

TRABAJO FIN DE GRADO

Indice:

1. Introducción
 - 1.1. Presentación del proyecto
 - 1.2. Objetivos del Proyecto
 - 1.3. Justificación del proyecto
2. Análisis de Requerimientos
 - 2.1. Identificación de necesidades y requerimientos
 - 2.2. Identificación de público.
 - 2.3. Estudio de mercado y competencia.
3. Diseño y Planificación
 - 3.1. Definición de arquitectura del proyecto
 - 3.2. Diseño de la interfaz de usuario
 - 3.3. Planificación de las tareas y los recursos necesarios
4. Implementación y pruebas
 - 4.1. Desarrollo de las funcionalidades del proyecto
 - 4.2. Pruebas unitarias y de integración
 - 4.3. Corrección de errores y optimización del rendimiento
5. Documentación
 - 5.1. Documentación técnica
 - 5.2. Documentación de usuario
 - 5.3. Manual de instalación y configuración
6. Mantenimiento y evolución
 - 6.1. Plan de mantenimiento y soporte
 - 6.2. Identificación de posibles mejoras y evolución del proyecto
 - 6.3. Actualizaciones y mejoras futuras
7. Conclusiones
 - 7.1. Evaluación del proyecto
 - 7.2. Cumplimiento de objetivos y requisitos
 - 7.3. Lecciones aprendidas y recomendaciones para futuros proyectos
8. Referencias y enlaces de interés

1. Introducción

1.1. Presentación del proyecto

La idea principal de este proyecto es la creación de una aplicación web full stack para la gestión de una tienda de camisetas de fútbol.

La aplicación será muy interactiva y fácil de usar para que así cualquier tipo de usuario pueda acceder a cualquier sitio que quiera y encuentre lo que busque lo más fácil posible.

1.2. Objetivos del Proyecto

Los objetivos principales de la aplicación serán:

- Alcanzar el mayor número de clientes posibles, dando a conocer la aplicación.
- Garantizar una buena experiencia al usuario al usar la aplicación.
- Ofrecer las camisetas que los clientes busquen con un alto inventario de stock para que todos puedan tener oportunidades de comprar.
- Frecuente actualización de la página con nuevas camisetas para estar siempre lo más adelantados posible al resto de tiendas.

1.3. Justificación del proyecto

La principal razón de la creación de esta aplicación es que la gente que no puede permitirse comprar una camiseta de su equipo favorito debido a los elevados precios de estas, puedan optar por una opción más económica y casi de la misma calidad.

Además, que los clientes puedan comprar o ver sus camisetas favoritas sin necesidad de ir a una tienda física.

2. Análisis de requerimientos

2.1. Identificación de necesidades y requerimientos

Para la creación de esta aplicación necesitaremos lo siguiente:

- Una buena base de datos para almacenar todos los datos necesarios.
- Un amplio catálogo de camisetas con toda la información, tallas,etc.
- Que el proceso de compra sea fácil y rápido, con un carrito en el que ir añadiendo los productos que se desean comprar.
- Una buena gestión de usuarios, para que cualquier usuario pueda iniciar sesión, registrarse, ver todos sus pedidos, sus datos personales, etc.
- También necesitaremos un rol de usuario especial para que los administradores de la página puedan añadir nuevos productos, eliminar, poner ofertas, gestionar los pedidos,etc.
- Necesitaremos una buena seguridad en la aplicación para que un usuario normal no pueda acceder al apartado de los administradores y así solo les permita comprar.

2.2. Identificación del público

El público principal de la tienda será el siguiente:

- Aficionados al fútbol de cualquier edad y género

- Personas que no pueden permitirse camisetas de fútbol muy caras.
- Personas que buscan un regalo para alguien que le gusta el fútbol.

2.3. Estudio del mercado y competencia

Se hará un minucioso estudio del mercado, otras tiendas de camisetas y buscaremos la mejor manera de diferenciarnos del resto y así llamar más la atención que el resto de tiendas del mismo sector que nosotros.

Actualmente ya existen varias páginas de ventas de camisetas de fútbol por lo que tendremos que hacerlo muy bien para destacar por encima de ellas.

3. Diseño y Planificación

3.1. Definición de arquitectura del proyecto

La arquitectura del proyecto se basará en una aplicación web full stack:

- En cuanto al BackEnd usaré Spring Boot, el cual es un framework de Java.
- En cuanto al FrontEnd usaré Angular.
- En cuanto a la Base de Datos, usaré una base de datos MySQL.
- Y finalmente para conectar todo usaré un API REST para conectar front con back.

3.2. Diseño de la interfaz de usuario

La interfaz de usuario será responsive, para que se vea bien en cualquier dispositivo, tanto en móviles como ordenadores como tablets.

La navegación dentro de la página será intuitiva para que los usuarios no tengan dudas para buscar lo que quieran.

Tendrá un diseño moderno y agradable a la vista para que sea agradable de ver.

3.3. Planificación de las tareas y los recursos necesarios

Realizaré un análisis para ver todos los requerimientos de la página, y que a la hora de ponerme a desarrollarla no se me olvide nada.

Una vez haya analizado todo me pondré a trabajar en primer lugar en la parte del back para crear las clases, repositorios, etc y conectarme a la base de datos.

Después pasaré a la parte del front en la que me conectaré al back por medio de una API Rest y desarrollaré toda la parte visual de la página usando bootstrap y css.

Finalmente realizaré las pruebas necesarias para comprobar que todo funciona correctamente y desplegaré la página.

4.1. Desarrollo de las funcionalidades del proyecto.

El desarrollo de esta aplicación se llevará a cabo de la siguiente manera:

En primer lugar he creado los proyectos correspondientes al Front y al Back.

Dentro de ellos he ido desarrollando cada una de las funcionalidades que son necesarias para realizar la aplicación correctamente, he ordenado las carpetas para que todo sea fácil de encontrar a la hora de buscar alguna clase.

El método que voy a utilizar para llevar a cabo la aplicación va a ser SCRUM ya que me parece el más útil y ordenado.

Iré realizando las tareas que quiera ir realizando dividiéndola a su vez en más subtareas para que todo sea más fácil.

En el front utilizaré Angular y para que toda la aplicación sea responsive usare Bootstrap, además utilizaré CSS para dar estilos más concretos y TypeScript para desarrollar las funcionalidades Angular.

En el Back utilizar Java con el framework de Spring Boot para conectarme más ágilmente a la Base de Datos y crearla más fácilmente y para conectar el Front con el Back utilizaré un API Rest.

En cuanto a IDEs, he utilizado Visual Studio Code para el front e IntelliJ IDEA para el Back, para lanzar la base de datos he utilizado XAMPP para crear una base de datos en mi localhost y así ir haciendo pruebas.

4.2. Pruebas unitarias y de integración

En cuanto a las pruebas unitarias y de integración las he ido haciendo manualmente yo comprobando todo lo que iba haciendo y además he usado postman para ir viendo que las solicitudes devuelven lo que quiero en mi back.

4.3. Corrección de errores y optimización del rendimiento

Para la corrección de errores, he ido usando la consola del navegador para ir viendo los errores que me van dando y así poder solucionarlo, a su vez también he usado la consola de IntelliJ IDEA para ver los errores internos que pueden producirse en el back.

Finalmente para mejorar el rendimiento he intentado que la base de datos no esté nunca muy sobrecargada, para que a medida de mi desarrollo no tarde mucho en devolverme los datos que yo le solicite.

5.Documentación

5.1. Documentación técnica

En primer lugar, mi aplicación se puede dividir en dos partes, por un lado tenemos el BackEnd el cual ha sido desarrollado utilizando Spring Boot y por otro lado tenemos la parte del FrontEnd, la cual ha sido desarrollada en Angular.

BackEnd



Como ya bien he dicho la parte del back está desarrollada en Java utilizando Spring Boot y la estructura de este proyecto es de la siguiente manera:

El **primer paquete** que nos encontramos en el proyecto es el de **BackCamisetas**. Dentro de este paquete, se encuentra una clase utilizada para que, al arrancar la aplicación por

primera vez, se persistan datos de prueba en la base de datos. Esto nos permite trabajar con ellos y verificar el correcto funcionamiento de la aplicación.

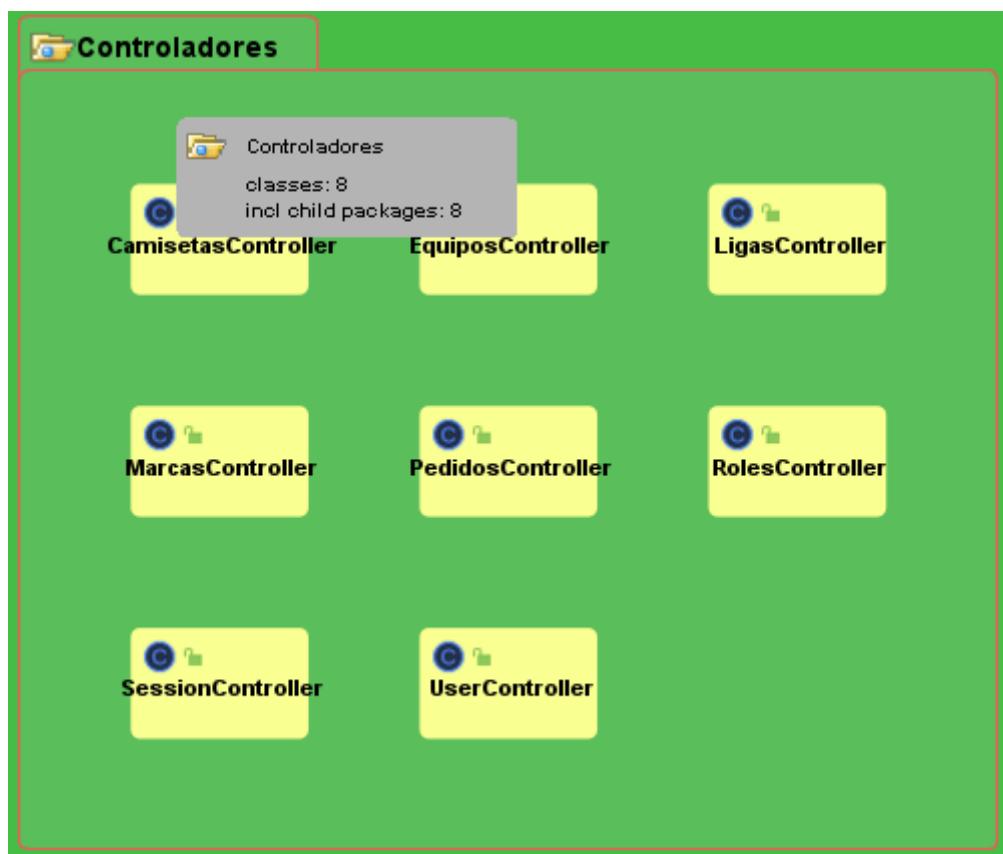
1. **Roles:** La primera función introduce los roles, los cuales pueden ser **cliente**, **admin** o **superadmin**. He dividido los roles de esta manera para que cada uno pueda desempeñar una acción diferente.
2. **Usuarios de prueba:** La segunda función está dedicada a crear los usuarios de prueba, generando uno para cada rol: un admin, un cliente y un superadmin.
3. **Marcas:** La tercera función crea las marcas que más tarde se asignarán a las camisetas, junto con sus respectivas fotos y nombres.
4. **Ligas:** La cuarta función se encarga de crear cinco diferentes ligas, las cuales también se asignarán posteriormente a las camisetas y a los equipos que se creen.
5. **Equipos:** La quinta función es para crear equipos. En esta función, obtengo todas las ligas y asingo a cada equipo su respectiva liga y foto.
6. **Camisetas:** La sexta función se dedica a crear camisetas. Aquí obtengo todos los equipos y marcas, y creo 10 camisetas diferentes, asignando a cada una un equipo y una marca, además del resto de datos que tiene una camiseta.
7. **Pedidos de prueba:** Finalmente, la última función sirve para crear unos pedidos de prueba. Esto permite comprobar que los pedidos se asignan correctamente a un usuario y que se les asigna también una lista de camisetas.

El **segundo paquete** que nos encontramos es el de controladores, en el cual hay un controlador para cada una de las entidades que tiene mi aplicación para así tener un mejor orden y acceder más rápido a cada una de las acciones que quiera realizar, en cada uno de los controladores nos conectamos a un servicio diferente del Front utilizando un API Rest.

- En primer lugar tenemos el **controlador de camisetas**, la primera función que nos encontramos es la de listar todas las camisetas que existen, en segundo lugar tenemos la función de eliminar camisetas la cual a través de un id, busca la que quiero eliminar y la elimina. La tercera función es la de insertar camisetas, a la cual se le pasa una camiseta y se inserta en la base de datos. Luego tenemos una función para obtener una camiseta por su id, la cual busca la camiseta por el id que se pasa como parámetro y te la devuelve. Finalmente en este controlador, encontramos la función para actualizar camisetas, la cual mediante un id y una camiseta nueva actualiza la camiseta que queremos.
- En segundo lugar tenemos el **controlador de equipos**, las funciones son similares en todos los controladores, aquí encontramos insertar equipos que actúa igual en todos los controladores, listar todos los equipos que te devuelve una lista con todos los equipos, eliminar equipos la cual elimina también todas las camisetas que pertenezcan a ese equipo eliminado, obtener equipo mediante su id y finalmente actualizar equipo.
- En tercer lugar tenemos el **controlador de ligas**, con funciones como insertar liga, listar todas las ligas, eliminar liga la cual elimina también todos los equipos que se encuentren asociados a la liga que se intenta eliminar y además se eliminan también todas las camisetas asociadas a dichas camisetas, lugar tenemos el obtener liga por el id y finalmente el método de actualizar liga.
- En cuarto lugar se encuentra el **controlador de marcas**, el cual es más de los mismo, tenemos el método de insertar marca, listar todas las marcas, eliminar

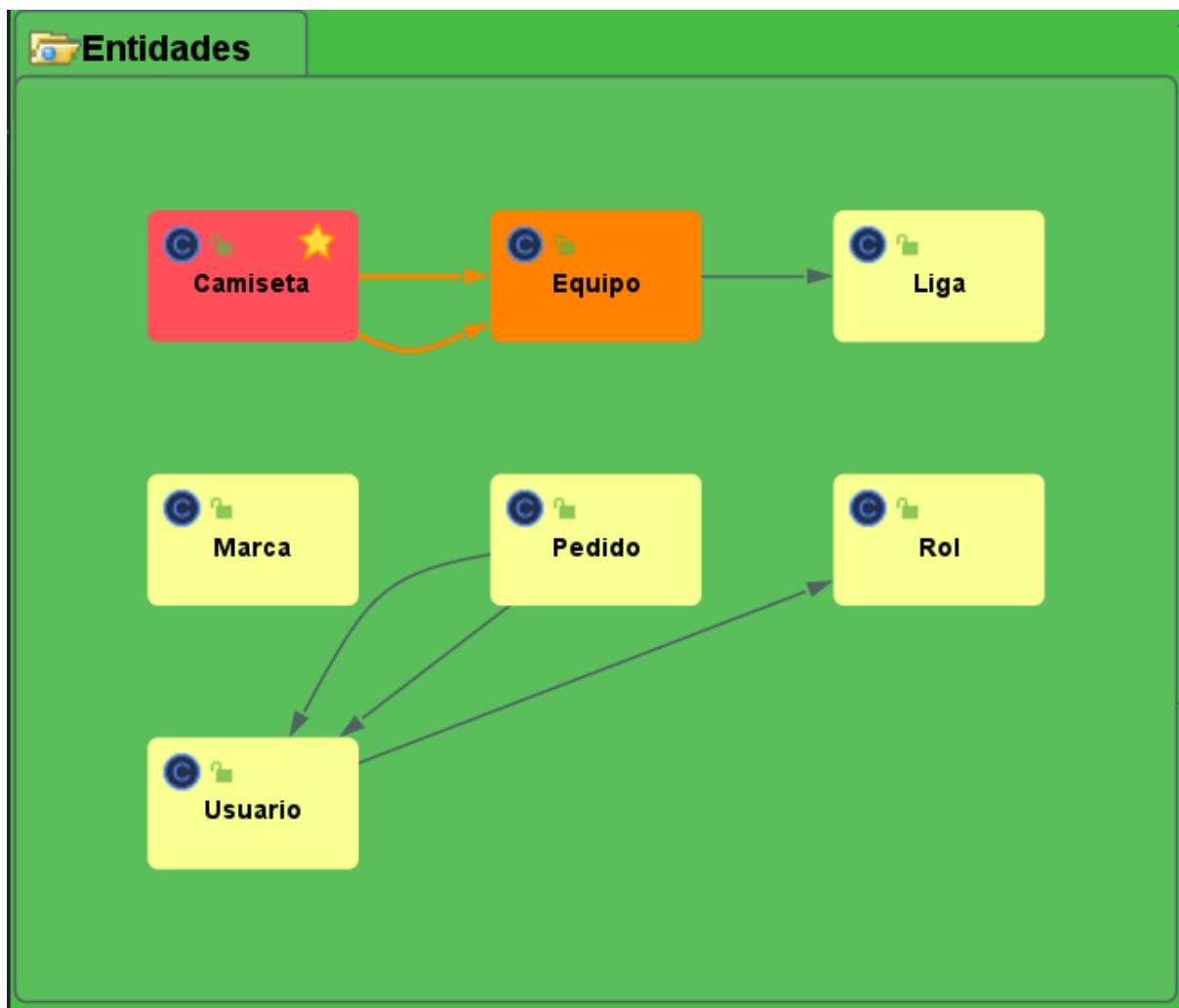
camisetas, el cual elimina también las camisetas con la marca que queremos eliminar, obtener marca por id y actualizar marca.

- En quinto lugar tenemos el **controlador de pedidos**, en el cual encontramos métodos como, listar todos los pedidos, obtener los pedidos de un usuario, realizar pedido, cambiar estado del pedido, que sirve para cambiar el estado del pedido de pendiente de pago a pagado.
- En sexto lugar tenemos el **controlador de roles**, en el cual solamente encontraremos una única función que es la de listar los roles existentes.
- En séptimo lugar tenemos el **controlador de sesión** que tiene tres métodos, el primer método es iniciar sesión que sirve para iniciar sesión con un usuario dentro de la aplicación y guardar la sesión hasta que se cierre, en segundo lugar tenemos el método para hacer logout, es decir, para cerrar sesión con el usuario que está iniciado y finalmente el get usuario que sirve para recuperar el usuario con todos sus datos que está iniciado sesión.
- Finalmente tenemos el **controlador de usuarios** con las funciones típicas, tenemos insertar usuario, comprobar si un usuario existe mediante su email y actualizar un usuario.



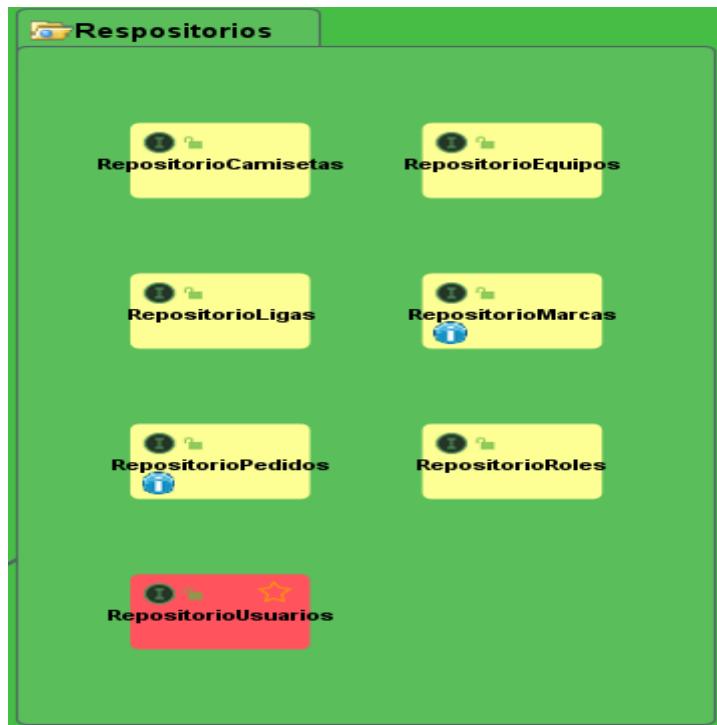
El **tercer paquete** que nos encontramos es el de entidades, el cual guarda todos los objetos que va a tener mi aplicación con todos sus atributos necesarios, nos encontramos siete clases:

- La primera entidad que encontramos es la de **Camiseta**, cuyos atributos son id, nombre, precio, descripción foto, equipo y marca.
- La segunda entidad es **Equipo**, con atributos como id, nombre, foto y liga.
- También tenemos la entidad **Liga**, con id, nombre y foto.
- Después encontramos **Marca** con los atributos id, nombre y foto.
- Ahora tenemos **Pedido**, con atributos como id, fecha, estado, importe, usuario y camisetas.
- Tenemos la entidad **Rol** con los atributos id y nombre.
- Finalmente encontramos la entidad **Usuario** con atributos como id, nombre, apellido, código postal, teléfono, ciudad, dirección, email, password, foto y rol.



El **cuarto paquete** que encontramos se trata del paquete de repositorios, en el que encontramos un repositorio por entidad que hay y todos los repositorios extienden de JpaRepository, para obtener todos los métodos y funciones que nos ofrece esta interfaz.

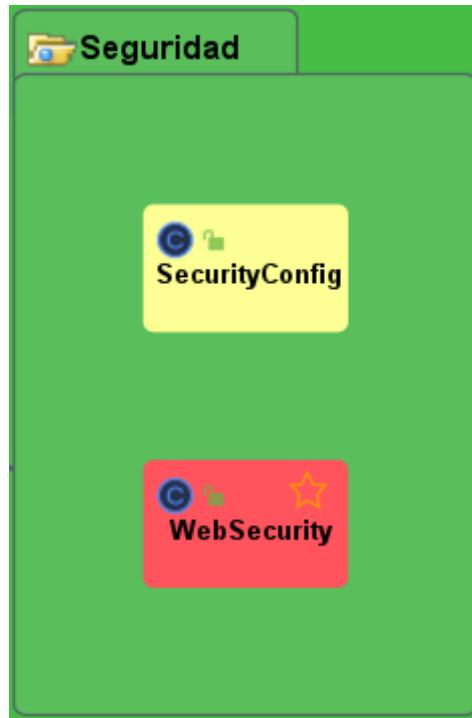
- En **Repositorio Camisetas** encontramos dos métodos adicionales, uno para buscar todas las camisetas según el id de su marca y otro para buscar todas las camisetas en función del id de su equipo.
- En **Repositorio Equipos** encontramos tres métodos adicionales, uno para eliminar equipos según la liga a la que pertenecen, otro para encontrar una liga por su nombre y otro para encontrar una liga por su id.
- En **Repositorio Ligas** solo encontramos un método adicional para encontrar ligas por su nombre.
- En **Repositorio Marcas** encontramos un método para encontrar marcas por su nombre.
- En **Repositorio Pedidos** encontramos un método para encontrar todos los pedidos de un usuario y otro para eliminar los pedidos que contengan una camiseta específica.
- En **Repositorio Roles** encontramos un único método para encontrar roles por su nombre.
- En **Repositorio Usuarios** encontramos un método para encontrar usuarios por su email.



El **quinto paquete** que encontramos es el de seguridad, en el cual encontramos solamente dos clases que son WebSecurity y SecurityConfig.

- En **SecurityConfig**, encontramos dos método, securityFilterChain, lo que hace es permitir el acceso a ciertas partes de la página según las reglas que nosotros pongamos, en mi caso, permito el acceso a todos sitios, ya que es en los controladores y en el front donde restringo el acceso a ciertas acciones a los usuarios. También encontramos passwordEncoder, que lo utilizo para encriptar las contraseñas que quiero guardar o desencriptar las contraseñas que quiero comparar con otras.

- En **WebSecurity** configuramos el CORS para permitir solicitudes desde la dirección <http://localhost:4200> y habilita los métodos GET, POST, PUT, OPTIONS, DELETE para cualquier ruta. También permite todas las cabeceras y soporta el uso de credenciales en las solicitudes.



El **sexto paquete** y último, solamente contiene la clase session, la cual se utiliza para almacenar la sesión de usuario y guardarla durante un tiempo determinado y así se pueda iniciar sesión correctamente y almacenar un usuario para recuperar todos sus datos de manera rápida y sencilla. Los métodos que tiene esta clase son iniciar sesión, logout y obtener usuario.

Mi aplicación, también cuenta con una carpeta dentro del paquete de resources en la que almaceno todas las imágenes que va a coger mi aplicación cuando la inicio para introducir los datos de prueba en la base de datos.

Además, en resources también se encuentra el archivo de "application.properties" en el cual se encuentra la url de la base de datos, el dialecto, el nombre de usuario, la contraseña y el driver entre otras cosas. Este archivo será necesario modificarlo si en lugar de querer conectarnos a una base de datos mysql nos quisieramos conectar a una oracle, postgresql o cualquier otra.

FrontEnd

La estructura de mi FrontEnd es muy similar a la que utilizo en el Back, mi proyecto consta de cinco carpetas principales, además de los archivos y carpetas que se generan al crear un proyecto angular.

La **primera carpeta** se trata de la carpeta de guards en la que nos encontramos con un auth guard que utilizo para que en el app-routing comparar si el rol del usuario que está con la sesión iniciada es el mismo que el que espero y si lo es, le da acceso a esa ruta y permite al usuario acceder, en caso de que no lo sea, envía al usuario a la ruta de error.

Aquí se puede ver un ejemplo de como utilizo el authguard en mi app-routing:

```
{  
  path: 'gestionLigas',  
  component: GestionLigasComponent,  
  canActivate: [authGuard], data: { expectedRole: 'Admin' }  
},  
{
```

Así, permito que a la ruta de gestionLigas solo puedan acceder los usuarios cuyo rol sea Admin, si no tiene ese rol, aparecerá esta pantalla:

[Camisetas](#) [Marcas](#) [Ligas](#)

Oops!

Parece que no tienes los permisos suficientes para acceder o la página no existe.

Contacta con el soporte técnico para más información.

[Volver](#)

Luego nos encontramos la carpeta includes en la que podemos encontrar dos componentes que son el navbar y el footer. Ambos componentes se encontrarán presentes todo el rato en la página y entre medias de estos se irán poniendo el resto de componentes según la ruta en la que estemos.

En el **navbar** tenemos varios elementos, entre ellos se encuentran:

- A la izquierda del todo tenemos el logo de la tienda, el cual es un enlace al inicio de la página que en tamaño para móviles pasa a estar centrado en la parte de arriba de la página.
- Un poco más a la derecha el nombre de la tienda que es “Spain Camisetas”, que en tamaño para móviles pasa a estar debajo del logo.
- Y a la derecha encontramos el botón de iniciar sesión que cuando se inicia sesión se convierte en un desplegable con mi perfil y si el usuario que ha iniciado sesión es cliente también aparece “mi cesta” para ver las camisetas que tengo en la cesta, y todo esto en tamaño para móviles pasa a estar debajo de lo anterior.
- Y finalmente debajo de todo esto tenemos un menú horizontal en el que encontramos “camisetas” para ver las camisetas disponibles, “ligas” para ver todas las ligas de fútbol que hay disponibles en la página, “marcas” al igual que ligas para ver las marcas disponibles en la página. También, si el usuario que ha iniciado sesión es admin aparecerá un nuevo elemento que se llamará “gestiones internas” el cual será un desplegable con cuatro posibles rutas:
 - **Gestión marcas:** En esta ruta podremos hacer un mantenimiento o CRUD de las marcas.
 - **Gestión de ligas:** En esta ruta podremos hacer un mantenimiento o CRUD de las ligas.
 - **Gestión de camisetas:** En esta ruta podremos hacer un mantenimiento o CRUD de las camisetas.
 - **Gestión de equipos:** En esta ruta podremos hacer un mantenimiento o CRUD de los equipos.

Y si iniciamos sesión como superadmin, aparecerá un apartado en la barra de navegación para crear usuarios de cualquier rol.

Aquí dejo unos ejemplos de cada una de las gestiones internas:

Gestión de Marcas

Gestion de Marcas

Nombre:

Ningún archivo seleccionado

ID	Nombre	Foto	Operaciones
1	Adidas		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
2	Nike		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

Gestión de Ligas

Gestion de Ligas

Nombre:

Ningún archivo seleccionado

ID	Nombre	Foto	Operaciones
1	LaLiga		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
2	Premier League		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

Gestión de Camisetas

Gestión de Camisetas

Nombre:

Precio:

Descripción:

Marca:

Equipo:

Ningún archivo seleccionado

ID	Nombre	Precio	Descripción	Marca	Equipo	Foto	Operaciones
2	Camiseta FC Barcelona 2023/2024	34	1ª Equipación del FC Barcelona de la temporada 2023/2024				<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
4	Camiseta Liverpool 2023/2024	35	1ª Equipación del Liverpool de la temporada 2023/2024				<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

Gestión de Equipos

Gestión de Equipos

Nombre:

Liga:

Ningún archivo seleccionado

ID	Nombre	Liga	Foto	Operaciones
1	Real Madrid			<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
2	FC Barcelona			<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

Aquí te muestro como se vería el navbar en diferentes situaciones:

Situación normal, al entrar a la página:



Al iniciar sesión como cliente:



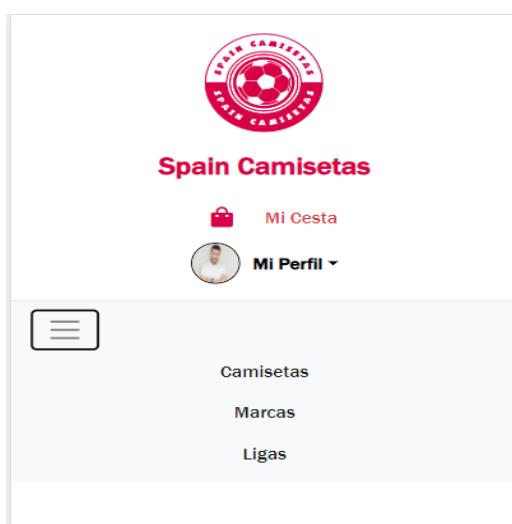
Al iniciar sesión como admin:



Al iniciar sesión como superadmin:

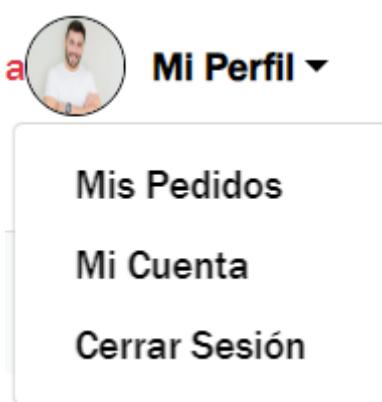


En tamaño para teléfonos:



Además, si desplegamos mi perfil también habrá varias opciones las cuales podrán ser:

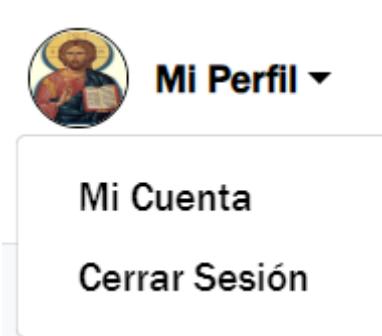
- Si eres **cliente** aparecerá mis pedidos, mi cuenta y cerrar sesión.



- Si eres admin aparecerá todos los pedidos, mi cuenta y cerrar sesión.



- Si eres superadmin aparecerá mi cuenta y cerrar sesión.



Y eso es todo acerca el navbar, más adelante entraremos en profundidad en cada una de las páginas que se encuentran en él y que hace cada una de ellos.

En cuanto al footer, es un footer simple con dos apartados, enlaces rápidos y redes sociales con enlaces que llevan a sus respectivos sitios, además aquí podemos ver el apartado sobre mi, para que la gente me conozca.

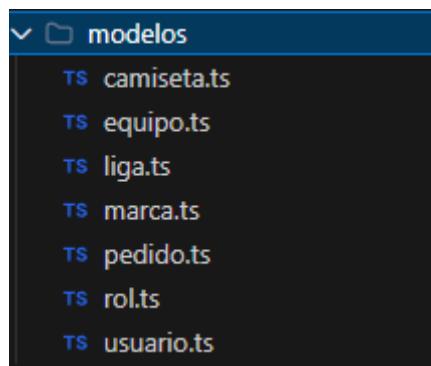
Enlaces rápidos

[Inicio](#)
[Ligas](#)
[Marcas](#)
[Sobre mi](#)

Redes Sociales

[Facebook](#)
 [Twitter](#)
 [Instagram](#)

La **segunda carpeta** del proyecto angular, es la carpeta modelos en la que almaceno todos los objetos que voy a utilizar en mi aplicación con sus atributos, exactamente los mismos que he utilizado en el back:



Aquí te dejo un ejemplo de uno de ellos para que puedas ver como son, son todos iguales pero cada uno con sus atributos:

```
import { Equipo } from "./equipo";
import { Marca } from "./marca";

export class Camiseta {
    id!:number;
    nombre!:string;
    precio!:number;
    descripcion!:string;
    foto!:string;
    equipo!:Equipo;
    marca!:Marca;

}
```

Ahora pasamos a la **tercera carpeta** que se trata de la carpeta de páginas, en la que encontraremos todos y cada uno de los componentes necesarios que hay en nuestra página web, en esta encontramos:

- **Camisetas:** Es la página inicial de mi aplicación, la que nos aparece al iniciar la aplicación, en esta podemos ver un listado de todas las camisetas que hay

disponibles en la tienda, además en la parte superior, nos encontraremos un buscador para buscar camisetas por su nombre y un filtro para encontrar todas las camisetas de una marca específica. También tengo una paginación para que en cada página solamente hay seis camisetas como máximo.

Filtrar por marca



Camiseta FC Barcelona 2023/2024
Precio: 34€



Camiseta Liverpool 2023/2024
Precio: 35€



Camiseta Paris Saint Germain 2023/2024
Precio: 34€



Camiseta Monaco 2023/2024
Precio: 37.5€



Camiseta Juventus 2023/2024
Precio: 34.7€



Camiseta Inter de Milán 2023/2024
Precio: 34.99€

1 2

- **Carrito:** El componente carrito, es un componente que utilizo para procesar los pedidos de camisetas, las camisetas que voy añadiendo al carrito se almacenan aquí y cuando entro puedo realizar un pedido de las camisetas que tengo añadidas en el mismo o si alguna no la quiero puedo eliminarla:

Carrito de Compras

	Camiseta FC Barcelona 2023/2024 1ª Equipación del FC Barcelona de la temporada 2023/2024	34.00€	
	Camiseta Inter de Milán 2023/2024 1ª Equipación del Inter de Milán de la temporada 2023/2024	34.99€	
Total: 68.99€			Comprar

- **Creación Usuarios:** Este componente es único para los superadmin, para que puedan crear usuarios con cualquier rol con todos los datos necesarios:

Crear Usuario

Nombre:	Apellido:
<input type="text"/>	<input type="text"/>
Código Postal:	Teléfono:
<input type="text"/>	<input type="text"/>
Ciudad:	Dirección:
<input type="text"/>	<input type="text"/>
Email:	
<input type="text"/>	
Contraseña:	
<input type="text"/>	
Rol:	
<input style="width: 100%; height: 30px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 5px;" type="text"/> Foto: <div style="display: flex; justify-content: space-between; align-items: center;"> <input style="border: 1px solid #ccc; padding: 2px 10px; margin-right: 10px;" type="button" value="Seleccionar archivo"/> <input style="border: 1px solid #ccc; padding: 2px 10px; background-color: #ccc; color: inherit; border-radius: 5px; font-size: 0.8em; margin-right: 10px;" type="button" value="Browse"/> </div>	
Crear Usuario	

- **Error:** Este componente lo uso para que en caso de que un usuario quiera acceder a una página que no tiene permiso o que no existe le salte en la pantalla para decirle que no puede acceder o que no existe:

Oops!

Parece que no tienes los permisos suficientes para acceder o la página no existe.

Contacta con el soporte técnico para más información.

[Volver](#)

- **GestionCamisetas:** Como bien he dicho anteriormente, este componente se encuentra en el navbar y sirve para hacer un CRUD de todas las camisetas que existen en la página:

Gestión de Camisetas

Nombre:

Precio:

Descripción:

Marca:

Equipo:

Ningún archivo seleccionado

ID	Nombre	Precio	Descripción	Marca	Equipo	Foto	Operaciones
2	Camiseta FC Barcelona 2023/2024	34	1ª Equipación del FC Barcelona de la temporada 2023/2024				Eliminar Editar
4	Camiseta Liverpool 2023/2024	35	1ª Equipación del Liverpool de la temporada 2023/2024				Eliminar Editar

- **GestionEquipos:** Al igual que todas las gestiones sirve para hacer un CRUD de los equipos disponibles:

Gestión de Equipos

Nombre:

Liga:

Ningún archivo seleccionado

ID	Nombre	Liga	Foto	Operaciones
1	Real Madrid			<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
2	FC Barcelona			<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

- **GestionLigas:** También es para hacer un CRUD de todas las ligas disponibles en la página:

Gestion de Ligas

Nombre:

Ningún archivo seleccionado

ID	Nombre	Foto	Operaciones
1	LaLiga		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
2	Premier League		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

- **GestionMarcas:** Finalmente el último CRUD es el de marcas que funciona igual que los anteriores:

Gestion de Marcas

Nombre:

Ningún archivo seleccionado

ID	Nombre	Foto	Operaciones
1	Adidas		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>
2	Nike		<input type="button" value="Eliminar"/> <input type="button" value="Editar"/>

- **Ligas:** El componente de ligas lo utilizo para mostrar a los usuarios todas las ligas disponibles en la página para que así puedan hacerse una idea de todas las ligas que hay, si entramos a este componente siendo administradores también podremos eliminar la liga que queramos y si es como cliente no aparecerá el botón:

Camisetas Marcas Ligas Gestiones Internas ▾



LaLiga



Premier League



Serie A

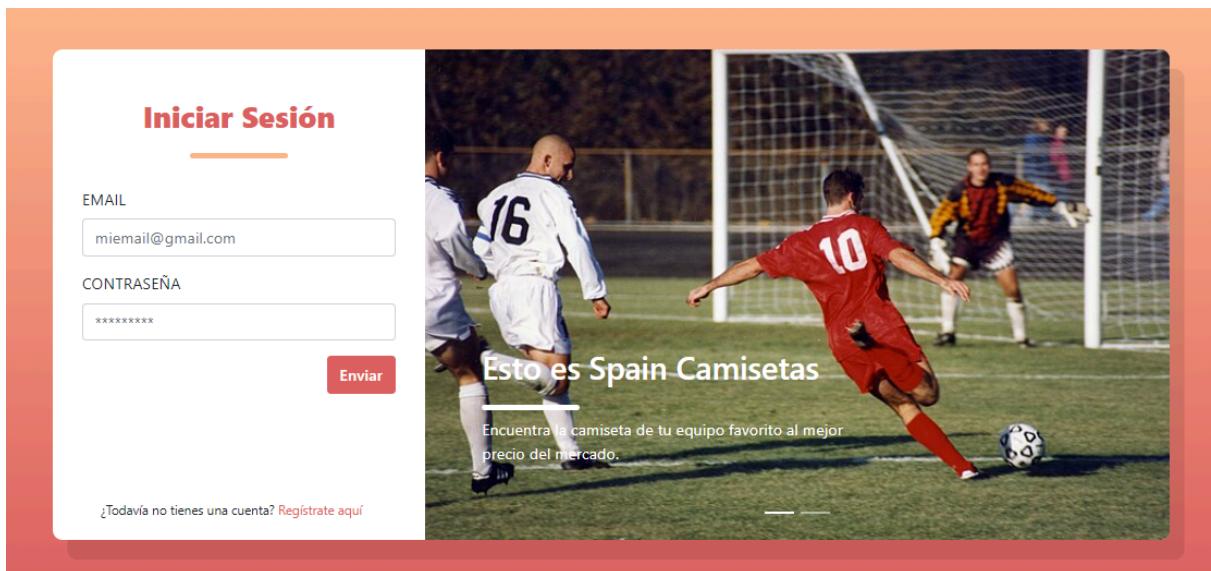


Bundesliga

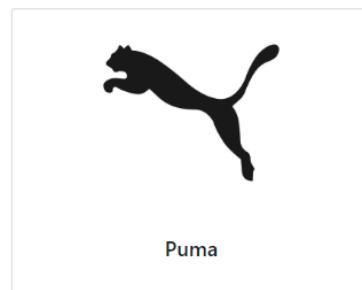


Ligue 1

- **Login:** El componente login, lo utilizo para iniciar sesión con un usuario y que ese usuario se guarde en el back en una clase sesion y así poder acceder a él cuando quiera:



- **Marcas:** El componente marcas es igual que el de ligas, tienen la misma función, mostrar a el cliente las marcas disponibles en la página y si entro como admin poder borrarlas directamente desde ahí:



- **MiCuenta:** El componente de mi cuenta es un componente que se encuentra en el desplegable de miperfil en el navbar y sirve para que el usuario pueda acceder a sus datos personales y modificarlos a su gusto si quiere cambiar la dirección, la foto de perfil o lo que sea:



Perfil de Usuario

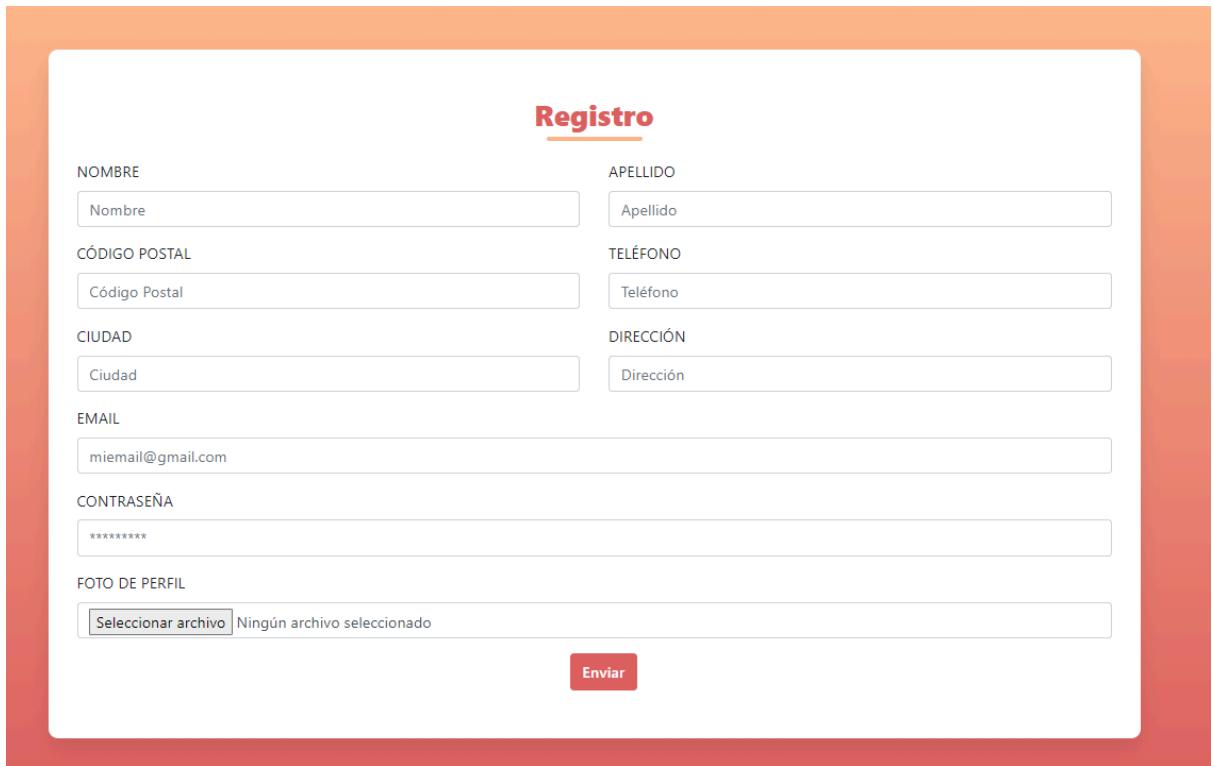
Nombre:	Ciudad:
<input type="text" value="Pepe"/>	<input type="text" value="Madrid"/>
Apellido:	Código Postal:
<input type="text" value="Diaz"/>	<input type="text" value="42000"/>
Email:	Dirección:
<input type="text" value="pepe@gmail.com"/>	<input type="text" value="Avenida de los Gobernadores"/>
Teléfono:	
<input type="text" value="66666666"/>	
<input type="button" value="Editar Perfil"/>	

- **Pedidos:** El componente de pedidos, sirve para que cada usuario pueda ver todos los pedidos que ha hecho a lo largo del tiempo, ahí podrá ver las camisetas que compró, la fecha del pedido, el importe, etc:

Aquí tienes tus pedidos Pepe

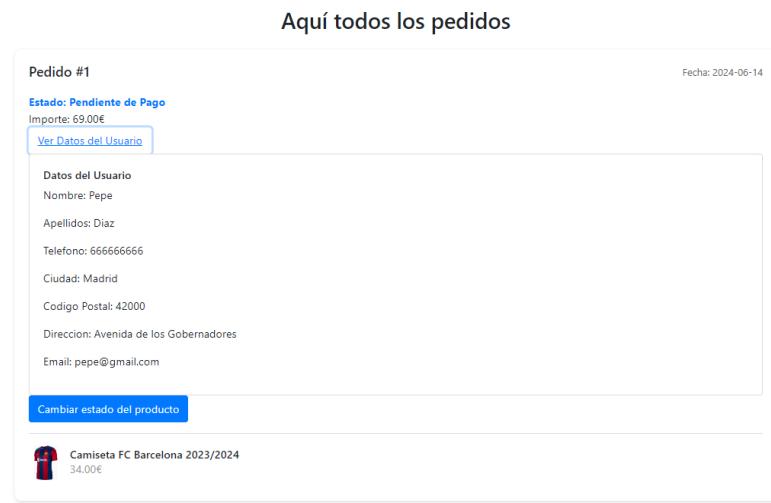
Pedido #1		Fecha: 2024-06-14
Estado: Pendiente de Pago		
Importe: 69.00€		
 Camiseta FC Barcelona 2023/2024 34€		
Pedido #3		Fecha: 2024-06-14
Estado: Pendiente de Pago		
Importe: 69.99€		
 Camiseta Liverpool 2023/2024 35€		
 Camiseta Inter de Milán 2023/2024 34.99€		

- **Registro:** El componente registro sirve para que un usuario cualquiera pueda darse de alta en la aplicación con sus datos, pero siempre creará un usuario con el rol de cliente, para que no tenga privilegios de más:



The image shows a registration form titled "Registro". It consists of several input fields arranged in pairs across two columns. The first column contains fields for "NOMBRE" (Name) and "CÓDIGO POSTAL" (Postal Code). The second column contains fields for "APELLIDO" (Last Name) and "TELÉFONO" (Phone Number). Below these are fields for "CIUDAD" (City) and "DIRECCIÓN" (Address). Further down are fields for "EMAIL" (Email) containing "miemail@gmail.com" and "CONTRASEÑA" (Password) containing "*****". A file upload field for "FOTO DE PERFIL" (Profile Photo) shows "Seleccionar archivo Ningún archivo seleccionado". At the bottom right is a red "Enviar" (Send) button.

- **TodosPedidos:** Este componente sirve para que los administradores puedan ver todos los pedidos que se han hecho en la aplicación junto con los datos del usuario que lo ha hecho y con un botón para marcar el producto como pagado o como pendiente de pago y así poder organizar los pedidos correctamente



The image shows a detailed view of an order. At the top, it says "Aquí todos los pedidos" (Here all the orders). Below that, it displays "Pedido #1" (Order #1) with the date "Fecha: 2024-06-14". It shows the order status as "Estado: Pendiente de Pago" (Status: Pending Payment) and the total amount "Importe: 69.00€". There is a blue "Ver Datos del Usuario" (View User Data) button. The user's data is listed below: Nombre: Pepe, Apellidos: Diaz, Telefono: 666666666, Ciudad: Madrid, Codigo Postal: 42000, Direccion: Avenida de los Gobernadores, Email: pepe@gmail.com. At the bottom, there is a blue "Cambiar estado del producto" (Change product status) button and a product item shown: "Camiseta FC Barcelona 2023/2024" at "34.00€".

- **VerCamisetas:** Finalmente, el último componente que tengo es el de ver camiseta, que sirve para ver la camiseta que se ha seleccionado en el componente camisetas y poder ver sus especificaciones y así poder añadirla al carrito de la compra para posteriormente comprarla:



Camiseta Liverpool 2023/2024

1ª Equipación del Liverpool de la temporada 2023/2024

Precio: 35€

Marca: Kappa

Equipo: Liverpool

Añadir al carrito
 Volver

- **Sobre mí:** Este componente no tiene ninguna funcionalidad, simplemente es una especie de carta de presentación para la gente que no me conozca, con un pequeño currículum:

Presentación

Hola, mi nombre es José Campillo, tengo 20 años y soy de San Carlos del Valle. Acabo de finalizar mi formación profesional en el módulo de Desarrollo de Aplicaciones Web. Me encantaría seguir formándome en el mundo del desarrollo y de la programación. Soy una persona apasionada por la tecnología, siempre en busca de nuevos retos y oportunidades para mejorar mis habilidades. Me considero proactivo, autodidacta y con una gran capacidad de adaptación a nuevos entornos y equipos de trabajo.

Curriculum

Educación

- IES Modesto Navarro
Bachillerato - 2020/2022
- IES Juan Bosco
Desarrollo de Aplicaciones Web - 2022/2024

Experiencia Profesional

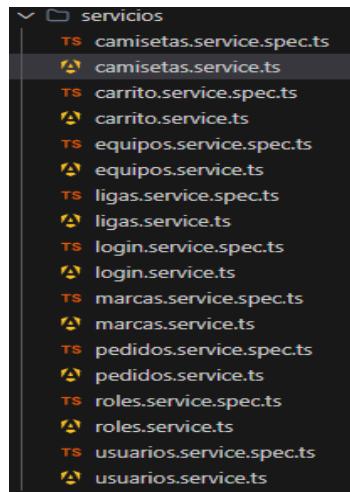
- Grupo de Desarrollo y mejora constante del SESCAM
FCT - 14/04/2024-21/06/2024
En el SESCAM hemos aprendido nuevas tecnologías como Angular, métodos de trabajo como SCRUM y hemos tocado un poco de KEOS.

Habilidades

- Desarrollo web (HTML, CSS, JavaScript, Angular)
- Programación en Java y PHP
- Base de datos (MySQL, Oracle)
- Control de versiones (Git)
- Trabajo en equipo y gestión de proyectos

Finalmente, la **quinta carpeta** es la de los servicios, que utilizo para conectar el front con el Back, tengo nueve servicios diferentes para saber que hace cada cosa y así poder organizarme mejor, los servicios son camisetas, carrito, equipo, ligas, login, marcas, pedidos, roles y usuarios, todos siguen la misma estructura, un url para conectarme con el back, se importa la clase httpClient para poder conectarme y los métodos necesarios que

requiera cada servicio, pongo un ejemplo de uno de ellos ya que todos son prácticamente iguales:



```

import { HttpClient } from '@angular/common/http';
import { Injectable, OnInit } from '@angular/core';
import { Observable } from 'rxjs';
import { Camiseta } from '../modelos/camiseta';

@Injectable({
  providedIn: 'root'
})
export class CamisetasService implements OnInit{
  APIurl = "http://localhost:8080/camisetas/";

  constructor(private httpClient:HttpClient) { }

  ngOnInit(): void {
    throw new Error('Method not implemented.');
  }

  public obtenerCamisetas():Observable<Camiseta[]>{
    return this.httpClient.get<Camiseta[]>(` ${this.APIurl}`);
  }

  public eliminarCamiseta(id:number):Observable<Camiseta>{
    return this.httpClient.delete<Camiseta>(` ${this.APIurl}${id}`);
  }

  public obtenerCamisetaById(id: number): Observable<Camiseta> {
    return this.httpClient.get<Camiseta>(` ${this.APIurl}${id}`);
  }

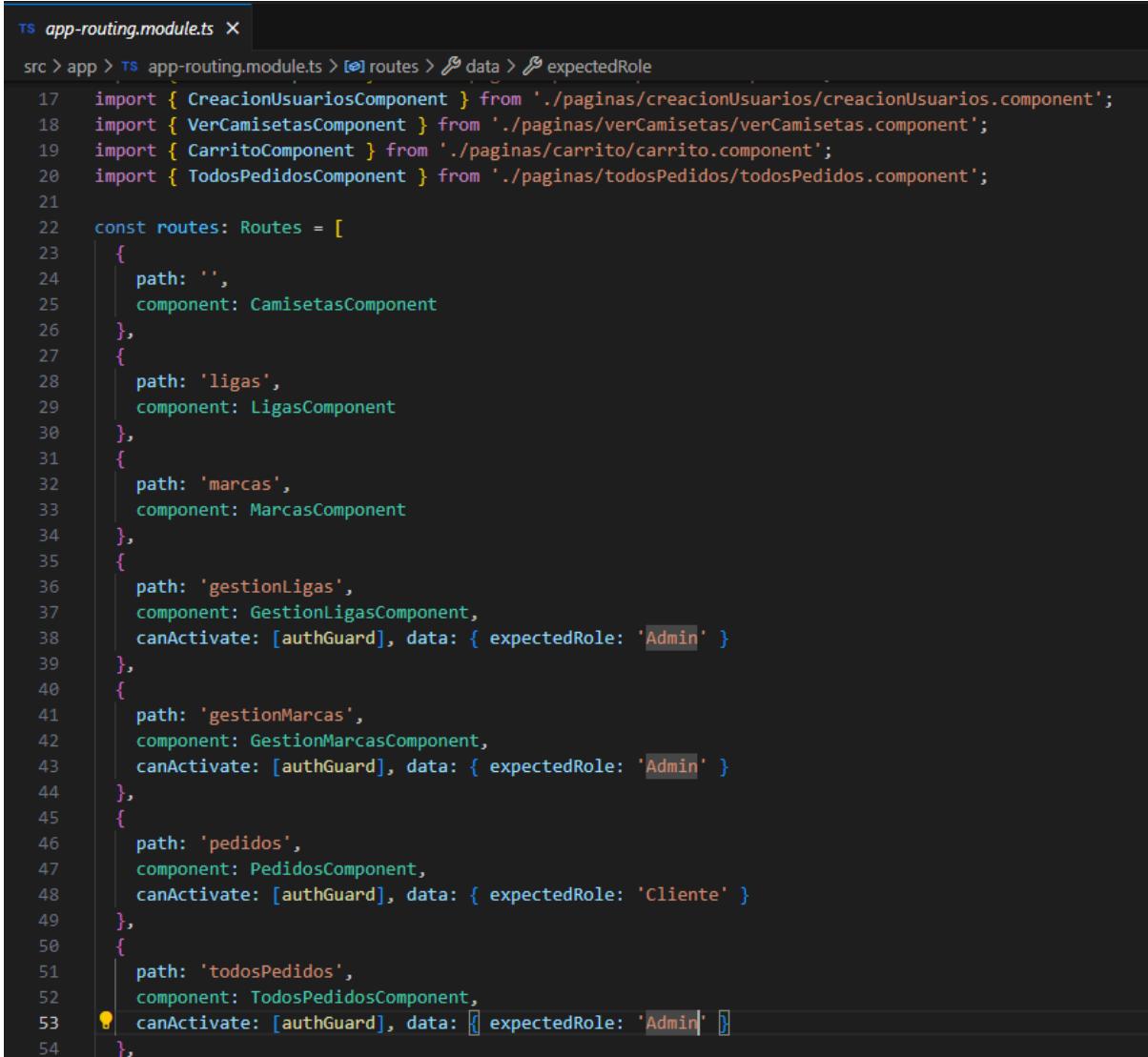
  public insertarCamiseta(camiseta: Camiseta): Observable<Camiseta> {
    return this.httpClient.post<Camiseta>(` ${this.APIurl}`, camiseta);
  }

  public actualizarCamiseta(camiseta: Camiseta): Observable<Camiseta> {
    return this.httpClient.put<Camiseta>(` ${this.APIurl}${camiseta.id}`, camiseta);
  }
}

```

Una vez finalizada la explicación de las carpetas me gustaría explicar los caminos que he realizado en los archivos que vienen con el proyecto angular como son app-routing, app.component.html, etc.

En **app-routing.module.ts** he declarado todas y cada una de las rutas necesarias para el correcto funcionamiento para poder acceder a los componentes:



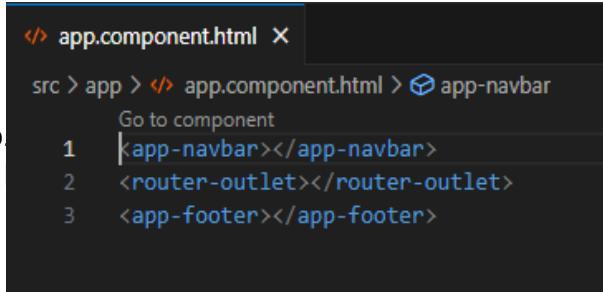
```

ts app-routing.module.ts ×
src > app > ts app-routing.module.ts > routes > data > expectedRole
17 import { CreacionUsuariosComponent } from './paginas/creacionUsuarios/creacionUsuarios.component';
18 import { VerCamisetasComponent } from './paginas/verCamisetas/verCamisetas.component';
19 import { CarritoComponent } from './paginas/carrito/carrito.component';
20 import { TodosPedidosComponent } from './paginas/todosPedidos/todosPedidos.component';
21
22 const routes: Routes = [
23   {
24     path: '',
25     component: CamisetasComponent
26   },
27   {
28     path: 'ligas',
29     component: LigasComponent
30   },
31   {
32     path: 'marcas',
33     component: MarcasComponent
34   },
35   {
36     path: 'gestionLigas',
37     component: GestionLigasComponent,
38     canActivate: [authGuard], data: { expectedRole: 'Admin' }
39   },
40   {
41     path: 'gestionMarcas',
42     component: GestionMarcasComponent,
43     canActivate: [authGuard], data: { expectedRole: 'Admin' }
44   },
45   {
46     path: 'pedidos',
47     component: PedidosComponent,
48     canActivate: [authGuard], data: { expectedRole: 'Cliente' }
49   },
50   {
51     path: 'todosPedidos',
52     component: TodosPedidosComponent,
53     canActivate: [authGuard], data: { expectedRole: 'Admin' }
54   },
55 ]

```

En **app.component.html** he dejado solamente como fijos el navbar y el footer y he añadido la etiqueta router-outlet para que las rutas que se van seleccionando aparezcan en medio de los mismos:

Finalmente en app.component.html



```

app.component.html ×
src > app > app.component.html > app-navbar
1 <app-navbar></app-navbar>
2 <router-outlet></router-outlet>
3 <app-footer></app-footer>

```

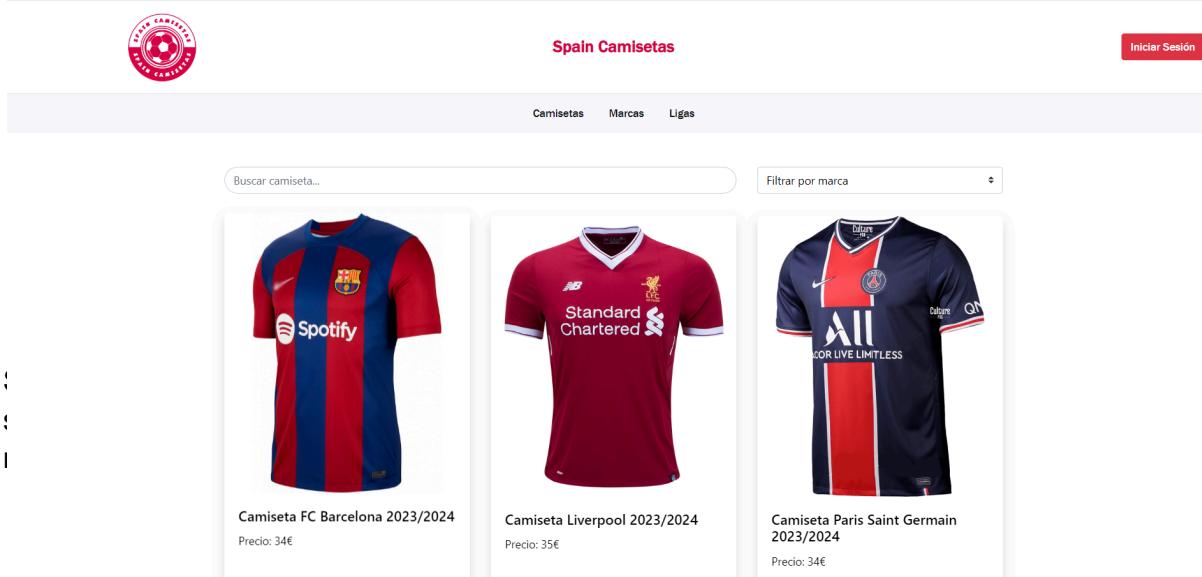
```
ts app.module.ts
src > app > ts app.module.ts > AppModule
  27 import { CarritoComponent } from './paginas/carrito/carrito.component';
  28 import { CarritoService } from './servicios/carrito.service';
  29 import { TodosPedidosComponent } from './paginas/todosPedidos/todosPedidos.component';
  30
  31 @NgModule({
  32   declarations: [
  33     AppComponent, NavbarComponent, FooterComponent, CamisetasComponent,
  34     LigasComponent, MarcasComponent, GestionCamisetasComponent, GestionEquiposComponent,
  35     GestionMarcasComponent, GestionLigasComponent, LoginComponent, RegistroComponent,
  36     ErrorComponent, MiCuentaComponent, PedidosComponent, CreacionUsuariosComponent, CarritoComponent,
  37     VerCamisetasComponent, TodosPedidosComponent
  38   ],
  39   imports: [
  40     BrowserModule,
  41     AppRoutingModule,
  42     NgbModule,
  43     HttpClientModule,
  44     FormsModule,
  45     ReactiveFormsModule,
  46     NgxPaginationModule
  47   ],
  48   providers: [
  49     provideClientHydration(), CarritoService
  50   ],
  51   bootstrap: [AppComponent]
  52 })
  53 export class AppModule { }
```

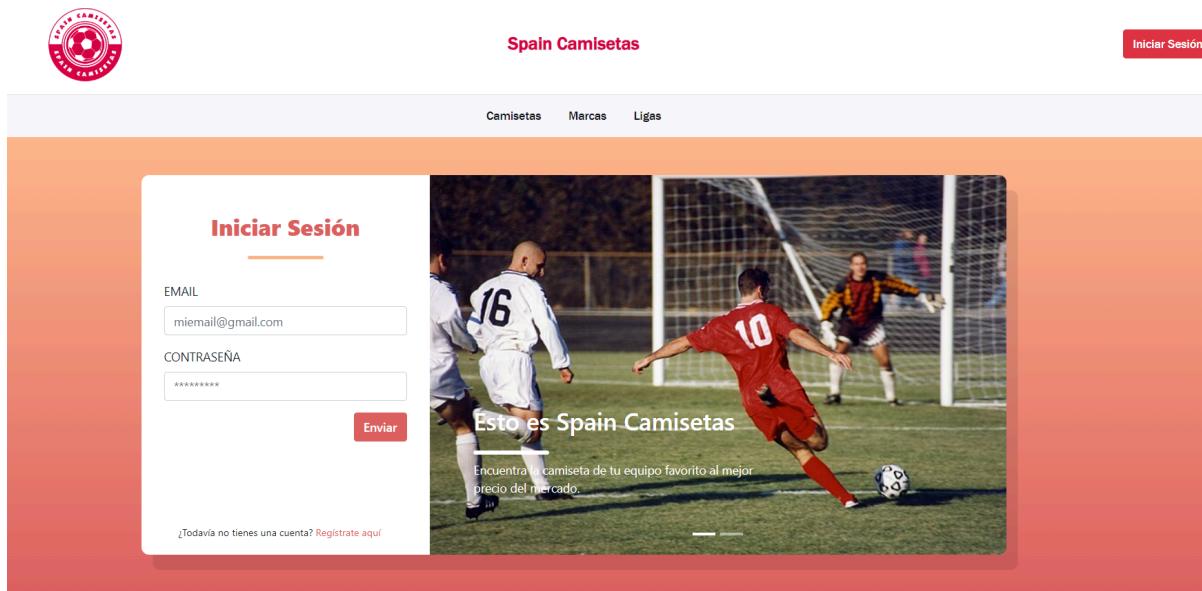
5.1. Documentación de usuario

En este apartado voy a explicar el proceso a seguir para hacer un buen uso de la aplicación por cualquier usuario que la utilice por primera vez.

Para empezar, al entrar a la página nos encontramos un listado de todas las camisetas que hay en la tienda las cuales podremos ver pero no comprar ya que si no estás iniciado sesión no podrás.

Arriba a la derecha encontraremos el botón para iniciar sesión:





Se nos abrirá un formulario para llenar con todos nuestros datos personales, la contraseña deberá de tener más de 6 caracteres y deberán estar todos los campos llenados para darse de alta en la aplicación:

Una vez registrado si todo está bien nos saltará una alerta indicando que todo ha ido correctamente e iniciamos sesión:

localhost:4200 dice
¡¡Usuario Registrado Correctamente!!

Aceptar

Spain Camisetas

Camisetas Marcas Ligas

Buscar camiseta... Filtrar por marca

Camiseta FC Barcelona 2023/2024 Precio: 34€

Camiseta Liverpool 2023/2024 Precio: 35€

Camiseta Paris Saint Germain 2023/2024 Precio: 34€

Y una vez dentro ya nos dejará comprar todas las camisetas que queramos, añadiéndolas al carrito y tramitando el pedido:

Camiseta FC Barcelona 2023/2024

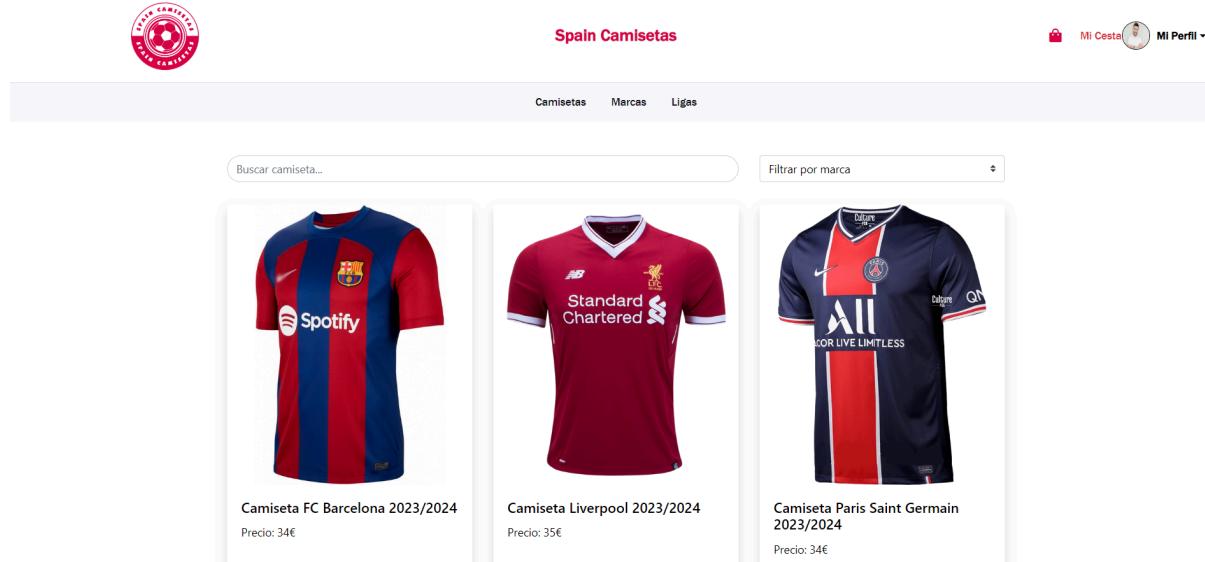
1ª Equipación del FC Barcelona de la temporada 2023/2024

Precio: 34€

Marca: Nike

Equipo: FC Barcelona

Añadir al carrito Volver



The screenshot shows the homepage of Spain Camisetas. At the top, there's a logo and a navigation bar with links for 'Camisetas', 'Marcas', and 'Ligas'. Below the header, a large section titled 'Carrito de Compras' (Shopping Cart) displays a single item: a red and blue striped FC Barcelona jersey. The product details say it's the '1ª Equipación del FC Barcelona de la temporada 2023/2024'. To the right of the product, the price '34.00€' is listed next to a delete icon. At the bottom of the cart section, it says 'Total: 34.00€' and has a blue 'Comprar' (Buy) button.

Aquí tienes tus pedidos Prueba

This screenshot shows the user's order history. It starts with 'Pedido #7' and indicates the status is 'Estado: Pendiente de Pago'. The date 'Fecha: 2024-06-17' is also shown. Below this, it lists the purchased item: a 'Camiseta FC Barcelona 2023/2024' at '34€'. There is a small thumbnail image of the jersey to the left of the product details.

Y así podremos hacer pedidos de las camisetas que queramos.
Si queremos editar los datos de nuestro usuario, solamente tenemos que ir a mi perfil y editarlos a tu gusto:



Perfil de Usuario

Nombre:	Ciudad:
<input type="text" value="Prueba"/>	<input type="text" value="prueba"/>
Apellido:	Código Postal:
<input type="text" value="Prueba"/>	<input type="text" value="11111"/>
Email:	Dirección:
<input type="text" value="prueba@gmail.com"/>	<input type="text" value="prueba, 56"/>
Teléfono:	
<input type="text" value="66666666"/>	

[Editar Perfil](#)

5.3. Manual de instalación y configuración

Para instalar y poder usar la aplicación en tu equipo local, deberás seguir los siguientes pasos:

1. En primer lugar deberemos de clonar el repositorio en tu ordenador y deberemos de tener instalado angular, node.js y una base de datos mysql como puede ser xampp.
2. Después, deberemos de crear una base de datos en phpmyadmin llamada camisetas.
3. Una vez creada la base de datos, deberemos arrancar el proyecto Spring Boot, el cual detectará la base de datos e introducirá todos los datos de prueba en las respectivas tablas.
4. Con las tablas ya creadas, es momento de iniciar nuestra aplicación angular, la cual obtendrá los datos de la base de datos y se conectará al Back para realizar todas las solicitudes que el usuario pueda hacer.
5. Nos creará tres usuarios un cliente, un admin y un superadmin, los datos de cada uno son:
 - a. Cliente: pepe@gmail.com 123456
 - b. Admin: admin@gmail.com admin123
 - c. Super Admin: superadmin@gmail.com superadmin123

Si has seguido todos los pasos, debería de funcionar la aplicación sin ningún problema, recomiendo usar el jdk 17 en nuestro IDE, para que descargue todas las dependencias correctamente.

6. Mantenimiento y evolución

6.1. Plan de mantenimiento y soporte

Para mantener correctamente la aplicación, deberemos llevar a cabo una revisión de todos los metodos y funcionalidades que desempeña.

Además, la aplicación se debería de ir actualizando cada poco tiempo ya que si quieres estar siempre con las camisetas más novedosas hay que añadirlas al catalogo de la tienda.

6.2. Identificación de posibles mejoras y evolución del proyecto

Las posibles mejoras que yo veo para esta aplicación son las siguientes:

1. Mejoraría la organización de los proyectos usando arquitecturas mejores como pueden ser la arquitectura hexagonal.
2. Mejoraría el estilo de la página, ya que en algunos aspectos no me he quedado conforme con este.

6.3. Actualizaciones y mejoras futuras

Las actualizaciones y mejoras futuras que se podrían añadir a la pagina serian:

- Un sistema de valoración de las camisetas, para que así los clientes puedan valorarlas.
- Un sistema de procesado de compra para que el cliente pague el importe del pedido por medio de la página.
- Más filtros para encontrar las camisetas que quieres más rápido.
- Sección de comentarios en las camisetas.

7. Conclusiones

7.1. Evaluación del proyecto

Me parece que el proyecto ha sido llevado a cabo correctamente, cumpliendo con mis objetivos que eran llevar a cabo una aplicación de venta de camisetas de fútbol de manera rápida e intuitiva.

He cumplido con todos los requisitos que se nos pedían como crear un FRONT, un BACK y una BDD que tuviera al menos 4 tablas, el apartado sobre mí para darme a conocer, etc.

He desarrollado una interfaz de usuario muy intuitiva, para que cualquier persona pueda acceder a la aplicación sin ningún problema y pueda tramitar su pedido de camisetas fácil y rápido.

7.2. Cumplimiento de objetivos y requisitos

1. **Base de Datos:** He cumplido con los requisitos de la base de datos de tener al menos 4 tablas, cuando se persisten las entidades del back se crean 8 tablas diferentes.
2. **BackEnd:** La aplicación se ha llevado a cabo con Java, haciendo uso de Spring Boot e Hibernate. Las contraseñas se encriptan para mejorar la seguridad de la aplicación.
3. **FrontEnd:** Se ha realizado el front usando angular, con lo que esto implica usar tambien TypeScript, HTML y CSS e incluso he usado bootstrap para una mejor disposición de la página.
4. La página es **responsive**, es decir, la puedes utilizar tanto en dispositivos grandes como en pequeños.
5. Los **formularios** están correctamente hechos, para que no puedan insertar cosas que no son en cada campo del formulario.

7.3. Lecciones aprendidas y recomendaciones para futuros proyectos

Gracias a este proyecto he conseguido profundizar mucho más en Angular, he conseguido entender mejor Spring Boot, he conseguido levantar API Rest sin ningún tipo de problemas y que todo funcione correctamente.

He mejorado en cuanto a GIT, ya que cada dia que hacía un cambio en la aplicación tenía que subir el cambio a GIT para que todo se me fuera guardando en mi repositorio.

Para futuros proyectos me gustaría organizarme de otra manera, haciendo uso de otra disposición de los archivos y probar otros lenguajes de programación para ampliar mi campo de conocimiento.

8. Referencias y enlaces de interés

Las páginas que más he consultado para obtener información son:

- <https://www.youtube.com/>
- <https://stackoverflow.com/>
- <https://openwebinars.net/>