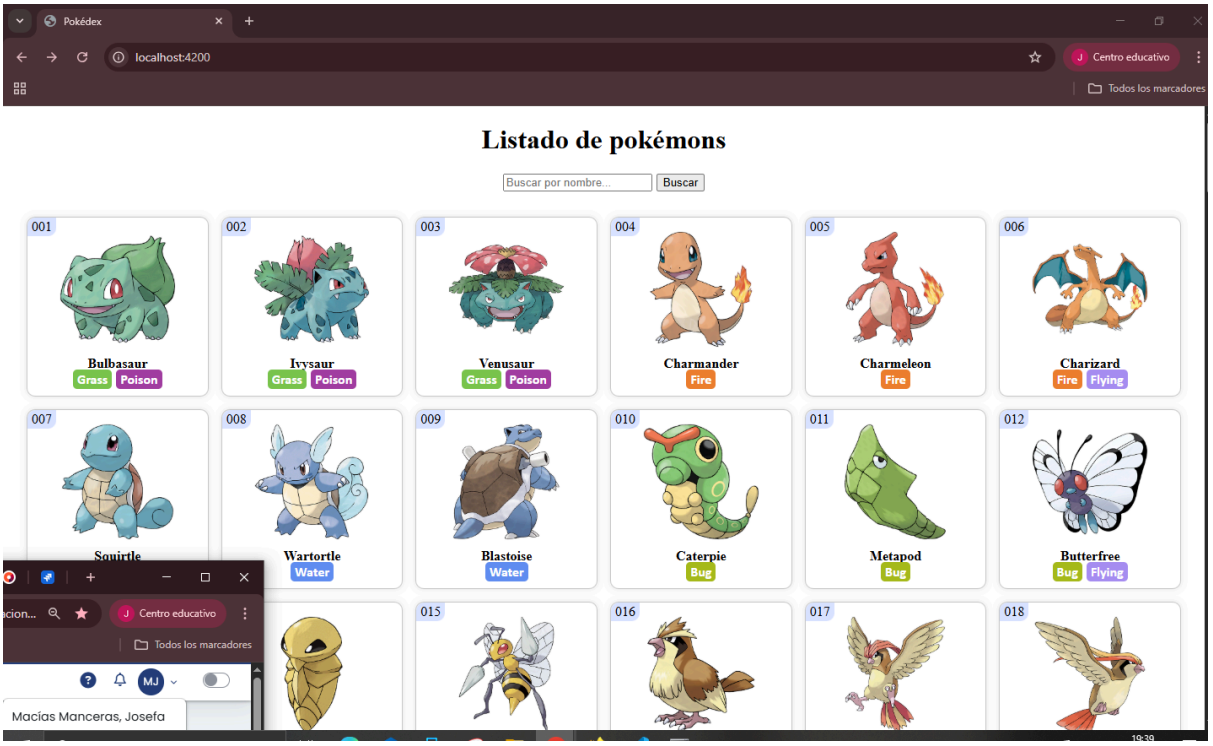
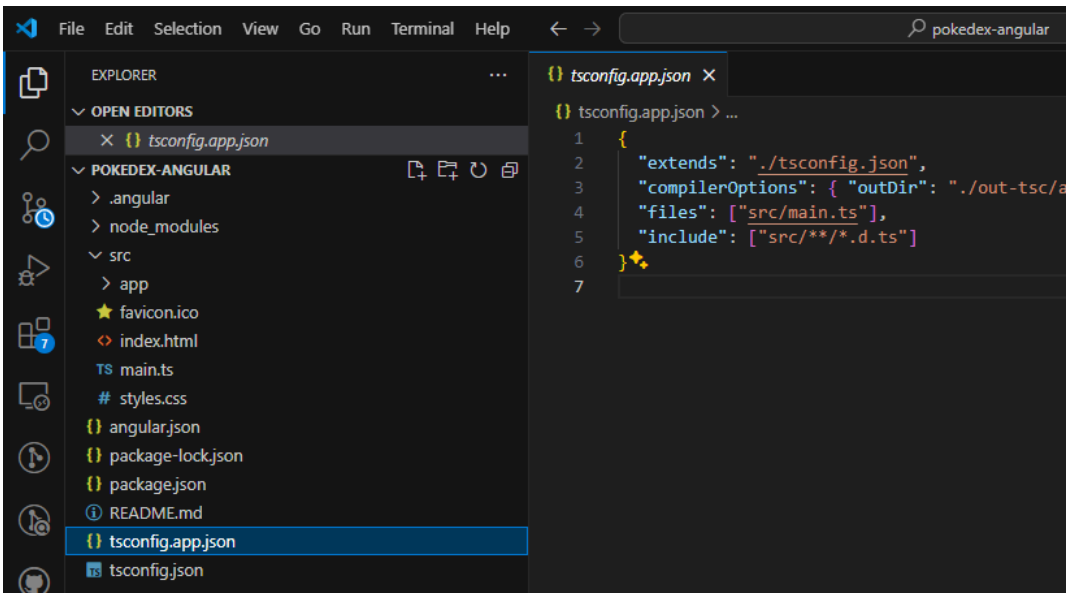
Resumen:

- 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:
Seleccione las opciones: Hojas de estilo CSS - No habilitar SSG/Prerendering.
Compruebo que se ejecute en Visual Studio Code.

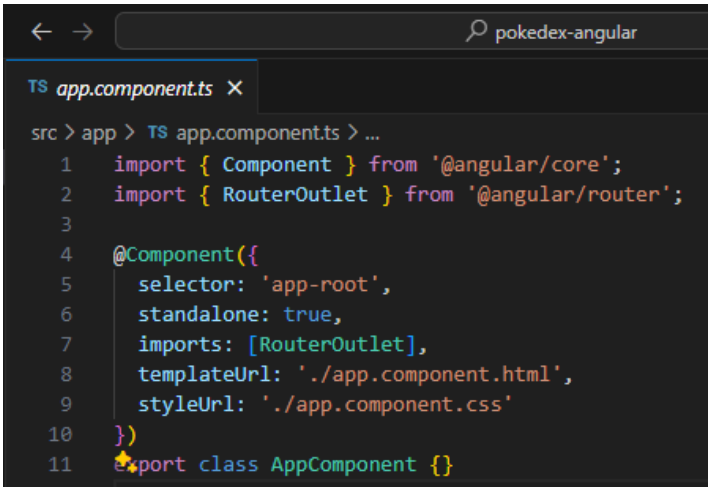


2. Vista Listado de pokémons: Con el componente: ng g c pages/pokemon-list. Y se configura el enrutador y la salida:

src/app/app.routes.ts:

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

src/app/app.component.html: <router-outlet></router-outlet>:
Esto fuerza a que se renderice el nuevo componente.

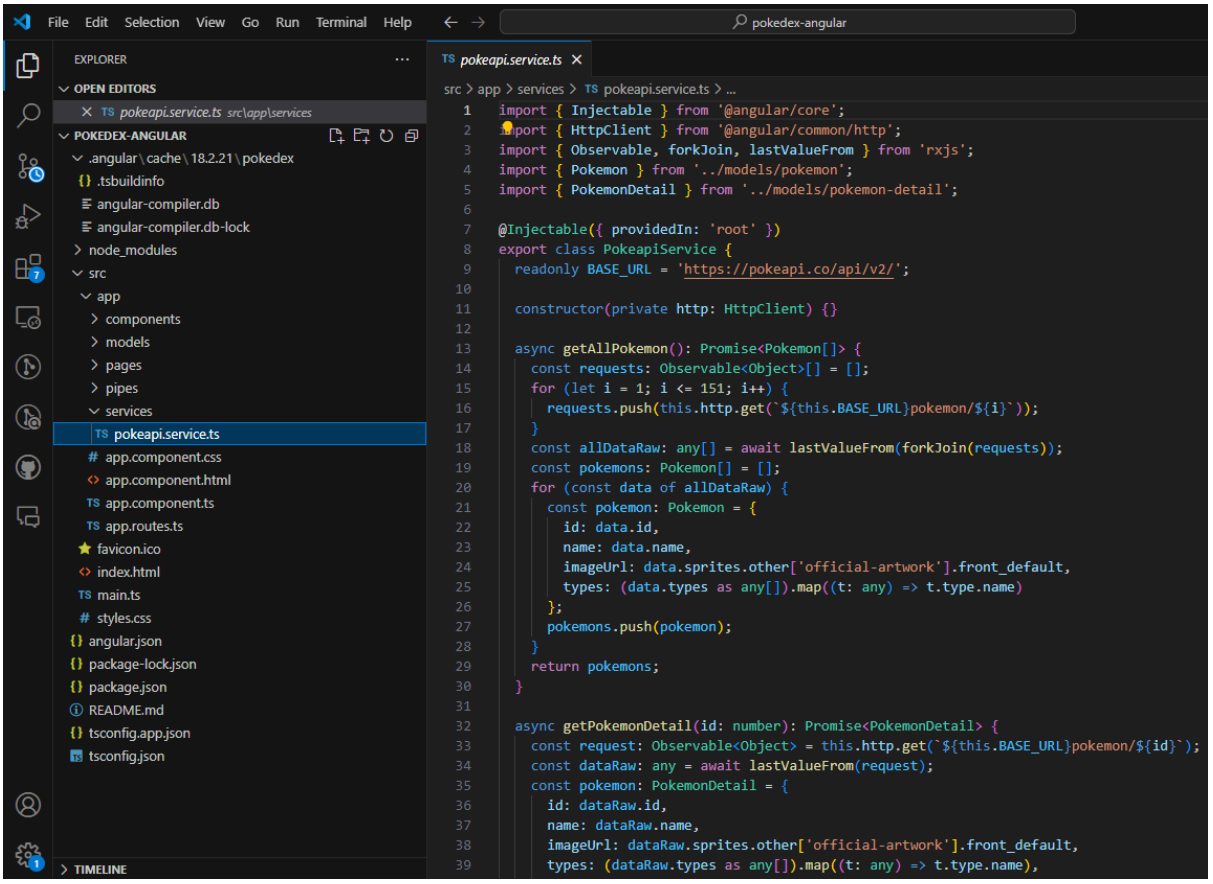


3. Modelos y servicio:

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

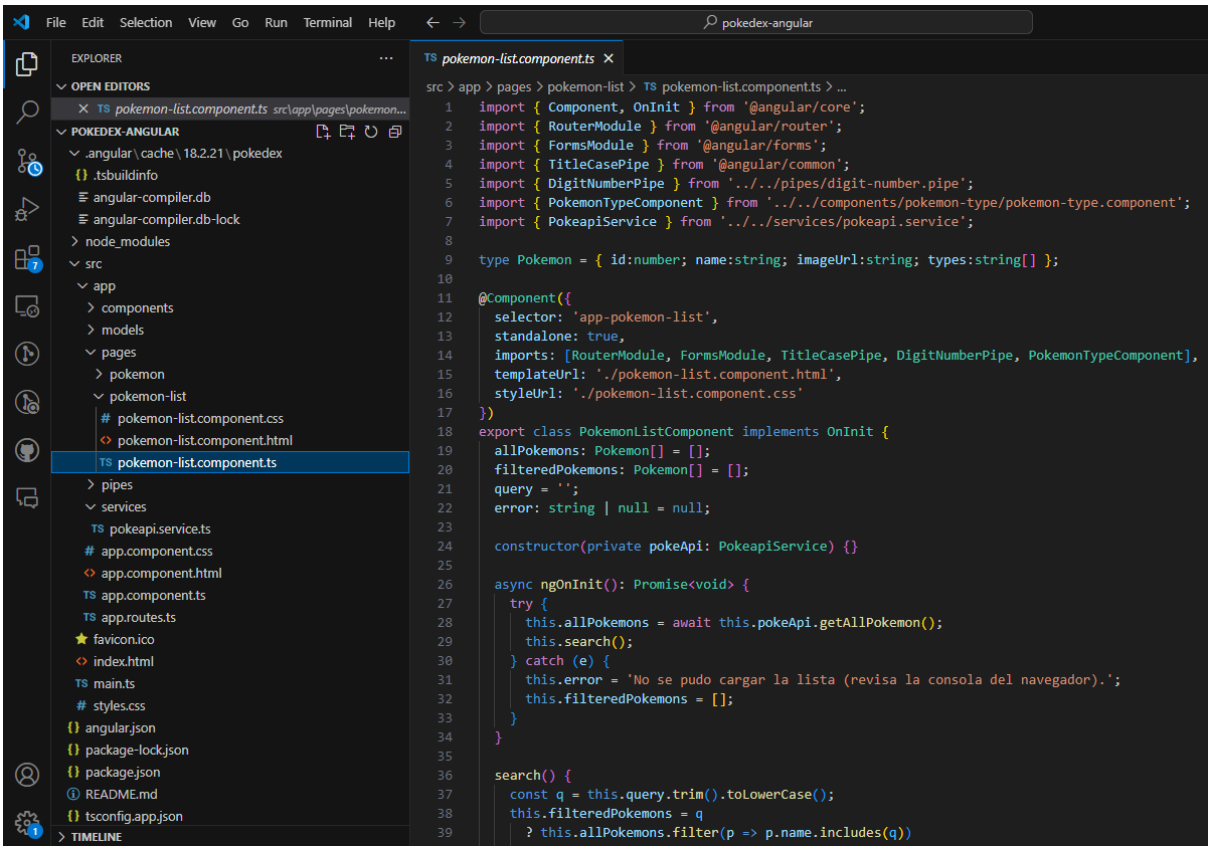
```
export interface Pokemon {
  id: number;
  name: string;
  imageUrl: string;
  types: string[];
}
```

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

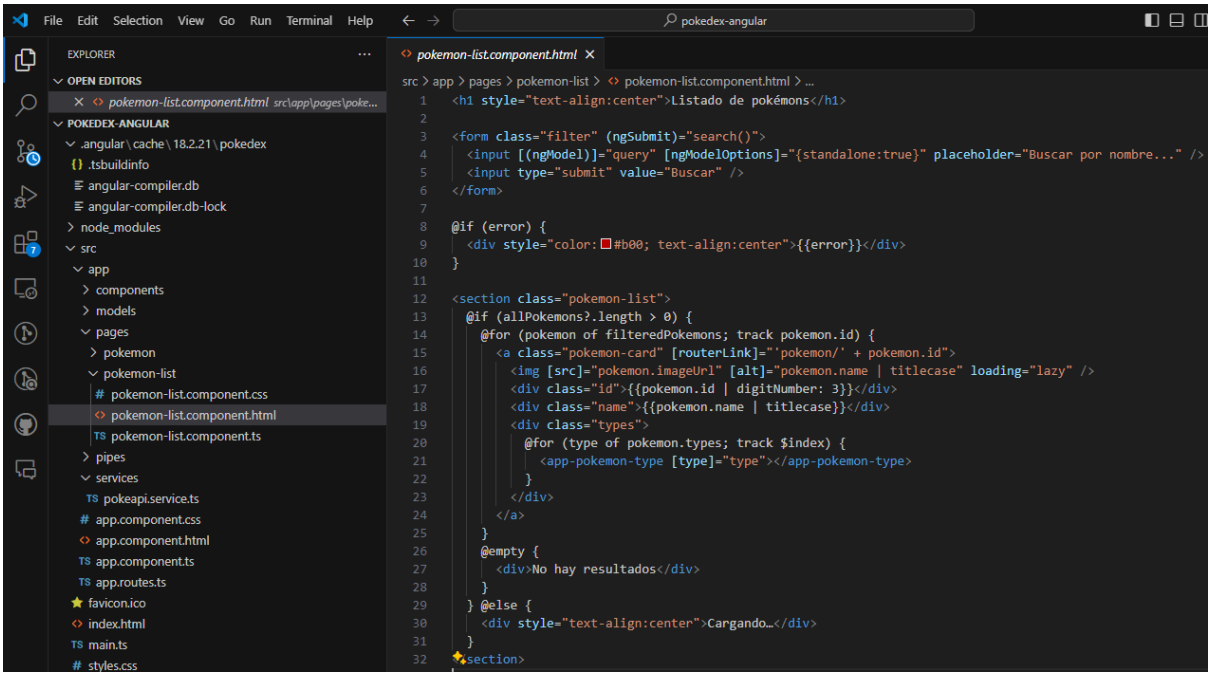


4. Listado: lógica y vista:

4.1 Lógica del componente:

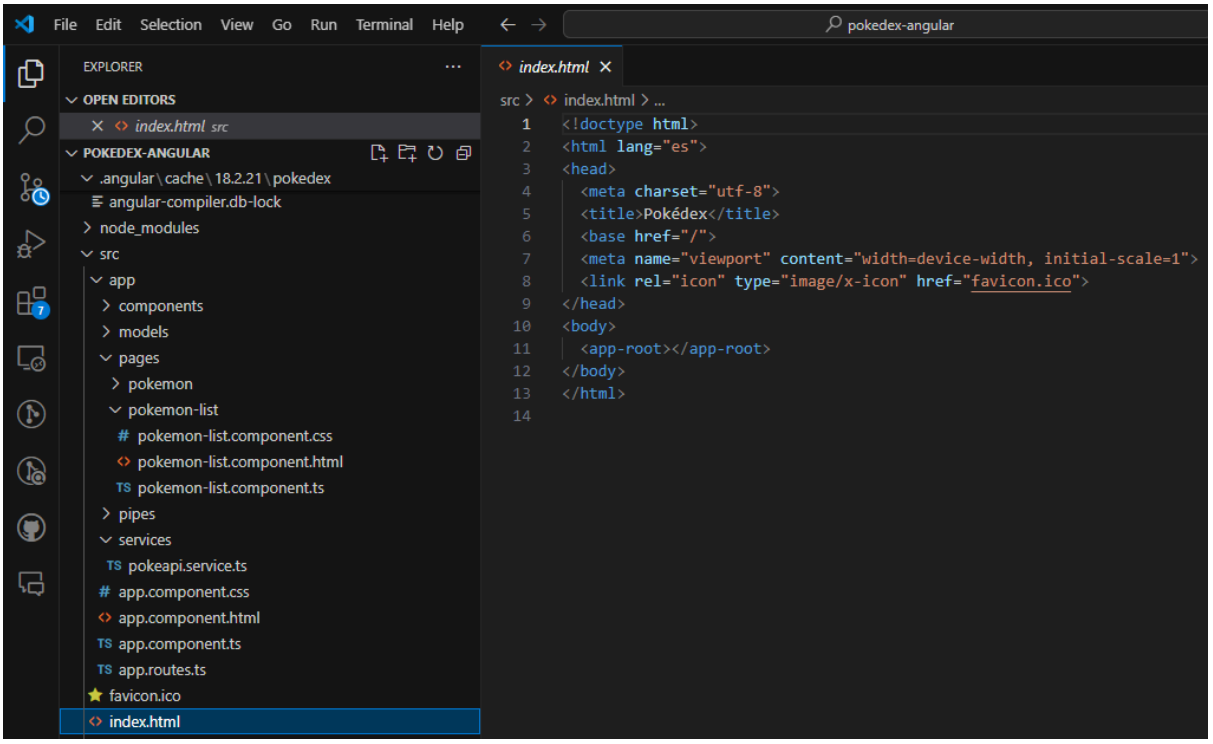


4.2 HTML básico:

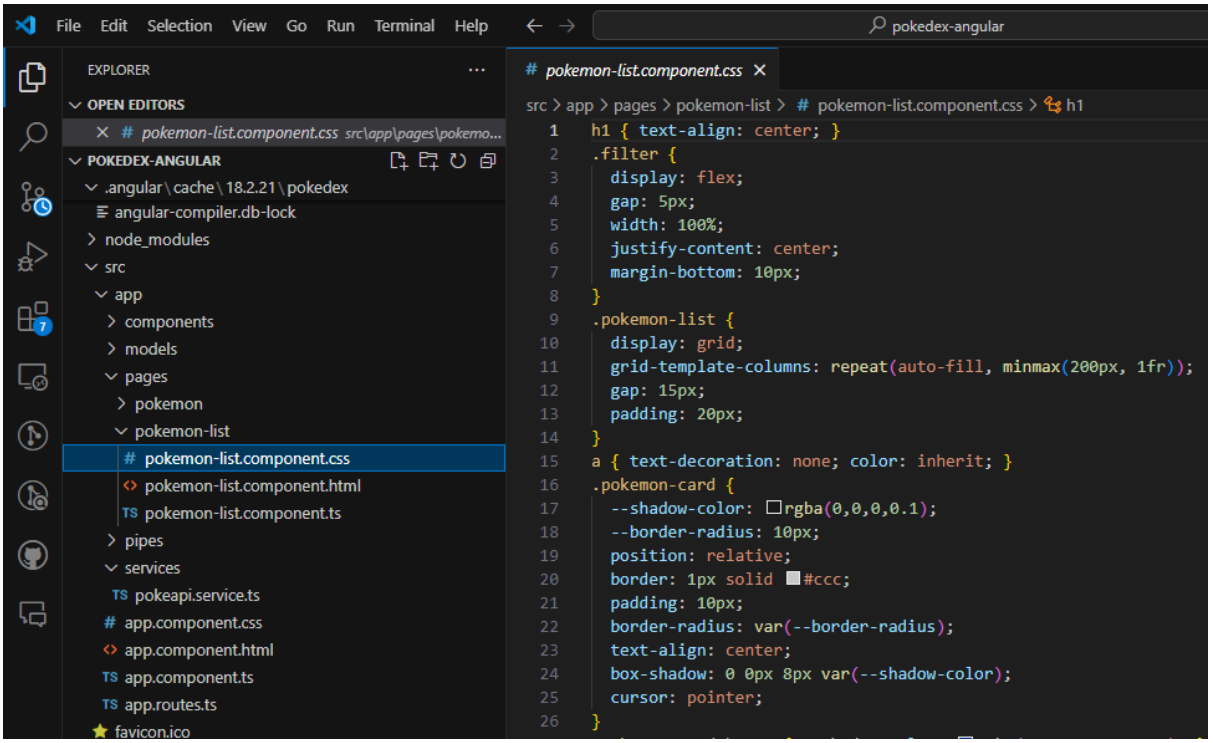


5. Mejoras de presentación:

5.1 Título de la pestaña:

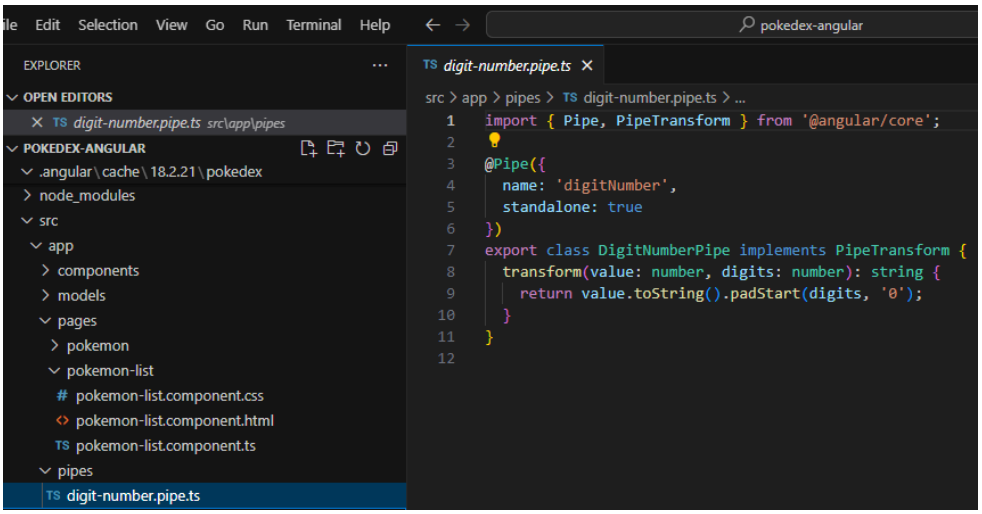


5.2 Estilos del grid:



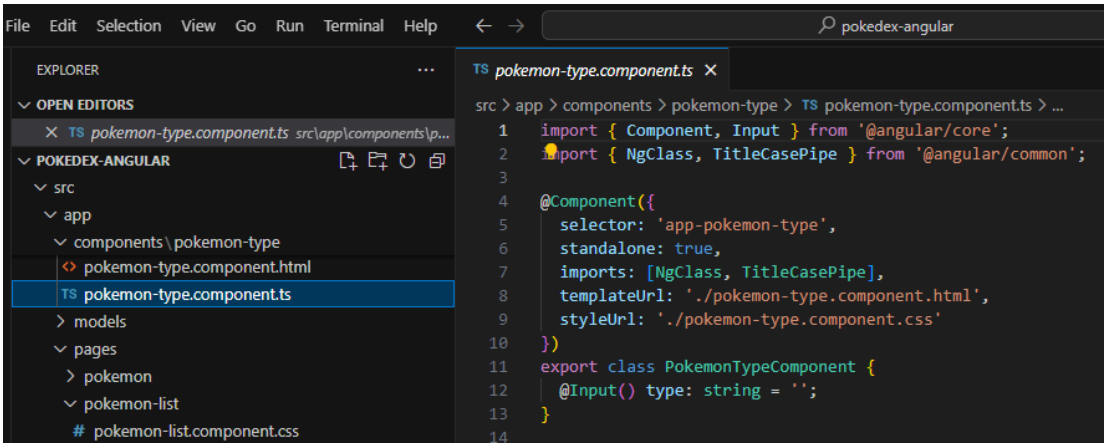
5.3 Nombre con mayúscula inicial: En el HTML:
{{ pokemon.name | titlecase }}

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



Uso en HTML: {{ pokemon.id | digitNumber:3 }}

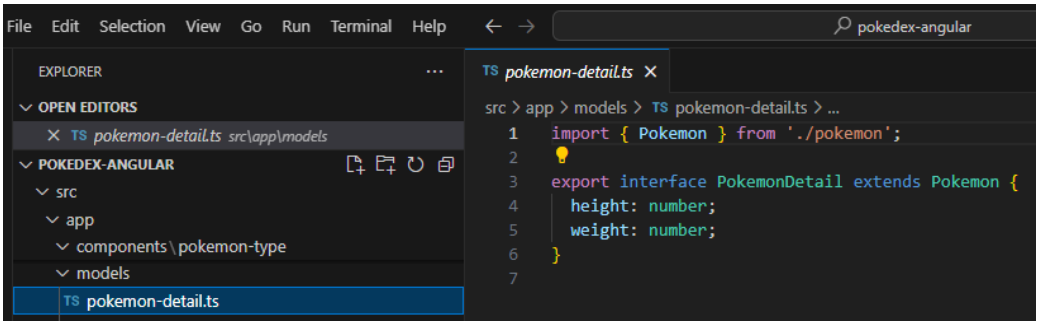
6. Componente de tipo (colores): ng g c components/pokemon-type:



HTML del tipo: <span [ngClass]="type">{{ 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



7.2 HTML y lógica del detalle:

pokedex-angular

pokemon-list.component.ts

src > app > pages > pokemon-list > pokemon-list.component.ts > ...

```ts
import { PokemonTypeComponent } from '../../components/pokemon-type/pokemon-type.component';
import { PokeapiService } from '../../services/pokeapi.service';

type Pokemon = { id:number; name:string; imageUrl:string; types:string[] };

@Component({
  selector: 'app-pokemon-list',
  standalone: true,
  imports: [RouterModule, FormsModule, TitleCasePipe, DigitNumberPipe, PokemonTypeComponent],
  templateUrl: './pokemon-list.component.html',
  styleUrl: './pokemon-list.component.css'
})
export class PokemonListComponent implements OnInit {
  allPokemons: Pokemon[] = [];
  filteredPokemons: Pokemon[] = [];
  query = '';
  error: string | null = null;

  constructor(private pokeApi: PokeapiService) {}

  async ngOnInit(): Promise<void> {
    try {
      this.allPokemons = await this.pokeApi.getAllPokemon();
      this.search();
    } catch (e) {
      this.error = 'No se pudo cargar la lista (revisa la consola del navegador).';
      this.filteredPokemons = [];
    }
  }

  search() {
    const q = this.query.trim().toLowerCase();
    this.filteredPokemons = q
      ? this.allPokemons.filter(p => p.name.includes(q))
      : this.allPokemons;
  }
}
```

8. Rutas del detalle y enlace desde la lista:

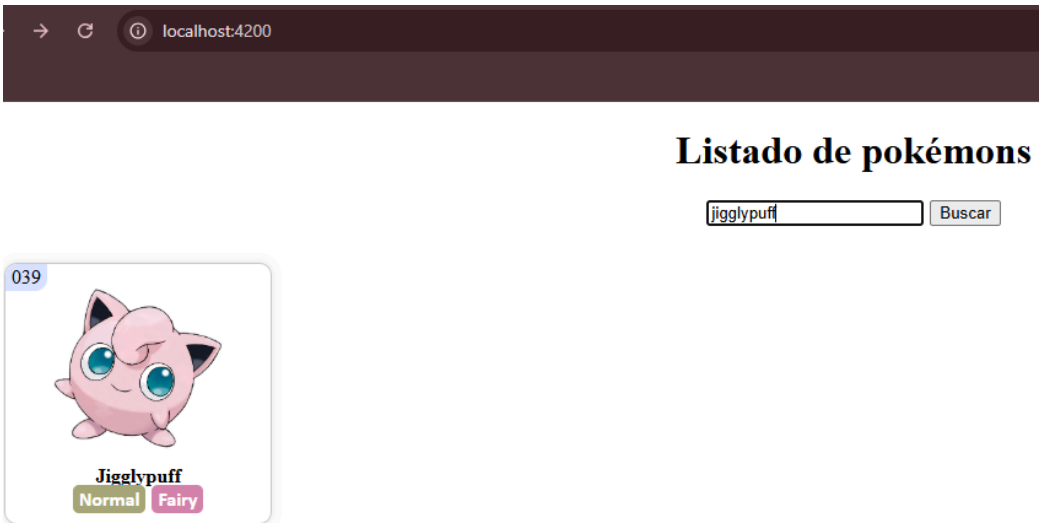
app.routes.ts

src > app > app.routes.ts > ...

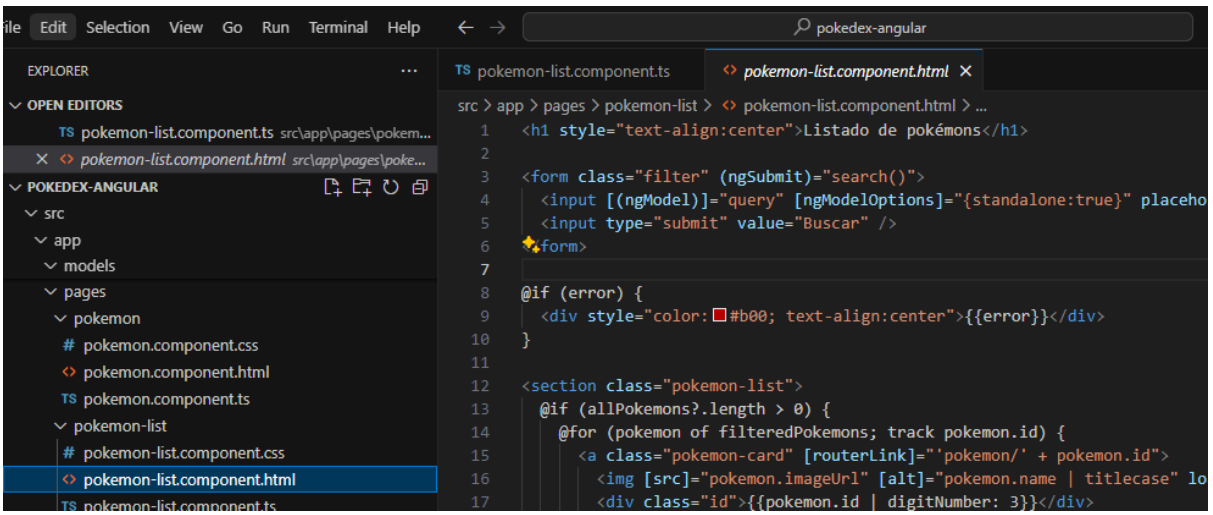
```ts
import { Routes } from '@angular/router';
import { PokemonListComponent } from './pages/pokemon-list/pokemon-list.component';
import { PokemonComponent } from './pages/pokemon/pokemon.component';

export const routes: Routes = [
  { path: '', component: PokemonListComponent },
  { path: 'pokemon/:id', component: PokemonComponent },
  { path: '**', redirectTo: '' }
];
```

9. Buscador por nombre:
Imports: usar FormsModule en el componente de lista.

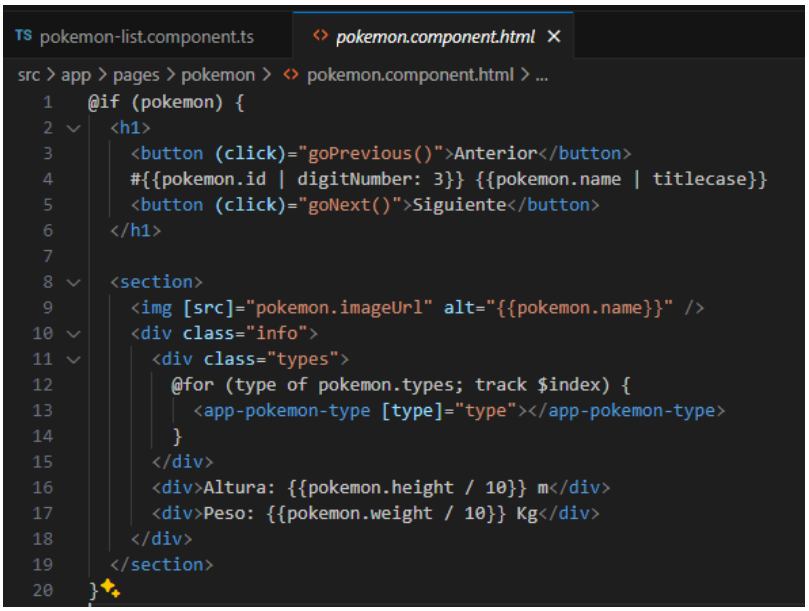
```
<form class="filter" (ngSubmit)="search()">
  <input [(ngModel)]="query" [ngModelOptions]="{standalone:true}" placeholder="Buscar por nombre..." />
  <input type="submit" value="Buscar" />
</form>
```

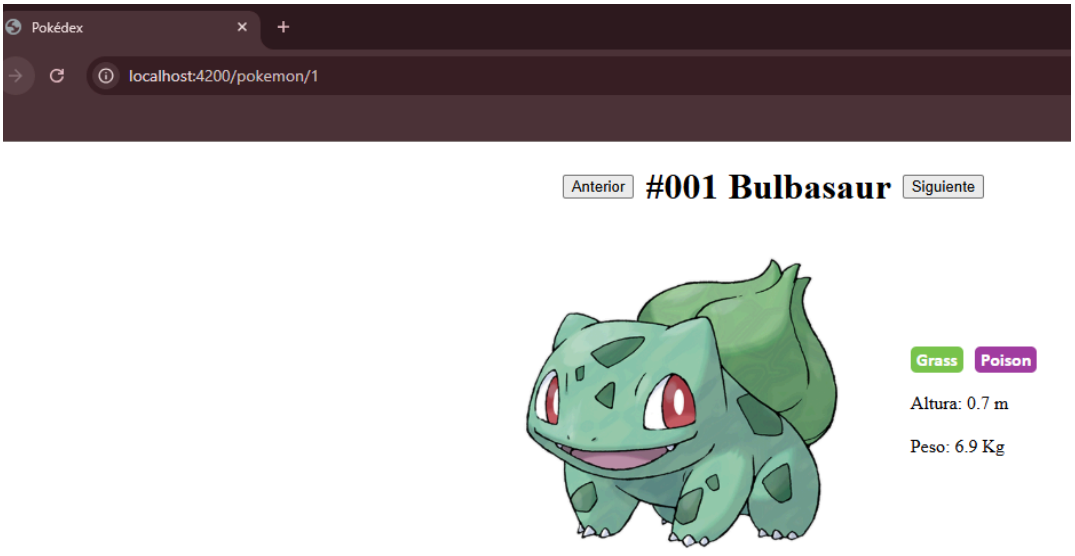


TS (lista):



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





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