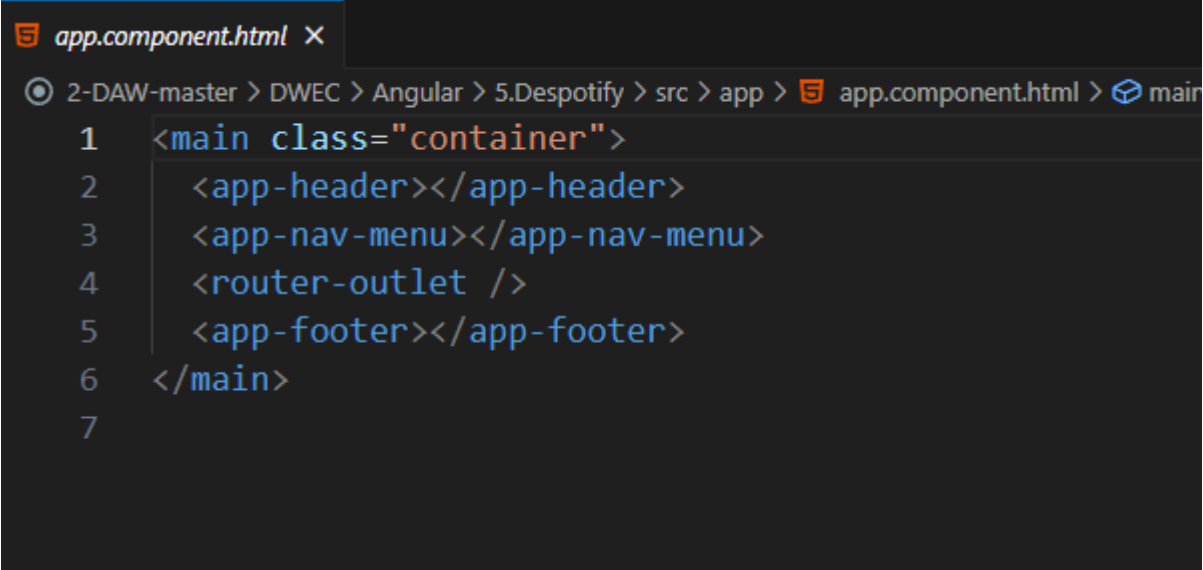


01. Crear componentes y servicios (ng g c nombre --skip-tests -s || ng g s nombre --skip-tests) Crear el page404
02. Insertar los templates
03. Index.html (  
    <link rel="stylesheet"  
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">  
    <script  
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></s  
    cript>)
04. app.component.html

a.



```
1 <main class="container">
2   <app-header></app-header>
3   <app-nav-menu></app-nav-menu>
4   <router-outlet />
5   <app-footer></app-footer>
6 </main>
7
```

05. app.component.ts



```
1 import { Component } from '@angular/core';
2 import { RouterOutlet } from '@angular/router';
3 import { HeaderComponent } from './header/header.component';
4 import { NavMenuComponent } from './nav-menu/nav-menu.component';
5 import { FooterComponent } from './footer/footer.component';
6
7 @Component({
8   selector: 'app-root',
9   imports: [RouterOutlet, HeaderComponent, NavMenuComponent, FooterComponent],
10  templateUrl: './app.component.html',
11  styleUrls: ['./app.component.css']
12 })
13 export class AppComponent {
14   title = '5.Despotify';
15 }
16
```

## app.routes.ts

```
app.routes.ts X
2-DAW-master > DWEC > Angular > 5.Despotify > src > app > app.routes.ts > ...
1 import { Routes } from '@angular/router';
2 import { PlaylistsComponent } from '../sections/playlists/playlists.component';
3 import { SongsComponent } from '../sections/songs/songs.component';
4 import { HomeComponent } from '../sections/home/home.component';
5 import { Page404Component } from '../page404/page404.component';
6
7 export const routes: Routes = [
8
9     { path: '', redirectTo: 'Home', pathMatch: 'full'},
10    { path: 'home', component: HomeComponent },
11    { path: 'playlists', component: PlaylistsComponent },
12    { path: 'songs', component: SongsComponent },
13    { path: '**', component: Page404Component}
14
15 ];
16
```

## nav-bar.component.html(defines el routerlink y el routerlinkactive)

```
nav-menu.component.html X
2-DAW-master > DWEC > Angular > 5.Despotify > src > app > nav-menu > nav-menu.component.html > div.row
Go to component
1 <div class="row">
2   <nav class="navbar navbar-expand-lg navbar-light bg-light">
3     <div class="container-fluid">
4       <div class="collapse navbar-collapse" id="navbarNav">
5         <ul class="navbar-nav">
6           <li class="nav-item">
7             <a class="nav-link" routerLinkActive="" aria-current="page" routerLink="/home">Home</a>
8           </li>
9           <li class="nav-item">
10            <a class="nav-link" routerLinkActive="" routerLink="/songs">Songs</a>
11          </li>
12          <li class="nav-item">
13            <a class="nav-link" routerLinkActive="" routerLink="/playlists">Playlists</a>
14          </li>
15        </ul>
16      </div>
17    </div>
18  </nav>
19 </div>
20
```

```
nav-menu.component.ts X
2-DAW-master > DWEC > Angular > 5.Despotify > src > app > nav-menu > nav-menu.component.ts > ...
1 import { Component } from '@angular/core';
2 import { RouterLink, RouterLinkActive } from '@angular/router';
3
4 @Component({
5   selector: 'app-nav-menu',
6   imports: [RouterLink, RouterLinkActive],
7   templateUrl: './nav-menu.component.html',
8   styles: ``
9 })
10 export class NavMenuComponent {
11
12 }
13
```

En el servicio  
(Llamamos a la apy)

```
favoritos.service.ts base-datos-cine.service.ts X search.component.ts search.component.html card-movie.component.html card-movie.component.ts
cineprueba > src > app > services > base-datos-cine.service.ts > BaseDatosCineService > urlBase
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class BaseDatosCineService {
8
9   constructor(private http: HttpClient) {
10
11   }
12   private urlBase= "https://www.omdbapi.com/?apikey=bbc3c930&";
13
14   getMoviesByTitle(title: string) {
15     return this.http.get(this.urlBase + "s=" + title );
16   }
17
18   getMovieById(id: string) {
19     return this.http.get(this.urlBase + "i=" + id);
20   }
21
22 }
23
24
25
```

```
base-datos-cine.service.ts X app.config.ts X
cineprueba > src > app > app.config.ts > ...
1 import { ApplicationConfig, provideZoneChangeDetection } from '@angular/core';
2 import { provideRouter } from '@angular/router';
3
4 import { routes } from './app.routes';
5 import { provideHttpClient } from '@angular/common/http';
6
7 export const appConfig: ApplicationConfig = {
8   providers: [provideZoneChangeDetection({ eventCoalescing: true }), provideRouter(routes), provideHttpClient()]
9 };
10
```

En el otro servicio

(creas un array para guardar los datos del servicio y creas un metodo para realizar la búsqueda)

```
favoritos.service.ts x search.component.ts search.component.html 1 base-datos-cine.service.ts
cineprueba > src > app > services > favoritos.service.ts > FavoritosService
1 import { Injectable } from '@angular/core';
2 import { BaseDatosCineService } from '../base-datos-cine.service';
3
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class FavoritosService {
9
10   constructor(private dbservice: BaseDatosCineService) { }
11
12   //creo un array para guardar los datos del servicio
13   private datosjson: any[] = []
14
15   getdatosjson() {
16     return this.datosjson
17   }
18
19   //metodo para realizar la busqueda por nombre de las peliculas llamando al servicio
20   searchbyname(name: string) {
21     this.dbservice.getMoviesByTitle(name).subscribe(json => {
22       let data : any = json
23       this.datosjson = data.Search
24     })
25   }
26
27
28
29 }
30
```

Ahora para utilizar ese método, nos metemos en el componente de search y le pasamos el método de buscar al input y lo pinta a través de un bucle for

```
favoritos.service.ts x search.component.ts search.component.html x card-movie.component.ts
cineprueba > src > app > sections > search > search.component.html > nav.navbar.bg-body-tertiary > div.container-fluid > form.d-flex >
Go to component
1 <!--el #search es para nombrar al input-->
2 <nav class="navbar bg-body-tertiary">
3   <div class="container-fluid">
4     <a class="navbar-brand">Navbar</a>
5     <form class="d-flex" role="search">
6       <input #search (keyup.enter)="searchbyname(search.value)" class="form-control" type="text">
7       <button class="btn btn-outline-success" type="submit">Search</button>
8     </form>
9   </div>
10 </nav>
11 @for (movies of getdatosjson(); track $index) {
12   <app-card-movie [movie] = "movies"></app-card-movie>
13 }
```

.En el componente de search, en el constructor llamamos al servicio en el que creamos el metodo.

-Despues le decimos el valor que queremos que nos devuelva la funcion(searchbyname).

-Y despues recuperamos el array que tiene guardado todos los datos del servicio

-Por ultimo como lo estamos como lo pintaremos en el card-movie, le importamos su componente

```
favorites.service.ts | search.component.ts x | search.component.html | card-movie.component.ts
cineprueba > src > app > sections > search > search.component.ts > SearchComponent > searchbyName
1  import { Component } from '@angular/core';
2  import { FavoritosService } from '../../services/favoritos.service';
3  import { CardMovieComponent } from '../card-movie/card-movie.component';
4
5  @Component({
6    selector: 'app-search',
7    imports: [CardMovieComponent],
8    templateUrl: './search.component.html',
9    styles: ``
10 })
11 export class SearchComponent {
12   constructor(private favservice : FavoritosService) { }
13
14   searchbyName(name: string) {
15     this.favservice.searchbyName(name)
16   }
17
18   getdatosjson() {
19     return this.favservice.getdatosjson()
20   }
21
22
23 }
```

Y por último la variable en la que guardamos el for del search.component.html, la creamos en el ts del card-movie.ts

```
favorites.service.ts x | search.component.ts | search.component.html | card-movie.component.ts x
cineprueba > src > app > sections > card-movie > card-movie.component.ts > ...
1  import { Component, Input } from '@angular/core';
2
3  @Component({
4    selector: 'app-card-movie',
5    imports: [],
6    templateUrl: './card-movie.component.html',
7    styles: ``
8  })
9  export class CardMovieComponent {
10    @Input() movie: any
11  }
12
```

Y ahora pintamos en la card-movie.component.html

```
favoritos.service.ts x base-datos-cine.service.ts search.component.ts search.component.html card-movie.component.html x card-r
cineprueba > src > app > sections > card-movie > card-movie.component.html > ...
Go to component
1 <div class="card mt-4 mx-auto" style="max-width: 400px;" >
2   <img [src]="movie.Poster" class="card-img-top img-fluid" alt="{{ movie.Title }}"
3     style="height: 500px; object-fit: cover; ">
4   <div class="card-body text-center">
5     <h5 class="card-title">{{ movie.Title }}</h5>
6     <p class="card-text">{{ movie.Year }}</p>
7     <!-- Botón para agregar a favoritos -->
8     <!-- <button class="btn btn-warning mt-2" (click)="addToFavorites(movie)">
9       ★ Añadir a favoritos
10    </button> -->
11   </div>
12 </div>
13
```

creamos el localStorage, un array donde se guardaran las peliculas favoritas, una funcion para comprobar si esta en favorito, otra para añadir y otra para borrar de favorito

```
favoritos.service.ts x card-movie.component.html card-movie.component.ts favorite.component.html favorite.component.ts
cineprueba > src > app > services > favoritos.service.ts > FavoritosService
8 export class FavoritosService {
22   searchbyName(name: string) {
28   })
29 }
30
31 //localStorage
32 saveAll(){
33   localStorage.setItem('favoritos', JSON.stringify(this.favoritos))
34 }
35
36 //creo un array favoritos para guardar las peliculas favoritas
37 private favoritos: any[] = []
38
39 getFavoritos() {
40   return this.favoritos
41 }
42
43 //funcion para comprobar si ya es favorito
44 isAlreadyFavorito(movie: any) {
45   return this.favoritos.find(fav => fav.imdbID === movie.imdbID)
46 }
47
48 //funcion para añadir a favoritos
49 addfavoritos(movie: any) {
50   if(!this.isAlreadyFavorito(movie)){
51     this.favoritos.push(movie)
52     this.saveAll()
53   }
54 }
55
56 //funcion para borrar de favoritos
57 borrarFavoritos(movie: any) {
58   let index = this.favoritos.indexOf(movie)
59   this.favoritos.splice(index, 1)
60   this.saveAll()
61 }
62
```

Despues muestro el titulo y año de las peliculas e implemento un if para que compruebe si la peli ya esta en favoritos o no.

```
favoritos.service.ts | card-movie.component.html x | card-movie.component.ts | favorite.component.html | favorite.component.ts
Go to component
1 <div class="card mt-4 mx-auto" style="max-width: 400px;" >
2   <img [src]="movie.Poster" class="card-img-top img-fluid" alt="{{ movie.Title }}"
3     style="height: 500px; object-fit: cover; ">
4   <div class="card-body text-center">
5     <h5 class="card-title">{{ movie.Title }}</h5>
6     <p class="card-text">{{ movie.Year }}</p>
7     @if (!isAlreadyFavorito(movie)) {
8       <button class="btn btn-warning mt-2" (click)="addfavoritos(movie)">
9         ★ Añadir a favoritos
10      </button>
11    }
12    @else {
13      <button class="btn btn-danger mt-2">
14        Ya esta en favoritos
15      </button>
16    }
17  </div>
18 </div>
19 </div>
20
```

Y en el card-movie.ts conecto con el servicio favoritos.service, añado las funciones de añadir y comprobar (el input movie es para que lo compruebe en el if )

```
favoritos.service.ts | card-movie.component.html | card-movie.component.ts x | favorite.component.html
Go to component
1 import { Component, Input } from '@angular/core';
2 import { FavoritosService } from '../services/favoritos.service';
3
4 @Component({
5   selector: 'app-card-movie',
6   imports: [],
7   templateUrl: './card-movie.component.html',
8   styles: ``
9 })
10 export class CardMovieComponent {
11   constructor(private favorito:FavoritosService) { }
12   @Input() movie: any
13
14   addfavoritos(movie : any){
15     this.favorito.addfavoritos(movie)
16   }
17   isAlreadyFavorito(movie: any){
18     return this.favorito.isAlreadyFavorito(movie)
19   }
20
21 }
22
```

Y en `favortie.component.html` le haces un bucle del array(`getfavoritos`) de las películas favoritas y en su `ts` recuperas el array y la función de borrar

```
favorites.service.ts  card-movie.component.html  card-movie.component.ts  favorite.component.html  favorite.component.ts x
cineprueba > src > app > sections > favorite > favorite.component.ts > FavoriteComponent > getFavoritos
1  import { Component } from '@angular/core';
2  import { FavoritosService } from '../../services/favorites.service';
3
4  @Component({
5    selector: 'app-favorite',
6    imports: [],
7    templateUrl: './favorite.component.html',
8    styles: ``
9  })
10 export class FavoriteComponent {
11
12   constructor(private favoritoss:FavoritosService) { }
13
14   getFavoritos(){
15     return this.favoritoss.getFavoritos()
16   }
17
18   borrarFavoritos(movie: any){
19     this.favoritoss.borrarFavoritos(movie)
20   }
21 }
22
```