



S P S
TECH IS NOW

DESARROLLADOR
FRONT-END
SENIOR
PRÁCTICA
TÉCNICA

19 de agosto de 2024

por:

MARIO TOMAS ISLAS CASTRO

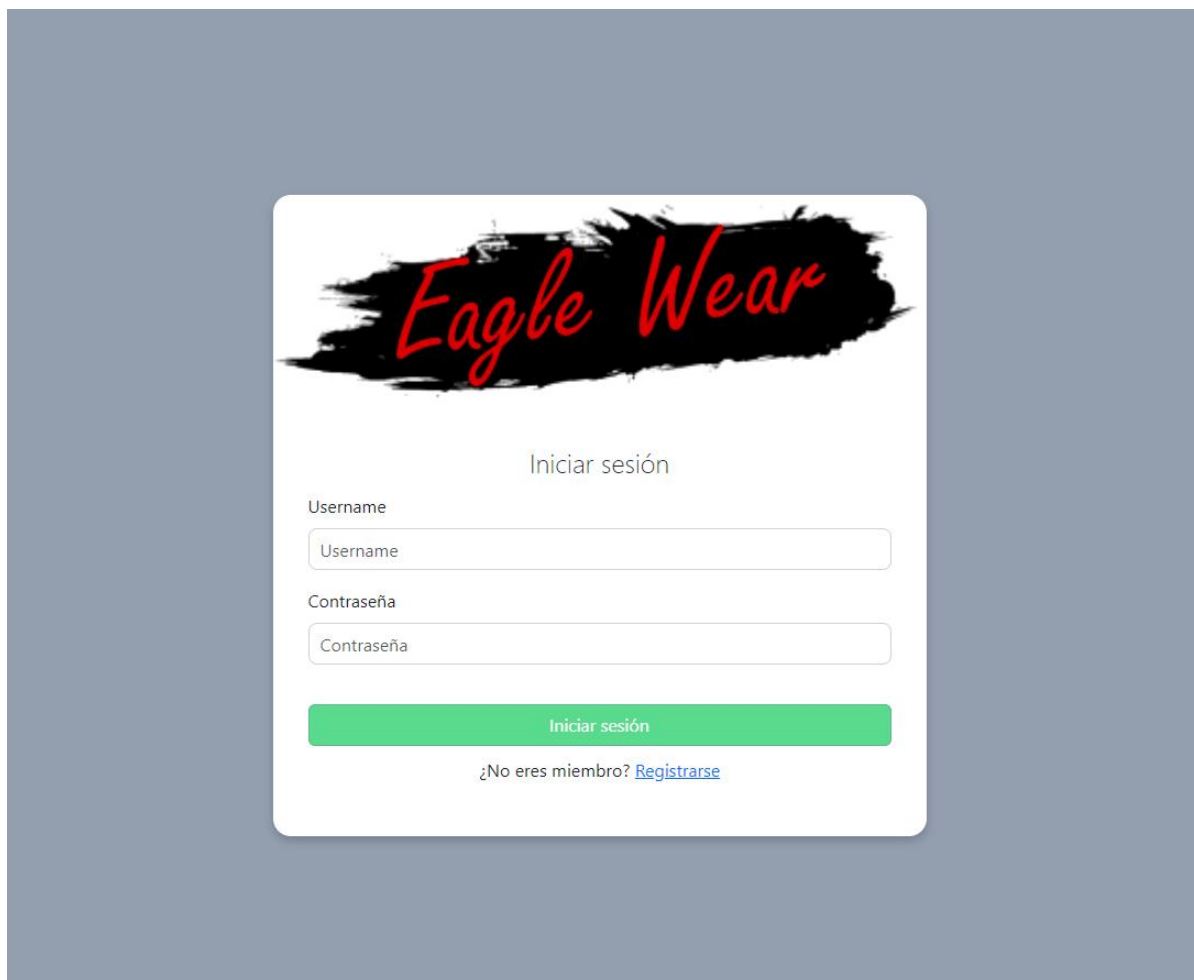
Descripción General


El proyecto **sps-web-shop-mario-tomas** es una aplicación web desarrollada en Angular que simula una tienda en línea. Esta aplicación consume una API de prueba disponible en [FakeStoreAPI](#), que proporciona datos ficticios de productos, carritos de compra y usuarios. El proyecto se ha desplegado en la plataforma [Netlify](#), lo que permite que esté disponible públicamente a través de la URL <https://master--storied-hummingbird-7a274f.netlify.app/login>.

Estructura del Proyecto

El proyecto se organiza en varios componentes de Angular que manejan las diferentes funcionalidades de la tienda:

- **Login:** Permite a los usuarios autenticarse.



- 


Registrarse



Nombre	Apellido
<input type="text"/>	<input type="text"/>
Nombre de usuario	Email
<input type="text"/>	<input type="text"/>
Contraseña	Confirmar Contraseña
<input type="password"/>	<input type="password"/>
Ciudad	Calle
<input type="text"/>	<input type="text"/>
Numero	Código postal
<input type="text"/>	<input type="text"/>
Latitud	Longitud
<input type="text"/>	<input type="text"/>
Teléfono	
<input type="text"/>	


Registrarse


¿Ya tienes una cuenta? [Iniciar sesión](#)

- master--stored-hummingbird-7a274f.netlify.app/index

 Inicio Acerca de Administrador

 Carrito (11)  Cerrar sesión







Fjallraven - Foldback No. 1 Backpack, Fits 15 Laptops

\$109.95

Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday


 Agregar al Carrito




Mens Casual Premium Slim Fit T-Shirts

\$22.30

Slim-fitting style, contrast raglan long sleeves, three-button henley placket, light weight & soft fabric for breathable and comfortable wearing. And Solid stitched shirts with round neck made for durability and a great fit for casual fashion wear and beloved baseball fans. The Henley style round neckline includes a three-button placket.


 Agregar al Carrito





Mens Cotton Jacket


\$55.99

great outdoorwear jacket for Spring/Summer/Winter, suitable for many occasions, such as working, hiking, camping, mountain/rock climbing, cycling, traveling or other outdoor. Good gift choice for you or your family member. A warm hearted love to father, husband or son in this thanksgiving or Christmas Day.

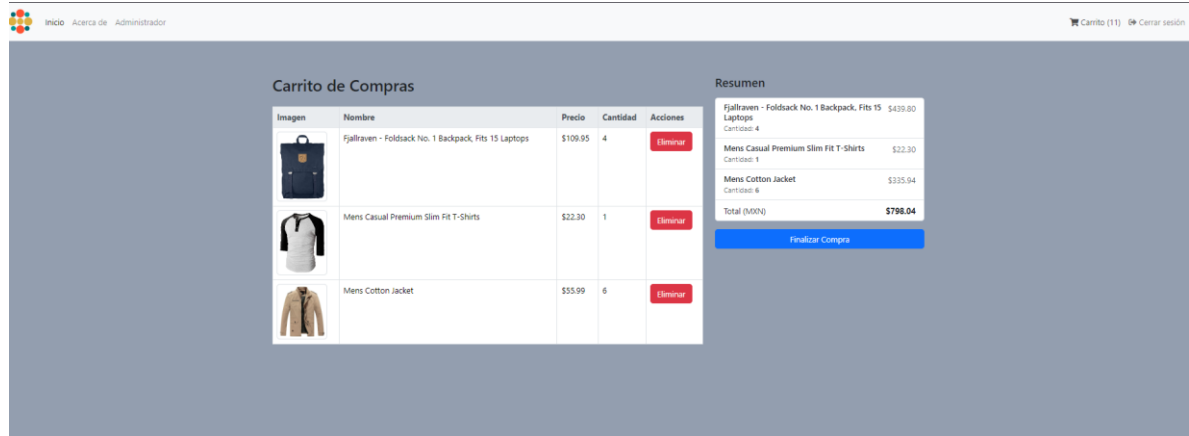
 Agregar al Carrito



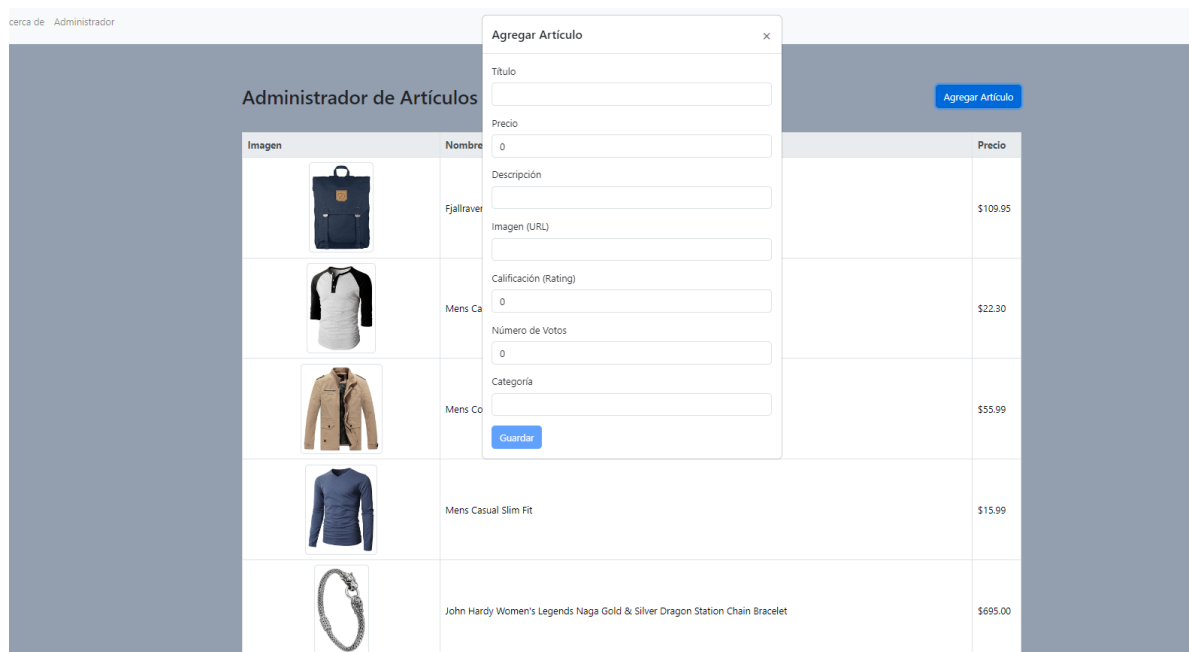




- **Carrito de compras:** Permite agregar y gestionar productos en el carrito.



- **Administrador:** Permite agregar y gestionar productos en la tienda.



Descripción de los Componentes y Servicios

1. Componente LoginComponent:

- **Función:** Gestiona el formulario de inicio de sesión. Se comunica con el AuthService para autenticar al usuario y guardar el token JWT en localStorage.

- **Desafíos:** Durante el proceso de inicio de sesión, se implementó un spinner para mejorar la experiencia del usuario mientras se procesa la autenticación. Este spinner se muestra utilizando una imagen rotatoria mientras la pantalla se oscurece.

2. Componente CarritoComponent:

- **Función:** Muestra los productos que el usuario ha agregado al carrito. Permite eliminar productos y finalizar la compra.
- **Desafíos:** La API sólo proporciona el productId y la quantity de los productos en el carrito, por lo que fue necesario realizar llamadas adicionales a la API para obtener detalles como el nombre, precio e imagen de cada producto.

3. Componente AdministratorComponent:

- **Función:** Permite a un administrador agregar nuevos productos a la tienda. Incluye un formulario que se abre en un modal, donde se ingresan detalles del producto.
- **Desafíos:** Después de guardar un nuevo producto, se implementó un modal de confirmación que se muestra al usuario para indicar que el producto fue agregado correctamente.

4. Servicio AuthService:

- **Función:** Gestiona la autenticación de los usuarios. Guarda el token JWT en el almacenamiento local, lo recupera, y extrae el userId del token para otras operaciones.
- **Desafíos:** Fue necesario extraer el userId del token para vincular correctamente el carrito de compras al usuario autenticado.

5. Servicio CarritoService:

- **Función:** Gestiona las operaciones relacionadas con el carrito de compras, como la obtención de productos, el agregado de nuevos productos y la finalización de la compra.
- **Desafíos:** Dado que la API no realiza cambios reales en el servidor (sólo simula respuestas), se tuvo que manejar este comportamiento en el frontend para que la aplicación respondiera como si la operación hubiera sido exitosa.

Problemas y Soluciones

1. **API con Respuestas Sin Persistencia:** Uno de los principales desafíos fue trabajar con una API que no persiste los cambios en sus datos. Aunque la API proporciona respuestas simuladas, los cambios realizados (como la adición o eliminación de productos) no se reflejan en una base de datos real. Esto requería manejar adecuadamente la interfaz de usuario para simular el comportamiento de una aplicación de comercio electrónico sin una verdadera manipulación de datos en el backend. La solución fue concentrar la lógica en el frontend y simular el comportamiento esperado del sistema.
2. **Obtención de Detalles de Productos en el Carrito:** La API devuelve el carrito del usuario solo con el id y la cantidad de cada producto, sin información adicional. Para solucionar esto, se tuvo que hacer una segunda llamada a la API para obtener detalles completos de cada producto (nombre, precio, imagen, etc.) a partir de su id. Este proceso implicó diseñar una función que, al recibir el carrito, itera sobre los productos y realiza las llamadas adicionales necesarias para enriquecer los datos del carrito.
3. **Manejo del ID de Usuario a partir del Token:** Dado que el carrito está asociado a un usuario específico, fue necesario extraer el id del usuario del token de autenticación proporcionado por la API. Este token es un JWT (JSON Web Token), y para obtener el id se implementó un método en el servicio de autenticación que decodifica el token y extrae la información necesaria. Esto permitió gestionar correctamente las solicitudes relacionadas con el carrito para cada usuario autenticado.

4. **Despliegue en Netlify:** El proyecto se desplegó en Netlify, una plataforma que ofrece despliegue continuo a partir de un repositorio de GitHub. Netlify fue elegido por su facilidad de uso, integración fluida con GitHub y su capacidad para manejar aplicaciones frontend sin necesidad de un backend dedicado. Durante el despliegue, se resolvieron varios problemas como la configuración de rutas, asegurándose de que la aplicación Angular manejara las rutas correctamente mediante el uso de un archivo `netlify.toml` que redirige todas las rutas al archivo `index.html`.
5. **Solución de Problemas de Construcción:** Durante el proceso de despliegue, se presentaron problemas con las versiones de Node.js y dependencias incompatibles. Esto se resolvió añadiendo especificaciones en el archivo `package.json` para garantizar que Netlify utilizara la versión adecuada de Node.js y asegurando que todas las dependencias estuvieran actualizadas y fueran compatibles con Angular 11.

URL del Despliegue

El proyecto está disponible públicamente en la siguiente URL:
<https://master--storied-hummingbird-7a274f.netlify.app>

Repositorio de Código

El código fuente del proyecto está disponible en GitHub en el siguiente enlace:
<https://github.com/RayEmer/sps-web-shop-mario-tomas>

Video demostrativo

El código fuente del proyecto está disponible en GitHub en el siguiente enlace:
<https://youtu.be/jsQEqA4OXl4>