



DESARROLLO DE SOFTWARE FRONTEND II

ESPINOSA MARTINEZ CARLOS LEONEL

Aarón Hernández García

Avance de proyecto final

19/02/2025

Introducción

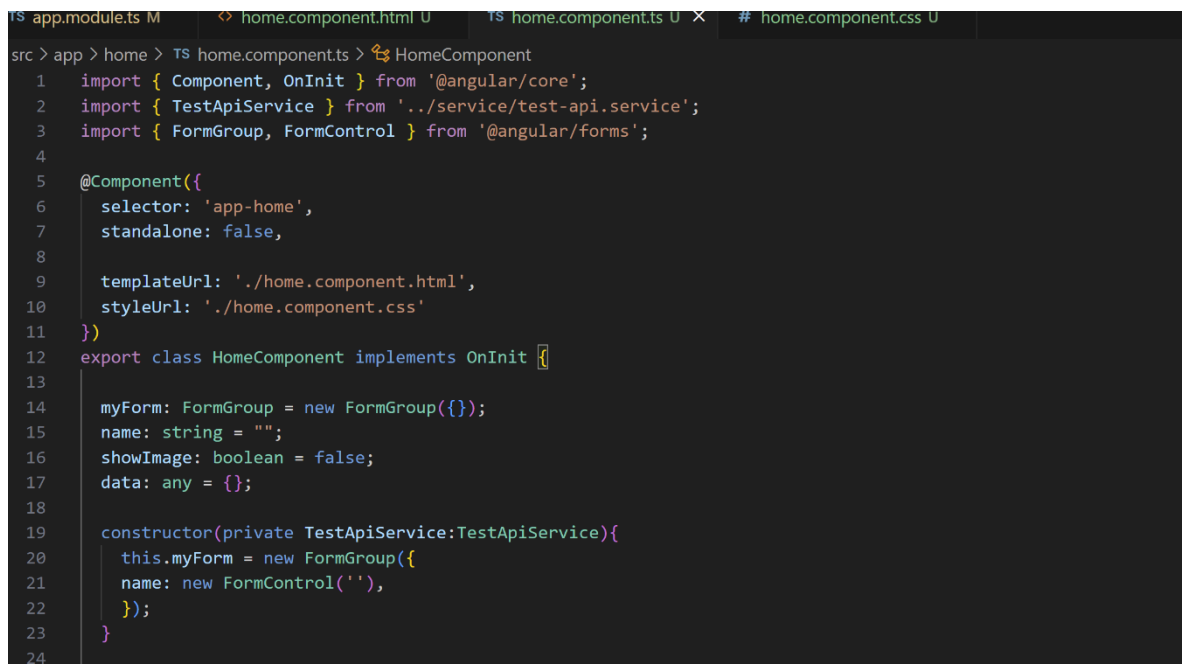
En este proyecto, desarrollaremos una aplicación web utilizando Angular para consumir una API de países. La aplicación permitirá visualizar información relevante como nombres y otras características de diferentes países del mundo.

Para lograrlo, utilizaremos el servicio HttpClient de Angular, que nos permitirá realizar peticiones HTTP a la API seleccionada y obtener los datos en formato JSON. Implementaremos una interfaz intuitiva y dinámica, facilitando la búsqueda y filtrado de información de los países de manera eficiente.

Este proyecto tiene como objetivo principal mejorar la comprensión del consumo de APIs en Angular, aplicando buenas prácticas en el desarrollo de servicios y en la estructura modular de la aplicación.

El proceso de desarrollo sigue los siguientes pasos:

Desarrollo de la aplicación: Creación de la interfaz y configuración del servicio para consumir la API.



```
src > app > home > TS home.component.ts > HomeComponent
1  import { Component, OnInit } from '@angular/core';
2  import { TestApiService } from '../service/test-api.service';
3  import { FormGroup, FormControl } from '@angular/forms';
4
5  @Component({
6    selector: 'app-home',
7    standalone: false,
8
9    templateUrl: './home.component.html',
10   styleUrls: ['./home.component.css']
11 })
12 export class HomeComponent implements OnInit {
13
14   myForm: FormGroup = new FormGroup({});
15   name: string = "";
16   showImage: boolean = false;
17   data: any = {};
18
19   constructor(private TestApiService: TestApiService) {
20     this.myForm = new FormGroup({
21       name: new FormControl(''),
22     });
23   }
24 }
```

Codigo

```
TS app.module.ts M X  <> home.component.html U  TS home.component.ts U  # home.component.css U
src > app > TS app.module.ts > AppModule
22 import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
23 import { MatIconModule } from '@angular/material/icon';
24 import { MatSelectModule } from '@angular/material/select';
25 import { MatInputModule } from '@angular/material/input';
26 import { MatFormFieldModule } from '@angular/material/form-field';
27 import { MatButtonModule } from '@angular/material/button';
28 import { MatCardModule } from '@angular/material/card';
29
30 import { CountrysComponent } from '../countrys/countrys.component';
31
32
33 @NgModule({
34   declarations: [
35     AppComponent,
36     PeliculasComponent,
37     DocumentalesComponent,
38     SeriesComponent,
39     MusicaComponent,
40     VideosComponent,
41     PecesComponent,
42     AnfibiosComponent,
43     ReptilesComponent,
44     AvesComponent,
45     MamiferosComponent,
46     HomeComponent,
47     CountrysComponent
48   ],
```

```

<form [formGroup]="myForm" (ngSubmit)="onSubmit()">
  <label for="name">Nombre del País:</label>
  <input id="name" formControlName="name" type="text">
  <button type="submit">Buscar</button>
</form>

<!-- Mostrar resultados si existe información -->
<div *ngIf="data.name">
  <h2>{{ data.name.common }} ({{ data.name.official }})</h2>
  <h3>Dominio de Nivel Superior: {{ data.tld?.join(', ') }}</h3>
  <h3>Moneda: {{ data.currencies | json }}</h3>
  <h3>Continente: {{ data.continents?.join(', ') }}</h3>
  <h3>Región: {{ data.region }}</h3>
  <h3>Subregión: {{ data.subregion }}</h3>
  <h3>Mapa: <a [href]="data.maps?.googleMaps" target="_blank">Ver en Google Maps</a></h3>

  <img *ngIf="showImage" [src]="data.flags?.png" [alt]="data.name.common + ' flag'" width="400">
</div>

```

```

ngOnInit(): void {
}

onSubmit(){
  const formValue = this.myForm.value;
  this.name = formValue.name;
}

```

TS app.module.ts M

<> home.component.html U

TS home.component.ts U

home.component.css U

TS d

src > app > countrys > TS countrys.component.spec.ts > ...

```

1  import { ComponentFixture, TestBed } from '@angular/core/testing';
2  import { CountrysComponent } from './countrys.component';
3
4  describe('CountrysComponent', () => {
5    let component: CountrysComponent;
6    let fixture: ComponentFixture<CountrysComponent>;
7
8    beforeEach(async () => {
9      await TestBed.configureTestingModule({
10       declarations: [CountrysComponent]
11     })
12     .compileComponents();
13
14     fixture = TestBed.createComponent(CountrysComponent);
15     component = fixture.componentInstance;
16     fixture.detectChanges();
17   });
18
19   it('should create', () => {
20     expect(component).toBeTruthy();
21   });
22 });
23

```

```
TS app.module.ts M  home.component.html U  TS home.component.ts U  # home.component.css U  TS countrys.component.ts U X
src > app > countrys > TS countrys.component.ts > ...
1  import { Component } from '@angular/core';
2  import { TestApiService } from '../service/test-api.service';
3
4  @Component({
5    selector: 'app-countrys',
6    standalone: false,
7
8    templateUrl: './countrys.component.html',
9    styleUrls: ['./countrys.component.css']
10 })
11 export class CountrysComponent {
12   pais: any={};
13
14   constructor(private api:TestApiService) { }
15
16   ngOnInit(): void {
17     this.pais();
18   }
19
20   searchPokemon():void{
21     this.api.consultapais('zapdos').subscribe({
22       next:(infoPais)=>{
23         this.pais = infoPais;
24         console.log(this.pais);
25       }
26     })
27   }
28
29 }
30
31
```

Pruebas locales: Verificación de la ejecución de la aplicación en el entorno de desarrollo.

Nombre del País:

Buscar

Mexico (United Mexican States)

Dominio de Nivel Superior: .mx

Moneda: { "MXN": { "name": "Mexican peso", "symbol": "\$" } }

Continente: North America

Región: Americas

Subregión: North America

Mapa:[Ver en Google Maps](#)



Nombre del País:

Buscar

United States (United States of America)

Dominio de Nivel Superior: .us

Moneda: { "USD": { "name": "United States dollar", "symbol": "\$" } }

Continente: North America

Región: Americas

Subregión: North America

Mapa:[Ver en Google Maps](#)



Conclusión

El uso de Angular para consumir una API de países permite desarrollar aplicaciones web dinámicas y escalables con una arquitectura modular. Gracias a HttpClient, podemos gestionar peticiones HTTP de manera eficiente, optimizando la comunicación con el backend y garantizando una carga fluida de datos.

Además, la integración de una interfaz interactiva facilita la experiencia del usuario, permitiéndole explorar y filtrar información de manera intuitiva. Este enfoque no solo mejora la presentación de datos, sino que también refuerza el aprendizaje sobre el consumo de APIs, el manejo de estados en Angular y la estructuración de proyectos web modernos.