

Seguimos con la app países, agregamos FormsModule a países.module.ts

The screenshot shows the Visual Studio Code interface with the Explorer, Editor, and Output panels. The Explorer panel on the left shows the project structure with 'países' selected. The Editor panel shows the 'países.module.ts' file with the following code:

```
src > app > países > TS países.module.ts > PaisesModule
3 import { PorCapitalComponent } from './pages/por-capital/por-capital.component';
4 import { PorPaisComponent } from './pages/por-pais/por-pais.component';
5 import { PorRegionComponent } from './pages/por-region/por-region.component';
6 import { VerPaisComponent } from './pages/ver-pais/ver-pais.component';
7 import { FormsModule } from '@angular/forms';
8
9
10
11 @NgModule({
12   declarations: [
13     PorCapitalComponent,
14     PorPaisComponent,
15     PorRegionComponent,
16     VerPaisComponent
17   ],
18   exports: [
19     PorCapitalComponent,
20     PorPaisComponent,
21     PorRegionComponent,
22     VerPaisComponent
23   ],
24   imports: [
25     CommonModule,
26     FormsModule
27   ]
28 })
```

The 'FormsModule' import on line 7 and the 'FormsModule' entry in the 'imports' array on line 26 are highlighted with red boxes. The Output panel at the bottom shows the message 'Compiled successfully.'

module.ts U <> por-pais.component.html U TS por-pais.component.ts U X

c > app > paises > pages > por-pais > TS por-pais.component.ts > PorPaisComponent > buscar

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-por-pais',
5   templateUrl: './por-pais.component.html',
6   styles: [
7   ]
8 })
9 export class PorPaisComponent implements OnInit {
10
11   termino:string = "Hola Mundo";
12
13   constructor() { }
14
15   buscar(){
16     console.log(this.termino);
17   }
18   ngOnInit(): void {
19   }
20
21 }
22
```

<> por-pais.component.html U X

> app > paises > pages > por-pais > <> por-pais.component.html > div.row >

```
Go to component
1 <h2>Por Pais</h2>
2 <hr>
3
4 <div class="row">
5   <div class="col">
6     <form (ngSubmit)="buscar()" autocomplete="off">
7       <input
8         type="text"
9         name="termino"
10        class="form-control"
11        [(ngModel)]="termino"
12        placeholder="Buscar pais...">
13     </form>
14   </div>
15 </div>
16
17 <hr>
18
19 <div class=" alert alert-danger">
20   No se encontro nada con la busqueda ingresada
21 </div>
22
23
24 <div class="row">
25   <div class="col">
```

Al darle enter nos imprime lo que contiene el input

The screenshot shows a web browser window with the title 'PaisesApp' and the address 'localhost:4200'. The page displays a search interface for countries. On the left, there is a sidebar with a blue 'Buscar Pais' button and two filter options: 'Por Región' and 'Por Capital'. The main search input field contains the text 'Hola Mundo'. Below the input field, a red message box states: 'No se encontro nada con la busqueda ingresada'. Below this message is a table with the following structure:

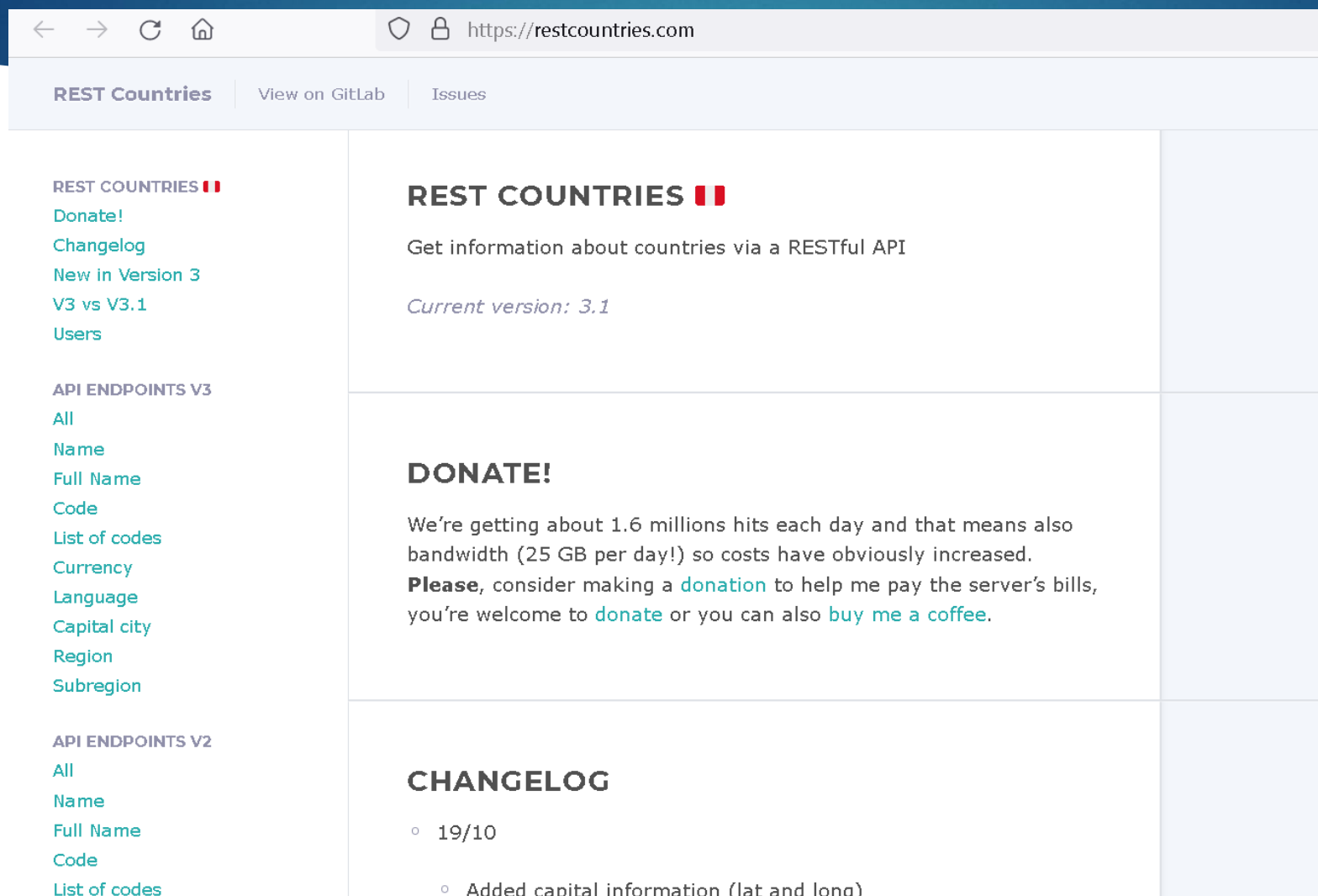
#	Bandera	Nombre	Poblacion
1	bandera	nombre	23232323

Each row in the table has a blue link labeled 'Ver...' to its right. The browser's developer console is open on the right side, showing the 'Consola' tab. It displays the following messages:

- [webpack-dev-server] Live Reloading enabled. (index.js)
- Angular is running in development mode. Call enableProdMode() core.mjs: to enable production mode.
- Hola Mundo (por-pais.component.ts:)

The console also shows a double right arrow '>>' at the bottom.

Api publica <https://restcountries.com/>



The screenshot shows a web browser displaying the REST Countries API website. The browser's address bar shows the URL <https://restcountries.com/>. The website has a light blue header with navigation links: "REST Countries", "View on GitLab", and "Issues". The main content area is divided into three sections. The first section, "REST COUNTRIES", includes links for "Donate!", "Changelog", "New in Version 3", "V3 vs V3.1", and "Users". The second section, "DONATE!", contains a paragraph about server costs and a request for donations, with links for "donation" and "buy me a coffee". The third section, "CHANGELOG", lists updates, including "Added capital information (lat and lon)". A sidebar on the left lists "API ENDPOINTS V3" and "API ENDPOINTS V2" with various filter options like "All", "Name", "Full Name", "Code", "List of codes", "Currency", "Language", "Capital city", "Region", and "Subregion".

REST Countries | View on GitLab | Issues

REST COUNTRIES

[Donate!](#)
[Changelog](#)
[New in Version 3](#)
[V3 vs V3.1](#)
[Users](#)

API ENDPOINTS V3

[All](#)
[Name](#)
[Full Name](#)
[Code](#)
[List of codes](#)
[Currency](#)
[Language](#)
[Capital city](#)
[Region](#)
[Subregion](#)

API ENDPOINTS V2

[All](#)
[Name](#)
[Full Name](#)
[Code](#)
[List of codes](#)

REST COUNTRIES

Get information about countries via a RESTful API

Current version: 3.1

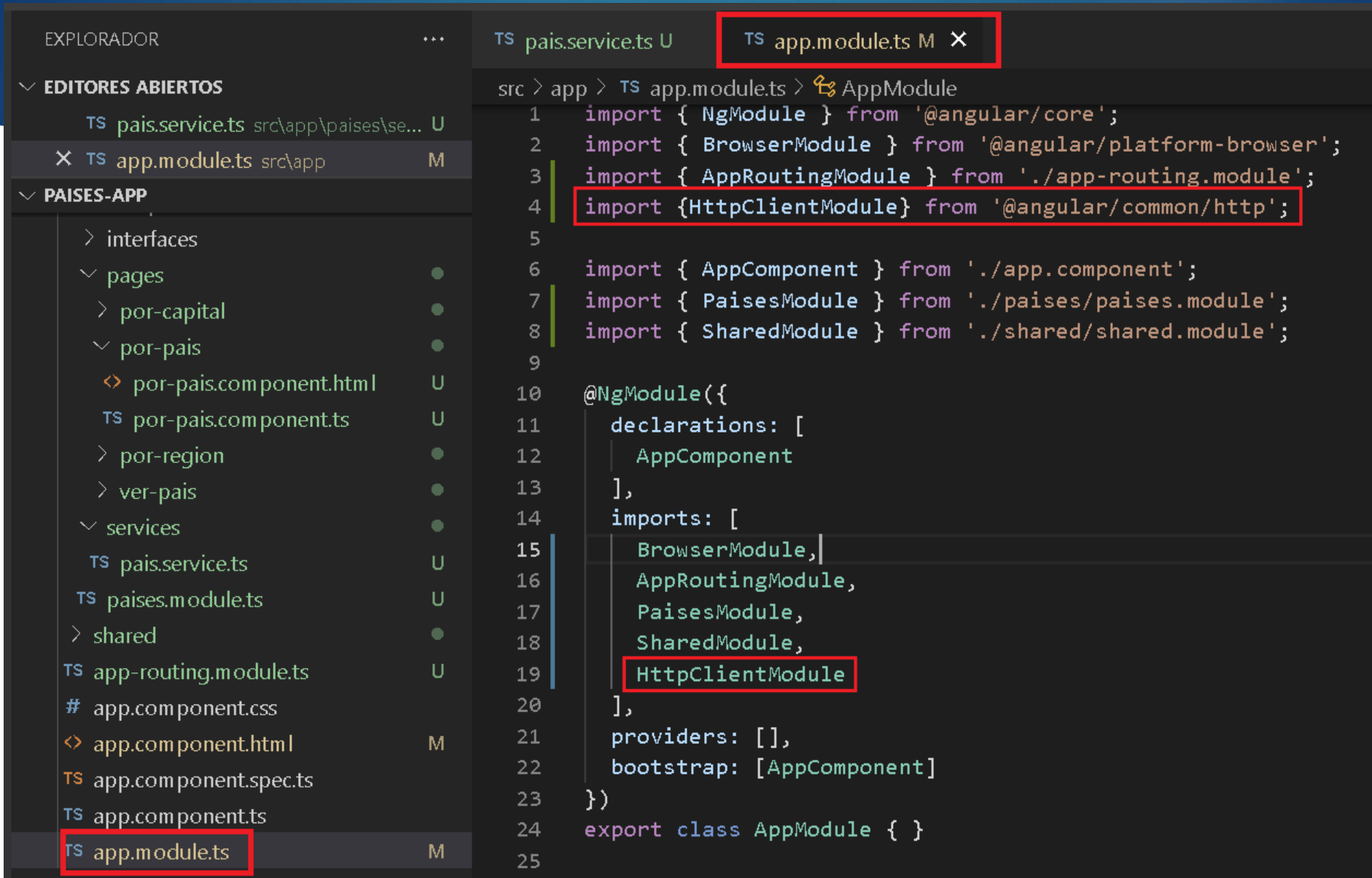
DONATE!

We're getting about 1.6 millions hits each day and that means also bandwidth (25 GB per day!) so costs have obviously increased.
Please, consider making a [donation](#) to help me pay the server's bills, you're welcome to [donate](#) or you can also [buy me a coffee](#).

CHANGELOG

- 19/10
 - Added capital information (lat and lon)

Llamamos al HttpClientModule para trabajar con peticiones http



The image shows a screenshot of the Visual Studio Code interface. On the left, the 'EXPLORADOR' (Explorer) sidebar is visible, showing the project structure. The 'EDITORES ABIERTOS' (Open Editors) section shows two files: 'TS pais.service.ts' and 'TS app.module.ts'. The 'PAISES-APP' section shows the project structure, including 'interfaces', 'pages', 'por-capital', 'por-pais', 'por-region', 'ver-pais', 'services', 'shared', and 'app-routing.module.ts'. The 'app.module.ts' file is highlighted in the Explorer. On the right, the 'TS app.module.ts' file is open in the editor. The code shows imports for 'NgModule', 'BrowserModule', 'AppRoutingModule', and 'HttpClientModule'. The 'HttpClientModule' import is highlighted with a red box. The '@NgModule' decorator is also visible, with 'HttpClientModule' listed in the 'imports' array, also highlighted with a red box.

```
src > app > TS app.module.ts > AppModule
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { AppRoutingModule } from './app-routing.module';
4  import { HttpClientModule } from '@angular/common/http';
5
6  import { AppComponent } from './app.component';
7  import { PaisesModule } from './paises/paises.module';
8  import { SharedModule } from './shared/shared.module';
9
10 @NgModule({
11   declarations: [
12     AppComponent
13   ],
14   imports: [
15     BrowserModule,
16     AppRoutingModule,
17     PaisesModule,
18     SharedModule,
19     HttpClientModule
20   ],
21   providers: [],
22   bootstrap: [AppComponent]
23 })
24 export class AppModule { }
25
```

EDITORES ABIERTOS

X TS pais.service.ts src\app\paises\se... U

TS app.module.ts src\app M

PAISES-APP

> interfaces

v pages ●

> por-capital ●

v por-pais ●

<> por-pais.component.html U

TS por-pais.component.ts U

> por-region ●

> ver-pais ●

v services ●

TS pais.service.ts U

TS paises.module.ts U

> shared ●

TS app-routing.module.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts M

> assets

src > app > paises > services > TS pais.service.ts > PaisService > buscarPais

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { Observable } from 'rxjs';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class PaisService {
9
10   //url base
11   private apiUrl: string = 'https://restcountries.com/v3.1';
12
13   constructor(private http:HttpClient) { }
14
15
16   buscarPais(termino:string): Observable<any>{
17     const url = `${ this.apiUrl }/name/${ termino }`;
18     return this.http.get(url);
19   }
20
21 }
22
```

EXPLORADOR

EDITORES ABIERTOS

TS pais.service.ts src\app\paises\se... U

X TS por-pais.component.ts src\ap... U

TS app.module.ts src\app M

PAISES-APP

> interfaces

v pages

> por-capital

v por-pais

<> por-pais.component.html U

TS por-pais.component.ts U

> por-region

> ver-pais

v services

TS pais.service.ts U

TS paises.module.ts U

> shared

TS app-routing.module.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts M

TS pais.service.ts U

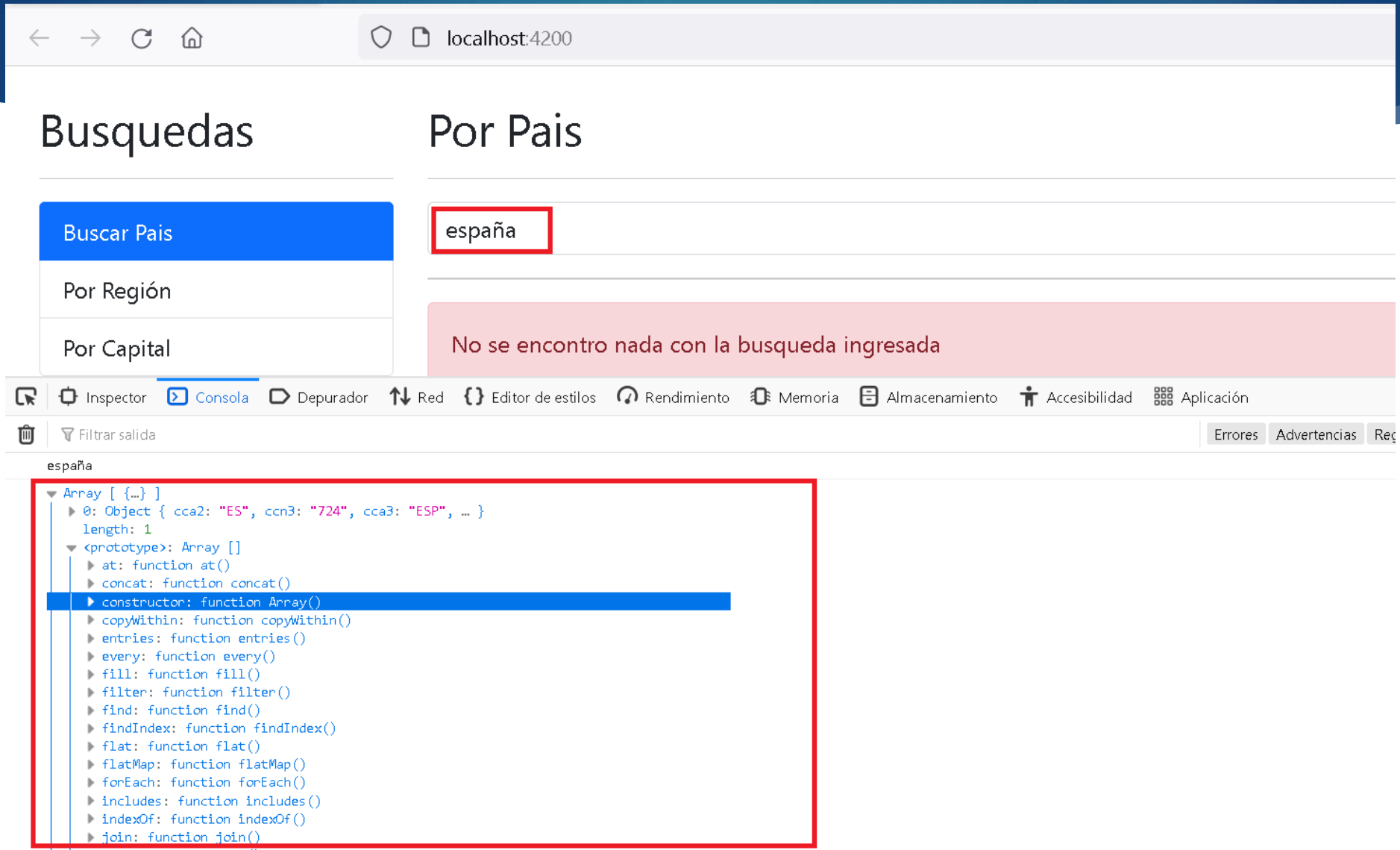
TS por-pais.component.ts U X

TS app.module.ts M

src > app > paises > pages > por-pais > TS por-pais.component.ts > PorPaisComponent > buscar >

```
1 import { Component } from '@angular/core';
2 import { PaisService } from '../../services/pais.service';
3
4 @Component({
5   selector: 'app-por-pais',
6   templateUrl: './por-pais.component.html',
7   styles: [
8   ]
9 })
10 export class PorPaisComponent {
11
12   termino:string = "";
13
14   constructor( private paiservice: PaisService) { }
15
16   buscar(){
17     console.log(this.termino);
18     this.paiservice.buscarPais(this.termino).subscribe(resp =>{
19       console.log(resp);
20     });
21   }
22
23
24 }
25
```

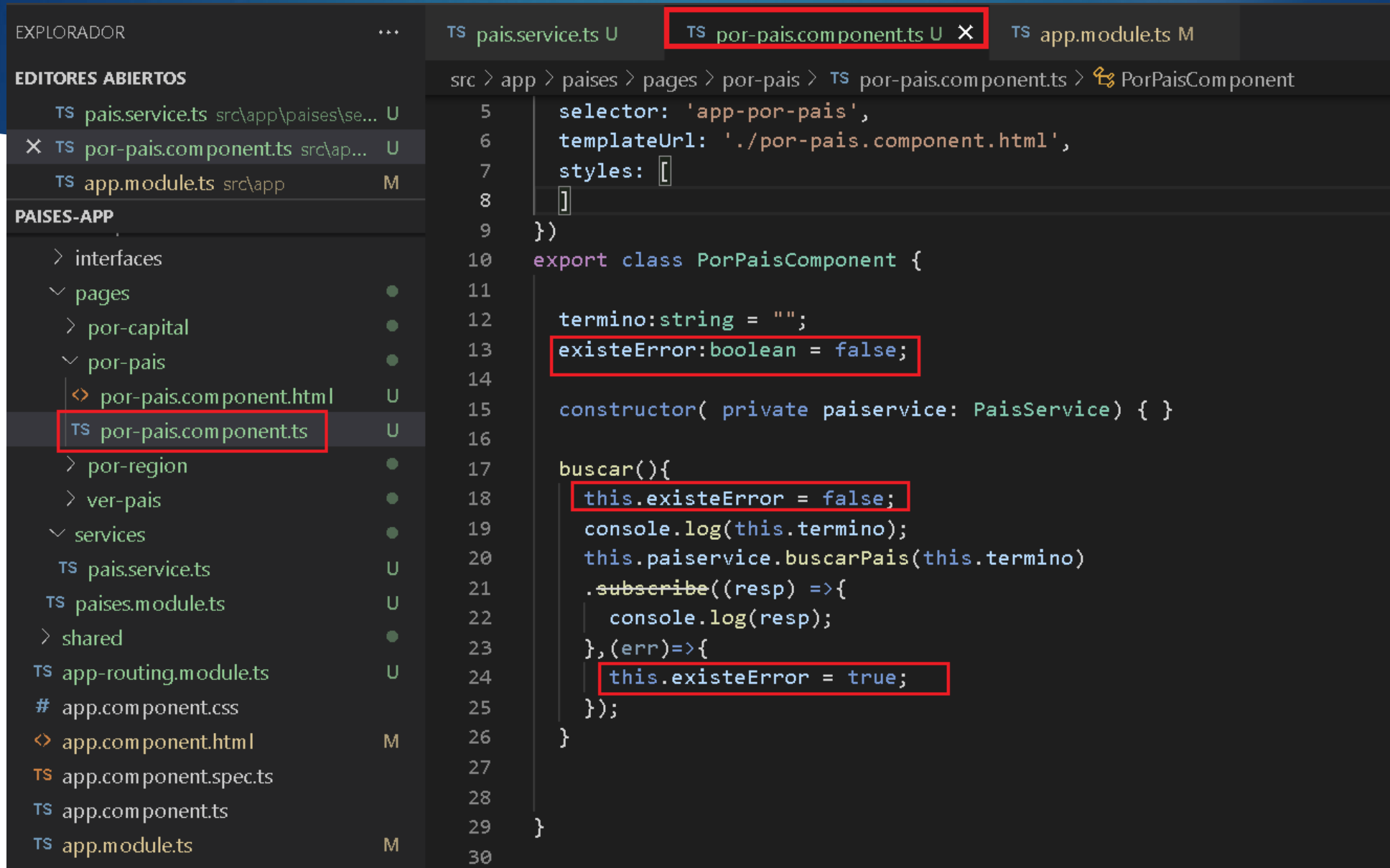
Probamos la petición http get



The screenshot shows a web browser at `localhost:4200` with a search interface. The left sidebar has three buttons: "Buscar Pais" (highlighted in blue), "Por Región", and "Por Capital". The main area has a search input with "españa" and a red border. Below the input is a red message: "No se encontro nada con la busqueda ingresada". The bottom of the browser shows the Chrome DevTools console with the following output:

```
españa
Array [ {...} ]
  0: Object { cca2: "ES", ccn3: "724", cca3: "ESP", ... }
    length: 1
  <prototype>: Array []
    at: function at()
    concat: function concat()
    constructor: function Array()
    copyWithin: function copyWithin()
    entries: function entries()
    every: function every()
    fill: function fill()
    filter: function filter()
    find: function find()
    findIndex: function findIndex()
    flat: function flat()
    flatMap: function flatMap()
    forEach: function forEach()
    includes: function includes()
    indexOf: function indexOf()
    join: function join()
```


Manejo de errores



The image shows a screenshot of an IDE with a dark theme. On the left is the Explorer sidebar showing a project structure for 'PAISES-APP'. The file 'TS por-pais.component.ts' is selected and highlighted with a red box. At the top, the file explorer shows three tabs: 'TS pais.service.ts U', 'TS por-pais.component.ts U' (highlighted with a red box), and 'TS app.module.ts M'. The main editor displays the code for 'PorPaisComponent'. The code includes a selector, templateUrl, and styles. It defines a class 'PorPaisComponent' with a 'termino:string' property, an 'existeError:boolean' property (highlighted with a red box), and a constructor that takes 'paiservice: PaisService'. The 'buscar()' method is defined, which sets 'this.existeError = false;' (highlighted with a red box), logs the termino, calls 'paiservice.buscarPais', and subscribes to the response. In the error handling block, it sets 'this.existeError = true;' (highlighted with a red box). The line numbers 5 through 30 are visible on the left side of the code editor.

```
5 selector: 'app-por-pais',
6 templateUrl: './por-pais.component.html',
7 styles: []
8 ]
9 })
10 export class PorPaisComponent {
11
12     termino:string = "";
13     existeError:boolean = false;
14
15     constructor( private paiservice: PaisService) { }
16
17     buscar(){
18         this.existeError = false;
19         console.log(this.termino);
20         this.paiservice.buscarPais(this.termino)
21             .subscribe((resp) =>{
22                 console.log(resp);
23             },(err)=>{
24                 this.existeError = true;
25             });
26     }
27
28
29 }
30
```

EXPLORADOR

EDITORES ABIERTOS

TS por-pais.component.ts src\ap... U

X <> por-pais.component.html src\... U

PAISES-APP

interfaces

pages

por-capital

por-pais

<> por-pais.component.html U

TS por-pais.component.ts U

por-region

ver-pais

services

TS pais.service.ts U

TS paises.module.ts U

shared

TS app-routing.module.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts M

assets

environments

★ favicon.ico

ESQUEMA

TS por-pais.component.ts U

<> por-pais.component.html U X

src > app > paises > pages > por-pais > <> por-pais.component.html > div.row

12placeholder="Buscar pais..."

13</form>

14

15</div>

16</div>

17

18<hr>

19

20<div *ngIf="existeError"

21class=" alert alert-danger">

22No se encontro nada con la busqueda ingresada

23</div>

24

25<div class="row">

26<div class="col">

27<table class="table table-hover">

28<thead>

29<tr>

30<th>#</th>

31<th>Bandera</th>

32<th>Nombre</th>

33<th>Poblacion</th>

34<th></th>

35</tr>

36</thead>

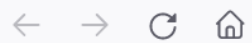
37<tbody>

PROBLEMAS

SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL



localhost:4200



Busquedas

Buscar Pais

Por Región

Por Capital

Por Pais

Buscar pais...

#	Bandera	Nombre	Poblacion	
1	bandera	nombre	23232323	Ver...

Inspector Consola Depurador Red Editor de estilos Rendimiento Memoria Almacenamiento Accesibilidad Aplicación

Filtrar salida

Errores Advertencias Registros Información Depurar

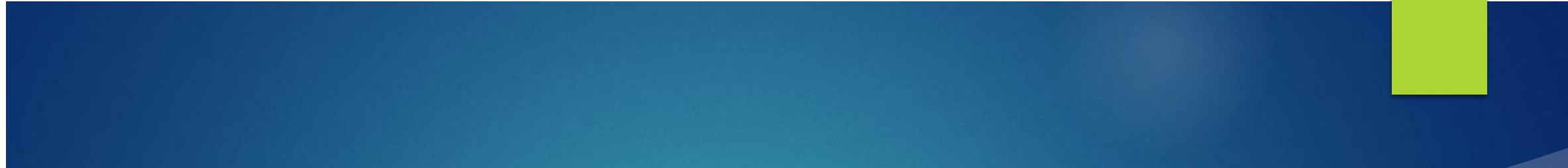
[webpack-dev-server] Disconnected!

[webpack-dev-server] Trying to reconnect...

Angular is running in development mode. Call enableProdMode() to enable production mode.

[webpack-dev-server] Live Reloading enabled.

>>



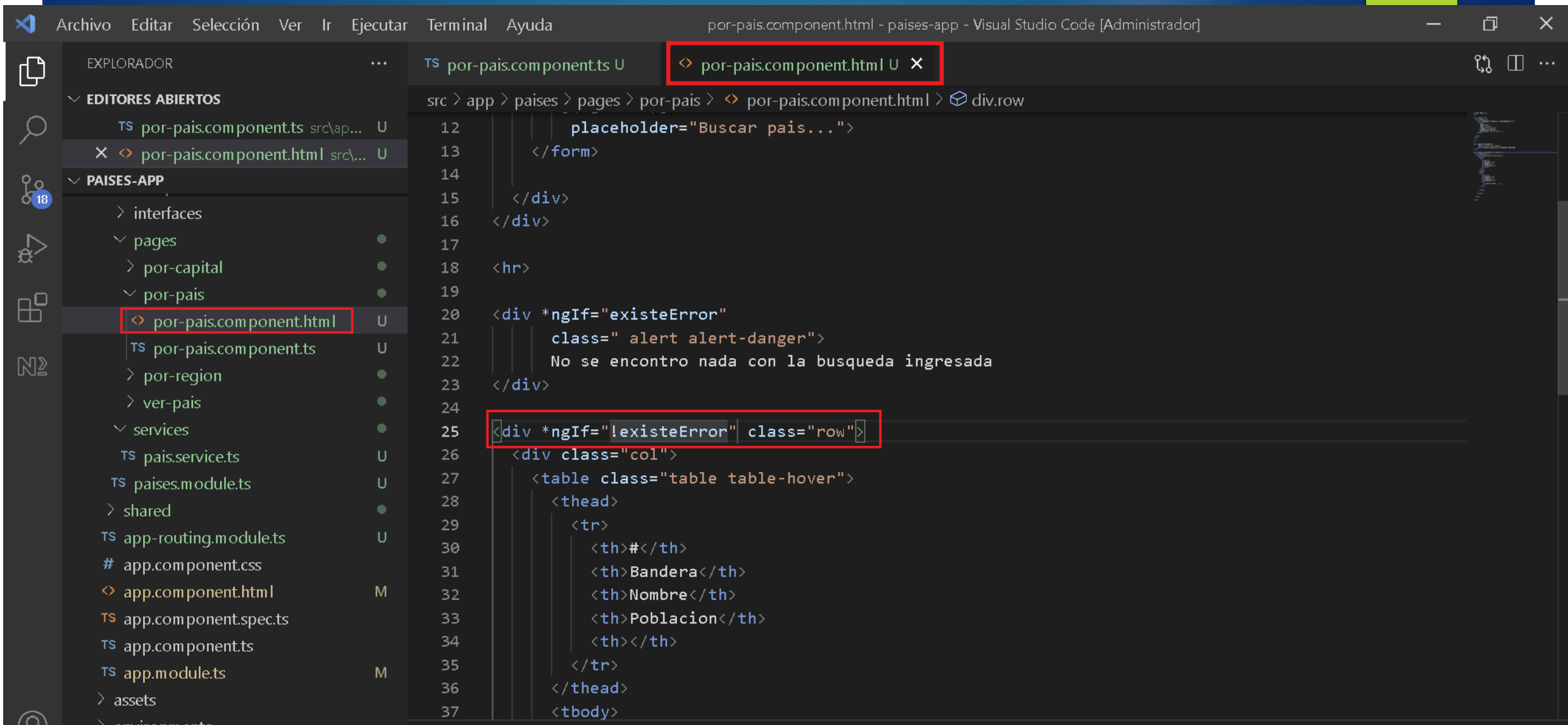
Busquedas

- Buscar Pais
- Por Región
- Por Capital

Por Pais

No se encontro nada con la busqueda ingresada

#	Bandera	Nombre	Poblacion	
1	bandera	nombre	23232323	Ver...



Busquedas

Buscar Pais

Por Región

Por Capital

Por Pais

lkjlkjslkasas

No se encontro nada con la busqueda ingresada

Tipado de la petición

The screenshot displays a REST client interface with a GET request to `https://restcountries.com/v3.1/name/espa`. The response is a JSON object with the following structure:

```
[
  {
    "name": {
      "common": "Spain",
      "official": "Kingdom of Spain",
      "nativeName": {
        "spa": {
          "official": "Reino de España",
          "common": "España"
        }
      }
    }
  }
],
```

The interface includes a top bar with request history, a main area for the current request (method, URL, and tabs for Params, Authorization, Headers, Body, Pre-request Script, Tests, and Settings), and a bottom section for the response (Body, Cookies, Headers, and Test Results). The response status is 200 OK, with a time of 436 ms and a size of 1.51 KB. The response is displayed in a JSON format, with a 'Pretty' button and a 'JSON' dropdown menu.

Formateamos con esta app la respuesta de la petición

←

→

↺

🏠

🔒 https://app.quicktype.io

☆

📁

☰

QT

quicktype

👤 Please Share!

👁 Options

?

☰

Name

Source type

Welcome

JSON

📁

```
{
  "name": {
    "common": "Spain",
    "official": "Kingdom of Spain",
    "nativeName": {
      "spa": {
        "official": "Reino de España",
        "common": "España"
      }
    }
  },
  "tld": [
    ".es"
  ],
  "cca2": "ES",
  "ccn3": "724",
  "cca3": "ESP",
  "cioc": "ESP",
  "independent": true,
  "status": "officially-assigned",
  "unMember": true,
  "currencies": {
    "EUR": {
      "name": "Euro",
      "symbol": "€"
    }
  }
}
```

```
export interface Welcome {
  name: Name;
  tld: string[];
  cca2: string;
  ccn3: string;
  cca3: string;
  cioc: string;
  independent: boolean;
  status: string;
  unMember: boolean;
  currencies: Currencies;
  idd: Id;
  capital: string[];
  altSpellings: string[];
  region: string;
  subregion: string;
  languages: Languages;
  translations: { [key: string]: Translation };
  latlng: number[];
  landlocked: boolean;
  borders: string[];
  area: number;
  demonyms: Demonyms;
  flag: string;
  maps: Maps;
  population: number;
  gini: Gini;
  fifa: string;
  car: Car;
  timezones: string[];
  continents: string[];
  flags: CostOfArms;
}
```

Language

Other

TypeScript

☒ Interfaces only

☐ Transform property names to be JavaScripty

☐ Explicitly name unions

☒ Verify JSON.parse results at runtime

☐ Make all properties optional ?

Copy Code

Visual Studio Code interface showing the Explorer, Editor, and Terminal panels.

Explorer Panel: Displays the file structure of the project. The `interfaces` folder is expanded, and the file `TS pais.interfaces.ts` is selected.

Editor Panel: Shows the content of `TS pais.interfaces.ts`. The file path is `src > app > paises > interfaces > TS pais.interfaces.ts`. The code defines an interface `Countries` with the following properties:

```
1 export interface Countries {  
2     name: Name;  
3     tld: string[];  
4     cca2: string;  
5     ccn3: string;  
6     cca3: string;  
7     cioc: string;  
8     independent: boolean;  
9     status: string;  
10    unMember: boolean;  
11    currencies: Currencies;  
12    idd: Idd;  
13    capital: string[];  
14    altSpellings: string[];  
15    region: string;  
16    subregion: string;  
17    languages: Languages;  
18    translations: { [key: string]: Translation };  
19    latlng: number[];  
20    landlocked: boolean;  
21    borders: string[];  
22    area: number;  
23    demonyms: Demonyms;  
24    flag: string;  
25    maps: Maps;  
26    population: number;
```

TS pais.interfaces.ts U

TS pais.service.ts U X

src > app > paises > services > TS pais.service.ts > ...

```
1  import { HttpClient } from '@angular/common/http';
2  import { Injectable } from '@angular/core';
3  import { Observable } from 'rxjs';
4  import { Countries } from '../interfaces/pais.interfaces';
5
6  @Injectable({
7    providedIn: 'root'
8  })
9  export class PaisService {
10
11    //url base
12    private apiUrl: string = 'https://restcountries.com/v3.1';
13
14    constructor(private http:HttpClient) { }
15
16
17    buscarPais(termino:string): Observable<Countries[]>{
18      const url = `${ this.apiUrl }/name/${ termino }`;
19      return this.http.get<Countries[]>(url);
20    }
21
22  }
23
```

TS por-pais.component.ts U X

<> por-pais.component.html U

TS pais.interfaces.ts U

src > app > paises > pages > por-pais > TS por-pais.component.ts > PorPaisComponent >

```
9      ]
10    })
11    export class PorPaisComponent {
12
13      termino:string = "";
14      existeError:boolean = false;
15      paises: Countries[]=[];
16
17      constructor( private paiservice: PaisService) { }
18
19      buscar(){
20        this.existeError = false;
21        console.log(this.termino);
22        this.paiservice.buscarPais(this.termino)
23          .subscribe((resp) =>{
24            console.log(resp);
25            this.paises = resp;
26
27          },(err)=>{
28            this.existeError = true;
29          });
30      }
31
32    }
```

TS por-pais.component.ts U

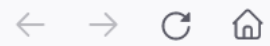
<> por-pais.component.html U X

TS pais.interfaces.ts U

src > app > paises > pages > por-pais > <> por-pais.component.html > div.row > div.col > table.table.table-t

```
28     <thead>
29       <tr>
30         <th>#</th>
31         <th>Bandera</th>
32         <th>Nombre</th>
33         <th>Poblacion</th>
34         <th></th>
35       </tr>
36     </thead>
37     <tbody>
38       <tr *ngFor="let item of paises; let i = index">
39         <td>{{ i+1 }}</td>
40         <td>
41           <img style="width: 50px;" [src]="item.flags.png" alt="">
42         </td>
43         <td>{{item.name.common}}</td>
44         <td>{{item.population | number}}</td>
45         <td>
46           <a href="#">Ver...</a>
47         </td>
48       </tr>
49     </tbody>
50   </table>
51
52
53
```

Probamos si todo funciona



localhost:4200

Busquedas

Buscar Pais

Por Región

Por Capital

Por Pais

#	Bandera	Nombre	Poblacion	
1		Spain	47,351,567	Ver...

EXPLORADOR

EDITORES ABIERTOS

TS paises.module.ts src\app\paises U

PAISES-APP

> .angular

> .vscode

> node_modules

src

app

paises

> components

> interfaces

> pages

services

TS pais.service.ts U

TS paises.module.ts U

> shared

TS app-routing.module.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts M

> assets

> environments

TS paises.module.ts U X

src > app > paises > TS paises.module.ts > PaisesModule

```
4 import { PorPaisComponent } from '../pages/por-pais/por-pais.component';
5 import { PorRegionComponent } from '../pages/por-region/por-region.component';
6 import { VerPaisComponent } from '../pages/ver-pais/ver-pais.component';
7 import { FormsModule } from '@angular/forms';
8 import { RouterModule } from '@angular/router';
```

```
12 @NgModule({
13   declarations: [
14     PorCapitalComponent,
15     PorPaisComponent,
16     PorRegionComponent,
17     VerPaisComponent
18   ],
19   exports: [
20     PorCapitalComponent,
21     PorPaisComponent,
22     PorRegionComponent,
23     VerPaisComponent
24   ],
25   imports: [
26     CommonModule,
27     FormsModule,
28     RouterModule
29   ]
```

<> por-pais.component.html U X

src > app > paises > pages > por-pais > <> por-pais.component.html > div.row > div.col > table

```
33     <th>Poblacion</th>
34     <th></th>
35   </tr>
36 </thead>
37 <tbody>
38   <tr *ngFor="let item of paises; let i = index">
39     <td>{{ i+1 }}</td>
40     <td>
41       <img style="width: 50px;" [src]="item.flags.png" alt="">
42     </td>
43     <td>{{item.name.common}}</td>
44     <td>{{item.population | number}}</td>
45     <td>
46       <a [routerLink] = "['/pais',item.cca2]">Ver...</a>
47     </td>
48   </tr>
49
50 </tbody>
51
52 </table>
```

Separamos en mas componentes

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

> node + v [] [] ^ X

```
PS C:\Users\dell\Desktop\pruebas\países-app> ng g c países/components/paisTabla --skipTests -is
Support for camel case arguments has been deprecated and will be removed in a future major version.
```

Use '--skip-tests' instead of '--skipTests'.

```
CREATE src/app/países/components/pais-tabla/pais-tabla.component.html (25 bytes)
```

```
CREATE src/app/países/components/pais-tabla/pais-tabla.component.ts (262 bytes)
```

```
UPDATE src/app/países/países.module.ts (935 bytes)
```

```
PS C:\Users\dell\Desktop\pruebas\países-app> [ ]
```


<> por-pais.component.html U X

<> pais-tabla.component.html U

src > app > paises > pages > por-pais > <> por-pais.component.html > div.row >

```

20 <div *ngIf="existeError"
21   |   |   class=" alert alert-danger">
22   |   |   No se encontro nada con la busqueda ingresad
23 </div>
24
25 <div *ngIf="!existeError" class="row">
26   <div class="col">
27 >   <table class="table table-hover"> ...
52   ...</table>
53
54   </div>
55
56 </div>
57

```

<> pais-tabla.component.html U X

src > app > paises > components > pais-tabla > <> pais-tabla.component.html >

Go to component

```

1 <p>pais-tabla works!</p>
2

```

<> por-pais.component.html U X

<> pais-tabla.component.html 1, U X

<> pais-tabla.component.html 1, U X

src > app > paises > pages > por-pais > <> por-pais.component.html > div.row

```

20 <div *ngIf="existeError"
21 |   class=" alert alert-danger">
22 |   No se encontro nada con la busqueda ingresad
23 </div>
24
25 <div *ngIf="!existeError" class="row">
26 |   <div class="col">
27 |     <app-pais-tabla></app-pais-tabla>
28 |   </div>
29 |
30 </div>
31
32 </div>
33

```

components > pais-tabla > <> pais-tabla.component.html > table.table.table-ho

Go to component

```

1 <table class="table table-hover">
2 |   <thead>
3 |     <tr>
4 |       <th>#</th>
5 |       <th>Bandera</th>
6 |       <th>Nombre</th>
7 |       <th>Poblacion</th>
8 |       <th></th>
9 |     </tr>
10 |   </thead>
11 |   <tbody>
12 |     <tr *ngFor="let item of paises; let i = index">
13 |       <td>{{ i+1 }}</td>
14 |       <td>
15 |         <img style="width: 50px;" [src]="item.flags
16 |       </td>
17 |       <td>{{item.name.common}}</td>
18 |       <td>{{item.population | number}}</td>
19 |       <td>

```

EXPLORADOR



EDITORES ABIERTOS

<> por-pais.component.html s... U

X TS pais-tabla.component.ts src... U

PAISES-APP

src

app

pais

components \ pais-tabla

<> pais-tabla.component.html 1, U

TS pais-tabla.component.ts U

> interfaces

pages

> por-capital

por-pais

<> por-pais.component.html U

TS por-pais.component.ts U

> por-region

> ver-pais

> services

<> por-pais.component.html U

TS pais-tabla.component.ts U X

src > app > pais > components > pais-tabla > TS pais-tabla.component.ts > PaisTablaComponent

```
1  import { Component, Input, OnInit } from '@angular/core';
2  import { Countries } from '../interfaces/pais.interfaces';
3
4  @Component({
5    selector: 'app-pais-tabla',
6    templateUrl: './pais-tabla.component.html',
7    styles: [
8    ]
9  })
10 export class PaisTablaComponent implements OnInit {
11
12    @Input() paises: Countries[]=[];
13
14    constructor() { }
15
16    ngOnInit(): void {
17    }
18
19  }
20
```

EXPLORADOR

<> por-pais.component.html U X

TS pais-tabla.component.ts U

✓ EDITORES ABIERTOS

X <> por-pais.component.html s... U

TS pais-tabla.component.ts src... U

✓ PAISES-APP

src

app

países

components \ pais-tabla

<> pais-tabla.component.html 1, U

TS pais-tabla.component.ts U

> interfaces

pages

> por-capital

por-pais

<> por-pais.component.html U

TS por-pais.component.ts U

> por-region

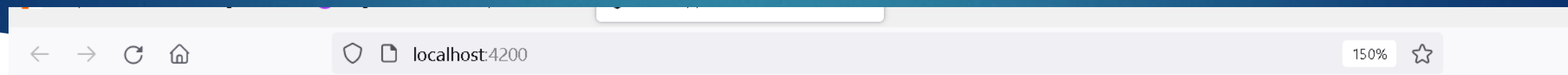
> ver-pais

> services

src > app > países > pages > por-pais > <> por-pais.component.html > div.row

```
20 <div *ngIf="existeError"
21     class=" alert alert-danger">
22     No se encontro nada con la busqueda ingresada
23 </div>
24
25 <div *ngIf="!existeError" class="row">
26     <div class="col">
27         <app-pais-tabla [países]="países"></app-pais-tabla>
28     </div>
29
30 </div>
31
32 </div>
33
```

Una vez pasado a un componente la tabla para reutilizarlo probamos que funcione todo




Busquedas

Buscar Pais

Por Región

Por Capital

Por Pais

#	Bandera	Nombre	Poblacion	
1		Portugal	10,305,564	Ver...

Reutilizamos el input

PROBLEMAS

SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL

> node + v [] [] ^ x

```
PS C:\Users\dell\Desktop\pruebas\países-app> ng g c países/components/paisInput --skipTests -is
Support for camel case arguments has been deprecated and will be removed in a future major version.
Use '--skip-tests' instead of '--skipTests'.
```

```
CREATE src/app/países/components/pais-input/pais-input.component.html (25 bytes)
```

```
CREATE src/app/países/components/pais-input/pais-input.component.ts (262 bytes)
```

```
UPDATE src/app/países/países.module.ts (1042 bytes)
```

```
PS C:\Users\dell\Desktop\pruebas\países-app> [ ]
```

<> por-pais.component.html U X

<> pais-input.component.html 3, U ...

por-pais > <> por-pais.component.html > div.row > div.col > app-pais-input

Go to component

```
1 <h2>Por Pais</h2>
2 <hr>
3
4 <div class="row">
5   <div class="col">
6     <app-pais-input></app-pais-input>
7   </div>
8 </div>
9
10 </div>
11
12 <hr>
13
14 <div *ngIf="existeError"
15   class="alert alert-danger">
16   No se encontro nada con la busqueda ingresada
17 </div>
18
19 <div *ngIf="!existeError" class="row">
```

<> pais-input.component.html 3, U X

src > app > paises > components > pais-input > <> pais-input.component.html > ...

Go to component

```
1 <form (ngSubmit)="buscar()" autocomplete="off">
2   <input
3     type="text"
4     name="termino"
5     class="form-control"
6     [(ngModel)]="termino"
7     placeholder="Buscar pais...">
8 </form>
9
```

TS pais-input.component.ts U X

<> por-pais.component.html U

TS por-pais.component.ts U

src > app > paises > components > pais-input > TS pais-input.component.ts > PaisInputComponent > buscar

```
1  import { Component, Output, EventEmitter } from '@angular/core';
```

```
2
```

```
3
```

```
4  @Component({
```

```
5    selector: 'app-pais-input',
```

```
6    templateUrl: './pais-input.component.html',
```

```
7    styles: [
```

```
8    ]
```

```
9  })
```

```
10 export class PaisInputComponent {
```

```
11
```

```
12   @Output() onEnter: EventEmitter<string> = new EventEmitter()
```

```
13
```

```
14   termino:string='';
```

```
15
```

```
16   buscar(){
```

```
17     this.onEnter.emit(this.termino);
```

```
18   }
```

```
19
```

```
20
```




TS pais-input.component.ts U

<> por-pais.component.html 1, U X



src > app > paises > pages > por-pais > <> por-pais.component.html > div.row > div.col > app-pais-inp

```
4 <div class="row">
5   <div class="col">
6     <app-pais-input (onEnter)="buscar($event)"></app-pais-input>
7   </div>
8 </div>
9
10 </div>
11
12 <hr>
13
14 <div *ngIf="existeError"
15   class="alert alert-danger">
16   No se encontro nada con la busqueda ingresada
17 </div>
18
19 <div *ngIf="!existeError" class="row">
20   <div class="col">
21     <app-pais-tabla [paises]="paises"></app-pais-tabla>
22   </div>
23 </div>
```

TS pais-input.component.ts U

<> por-pais.component.html U

TS por-pais.component.ts U X

src > app > paises > pages > por-pais > TS por-pais.component.ts > PorPaisComponent > buscar

```
17 constructor( private paiservice: PaisService) { }
```

```
18  
19 buscar(termino: string){
```

```
20  
21   this.termino = termino;
```

```
22  
23   this.existeError = false;
```

```
24   console.log(this.termino);
```

```
25   this.paiservice.buscarPais(this.termino)
```

```
26   .subscribe((resp) =>{
```

```
27     console.log(resp);
```

```
28     this.paises = resp;
```

```
29  
30   },(err)=>{
```

```
31     this.existeError = true;
```

```
32   });
```

```
33 }
```

```
34  
35  
36 }
```

Busquedas


Por Pais

Buscar Pais

Por Región

Por Capital

Germany

#	Bandera	Nombre	Poblacion	
1		Germany	83,240,525	Ver...

Creamos un nuevo servicio para capital

The screenshot shows the Visual Studio Code interface with the Explorer and Editor views. The Explorer view on the left shows the project structure with the file `TS pais.service.ts` selected and highlighted with a red box. The Editor view on the right shows the code for `PaisService` in `TS pais.service.ts`. The breadcrumb navigation at the top of the editor shows the path `src > app > paises > services > TS pais.service.ts > PaisService`. The code defines a class `PaisService` with a private `apiUrl` and two methods: `buscarPais` and `buscarCapital`. The `buscarCapital` method is highlighted with a red box.

EXPLORADOR

... TS pais.service.ts U X

src > app > paises > services > TS pais.service.ts > PaisService

EDITORES ABIERTOS

- X TS pais.service.ts src\app\paises... U

PAISES-APP

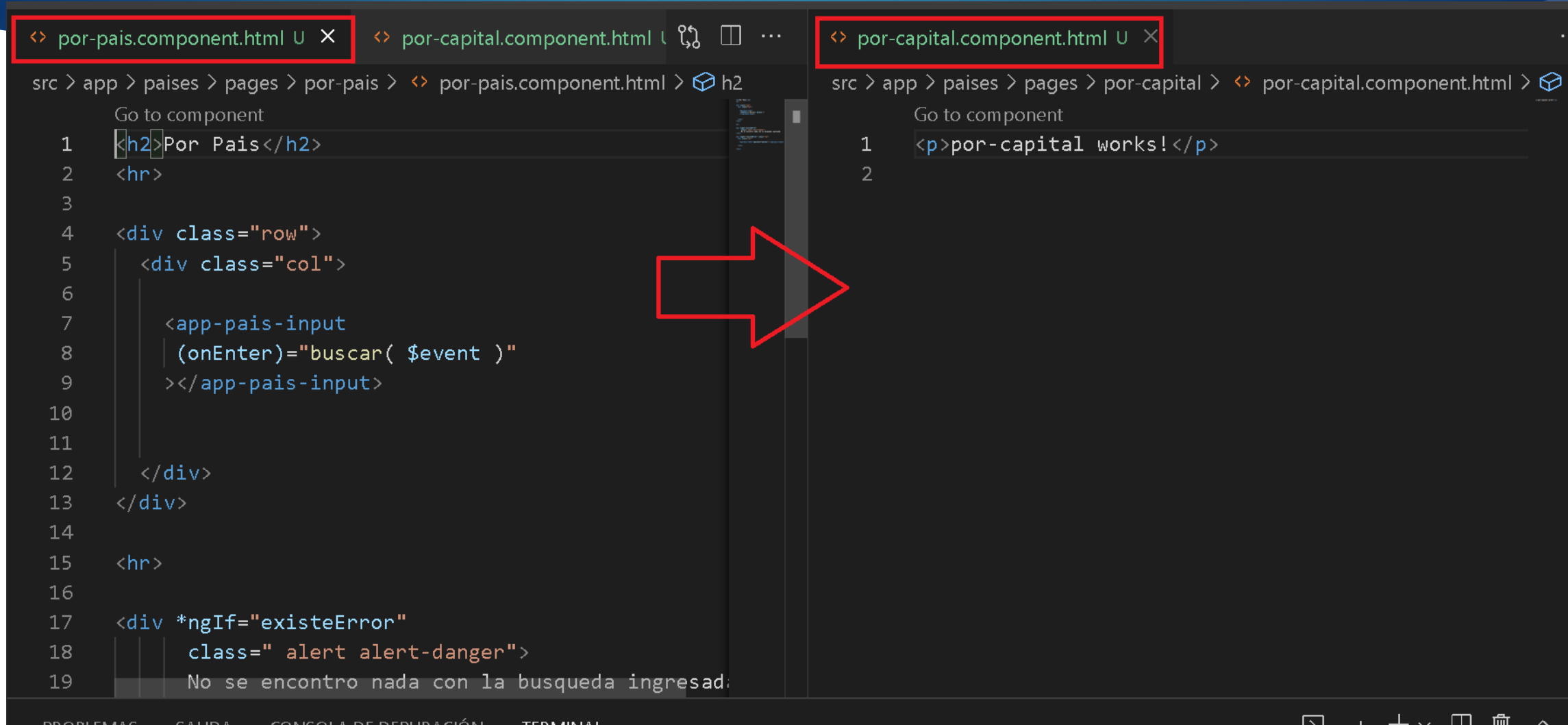
- > pais-tabla ●
- > interfaces ●
- ▼ pages ●
 - > por-capital ●
 - ▼ por-pais ●
 - <> por-pais.component.html U
 - TS por-pais.component.ts U
 - > por-region ●
 - > ver-pais ●
- ▼ services ●
 - TS pais.service.ts U
 - TS paises.module.ts U
 - > shared ●
 - TS app-routing.module.ts U
 - # app.component.css
 - <> app.component.html M

```
9 export class PaisService {
10
11     //url base
12     private apiUrl: string = 'https://restcountries.com/v3.1';
13
14     constructor(private http:HttpClient) { }
15
16
17     buscarPais(termino:string): Observable<Countries[]>{
18         const url = `${ this.apiUrl }/name/${ termino }`;
19         return this.http.get<Countries[]>(url);
20     }
21
22     buscarCapital(termino:string):Observable<Countries[]>{
23         const url = `${ this.apiUrl }/capital/${ termino }`;
24         return this.http.get<Countries[]>(url);
25     }
26
27 }
28
```

Ejercicio

- ▶ Crear la vista por capital reutilizando los componentes de tabla e input

Copiamos la misma estructura que tenemos en por países



```
<> por-pais.component.html U X <> por-capital.component.html U X <> por-capital.component.html U X
src > app > paises > pages > por-pais > <> por-pais.component.html > h2 src > app > paises > pages > por-capital > <> por-capital.component.html >
Go to component Go to component
1 <h2>Por Pais</h2> 1 <p>por-capital works!</p>
2 <hr> 2
3
4 <div class="row">
5   <div class="col">
6     <app-pais-input
7       (onEnter)="buscar( $event )"
8     ></app-pais-input>
9
10
11
12   </div>
13 </div>
14
15 <hr>
16
17 <div *ngIf="existeError"
18   class=" alert alert-danger">
19   No se encontro nada con la busqueda ingresad
```



<> por-capital.component.html 4, U X



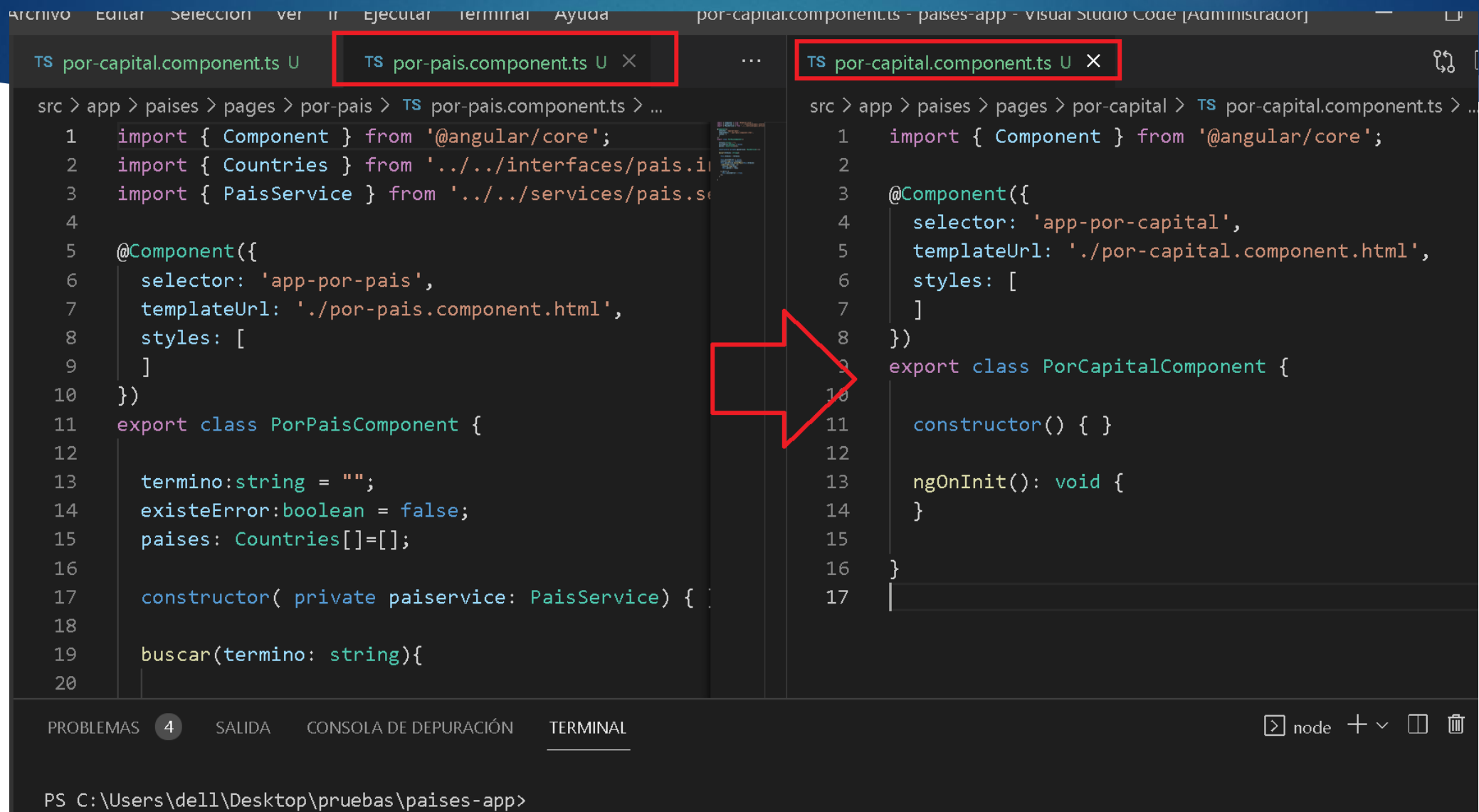
23

src > app > paises > pages > por-capital > <> por-capital.component.html >  div.row

Go to component

```
1 <h2>Por Capital</h2>
2 <hr>
3
4 <div class="row">
5   <div class="col">
6
7     <app-pais-input
8       (onEnter)="buscar( $event )"
9     ></app-pais-input>
10
11   </div>
12 </div>
13 </div>
14
15 <hr>
16
17 <div *ngIf="existeError"
18   class=" alert alert-danger">
19   No se encontro nada con la busqueda ingresada
```

También copiamos toda la estructura del ts país a capital



```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  por-capital.component.ts - paises-app - visual studio code [Administrador]
```

TS por-capital.component.ts U TS por-pais.component.ts U X TS por-capital.component.ts U X

src > app > paises > pages > por-pais > TS por-pais.component.ts > ...

```
1  import { Component } from '@angular/core';
2  import { Countries } from '../interfaces/pais.interface';
3  import { PaisService } from '../services/pais.service';
4
5  @Component({
6    selector: 'app-por-pais',
7    templateUrl: './por-pais.component.html',
8    styles: [
9    ]
10 })
11 export class PorPaisComponent {
12
13   termino:string = '';
14   existeError:boolean = false;
15   paises: Countries[]=[];
16
17   constructor( private paiservice: PaisService) {
18
19   }
20   buscar(termino: string){
```

src > app > paises > pages > por-capital > TS por-capital.component.ts > ...

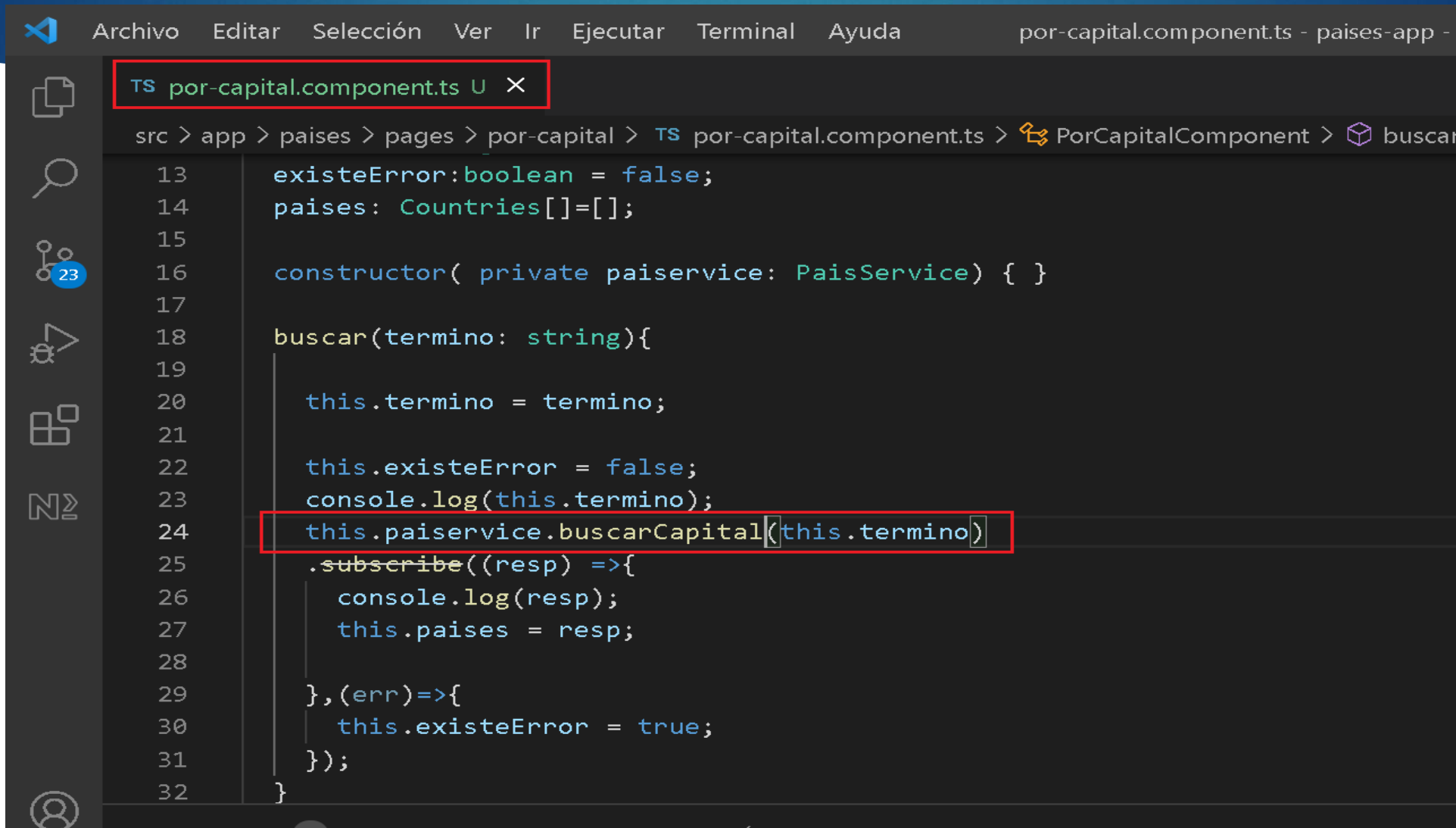
```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-por-capital',
5    templateUrl: './por-capital.component.html',
6    styles: [
7    ]
8  })
9  export class PorCapitalComponent {
10
11    constructor() { }
12
13    ngOnInit(): void {
14    }
15
16  }
17
```

PROBLEMAS 4 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

node + - [] [X]

PS C:\Users\dell\Desktop\pruebas\paises-app>

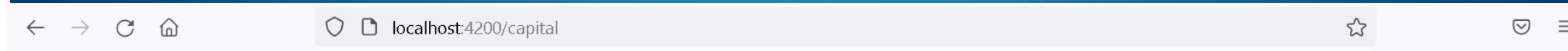
Lo único que nos toca cambiar es el método del servicio a buscarCapital



The screenshot shows a code editor with a dark theme. The top menu bar includes 'Archivo', 'Editar', 'Selección', 'Ver', 'Ir', 'Ejecutar', 'Terminal', and 'Ayuda'. The title bar of the active window reads 'por-capital.component.ts - paises-app -'. The breadcrumb navigation path is 'src > app > paises > pages > por-capital > TS por-capital.component.ts > PorCapitalComponent > buscarCapital'. The code is written in TypeScript and defines a class component. The 'constructor' takes a 'private paiservice: PaisService' as an argument. The 'buscar' method is defined with a 'termino: string' parameter. Inside 'buscar', the 'termino' is assigned to 'this.termino', 'this.existeError' is set to 'false', and 'console.log(this.termino)' is called. The core logic is on line 24, where 'this.paiservice.buscarCapital(this.termino)' is called, and this line is highlighted with a red box. This call is followed by a '.subscribe' method that handles the response and error. The code is as follows:

```
13 existeError:boolean = false;
14 paises: Countries[]=[];
15
16 constructor( private paiservice: PaisService) { }
17
18 buscar(termino: string){
19
20     this.termino = termino;
21
22     this.existeError = false;
23     console.log(this.termino);
24     this.paiservice.buscarCapital(this.termino)
25     .subscribe((resp) =>{
26         console.log(resp);
27         this.paises = resp;
28
29     },(err)=>{
30         this.existeError = true;
31     });
32 }
```

Con esto ya debería de funcionar



Busquedas

Buscar Pais

Por Región

Por Capital

Por Capital

Paris

#	Bandera	Nombre	Poblacion	
1		France	67,391,582	Ver...

Pipes

- Los *pipes* son una herramienta de Angular que nos permite **transformar** visualmente la información, por ejemplo, cambiar un texto a mayúsculas o minúsculas, o darle formato de fecha y hora.

Guía Oficial

Angular - Transforming Data Usin x +

angular.io/guide/pipes

Download Print Share Star X Home Angular Dev Community K R

ANGULAR

FEATURES DOCS RESOURCES EVENTS BLOG

Search

Twitter GitHub YouTube

Introduction

Getting Started >

Understanding Angular >

Overview

Components >

Templates >

Introduction

Text interpolation

Template statements

Pipes

Property binding

Attribute, class, and style bindings

Transforming Data Using Pipes

Use [pipes](#) to transform strings, currency amounts, dates, and other data for display. Pipes are simple functions to use in [template expressions](#) to accept an input value and return a transformed value. Pipes are useful because you can use them throughout your application, while only declaring each pipe once. For example, you would use a pipe to show a date as **April 15, 1988** rather than the raw string format.

For the sample application used in this topic, see the [live example](#) / [download example](#).

Angular provides built-in pipes for typical data transformations, including transformations for internationalization (i18n), which use locale information to format data. The following are commonly used built-in pipes for data formatting:

- Transforming Data Using Pipes
- Prerequisites
- Using a pipe in a template
- Transforming data with parameters and chained pipes
- Example: Formatting a date
- Example: Applying two formats by chaining pipes
- Creating pipes for custom data transformations
- Marking a class as a pipe
- Using the PipeTransform

Creamos un nuevo proyecto o reutilizamos alguno

- ▶ Creamos un componente para ejemplificar los pipes

uppercase y lowercase

- ▶ Se utiliza para pasar todo el parámetro a mayúscula con uppercase y minúscula con lowercase

uppercase y lowercase

TS pipes.component.ts U X

<> pipes.component.html U

src > app > pipes > TS pipes.component.ts > PipesComponent > name2

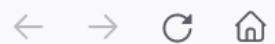
```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-pipes',
5   templateUrl: './pipes.component.html',
6   styles: [
7   ]
8 })
9 export class PipesComponent {
10   name:string = 'angular';
11
12   name2:string= 'TYPESCRIPT';
13
14 }
15
```

<> pipes.component.html U X

src > app > pipes > <> pipes.component.html > ...

Go to component

```
1 <h2>uppercase y lowercase</h2>
2 <p>{{name | uppercase}}</p>
3 <p>{{name2 | lowercase}}</p>
4
```



localhost:4200

240°

uppercase y lowercase

ANGULAR

typescript

slice

- ▶ El *slice* es un *pipe* que requiere mínimo un parámetro.
- ▶ El parámetro que agreguemos después de los dos puntos (un número) será la cantidad de elementos que **eliminará** de nuestra variable, empezando por el principio

Slice

TS pipes.component.ts U X <> pipes.component.html U

src > app > pipes > TS pipes.component.ts > PipesComponent

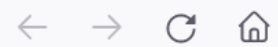
```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-pipes',
5    templateUrl: './pipes.component.html',
6    styles: [
7    ]
8  })
9  export class PipesComponent {
10    name:string = 'angular';
11    name2:string= 'TYPESCRIPT';
12
13
14  }
15
```

<> pipes.component.html U X

src > app > pipes > <> pipes.component.html > p

Go to component

```
1  <h2>uppercase y lowercase</h2>
2  <p>{{name | uppercase}}</p>
3  <p>{{name2 | lowercase}}</p>
4
5
6  <h2>slice</h2>
7  <p>{{name | slice:2}}</p>
8  <p>{{name2 | slice:0:2}}</p>
9
```



localhost:4200

uppercase y lowercase

ANGULAR

typescript

slice

gular

TY

decimal

- ▶ Podemos utilizar este *pipe* para trabajar con números y decimales

decimal

TS pipes.component.ts U X

<> pipes.component.html U



src > app > pipes > TS pipes.component.ts > PipesComponent

```
7   ]
8   })
9   export class PipesComponent {
10     name:string = 'angular';
11     name2:string= 'TYPESCRIPT';
12
13     //sintaxis {{valorpi | number:N1.N2-N3}}
14     //    N1 = 1  cantidad de números enteros
15     //N2 = 0 cantidad mínima de números decimales
16     //N3 = 3 cantidad máxima de números decimales
17     valorpi:number=3.141592;
18
19
20   }
21
```

<> pipes.component.html U X

src > app > pipes > <> pipes.component.html > p

Go to component

```
1   <h2>uppercase y lowercase</h2>
2   <p>{{name | uppercase}}</p>
3   <p>{{name2 | lowercase}}</p>
4
5
6   <h2>slice</h2>
7   <p>{{name | slice:2}}</p>
8   <p>{{name2 | slice:0:2}}</p>
9
10  <h2>decimal</h2>
11  <p>{{valorpi | number:'1.0-2' }}</p>
12
13
```



localhost:4200

150%



uppercase y lowercase

ANGULAR

typescript

slice

gular

TY

decimal

3.14

Percent

- ▶ Utilizamos este *pipe* para mostrar números en forma de porcentaje

Percent

TS pipes.component.ts U X <> pipes.component.html U

src > app > pipes > TS pipes.component.ts > PipesComponent

```
11 name2:string= 'TYPESCRIPT';
12
13 //sintaxis {{valorpi | number:N1.N2-N3}}
14 //    N1 = 1  cantidad de números enteros
15 //N2 = 0 cantidad mínima de números decimales
16 //N3 = 3 cantidad máxima de números decimales
17 valorpi:number=3.141592;
18
19 // {{ myNum | percent: '(string con 3 valores numéricos)
20 // N1.N2-N3' }}
21 //N1 = 1 --> cantidad de números enteros (valor por defecto)
22 //N2 = 0 --> cantidad mínima de números decimales (valor por defecto)
23 //N3 = 0 --> cantidad máxima de números decimales (valor por defecto)
24 myNum = 0.589;
25
26
27
```

<> pipes.component.html U X

src > app > pipes > <> pipes.component.html > p

```
10 <h2>decimal</h2>
11 <p>{{valorpi | number:'1.0-2' }}</p>
12
13
14 <h2>percent</h2>
15 <p>{{myNum | percent }}</p>
16 <p>{{myNum | percent:'1.0-2' }}</p>
17
18
```




localhost:4200

150%



ANGULAR

typescript

slice

gular

TY

decimal

3.14

percent

59%

58.9%

currency

- ▶ Este *pipe* se utiliza cuando queremos mostrar números acompañados de una **divisa** (euros, dólares, yenes, etc)

currency

TS pipes.component.ts U x <> pipes.component.html U

src > app > pipes > TS pipes.component.ts > PipesComponent

```
11 name2:string= 'TYPESCRIPT';
12
13 //sintaxis {{valorpi | number:N1.N2-N3}}
14 //    N1 = 1  cantidad de números enteros
15 //N2 = 0 cantidad mínima de números decimales
16 //N3 = 3 cantidad máxima de números decimales
17 valorpi:number=3.141592;
18
19 // {{ myNum | percent: '(string con 3 valores numéricos)
20 // N1.N2-N3' }}
21 //N1 = 1 --> cantidad de números enteros (valor por defecto)
22 //N2 = 0 --> cantidad mínima de números decimales (valor por defecto)
23 //N3 = 0 --> cantidad máxima de números decimales (valor por defecto)
24 myNum = 0.589;
25
26 //{{myVariable|currency:'currencyCharacter':'symbol/code':N1.N2-N3'}}
27 salary = 3500.5;
28
29
```

<> pipes.component.html U x

src > app > pipes > <> pipes.component.html > p

```
10 <h2>decimal</h2>
11 <p>{{valorpi | number:'1.0-2' }}</p>
12
13
14 <h2>percent</h2>
15 <p>{{myNum | percent }}</p>
16 <p>{{myNum | percent:'1.0-2' }}</p>
17
18 <h2>currency</h2>
19 <p>{{salary | currency }}</p>
20 <p>{{salary | currency:'currencyCharacter':'€' }}</p>
21
```



localhost:4200

150% ☆

gular

TY

decimal

3.14

percent

59%

58.9%

currency

\$3,500.50

€3,500.50

json

- El *pipe* *JSON* es muy útil combinado con la etiqueta **<pre>**, cuando queremos mostrar código en formato JSON. Es especialmente útil cuando queremos mostrar objetos en el navegador

TS pipes.component.ts U x <> pipes.component.html U

src > app > pipes > TS pipes.component.ts > PipesComponent > personaje

```
25 //{{myVariable|currency:'currencyCharacter':'symbol/code':N1.N2-N3}}
26 salary = 3500.5;
27
28
29 personaje= {
30   name: 'Mr. Bean',
31   alias: 'Bean',
32   song: 'Toxicity',
33   skills: ['eidetic memory', 'makes people nervous'],
34   youtubeChannel: 'Fun With Flags',
35   address: {
36     street: 'Elm Street',
37     number: 3,
38     city: 'Liverpool'
39   }
40 };
41 }
42
```

<> pipes.component.html U x

src > app > pipes > <> pipes.component.html > ...

```
10 <h2>decimal</h2>
11 <p>{{valorpi | number:'1.0-2' }}</p>
12
13
14 <h2>percent</h2>
15 <p>{{myNum | percent }}</p>
16 <p>{{myNum | percent:'1.0-2' }}</p>
17
18 <h2>currency</h2>
19 <p>{{salary | currency }}</p>
20 <p>{{salary | currency:'currencyCharacter':'€' }}</p>
21
22
23 <h2>json</h2>
24 <p>{{personaje | json }}</p>
25
```



localhost:4200

150%



3.14

percent

59%

58.9%

currency

\$3,500.50

€3,500.50

json

```
{ "name": "Mr. Bean", "alias": "Bean", "song": "Toxicity", "skills": [ "eidetic memory", "makes people nervous" ],  
  "youtubeChannel": "Fun With Flags", "address": { "street": "Elm Street", "number": 3, "city": "Liverpool" } }
```

async

- ▶ Este *pipe* nos permite mostrar información proveniente de código basado en ***promises***, entre otras cosas

TS pipes.component.ts U X

<> pipes.component.html U

pipes > TS pipes.component.ts > PipesComponent > promiseValue > <function> > setTimeout() callback

```
31   alias: 'Bean',
32   song: 'Toxicity',
33   skills: ['eidetic memory', 'makes people nervous'],
34   youtubeChannel: 'Fun With Flags',
35   address: {
36     street: 'Elm Street',
37     number: 3,
38     city: 'Liverpool'
39   }
40 };
41
```

```
42 promiseValue = new Promise((resolve, reject) => {
43   setTimeout(() => {
44     resolve('Esto funciona');
45   }, 3000);
46 });
47 }
48
```

<> pipes.component.html U X

src > app > pipes > pipes.component.html > pre

```
10 <h2>decimal</h2>
11 <p>{{valorpi | number:'1.0-2' }}</p>
12
13
14 <h2>percent</h2>
15 <p>{{myNum | percent }}</p>
16 <p>{{myNum | percent:'1.0-2' }}</p>
17
18 <h2>currency</h2>
19 <p>{{salary | currency }}</p>
20 <p>{{salary | currency:'currencyCharacter':'€' }}</p>
21
22
23 <h2>json</h2>
24 <p>{{personaje | json }}</p>
25
26 <h2>async</h2>
27 <pre>{{promiseValue | async }}</pre>
28
```



localhost:4200

150%



59%

58.9%

currency

\$3,500.50

€3,500.50

json

```
{ "name": "Mr. Bean", "alias": "Bean", "song": "Toxicity", "skills": [ "eidetic memory", "makes people nervous" ],  
  "youtubeChannel": "Fun With Flags", "address": { "street": "Elm Street", "number": 3, "city": "Liverpool" } }
```

async

Esto funciona



date

- ▶ Este *pipe* nos permite mostrar fechas de manera más legible, admitiendo múltiples formatos.

TS pipes.component.ts U X

<> pipes.component.html U

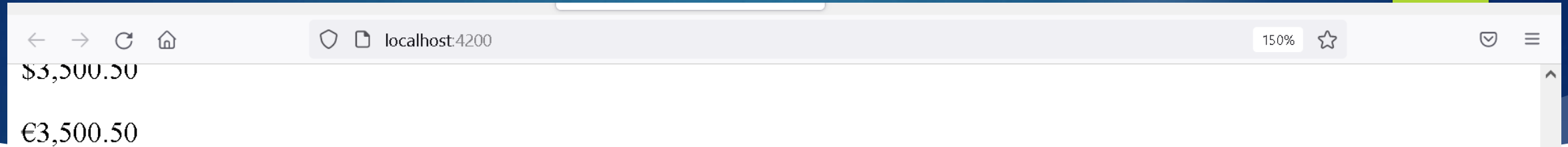
src > app > pipes > TS pipes.component.ts > PipesComponent > myDate

```
40   };
41
42   promiseValue = new Promise((resolve, reject) => {
43     setTimeout(() => {
44       resolve('Esto funciona');
45     }, 3000);
46   });
47
48   myDate = new Date();
49 }
50
```

<> pipes.component.html U X

src > app > pipes > <> pipes.component.html > pre

```
19 <p>{{salary | currency }}</p>
20 <p>{{salary | currency:'currencyCharacter':'€' }}</p>
21
22
23 <h2>json</h2>
24 <p>{{personaje | json }}</p>
25
26 <h2>async</h2>
27 <pre>{{promiseValue | async }}</pre>
28
29 <h2>date</h2>
30 <p>{{myDate }}</p>
31 <p>{{myDate | date }}</p>
32 <pre>{{ myDate | date:'EEE, d of LLL of yyyy' }} </pre>
33
```



json

```
{ "name": "Mr. Bean", "alias": "Bean", "song": "Toxicity", "skills": [ "eidetic memory", "makes people nervous" ],  
  "youtubeChannel": "Fun With Flags", "address": { "street": "Elm Street", "number": 3, "city": "Liverpool" } }
```

async

Esto funciona

date

Wed Dec 15 2021 06:24:42 GMT+0100 (hora estándar de Europa central)

Dec 15, 2021

Wed, 15 of Dec of 2021