



DESARROLLO DE SOFTWARE FRONTEND II
ESPINOSA MARTINEZ CARLOS LEONEL
Aarón Hernández García

Entrega de proyecto final

19/02/2025

Introducción

En este proyecto, desarrollaremos una aplicación web (App de Países) utilizando Angular para consumir una API seleccionada. La implementación incluye la creación de una interfaz que permita interactuar con los datos proporcionados por la API, utilizando un servicio de Angular para su consumo.

El proceso de desarrollo sigue los siguientes pasos:

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

Código

```
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;
  this.consultPais(this.name);
  this.myForm.reset();
}

consultPais(pais:string){
  this.TestApiService.consultapais(pais).subscribe({
    next: this.handleSuccessMethod.bind(this),
    error: this.handleErrorMethod.bind(this),
  })
}

handleSuccessMethod(data: any) {
  if (data && data.length > 0) {
    this.showImage = true;
    this.data = data[0]; // 🚀 Tomamos el primer pais devuelto por la API
  } else {
    this.handleErrorMethod();
  }
}

handleErrorMethod() {
  this.showImage = false;
  this.data = { name: { common: "País no encontrado" } };
}
}

```

```
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
```

```
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](#)



Compilación: Uso del comando ng build para generar los archivos de producción dentro de la carpeta dist.

```
C:\Users\windows\practical>
* History restored
C:\Users\windows\practical>ng build
Browser bundles
Initial chunk files | Names | Raw size | Estimated transfer size
chunk-SNZDF23C.js | - | 232.70 kB | 65.69 kB
main-GNOG5LCZ.js | main | 190.45 kB | 45.25 kB
polyfills-FFHMD2TL.js | polyfills | 34.52 kB | 11.28 kB
styles-JBK7QVY4.css | styles | 6.98 kB | 1.14 kB









| Initial total | 464.64 kB | 123.36 kB

Lazy chunk files | Names | Raw size | Estimated transfer size
chunk-U64SEDCR.js | browser | 864 bytes | 864 bytes

Server bundles
Initial chunk files | Names | Raw size
server.mjs | server | 845.45 kB
main.server.mjs | main.server | 638.58 kB
polyfills.server.mjs | polyfills.server | 268.60 kB
chunk-ALA5AGRC.mjs | - | 226.15 kB
chunk-X2SEQXRR.mjs | - | 2.41 kB
chunk-GFUZYSG.mjs | - | 1.24 kB

Lazy chunk files | Names | Raw size
chunk-EJSJJTUO.mjs | xhr2 | 12.14 kB
chunk-TN04MGIL.mjs | xhr2 | 12.07 kB
chunk-GHZTCRXO.mjs | browser | 927 bytes

Prerendered 1 static route.
Application bundle generation complete. [33.492 seconds]
```

<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño
 chunk-SNZDF23C.js	15/02/2025 06:00 p. m.	Archivo JavaScript	228 KB
<input checked="" type="checkbox"/>  chunk-U64SEDCR.js	15/02/2025 06:00 p. m.	Archivo JavaScript	1 KB
 favicon.ico	16/01/2025 09:38 p. m.	Icono	15 KB
 index.csr.html	15/02/2025 06:00 p. m.	Chrome HTML Do...	24 KB
 index.html	15/02/2025 06:00 p. m.	Chrome HTML Do...	26 KB
 main-GNOG5LCZ.js	15/02/2025 06:00 p. m.	Archivo JavaScript	186 KB
 polyfills-FFHMD2TL.js	15/02/2025 06:00 p. m.	Archivo JavaScript	34 KB
 styles-JBK7QVY4.css	15/02/2025 06:00 p. m.	Archivo de origen ...	7 KB

Validación de la compilación: Implementación de un servidor estático con node-static (npm install -g node-static) y pruebas de funcionamiento.

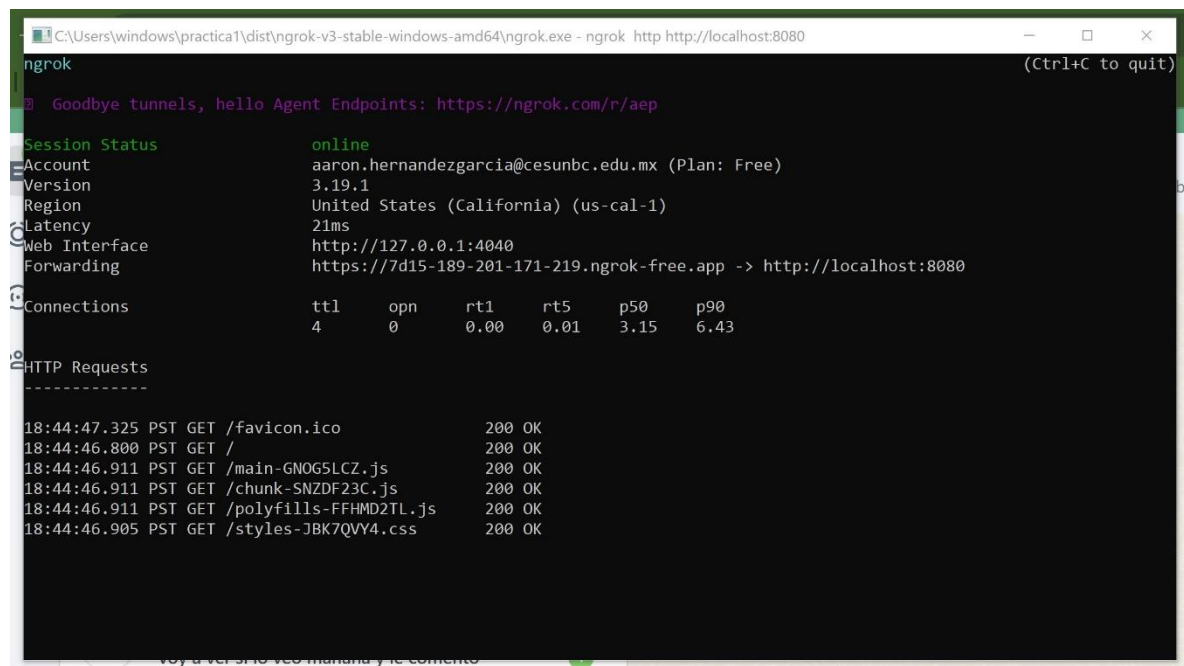
```
C:\Users\windows\practica1>npm install -g node-static

added 6 packages in 3s
npm notice
npm notice New major version of npm available! 10.9.2 -> 11.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.1.0
npm notice To update run: npm install -g npm@11.1.0
npm notice

C:\Users\windows\practica1>Static -v
node-static 0.7.9

C:\Users\windows\practica1>
```

Exposición de la aplicación: Uso de herramientas como ngrok para acceder a la aplicación desde internet.



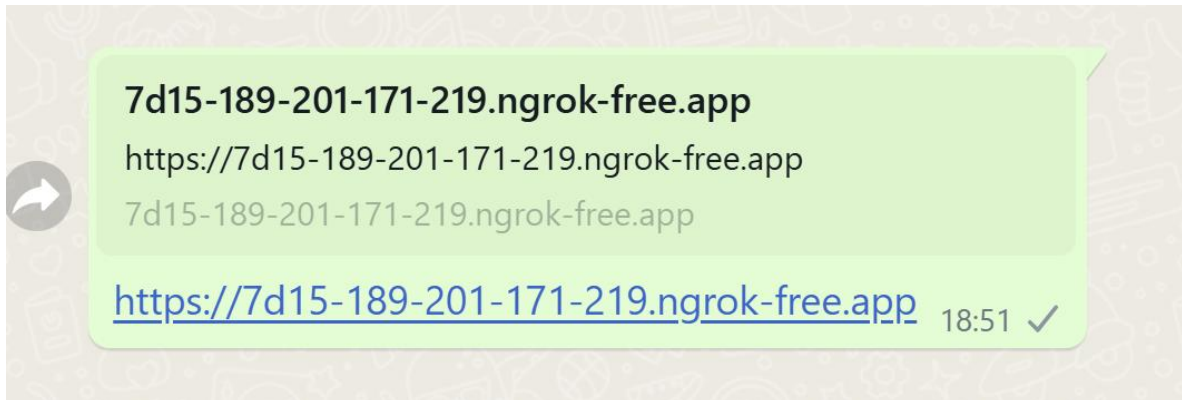
```
C:\Users\windows\practica1\dist\ngrok-v3-stable-windows-amd64\ngrok.exe - ngrok http http://localhost:8080
ngrok (Ctrl+C to quit)
Goodbye tunnels, hello Agent Endpoints: https://ngrok.com/r/aep

Session Status      online
Account             aaron.hernandezgarcia@cesunbc.edu.mx (Plan: Free)
Version             3.19.1
Region              United States (California) (us-cal-1)
Latency             21ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://7d15-189-201-171-219.ngrok-free.app -> http://localhost:8080

Connections
  ttl    opn    rt1    rt5    p50    p90
    4      0    0.00   0.01   3.15   6.43

HTTP Requests
-----
18:44:47.325 PST GET /favicon.ico           200 OK
18:44:46.800 PST GET /                     200 OK
18:44:46.911 PST GET /main-GNOG5LCZ.js      200 OK
18:44:46.911 PST GET /chunk-SNZDF23C.js     200 OK
18:44:46.911 PST GET /polyfills-FFHMD2TL.js  200 OK
18:44:46.905 PST GET /styles-JBK7QVY4.css   200 OK
```


Comprobación de la aplicación: Uso del link proporcionado por Ngrok y pasarlo a un compañero para su verificación



Conclusión

El uso de **Angular** para el desarrollo de aplicaciones web proporciona una estructura robusta y modular, permitiendo una mejor organización del código y facilitando el mantenimiento a largo plazo. Su capacidad para gestionar el consumo de APIs a través de servicios optimiza la comunicación con servidores y bases de datos.

Además, el proceso de **compilación y despliegue** con ng build permite generar archivos optimizados para producción, lo que mejora el rendimiento y la velocidad de carga de la aplicación. El uso de **node-static** facilita la ejecución local de la aplicación compilada, permitiendo pruebas antes de su implementación final.

Por otro lado, herramientas como **Ngrok** simplifican la exposición de la aplicación en internet sin necesidad de configuraciones complejas de servidores o redes, haciendo que el acceso y las pruebas remotas sean más eficientes.