

- Visualización de productos con imágenes, descripciones y precios.
- Un carrito de compras persistente en la sesión del usuario.
- Métodos de pago simulados para completar la compra.
- Integración con una fake API para obtener descripciones dinámicas de productos.
- Una interfaz adaptable y accesible desde distintos dispositivos.

### **1.3. Objetivos**

- Desarrollar una aplicación funcional en Angular.
- Implementar una experiencia de usuario fluida e interactiva.
- Implementar una buena interacción con el carrito de compra.
- implementar una buena navegación buena y sencilla

### **1.4. Modelado de la solución**

#### **1.4.1. Recursos humanos**

- Desarrollador Front end
- Diseñador CSS para una interfaz con lógica, creativa pero sencilla a la vez.

#### **1.4.2. Recursos hardware**

- Servidores locales y en la nube para desplegar la aplicación y para guardarla.
- Mi ordenador personal para toda la realización del proyecto.
































#### **1.4.3. Recursos software**

































- Angular para el desarrollo frontend.
- Json-server para la API fake de productos.
- Postman para la API de Youtube.
- GitHub para control de versiones.
- Gimp para edición de imágenes necesarias para la interfaz.

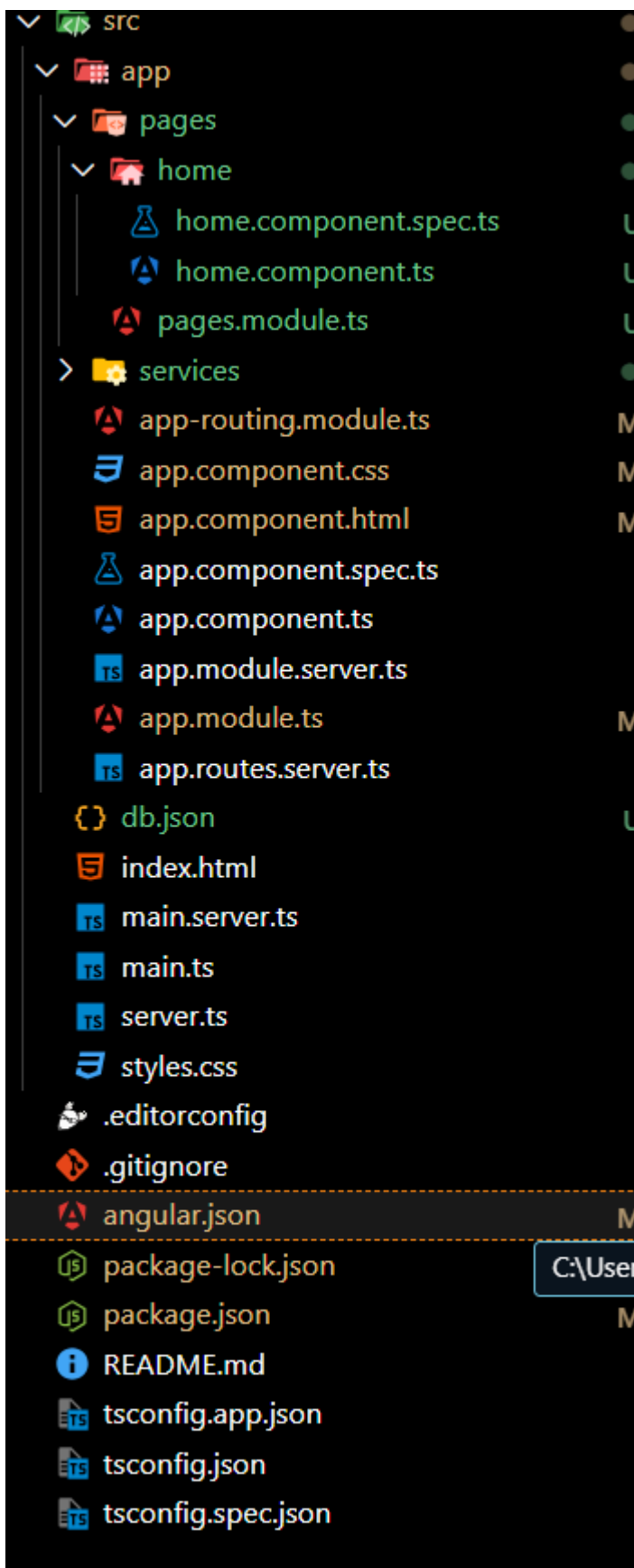
## **2. Ejecución de la práctica**

### **2.1. Documentación técnica e implementación de la aplicación**

Esta es la estructura que sigue mi proyecto:

- >  .angular
- >  .vscode
- >  documentation
- >  node\_modules
- ✓  public
  - >  image
    - ★ favicon.ico
- ✓  src
  - ✓  app
    - ✓  interface
      - ✓  interface
        -  interface.component.css
        -  interface.component.html
        -  interface.component.spec.ts
        -  interface.component.ts
        -  interface.module.ts
  - ✓  pages
    - ✓  catalogo
      -  catalogo.component.css
      -  catalogo.component.html
      -  catalogo.component.spec.ts
      -  catalogo.component.ts
    - ✓  cesta
      -  cesta.component.css
      -  cesta.component.html
      -  cesta.component.spec.ts
      -  cesta.component.ts
    - ✓  contacta
      -  contacta.component.css
      -  contacta.component.html
      -  contacta.component.spec.ts
      -  contacta.component.ts

- ▼  src
  - ▼  app
    - ▼  pages
      - ▼  contacta
        -  contacta.component.ts
      - ▼  detalles-producto
        -  detalles-producto.component.css
        -  detalles-producto.component.h...
        -  detalles-producto.component.s...
        -  detalles-producto.component.ts
      - ▼  footer
        -  footer.component.css
        -  footer.component.html
        -  footer.component.spec.ts
        -  footer.component.ts
      - ▼  header
        -  header.component.css
        -  header.component.html
        -  header.component.spec.ts
        -  header.component.ts
      - ▼  home
        -  home.component.css
        -  home.component.html
        -  home.component.spec.ts
        -  home.component.ts
        -  pages.module.ts
    - >  services
      -  app-routing.module.ts
      -  app.component.css
      -  app.component.html
      -  app.component.spec.ts
      -  app.component.ts



Mi proyecto está estructurado de una forma organizada en la cual todos los componentes se encuentran en `app/src/pages` y los módulos importantes como `app.module.ts`, `app-routing.module.ts` se encuentra en `src/app`

## **3. Documentación del sistema**

### **3.1. Manual de instalación y configuración de la aplicación**

1. Clonar el repositorio del proyecto desde GitHub.
2. Instalar dependencias con "npm install "
3. Iniciar el servidor JSON con "json-server –watch db.json "
4. Ejecutar la aplicación Angular con "ng serve "

### **3.2. Manual de usuario**

Aquí se explica cómo usar la tienda en línea para explorar sneakers, agregarlos a tu cesta y comprarlos de forma rápida y segura.

Para que la experiencia sea óptima, te recomendamos usar un navegador actualizado como Chrome, Firefox, Edge o Safari y asegurarte de tener una conexión estable..

Desde la página de inicio, puedes acceder al catálogo haciendo clic en "Catálogo" en la barra de navegación. Ahí encontrarás todos los modelos disponibles con su nombre y precio. Si quieres más detalles sobre algún par, haz clic en la imagen para ver fotos adicionales, descripción, tallas y colores disponibles.

Cuando encuentres los sneakers que te gustan, presiona "Añadir a la cesta" y verás un mensaje de confirmación. Luego, puedes seguir explorando o ir a la cesta haciendo clic en su ícono, ubicado en la parte superior derecha. En la cesta podrás revisar los productos agregados, modificar cantidades con los botones - y +, eliminar artículos o vaciarla por completo si decides empezar de nuevo.

Para finalizar la compra, ve a la cesta y haz clic en "Pagar". Serás redirigido a la plataforma de pago para completar la transacción de manera segura. Una vez confirmado el pago, recibirás un mensaje de confirmación y podrás seguir navegando en la tienda.

Si tienes problemas con la página, prueba recargando o revisando tu conexión a Internet. Si los productos no se muestran en el catálogo, intenta limpiar la caché del navegador. Si no puedes agregar productos, revisa que haya stock disponible. Y si el pago no se procesa, asegúrate de que tu método de pago sea válido y que haya productos en tu cesta.

## **4. Conclusiones finales**

### **4.1. Grado de cumplimiento de los objetivos fijados**

Se lograron implementar todas las funcionalidades requeridas, proporcionando una experiencia de usuario fluida y efectiva.

## **4.2. Propuesta de modificaciones o ampliaciones futuras del sistema implementado**

- Implementar tallas y colores
- Implementar un pago real
- Ampliar el catálogo con más productos y un filtro avanzado.

## **5. Bibliografía**

- Documentación oficial de Angular (<https://angular.io/docs>)
- Guía de json-server (<https://github.com/typicode/json-server>)