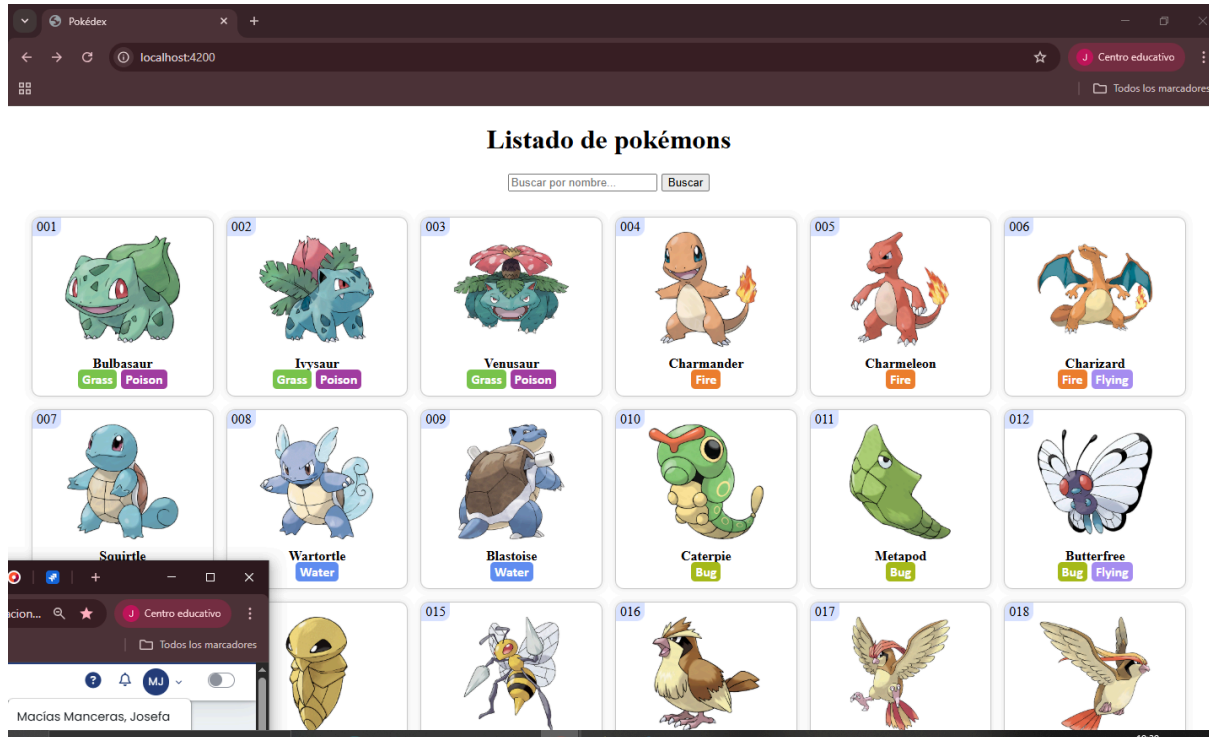


Resumen:

- Muestra los 151 pokémon con su imagen.
- Tiene un buscador por nombre.
- Al pinchar en el pokemon, se abre su ficha con más datos.
- Tiene un botón para darle a anterior/siguiente desde la ficha.

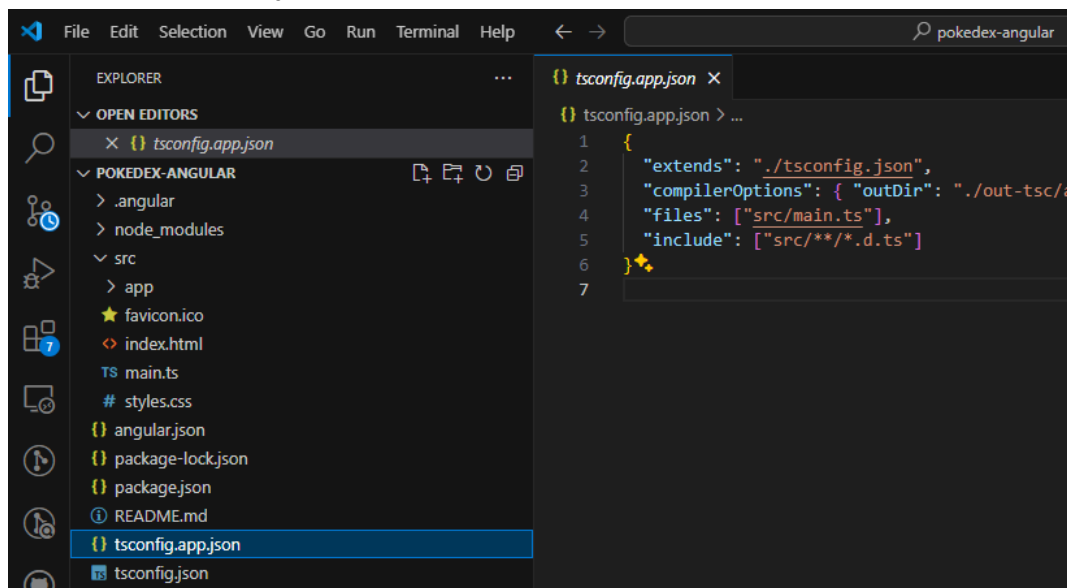


1. Creación del proyecto:

ng new pokedex --skip-tests:

Selecione las opciones: Hojas de estilo CSS - No habilitar SSG/Prerendering.

Compruebo que se ejecute en Visual Studio Code.



2. Vista Listado de pokémons: Con el componente: ng g c pages/pokemon-list. Y se configura el enrutador y la salida:

src/app/app.routes.ts:

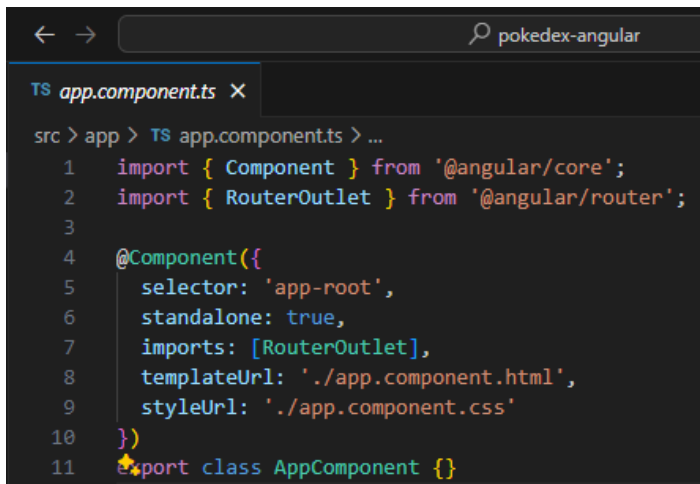
```
export const routes: Routes = [
  { path: '', component: PokemonListComponent }
];
```



```
TS app.routes.ts X
src > app > TS app.routes.ts > ...
1  import { Routes } from '@angular/router';
2  import { PokemonListComponent } from '../pages/pokemon-list/pokemon-list.component';
3  import { PokemonComponent } from '../pages/pokemon/pokemon.component';
4
5  export const routes: Routes = [
6    { path: '', component: PokemonListComponent },
7    { path: 'pokemon/:id', component: PokemonComponent },
8    { path: '**', redirectTo: '' }
9  ]
10
```

src/app/app.component.html: <router-outlet></router-outlet>:

Esto fuerza a que se renderice el nuevo componente.



```
TS app.component.ts X
src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3
4  @Component({
5    selector: 'app-root',
6    standalone: true,
7    imports: [RouterOutlet],
8    templateUrl: './app.component.html',
9    styleUrls: ['./app.component.css']
10 })
11 export class AppComponent {}
```

3. Modelos y servicio:

- **Modelo Pokemon:** ng g interface models/pokemon:

```
export interface Pokemon {
  id: number;
  name: string;
  imageUrl: string;
  types: string[];
}
```

- **Servicio PokeapiService:** ng g s services/pokeapi

```

src > app > services > TS pokeapi.service.ts > ...
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable, forkJoin, lastValueFrom } from 'rxjs';
4  import { Pokemon } from '../models/pokemon';
5  import { PokemonDetail } from '../models/pokemon-detail';
6
7  @Injectable({ providedIn: 'root' })
8  export class PokeapiService {
9      readonly BASE_URL = 'https://pokeapi.co/api/v2/';
10
11      constructor(private http: HttpClient) {}
12
13      async getAllPokemon(): Promise<Pokemon[]> {
14          const requests: Observable<Object>[] = [];
15          for (let i = 1; i <= 151; i++) {
16              requests.push(this.http.get(`${this.BASE_URL}pokemon/${i}`));
17          }
18          const allDataRaw: any[] = await lastValueFrom(forkJoin(requests));
19          const pokemons: Pokemon[] = [];
20          for (const data of allDataRaw) {
21              const pokemon: Pokemon = {
22                  id: data.id,
23                  name: data.name,
24                  imageUrl: data.sprites.other['official-artwork'].front_default,
25                  types: (data.types as any[]).map((t: any) => t.type.name)
26              };
27              pokemons.push(pokemon);
28          }
29          return pokemons;
30      }
31
32      async getPokemonDetail(id: number): Promise<PokemonDetail> {
33          const request: Observable<Object> = this.http.get(`${this.BASE_URL}pokemon/${id}`);
34          const dataRaw: any = await lastValueFrom(request);
35          const pokemon: PokemonDetail = {
36              id: dataRaw.id,
37              name: dataRaw.name,
38              imageUrl: dataRaw.sprites.other['official-artwork'].front_default,
39              types: (dataRaw.types as any[]).map((t: any) => t.type.name),

```

4. Listado: lógica y vista:

4.1 Lógica del componente:

```

src > app > pages > pokemon-list > TS pokemon-list.component.ts > ...
1  import { Component, OnInit } from '@angular/core';
2  import { RouterModule } from '@angular/router';
3  import { FormsModule } from '@angular/forms';
4  import { TitleCasePipe } from '@angular/common';
5  import { DigitNumberPipe } from '../pipes/digit-number.pipe';
6  import { PokemonTypeComponent } from '../components/pokemon-type.component';
7  import { PokeapiService } from '../services/pokeapi.service';
8
9  type Pokemon = { id:number; name:string; imageUrl:string; types:string[] };
10
11  @Component({
12      selector: 'app-pokemon-list',
13      standalone: true,
14      imports: [RouterModule, FormsModule, TitleCasePipe, DigitNumberPipe, PokemonTypeComponent],
15      templateUrl: './pokemon-list.component.html',
16      styleUrls: ['./pokemon-list.component.css']
17  })
18  export class PokemonListComponent implements OnInit {
19      allPokemons: Pokemon[] = [];
20      filteredPokemons: Pokemon[] = [];
21      query = '';
22      error: string | null = null;
23
24      constructor(private pokeApi: PokeapiService) {}
25
26      async ngOnInit(): Promise<void> {
27          try {
28              this.allPokemons = await this.pokeApi.getAllPokemon();
29              this.search();
30          } catch (e) {
31              this.error = 'No se pudo cargar la lista (revisa la consola del navegador).';
32              this.filteredPokemons = [];
33          }
34      }
35
36      search() {
37          const q = this.query.trim().toLowerCase();
38          this.filteredPokemons = q
39              ? this.allPokemons.filter(p => p.name.includes(q))

```

4.2 HTML básico:

```

1 <h1 style="text-align:center">Listado de pokémons</h1>
2
3 <form class="filter" (ngSubmit)="search()">
4   <input [(ngModel)]="query" [ngModelOptions]="{standalone:true}" placeholder="Buscar por nombre..." />
5   <input type="submit" value="Buscar" />
6 </form>
7
8 @if (error) {
9   <div style="color: #b00; text-align:center">{{error}}</div>
10 }
11
12 <section class="pokemon-list">
13   @if (allPokemons?.length > 0) {
14     @for (pokemon of filteredPokemons; track pokemon.id) {
15       <a class="pokemon-card" [routerLink]="pokemon/' + pokemon.id">
16         <img [src]="pokemon.imageUrl" [alt]="pokemon.name | titlecase" loading="lazy" />
17         <div class="id">{{pokemon.id | digitNumber: 3}}</div>
18         <div class="name">{{pokemon.name | titlecase}}</div>
19         <div class="types">
20           @for (type of pokemon.types; track $index) {
21             <app-pokemon-type [type]="type"></app-pokemon-type>
22           }
23         </div>
24       </a>
25     }
26   } @empty {
27     <div>No hay resultados</div>
28   } @else {
29     <div style="text-align:center">Cargando...</div>
30   }
31 </section>

```

5. Mejoras de presentación:

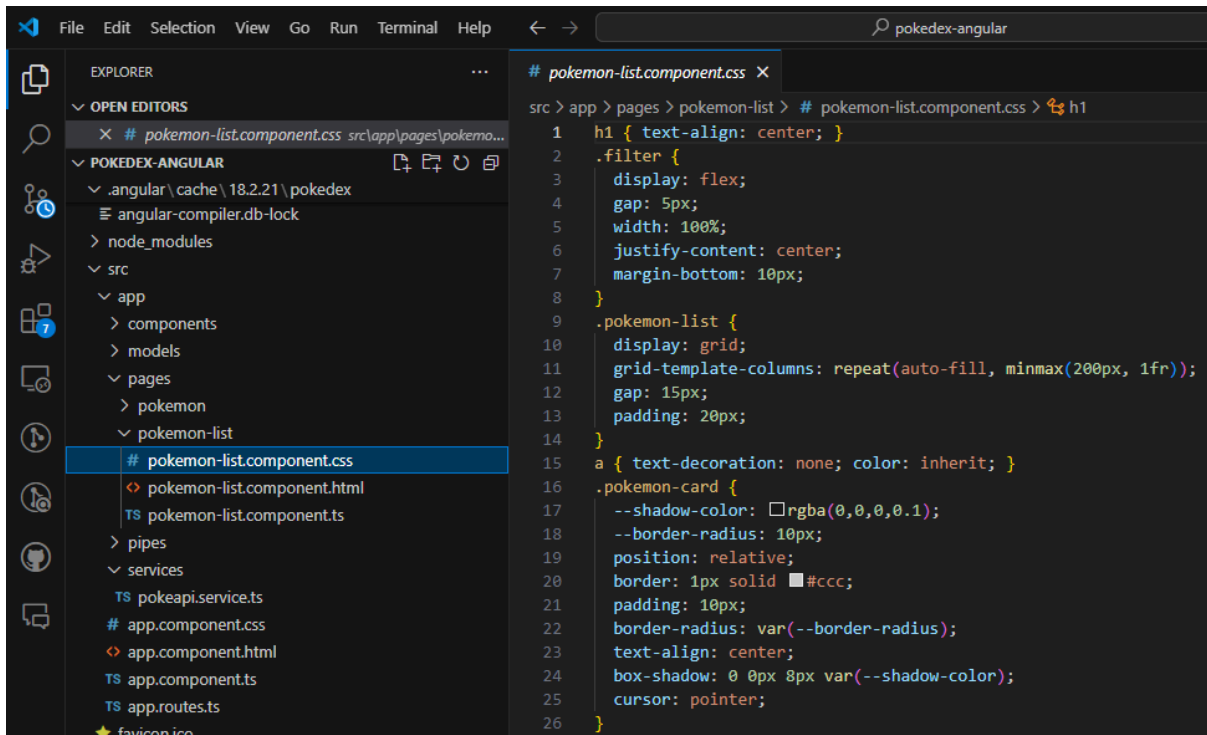
5.1 Título de la pestaña:

```

1 <!doctype html>
2 <html lang="es">
3 <head>
4   <meta charset="utf-8">
5   <title>Pokédex</title>
6   <base href="/">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="icon" type="image/x-icon" href="favicon.ico">
9 </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>

```

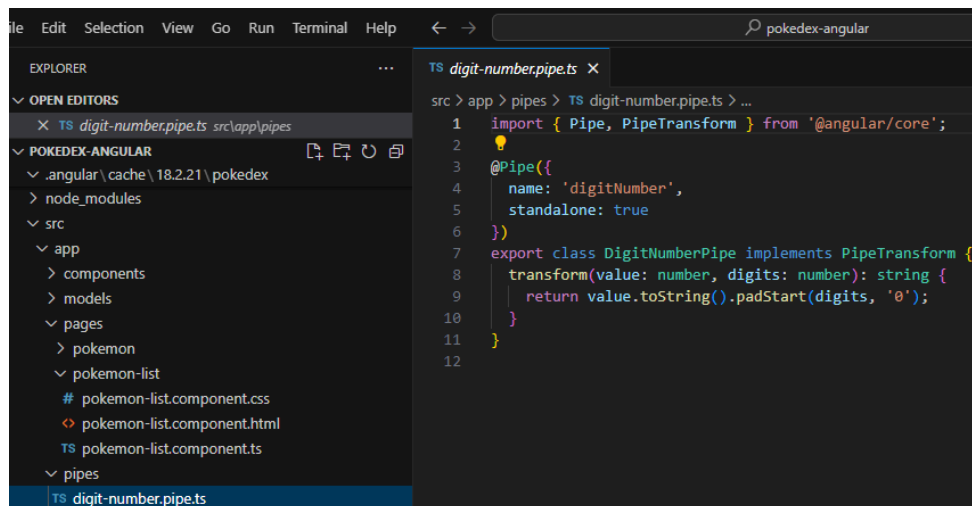
5.2 Estilos del grid:



5.3 Nombre con mayúscula inicial: En el HTML:

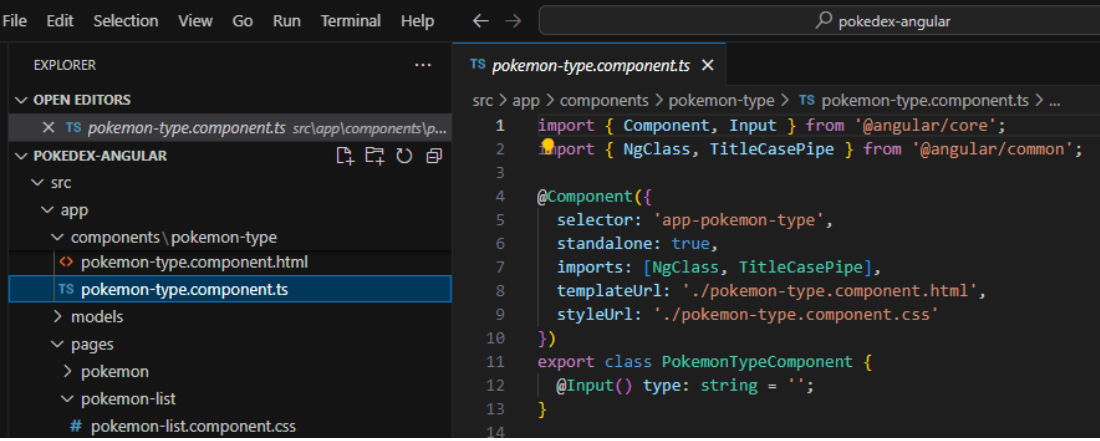
<div class="name">{{ pokemon.name | titlecase }}</div>

5.4 Pipe para id con 3 dígitos: ng g pipe pipes/digit-number.



Uso en HTML: <div class="id">{{ pokemon.id | digitNumber:3 }}</div>

6. Componente de tipo (colores): ng g c components/pokemon-type:



```

src > app > components > pokemon-type > TS pokemon-type.component.ts > ...
1  import { Component, Input } from '@angular/core';
2  import { NgClass, TitleCasePipe } from '@angular/common';
3
4  @Component({
5    selector: 'app-pokemon-type',
6    standalone: true,
7    imports: [NgClass, TitleCasePipe],
8    templateUrl: './pokemon-type.component.html',
9    styleUrls: ['./pokemon-type.component.css']
10 })
11 export class PokemonTypeComponent {
12   @Input() type: string = '';
13 }
14

```

HTML del tipo: `{{ type | titlecase }}`

Uso en la lista:

```

@for (type of pokemon.types; track $index) {
  <app-pokemon-type [type]="type"></app-pokemon-type>
}

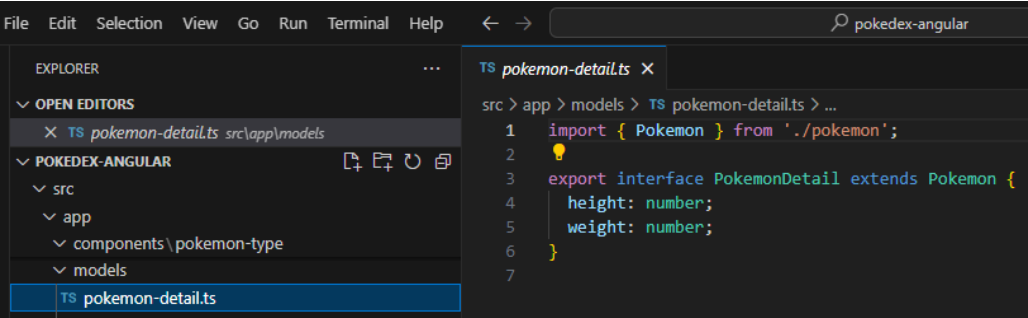
```

7. Vista “Detalle de pokémon”:

7.1 Modelo y método del servicio:

ng g c pages/pokemon

ng g interface models/pokemon-detail



```

src > app > models > TS pokemon-detail.ts > ...
1  import { Pokemon } from './pokemon';
2
3  export interface PokemonDetail extends Pokemon {
4    height: number;
5    weight: number;
6  }
7

```

7.2 HTML y lógica del detalle:

```

← → pokedex-angular
TS pokemon-list.component.ts X
src > app > pages > pokemon-list > TS pokemon-list.component.ts > ...
6 import { PokemonTypeComponent } from '../components/pokemon-type/pokemon-type.component';
7 import { PokeapiService } from '../services/pokeapi.service';
8
9 type Pokemon = { id:number; name:string; imageUrl:string; types:string[] };
10
11 @Component({
12   selector: 'app-pokemon-list',
13   standalone: true,
14   imports: [RouterModule, FormsModule, TitleCasePipe, DigitNumberPipe, PokemonTypeComponent],
15   templateUrl: './pokemon-list.component.html',
16   styleUrls: ['./pokemon-list.component.css']
17 })
18 export class PokemonListComponent implements OnInit {
19   allPokemons: Pokemon[] = [];
20   filteredPokemons: Pokemon[] = [];
21   query = '';
22   error: string | null = null;
23
24   constructor(private pokeApi: PokeapiService) {}
25
26   async ngOnInit(): Promise<void> {
27     try {
28       this.allPokemons = await this.pokeApi.getAllPokemon();
29       this.search();
30     } catch (e) {
31       this.error = 'No se pudo cargar la lista (revisa la consola del navegador).';
32       this.filteredPokemons = [];
33     }
34   }
35
36   search() {
37     const q = this.query.trim().toLowerCase();
38     this.filteredPokemons = q
39       ? this.allPokemons.filter(p => p.name.includes(q))
40       : this.allPokemons;
41   }
42 }
43

```

8. Rutas del detalle y enlace desde la lista:

```

TS app.routes.ts X
src > app > TS app.routes.ts > ...
1 import { Routes } from '@angular/router';
2 import { PokemonListComponent } from './pages/pokemon-list/pokemon-list.component';
3 import { PokemonComponent } from './pages/pokemon/pokemon.component';
4
5 export const routes: Routes = [
6   { path: '', component: PokemonListComponent },
7   { path: 'pokemon/:id', component: PokemonComponent },
8   { path: '**', redirectTo: '' }
9 ]

```

[Anterior](#) **#151 Mew** [Siguiente](#)



Psychic

Altura: 0.4 m

Peso: 4 Kg

9. Buscador por nombre:

Imports: usar FormsModule en el componente de lista.

```
<form class="filter" (ngSubmit)="search()">
```

```
  <input [(ngModel)]="query" [ngModelOptions]="{standalone:true}" placeholder="Buscar por nombre..." />
```

```
  <input type="submit" value="Buscar" />
```

```
</form>
```



Listado de pokémons



TS (lista):


```

1 <h1 style="text-align:center">Listado de pokémons</h1>
2
3 <form class="filter" (ngSubmit)="search()">
4   <input [(ngModel)]="query" [ngModelOptions]="{standalone:true}" placeholder="Buscar" />
5   <input type="submit" value="Buscar" />
6 </form>
7
8 @if (error) {
9   <div style="color: #b00; text-align:center">{{error}}</div>
10 }
11
12 <section class="pokemon-list">
13   @if (allPokemon?.length > 0) {
14     @for (pokemon of filteredPokemon; track pokemon.id) {
15       <a class="pokemon-card" [routerLink]="'/pokemon/' + pokemon.id">
16         <img [src]="pokemon.imageUrl" [alt]="pokemon.name | titlecase" />
17         <div class="id">{{pokemon.id | digitNumber: 3}}</div>

```

10. Botones Anterior/Siguiente y actualización al cambiar la URL Métodos goPrevious/goNext

```

1 @if (pokemon) {
2   <h1>
3     <button (click)="goPrevious()">Anterior</button>
4     #{{pokemon.id | digitNumber: 3}} {{pokemon.name | titlecase}}
5     <button (click)="goNext()">Siguiente</button>
6   </h1>
7
8   <section>
9     <img [src]="pokemon.imageUrl" alt="{{pokemon.name}}" />
10    <div class="info">
11      <div class="types">
12        @for (type of pokemon.types; track $index) {
13          <app-pokemon-type [type]="type"></app-pokemon-type>
14        }
15      </div>
16      <div>Altura: {{pokemon.height / 10}} m</div>
17      <div>Peso: {{pokemon.weight / 10}} Kg</div>
18    </div>
19  </section>
20 }

```



Anterior **#001 Bulbasaur** Siguiente



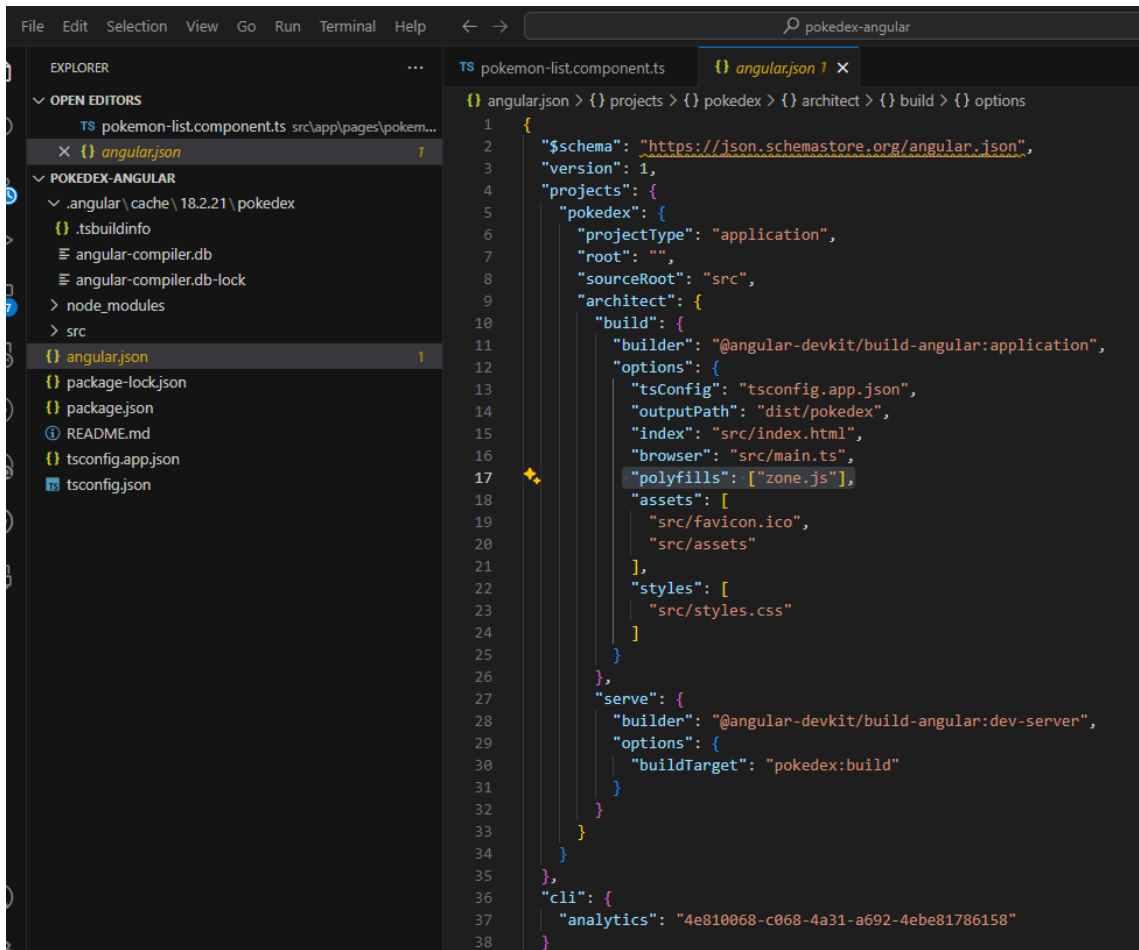
Grass Poison

Altura: 0.7 m

Peso: 6.9 Kg

11. Ajustes extra:

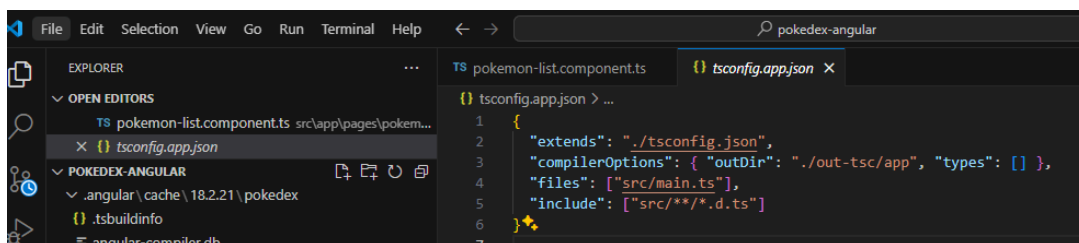
Me estaba dando problemas al cargar la página, así que creé tsconfig.app.json y añadí "tsConfig": "tsconfig.app.json" en angular.json y añadí zone.js



```

1  {
2    "$schema": "https://json.schemastore.org/angular.json",
3    "version": 1,
4    "projects": {
5      "pokedex": {
6        "projectType": "application",
7        "root": "",
8        "sourceRoot": "src",
9        "architect": {
10         "build": {
11           "builder": "@angular-devkit/build-angular:application",
12           "options": {
13             "tsConfig": "tsconfig.app.json",
14             "outputPath": "dist/pokedex",
15             "index": "src/index.html",
16             "browser": "src/main.ts",
17             "polyfills": ["zone.js"],
18             "assets": [
19               "src/favicon.ico",
20               "src/assets"
21             ],
22             "styles": [
23               "src/styles.css"
24             ]
25           },
26           "serve": {
27             "builder": "@angular-devkit/build-angular:dev-server",
28             "options": {
29               "buildTarget": "pokedex:build"
30             }
31           }
32         }
33       }
34     },
35     "cli": {
36       "analytics": "4e810068-c068-4a31-a692-4ebe81786158"
37     }
38   }

```



```

1  {
2    "extends": "./tsconfig.json",
3    "compilerOptions": { "outDir": "../out-tsc/app", "types": [] },
4    "files": ["src/main.ts"],
5    "include": ["src/**/*.d.ts"]
6  }
7

```