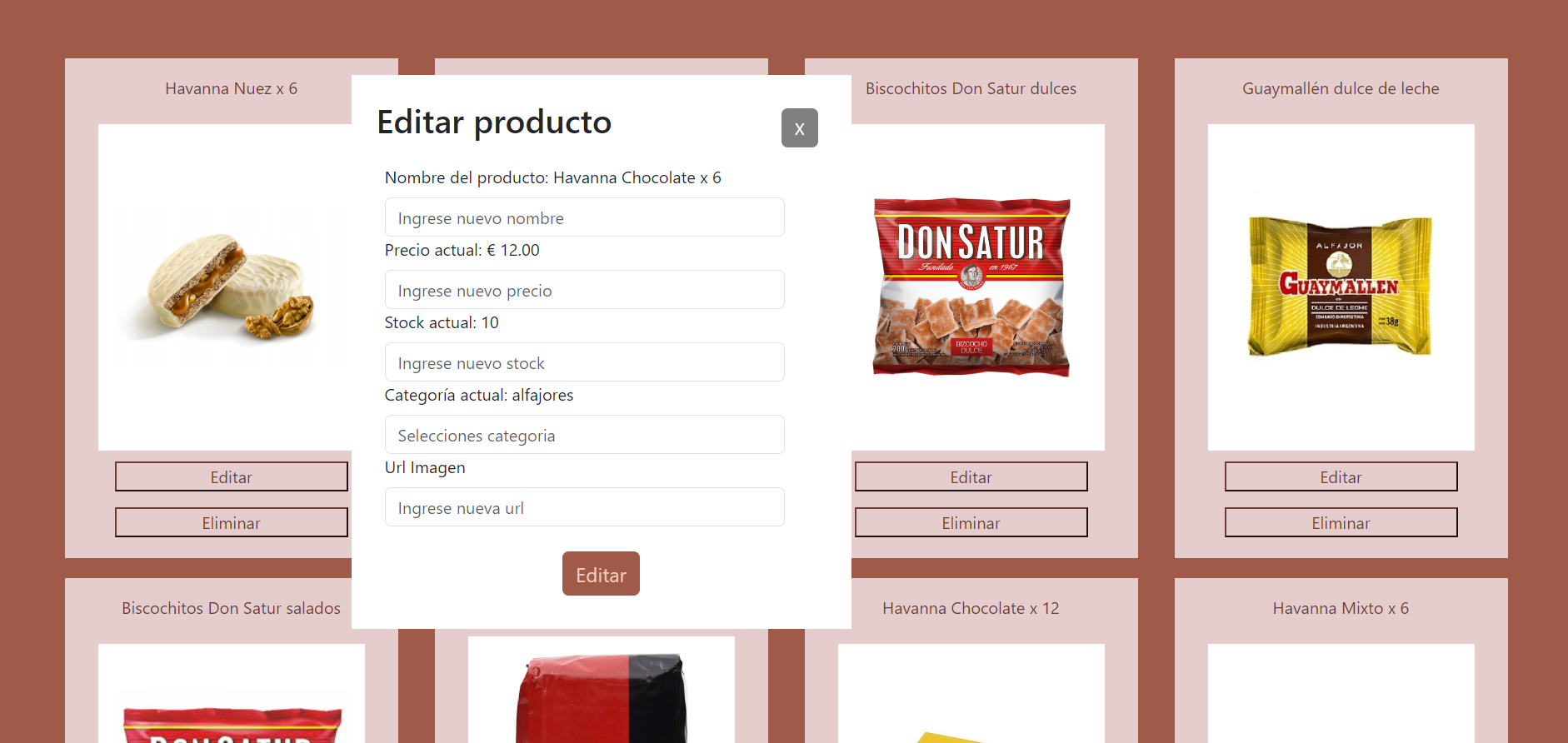
Al clickar “Editar o Eliminar productos” se muestran todos los productos con dos botones: “Editar” o “Eliminar”.

Para eliminar simplemente hacemos click en el producto que queremos eliminar. Para esto la api usa el endpoint: "localhost:4000/products/:productId" (MÉTODO DELETE API)

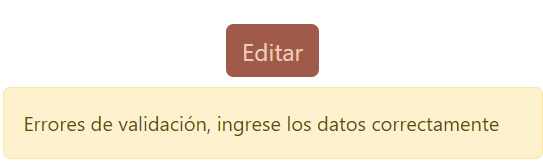


Al hacer clic en “Editar” aparece una ventana con los datos actuales, debemos ingresar los datos nuevos, no es necesario ingresar todos los datos, sólo los que nos interesa editar, luego hacemos click en el botón “Editar”.

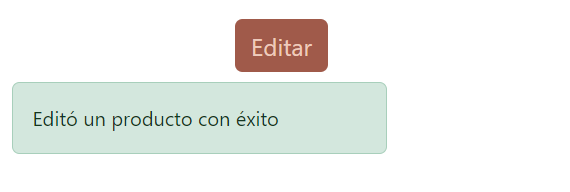
Para esto la api usa el endpoint: "localhost:4000/products/:productId" (MÉTODO PATCH API)



Si algún dato no se ingresó de forma correcta nos aparece un mensaje de alerta:

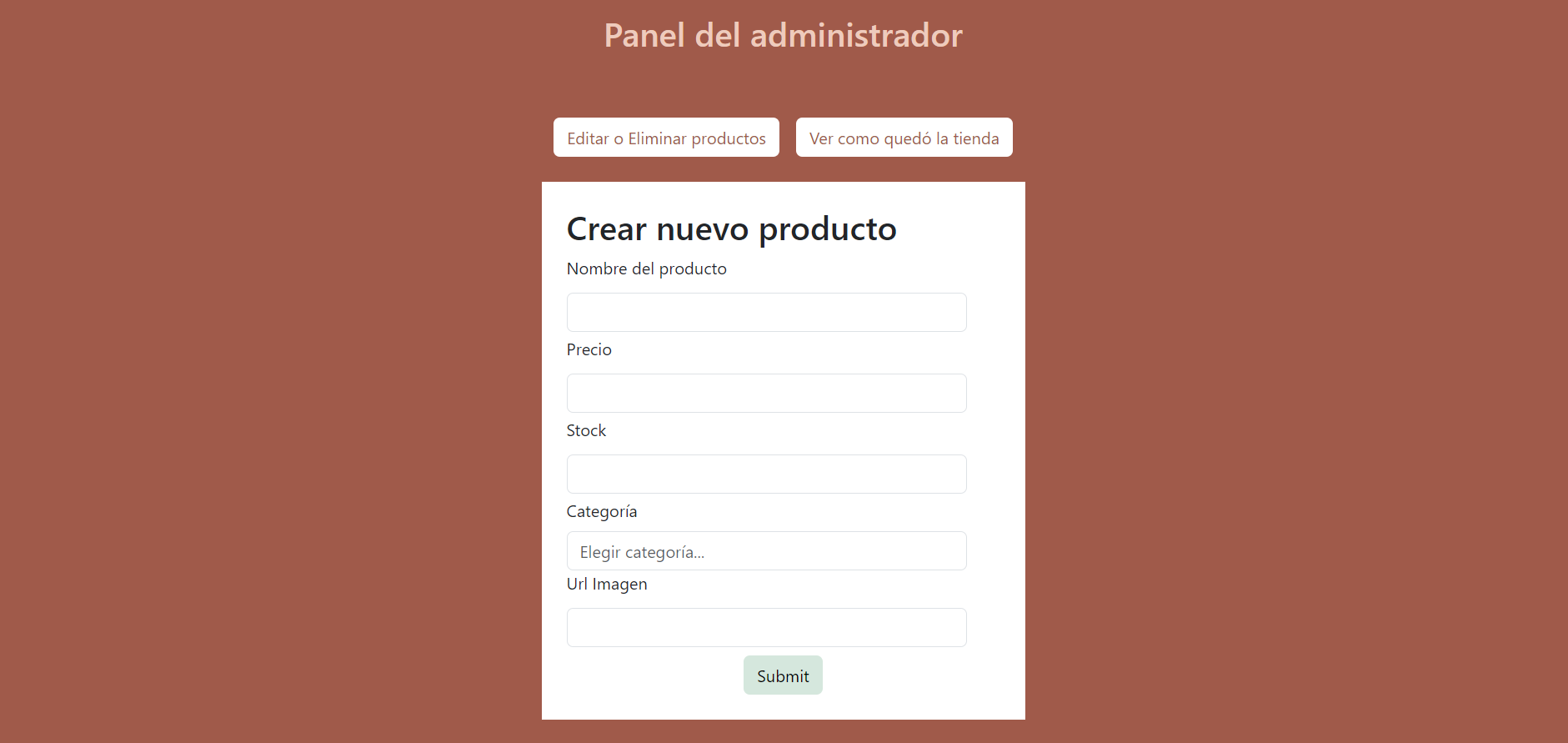


Si editamos el producto correctamente también nos aparece un mensaje :



Por último, para crear un nuevo producto vamos a esta ventana:

Para crear un nuevo producto la api usa el endpoint: "localhost:4000/products/new" (MÉTODO PUT API)



No necesitamos ingresar el ID porque la API lo crea de manera aleatoria con la siguiente función: const productId = crypto.randomUUID();

Tanto en el método PUT como en el método PATCH de la api agregué funciones de validación de datos antes de enviar la información a la base de datos SQL.

*Notas adicionales:*

*Me faltó hacer el deploy del proyecto y agregar una autenticación de usuario y contraseña para el administrador. En cuanto pueda voy a agregarlo, por trabajo no tuve tanto tiempo para dedicarle.*

*Quiero además agradecerles* ***Victoria y Marcelo*** *por la paciencia y dedicación que ponen en sus clases, me gustó mucho esta cursada y siento que aunque me falta muchísimo, fué un gran aprendizaje. Muchísimas gracias!!*