Práctica Angular

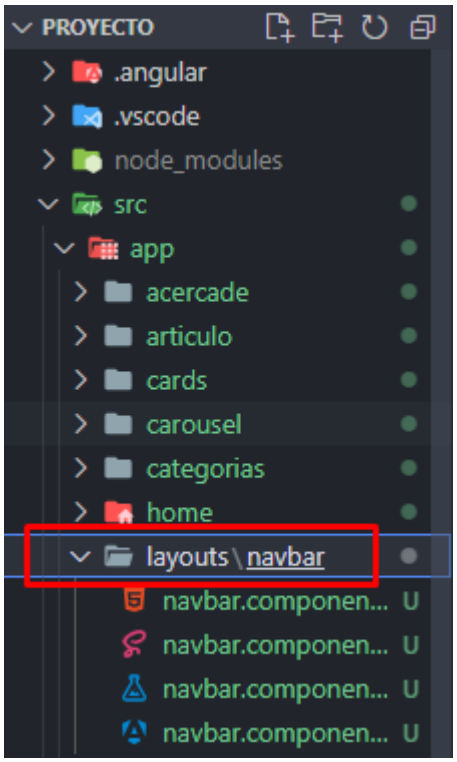
Daniel Sánchez Sánchez

Índice

A continuación voy a ir en orden paso a paso cubriendo todos los puntos que se piden realizar en la práctica y mostrando como los he resuelto.

- Navbar de Bootstrap con logotipo y enlaces

Para hacer el navbar cree un componente



Este es el código del navbar:

```
<nav class="navbar navbar-expand-lg bg-body-tertiary sticky-top">
    <div class="container-fluid">
        <a class="navbar-brand" [routerLink]="['']">
            <img src="/assets/kick.png" width="30" height="30" alt="" >
        </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item" [routerLinkActive]="['active']" [routerLinkActiveOptions]="{exact:true}">
            <a class="nav-link" [routerLink]="['']">Tienda</a>
          </li>
          <li class="nav-item" [routerLinkActive]="['active']" [routerLinkActiveOptions]="{exact:true}">
            <a class="nav-link" [routerLink]="['acerca']">Nosotros</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
```

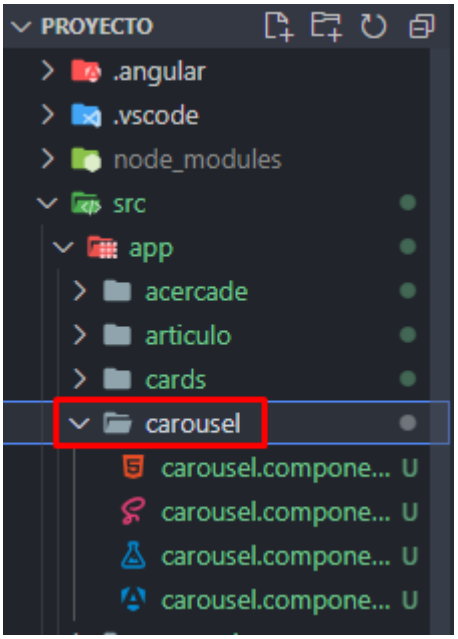
(los estilos podrás encontrarlos en el archivo scss del componente)

Y por último el enrutamiento en el archivo **app-routing.module** para que funcione:

```
const routes: Routes = [
  { path: '', component: HomeComponent},
  {path: 'acerca', component: AcercadeComponent},
  {path: 'articulos', component: ArticuloListComponent},
  {path: 'articulo-ficha/:idArticulo', component: ArticuloFormComponent}
];
```

- Carousel Bootstrap con 3 imágenes

Para ello cree un componente "carousel"



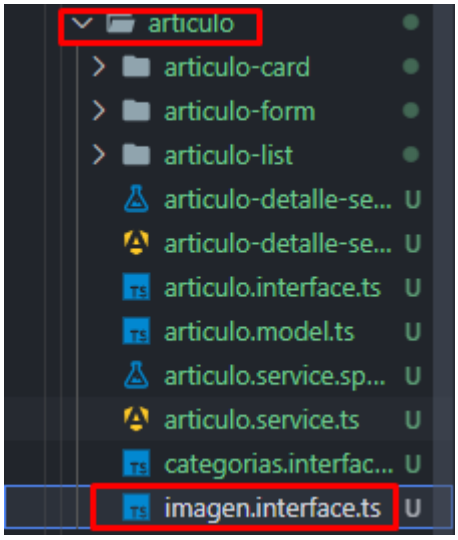
Este sería el código del carousel:

```
<script src="https://code.jquery.com/jquery-3.1.1.slim.min.js" integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed950n" crossorigin="anonymous"></script>

<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-inner">
      <div class="carousel-item" *ngFor="let imagen of imagenes; let i = index" [class.active]="i === 0">
        <img [src]="imagen.url" class="d-block w-100" alt="Imagen">
      </div>
    </div>
    <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Previous</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Next</span>
    </button>
  </div>
```

(los estilos podrás encontrarlos en el archivo scss del componente)

Como exigía que las imágenes se recuperasen de un backend (Mockoon) hice lo siguiente:

1. Cree una interfaz Imagen en la que tener una url de la imagen




```
export interface Imagen{
url: string;
}
```

2. En mi componente, importo la interfaz y HttpClient, para poder hacer las peticiones al backend. Creo un array de imágenes y lo lleno con las urls de las imágenes del backend. Al iniciar el componente se llamará a la función.

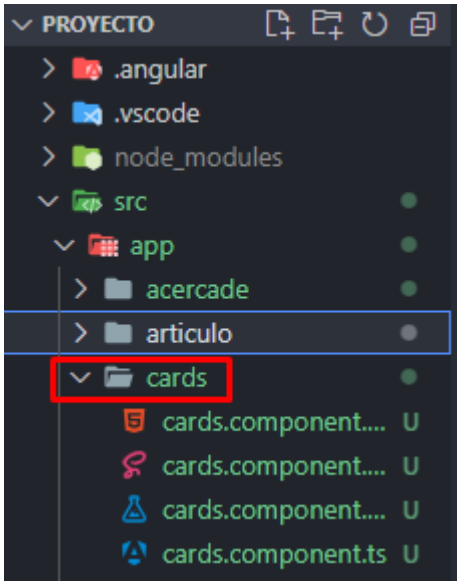
```
import { Component } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Imagen } from '../articulo/imagen.interface';

@Component({
  selector: 'app-carousel',
  templateUrl: './carousel.component.html',
  styleUrl: './carousel.component.scss'
})
export class CarouselComponent {

  imagenes: Imagen[] = [];

  constructor(private http: HttpClient){}

  ngOnInit():void {
    this.obtenerImagenes();
  }


  obtenerImagenes() {
    this.http.get<Imagen[]>("http://localhost:3001/carousel").subscribe(data => {
      this.imagenes = data;
    });
  }
}
```

3. Para poner las imágenes utilicé ngFor de Angular para pintar tantas imágenes (3) como hubiese en el array de imágenes previamente creado. Mediante [src] le asigno la imagen, atribuyéndole el valor de la url de cada objeto imagen.

```
<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-inner">
      <div class="carousel-item" *ngFor="let imagen of imagenes; let i = index" [class.active]="i === 0">
        <img [src]="imagen.url" class="d-block w-100" alt="Imagen">
      </div>
    </div>
```

- 3 tarjetas con imágenes

Para ello cree un componente "cards", el funcionamiento es exactamente el mismo que el del carrousel a la hora de recuperar las imágenes del backend, ya que utiliza la misma interfaz y función.



(los estilos podrás encontrarlos en el archivo scss del componente)

- Categorías

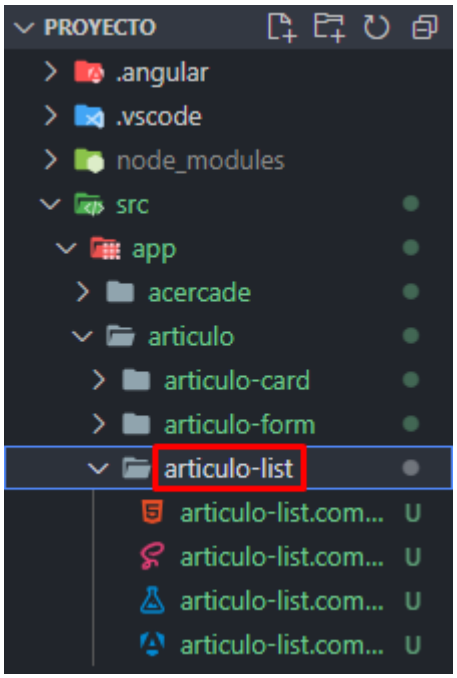
Para ello cree un componente "categorias", el funcionamiento es idéntico al anterior cambiando la interfaz por una que además de la url de la imagen añade el título de la categoría.

```
<div class="container">
    <a *ngFor="let card of imagenes" class="card text-bg-dark" [routerLink]="['articulos']">
      <img [src]="card.url" class="card-img" alt="...">
      <div class="card-img-overlay">
        <h5 class="card-title">{{ card.title }}</h5>
      </div>
    </a>
  </div>
```

Al hacer click en cualquiera de las categorías nos llevará a la lista de artículos (para todas las categorías es la misma para no alargar el backend innecesariamente).

-Lista de artículos

Para ello cree un componente "articulo-list", en el html, pinto tantos articulo-card (que ahora veremos) como elementos haya en el backend, además añado a cada artículo un evento que permitirá verlo más en detalle.



(los estilos podrás encontrarlos en el archivo scss del componente)

```
<div class="contenedor">
<app-articulo-card *ngFor="let articulo of articulos" [articulo]="articulo" [textoBoton]="'Ver más'" (click)="navegarFicha($event)"></app-articulo-card>
</div>
```

En la lógica del componente encontramos un array de artículos de la clase Articulo

```
export class Articulo {
    private id: number;
    private nombre: string;
    private precio: number;
    private descripcion?: string;
    private imagen: string;
    private fav?: boolean;
    private tallas?: string[];
```

```
export class ArticuloListComponent {
articulos: Articulo[] = [];
```

Además cree un servicio para llevar a cabo la llamada al backend, vamos a verlo:

```
@Injectable({
  providedIn: 'root'
})

export class ArticuloService {

  constructor(private http: HttpClient ) {}

   public obtenerArticulosRest(): Observable<IArticulo[]> {
    const urlEndPoint: string = "http://localhost:3001/items";
   return this.http.get<[IArticulo]>(urlEndPoint);
   }
}
```

Como se puede ver está utilizando una interfaz y no la clase Articulo, eso es porque no creaba bien los objetos e hice una interfaz para mapear el objeto:

```
export interface IArticulo{
    id: number;
    nombre: string;
    precio: number;
    imagen: string;
    descripcion?: string;
    fav?: boolean;
    tallas?: string[];
}
```

Una vez hecho esto volvemos al componente para ahora sí crear un objeto Articulo con los datos que hemos obtenido gracias al servicio:

```
private obtenerArticulosRest(){
this.articuloService.obtenerArticulosRest().subscribe(
  (data) => {
    data.forEach((articulo) => {
     const articuloAIncluir: Articulo = new Articulo(articulo.id, articulo.nombre, articulo.precio, articulo.imagen, articulo.descripcion, articulo.tallas, articulo.fav);
     this.articulos.push(articuloAIncluir);
    })
  }
)

}
```

Además añado la lógica para navegar a la ficha del artículo que seleccione, para ello cree otro servicio, vamos a verlo:

```
@Injectable({
  providedIn: 'root'
})
export class ArticuloDetalleService {
  private articuloSeleccionado?: Articulo;

  constructor() {}

  setArticuloSeleccionado(articulo: Articulo): void {
    this.articuloSeleccionado = articulo;
  }

  getArticuloSeleccionado(): Articulo | undefined {
    return this.articuloSeleccionado;
  }
}
```

Con esto ya podemos navegar a la ficha del articulo con la siguiente lógica:

```
public navegarFicha(idArticulo: number): void {
  const articuloSeleccionado = this.articulos.find(articulo => articulo.getId() === idArticulo);
  console.log(articuloSeleccionado);

  if (articuloSeleccionado)
   {
    console.log("entra");
    this.articuloDetalleService.setArticuloSeleccionado(articuloSeleccionado);
    this.router.navigate(['articulo-ficha', idArticulo]);
  }else{
    console.log("no entra")
  }
}
```

Como he dicho antes ahora vamos a ver la card de cada artículo en detalle:

```
import { Component, EventEmitter, Input, Output } from '@angular/core';
import { Articulo } from '../articulo.model';
import  localeES  from '@angular/common/locales/es';

import { registerLocaleData } from '@angular/common';

registerLocaleData(localeES)

@Component({
  selector: 'app-articulo-card',
  templateUrl: './articulo-card.component.html',
  styleUrl: './articulo-card.component.scss'
})
export class ArticuloCardComponent {
@Input() articulo?: Articulo;
@Output() click: EventEmitter<number> = new EventEmitter<number>();
@Input() textoBoton: string = "Ver más";

public pulsarBoton():void {

  this.click.emit(this.articulo?.getId());
}
}
```

En este componente hay muchas cosas a destacar, mediante Input le digo al componente que le va a llegar un articulo, con lo que relleno la card con los datos que vengan del backend de cada artículo. Con Output digo que devolverá (emitirá) un evento con un número (el id del articulo). También he importado "localeES" y "registerLocalData" para poder hacer un currencypipe para que el artículo tenga el precio con formato español como se pedía. Ahora veremos como he implementado todo esto en el html:

```
<div class="card" style="width: 18rem;" *ngIf="articulo">

    <img class="card-img-top" [src]="articulo.getImagen()" alt="Imagen de zapatilla {{articulo.getNombre}}">
    <div class="card-body">
      <h5 class="card-title">{{articulo.getNombre()}}</h5>
      <p class="card-text" id="precio">{{ articulo.getPrecio() | currency:'EUR':'symbol':'1.2-2':'es-ES'}}</p>
      <a class="btn btn-primary " (click)="pulsarBoton()" id="boton" >{{textoBoton}}</a>
      <div class="icono">
        <img *ngIf="articulo.getFav()?.toString() === 'true'" src="assets/heart1.png" alt="" class="corazon">
        <img *ngIf="articulo.getFav()?.toString() !== 'true'" src="assets/heart0.png" alt="" class="corazon">
      </div>
    </div>

  </div>
```

Lo primero que hago es pintar la card si encuentra un articulo, después con [src] coger la imagen del artículo, más adelante con la sintaxis de la doble llave "{{ }}" cojo el nombre del artículo. Hago lo mismo para el precio añadiéndole el pipe del que hablaba antes. Añado un evento click al botón de la card que llama a una función que lo acciona. Por último llamo a getFav() de articulo para saber si es true o no y así pintar una imagen u otra como se pedía (ya que tenía que el campo tenía que venir del backend) usando ngIf de Angular para hacer una condición.

Una vez hecho todo esto vamos a ver el componente del detalle del artículo.

- Ficha del artículo

Para ello cree un componente "articulo-form" con la siguiente estructura:

```
<div class="product-container">
    <div class="contenedor-imagen">
    <img *ngIf="articuloSeleccionado?.getImagen()" [src]="articuloSeleccionado?.getImagen()" alt="Imagen del artículo" class="imagen-grande">
    </div>
    <div class="product-details">
      <h2>{{articuloSeleccionado?.getNombre()}}</h2>
      <p class="precio">{{ articuloSeleccionado?.getPrecio() | currency:'EUR':'symbol':'1.2-2':'es-ES'}}</p>
      <p class="desc-label">Descripción:</p>
      <p class="desc">{{articuloSeleccionado?.getDescripcion()}}</p>
      <p class="color">Color:</p>

      <div class="small-product-image">
        <img *ngIf="articuloSeleccionado?.getImagen()" [src]="articuloSeleccionado?.getImagen()" alt="Imagen del artículo" class="imagen-pequeña">
      </div>
      <div *ngIf="articuloSeleccionado">
        <p class="talla-label">Escoge una talla: </p>
        <div class="tallas-container">
          <div *ngFor="let talla of articuloSeleccionado.getTallas()" class="talla-box">
            {{ talla }}
          </div>
        </div>

        <button type="button" class="btn btn-primary btn-lg avisame" >Avísame 📩</button>

        <button type="button" class="btn btn-primary btn-lg carrito">Añadir al carrito</button>
      </div>
    </div>

  </div>
```

Si vamos a la lógica del componente vemos que tenemos un campo idArticulo y un articuloSeleccionado, usando el servicio que antes mostré recupero los campos del artículo seleccionado usando su idArticulo.

```
export class ArticuloFormComponent {

  idArticulo?: string
  articuloSeleccionado?: Articulo


constructor
(private route: ActivatedRoute,
  private articuloDetalleService: ArticuloDetalleService)
{}

ngOnInit(): void {
this.idArticulo = this.route.snapshot.paramMap.get('idArticulo') ?? undefined;

this.articuloSeleccionado = this.articuloDetalleService.getArticuloSeleccionado();

this.articuloSeleccionado?.getImagen()
}



}
```

Una vez hecho esto ya puedo coger todos los atributos del objeto y pintarlos en mi página:

```
<div class="product-container">
    <div class="contenedor-imagen">
    <img *ngIf="articuloSeleccionado?.getImagen()" [src]="articuloSeleccionado?.getImagen()" alt="Imagen del artículo" class="imagen-grande">
    </div>
    <div class="product-details">
      <h2>{{articuloSeleccionado?.getNombre()}}</h2>
      <p class="precio">{{ articuloSeleccionado?.getPrecio() | currency:'EUR':'symbol':'1.2-2':'es-ES'}}</p>
      <p class="desc-label">Descripción:</p>
      <p class="desc">{{articuloSeleccionado?.getDescripcion()}}</p>
      <p class="color">Color:</p>

      <div class="small-product-image">
        <img *ngIf="articuloSeleccionado?.getImagen()" [src]="articuloSeleccionado?.getImagen()" alt="Imagen del artículo" class="imagen-pequeña">
      </div>
      <div *ngIf="articuloSeleccionado">
        <p class="talla-label">Escoge una talla: </p>
        <div class="tallas-container">
          <div *ngFor="let talla of articuloSeleccionado.getTallas()" class="talla-box">
            {{ talla }}
          </div>
        </div>

        <button type="button" class="btn btn-primary btn-lg avisame" >Avísame 📩</button>

        <button type="button" class="btn btn-primary btn-lg carrito">Añadir al carrito</button>
      </div>
    </div>

  </div>
```

De esta manera cumplo con el requisito de que los atributos fuesen dinámicos.

Estructura de la aplicación

A continuación veremos cómo separar la web por componentes como nos permite Angular nos ha simplificado el código, este es mi componente home (la pestaña tienda):

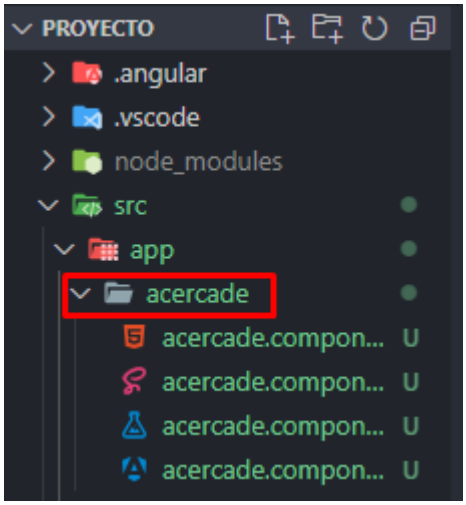
```
<app-carousel></app-carousel>
<h1>🔥 ¡Tendencia ahora mismo! 🔥</h1>
<app-cards></app-cards>
<h1> Categorías </h1>
<app-categorias></app-categorias>
```

Como vemos simplemente llamo a los componentes que hemos viste anteriormente usándolos como si fueran etiquetas html.

De esta misma forma mi componente app, el componente principal de la aplicación sólo tiene dos líneas, una en la que llama al navbar (que estará siempre presente) y otra en la que llamo a router-outlet que es la etiqueta que me permite ir cambiando los componentes.

```
<app-navbar></app-navbar>
<router-outlet></router-outlet>
```

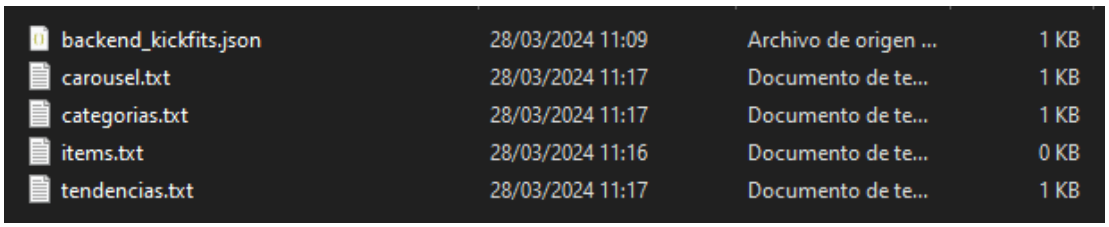
Para terminar la documentación y finalizar el apartado de tienda, volveré al principio para ver rápidamente la sección de "Nosotros":




```
<div class="container">
    <section id="historia">
        <h2>Nuestra Historia</h2>
        <p>Somos una tienda de ropa deportiva fundada en el año 2024. Desde nuestros inicios, nos hemos dedicado a proporcionar a nuestros clientes las mejores opciones en ropa deportiva para ayudarles a alcanzar sus objetivos de fitne
            Nuestra pasión por el deporte y el estilo de vida activo nos impulsa a ofrecer productos de alta calidad de las mejores marcas,
            así como un excelente servicio al cliente.
        </p>

        <img src="https://images.unsplash.com/photo-1448387473223-5c37445527e7?q=80&w=1800&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D">

    </section>

    <section id="tienda">
        <h2>Nuestra Tienda</h2>
        <p>Nuestra tienda ofrece una amplia gama de productos para satisfacer las necesidades de deportistas de todos los niveles y disciplinas.
            Desde ropa deportiva para correr, hacer yoga, entrenamiento en el gimnasio hasta calzado especializado, tenemos todo lo que necesitas para rendir al máximo y lucir bien mientras lo haces.
        </p>
        <img src="https://images.unsplash.com/photo-1562771379-eafdca7a02f8?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D">

    </section>
    <section id="equipo">
        <h2>Nuestro Equipo</h2>
        <p>Nuestro equipo está formado por apasionados del deporte y la moda que están comprometidos con proporcionar la mejor experiencia de compra a nuestros clientes. Con amplio conocimiento en fitness y moda deportiva, estamos aqu
            Disponemos de asesoría online 24h.
        </p>
        <img src="https://images.unsplash.com/photo-1483721310020-03333e577078?q=80&w=2128&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D">
    </section>

    <section id="valores">
        <h2>Nuestros Valores</h2>

            <p><strong>Calidad:</strong> Nos comprometemos a ofrecer productos de la más alta calidad para garantizar el rendimiento y la durabilidad.</p> <br>
            <p><strong>Servicio al Cliente:</strong> Nuestro objetivo es superar las expectativas de nuestros clientes con un servicio personalizado y amable.</p> <br>
            <p><strong>Comunidad:</strong> Creemos en construir una comunidad activa y positiva en torno al deporte y el estilo de vida saludable.</p>

            <img src="https://images.unsplash.com/photo-1518622358385-8ea7d0794bf6?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D">
    </section>

    <div class="cta">
        <h2>Conéctate con nosotros</h2>
        <p>Síguenos en nuestras redes sociales para mantenerte al día con las últimas novedades, ofertas especiales y eventos:</p>
        <p>
            <a href="https://www.facebook.com/"><img src="assets/face.png" alt="Facebook" id="face"></a> |
            <a href="https://www.instagram.com/"><img src="assets/instagram.png" alt="Instagram" id="insta"></a> |
            <a href="https://twitter.com/"><img src="assets/x.png"alt="Twitter" id="x"></a>
        </p>
    </div>
</div>
```

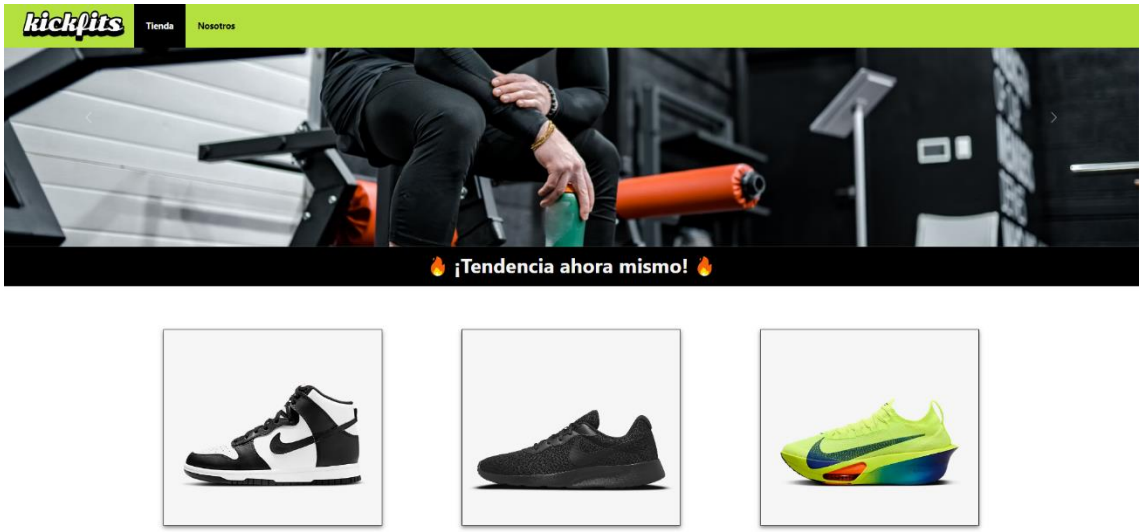
¡¡IMPORTANTE!!

Para que el backend funcione importa el archivo json y pega dentro de cada endpoint el contenido de su correspondiente .txt

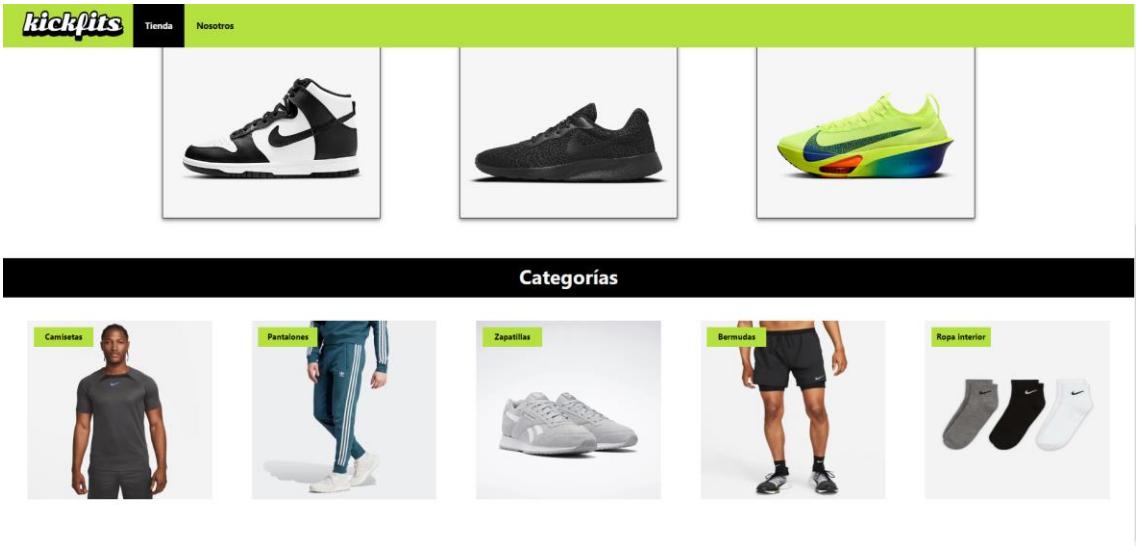


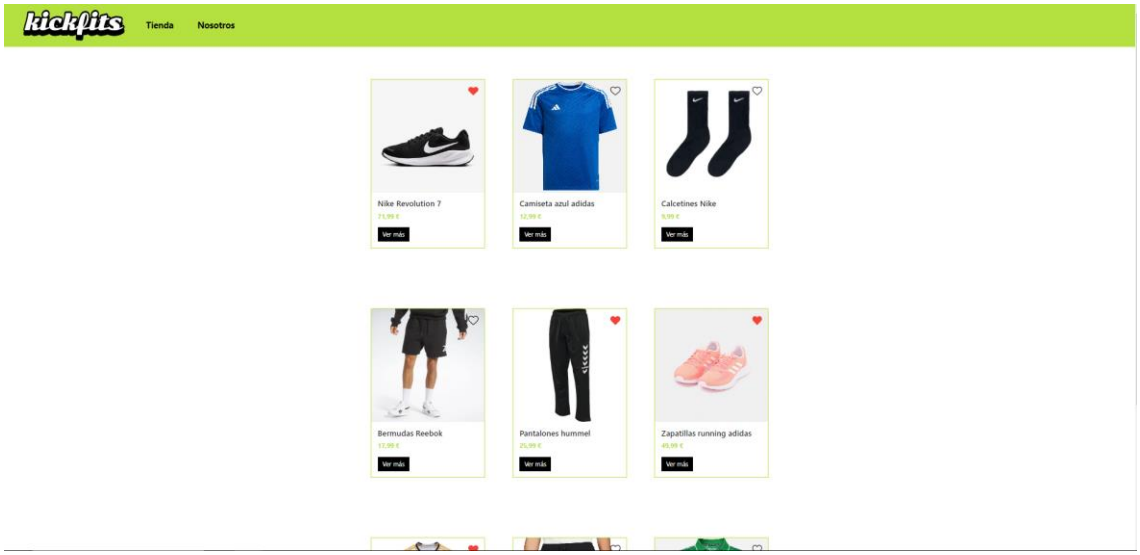
backend_kickfits.json	28/03/2024 11:09	Archivo de origen ...	1 KB
carousel.txt	28/03/2024 11:17	Documento de te...	1 KB
categorias.txt	28/03/2024 11:17	Documento de te...	1 KB
items.txt	28/03/2024 11:16	Documento de te...	0 KB
tendencias.txt	28/03/2024 11:17	Documento de te...	1 KB

[Galería de imágenes]



kickfits




kickfits
Tienda
Nosotros
Nike Revolution 7
71,99 €
Ver más
Camiseta azul adidas
32,99 €
Ver más
Calcetines Nike
9,99 €
Ver más
Bermudas Reebok
32,99 €
Ver más
Pantalones hummel
25,99 €
Ver más
Zapatillas running adidas
45,99 €
Ver más


kickfits
Tienda
Nosotros
Emirates
FLY BETTER
Camiseta Real Madrid
89,99 €
Descripción:
Primera equipación Real Madrid
Color:
Escoge una talla:
S
M
L
XL
XXL
Avísame
Añadir al carrito


kickfits
Tienda
Nosotros
Nuestros Valores
Calidad: Nos comprometemos a ofrecer productos de la más alta calidad para garantizar el rendimiento y la durabilidad.
Servicio al Cliente: Nuestro objetivo es superar las expectativas de nuestros clientes con un servicio personalizado y amable.
Comunidad: Creemos en construir una comunidad activa y positiva en torno al deporte y el estilo de vida saludable.
Conéctate con nosotros
Síguenos en nuestras redes sociales para mantenerte al día con las últimas novedades, ofertas especiales y eventos: