



**IONIC**

Tema 2 – My First App

## ■ Creación y configuración inicial

- Creación de un proyecto ionic en blanco con el framework Angular
  - ionic start myfirstapp.
- Agregar el servicio de acceso a los datos:
  - ionic g service services/peliculas
- Agregar la página de detalles:
  - ionic g page pages/detalles
- Agregar el proveedor (**provider**) de cliente HTTP en **src/main.ts**:
  - Agregar import:
    - `import { provideHttpClient } from '@angular/common/http';`
  - **Agregar al array de providers:**
    - **`provideHttpClient()`**

## ■ Modificando la configuración del proyecto

- Uso de ESBuild (a partir de la versión 17 de Angular), un empaquetador y minificador de JavaScript para agilizar la ejecución de la aplicación, modificando el fichero angular.json.

```
"build": {  
  "builder": "@angular-devkit/build-angular:browser",  
  "options": {  
    "outputPath": "www",  
    "index": "src/index.html",  
    "main": "src/main.ts",  
    "polyfills": "src/polyfills.ts",  
    "tsConfig": "tsconfig.app.json",  
    "inlineStyleLanguage": "scss",
```



```
"build": {  
  "builder": "@angular-devkit/build-angular:application",  
  "options": {  
    "outputPath": "www",  
    "index": "src/index.html",  
    "browser": "src/main.ts",  
    "polyfills": ["src/polyfills.ts"],  
    "tsConfig": "tsconfig.app.json",  
    "inlineStyleLanguage": "scss",
```

- Eliminar, además, los parámetros “**buildOptimizer**” y “**vendorChunk**”

# IONIC



- Configuración de constantes de desarrollo y producción:

- Ficheros:

- src/environment/environment.prod.ts → producción

- src/environment/environment.ts → desarrollo

```
export const environment = {  
  production: false,  
  apiKey: 'miapikey'  
};
```

# IONIC



- Creación de los interfaces para el intercambio de datos.
  - `ionic g interface ./interfaces/movie`

- Creación el sistema de 'routing': fichero **app.routes.ts**.

```
import { Routes } from '@angular/router';

export const routes: Routes = [
  {
    path: 'home',
    loadComponent: () => import('./home/home.page').then((m) => m.HomePage),
  },
  {
    path: '',
    redirectTo: 'home',
    pathMatch: 'full',
  },
  {
    path: 'detalles/:titulo',
    loadComponent: () => import('./pages/detalles/detalles.page').then(m =>
m.DetallesPage)
  },
];
```

## ■ Creación del servicio (1):

```
import { HttpClient } from '@angular/common/http';  
import { inject, Injectable } from '@angular/core';  
import { environment } from 'src/environments/environment';  
import { Movie } from '../interfaces/movie';  
import { delay, Observable } from 'rxjs';
```

```
const BASE_URL = 'https://www.omdbapi.com';  
const API_KEY = environment.apiKey;
```

## ■ Creación del servicio (2):

```
@Injectable({
  providedIn: 'root'
})
export class PeliculasService {
  private http = inject(HttpClient);
  public movies: Movie[] = [];
  constructor() { }
  getMovie(title: string): Observable<Movie> {
    return this.http
      .get<Movie>(`${BASE_URL}?apikey=${API_KEY}&t=${title}`)
      .pipe(
        delay(1000) //Simulación de retardo
      );
  }
  addMovie(movie: Movie) {
    this.movies.push(movie);
  }
}
```



# IONIC



- Definición de home page: [home.page.html](#)

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title>
      Superpelis
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">Superpelis</ion-title>
    </ion-toolbar>
  </ion-header>

  <ion-item class="ion-padding-end">
    <ion-input placeholder="Título" [(ngModel)]="titulo" name="titulo"></ion-input>
  </ion-item>

  <ion-button expand="full" class="ion-padding" (click)="buscar()">Buscar</ion-button>
```

Superpelis

Título

Buscar

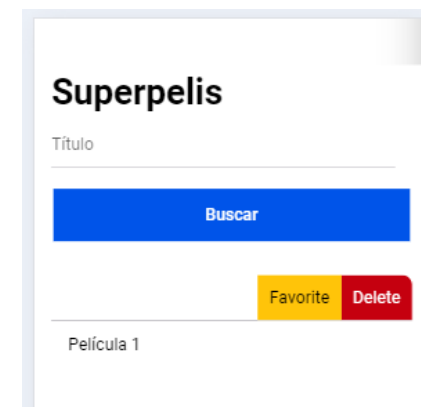
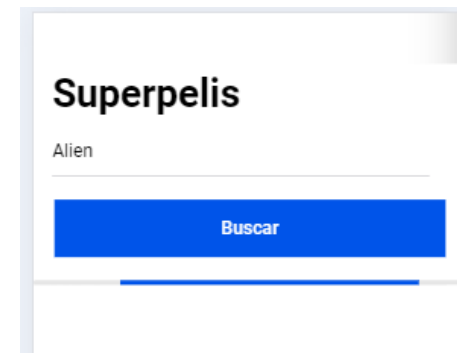
# IONIC



## ■ Definición de home page: [home.page.html](#)

```
<ion-progress-bar type="indeterminate" *ngIf="cargando"></ion-progress-bar>

<ion-list [inset]="true">
  <ion-item-sliding *ngFor="let movie of dataService.movies">
    <ion-item>
      <ion-label>{{movie.Title}}</ion-label>
    </ion-item>
    <ion-item-options side="start">
      <ion-item-option color="success" (click)="guardar(movie.Title)">Guardar</ion-
item-option>
    </ion-item-options>
    <ion-item-options>
      <ion-item-option color="danger">Eliminar</ion-item-option>
    </ion-item-options>
  </ion-item-sliding>
</ion-list>
</ion-content>
```



# IONIC



## ■ Definición de home page: [home.page.ts](#)

```
import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';
import {
  IonHeader, IonToolbar, IonTitle, IonContent,
  IonList, IonInput, IonItem, IonButton, IonLabel,
  IonItemOption, IonItemOptions, IonItemSliding, IonProgressBar
} from '@ionic/angular/standalone';
import { PeliculasService } from '../services/peliculas.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
  standalone: true,
  imports: [IonProgressBar, IonItemSliding, IonLabel,
    FormsModule, CommonModule, IonHeader, IonToolbar,
    IonTitle, IonContent, IonList, IonInput,
    IonItem, IonButton, IonItemOption, IonItemOptions
  ],
})
```

## ▪ Definición de home page: [home.page.ts](#)

```
export class HomePage {
  public titulo: string = "";
  public cargando: boolean = false;
  constructor(public dataService: PeliculasService, private router: Router) { }

  public buscar(): void {
    this.cargando = true;
    this.dataService.getMovie(this.titulo).subscribe(data => {
      this.cargando = false;
      if (data.Response === 'False') {
        console.error("Dato no encontrado");
      } else {
        this.dataService.addMovie(data);
      }
    })
  }

  public guardar(titulo: string): void {
    console.log("Guardando..." + titulo);
    this.router.navigate(['detalles/${titulo}']);
  }
}
```

## ■ Definición de la página de detalle: detalles.page.html

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title></ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-list [inset]="true">
    <ion-card>
      
      <ion-card-header>
        <ion-card-title>{{pelicula?.Title}}</ion-card-title>
        <ion-card-subtitle>{{pelicula?.Director}}</ion-card-subtitle>
      </ion-card-header>
      <ion-card-content>
        {{pelicula?.Plot}}
      </ion-card-content>
    </ion-card>
  </ion-list>
</ion-content>
```

## ■ Definición de la página de detalle: detalles.page.html

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title></ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-list [inset]="true">
    <ion-card>
      
      <ion-card-header>
        <ion-card-title>{{pelicula?.Title}}</ion-card-title>
        <ion-card-subtitle>{{pelicula?.Director}}</ion-card-subtitle>
      </ion-card-header>
      <ion-card-content>
        {{pelicula?.Plot}}
      </ion-card-content>
    </ion-card>
  </ion-list>
</ion-content>
```

## ■ Definición de la página de detalle: detalles.page.ts

```
@Component({
  selector: 'app-detalles',
  templateUrl: './detalles.page.html',
  styleUrls: ['./detalles.page.scss'],
  standalone: true,
  imports: [IonCardHeader, IonCardContent, IonCard, IonCardSubtitle, IonCardTitle, IonImg, IonLabel,
IonItem, IonList, IonContent, IonHeader, IonTitle, IonToolbar, CommonModule, FormsModule]
})
export class DetallesPage implements OnInit {
  titulo: string = "";
  pelicula: Movie | undefined;

  constructor(private ar: ActivatedRoute, private dataService: PeliculasService) { }

  ngOnInit() {
    this.ar.params.subscribe(params => {
      this.titulo = params['titulo'];
    });
    this.pelicula = this.dataService.movies.find(pelicula => pelicula.Title ===
this.titulo);
  }
}
```