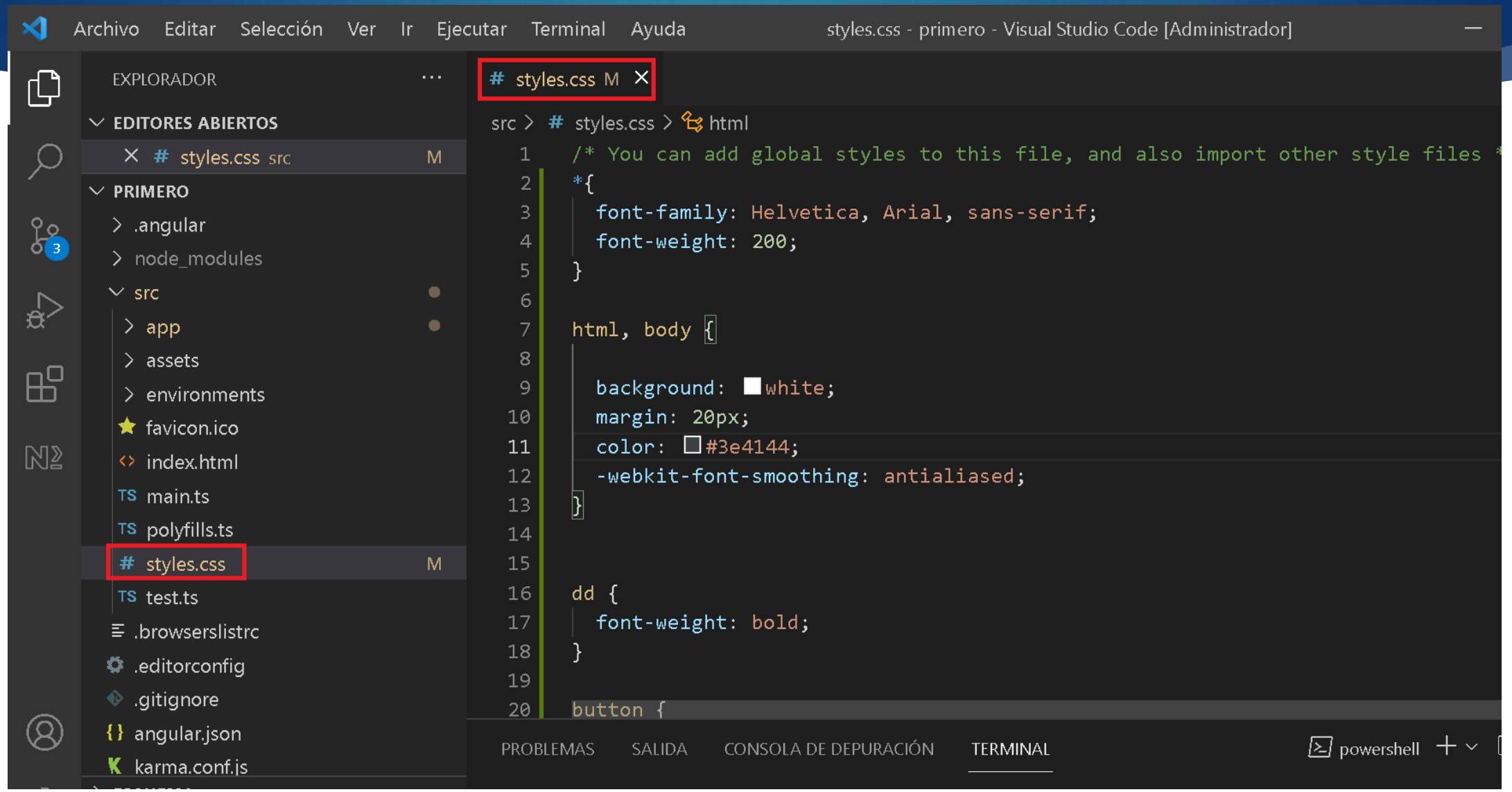
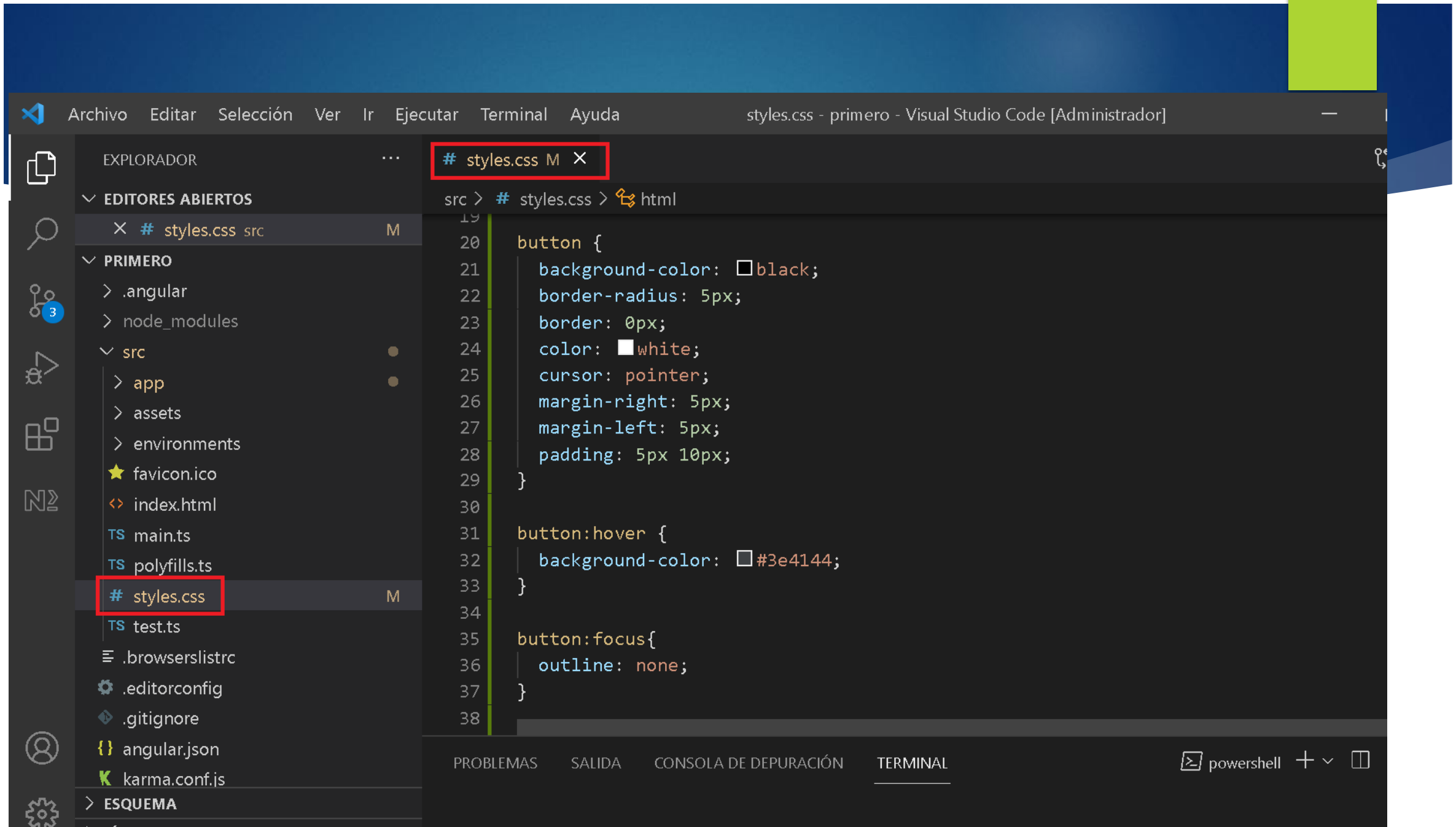
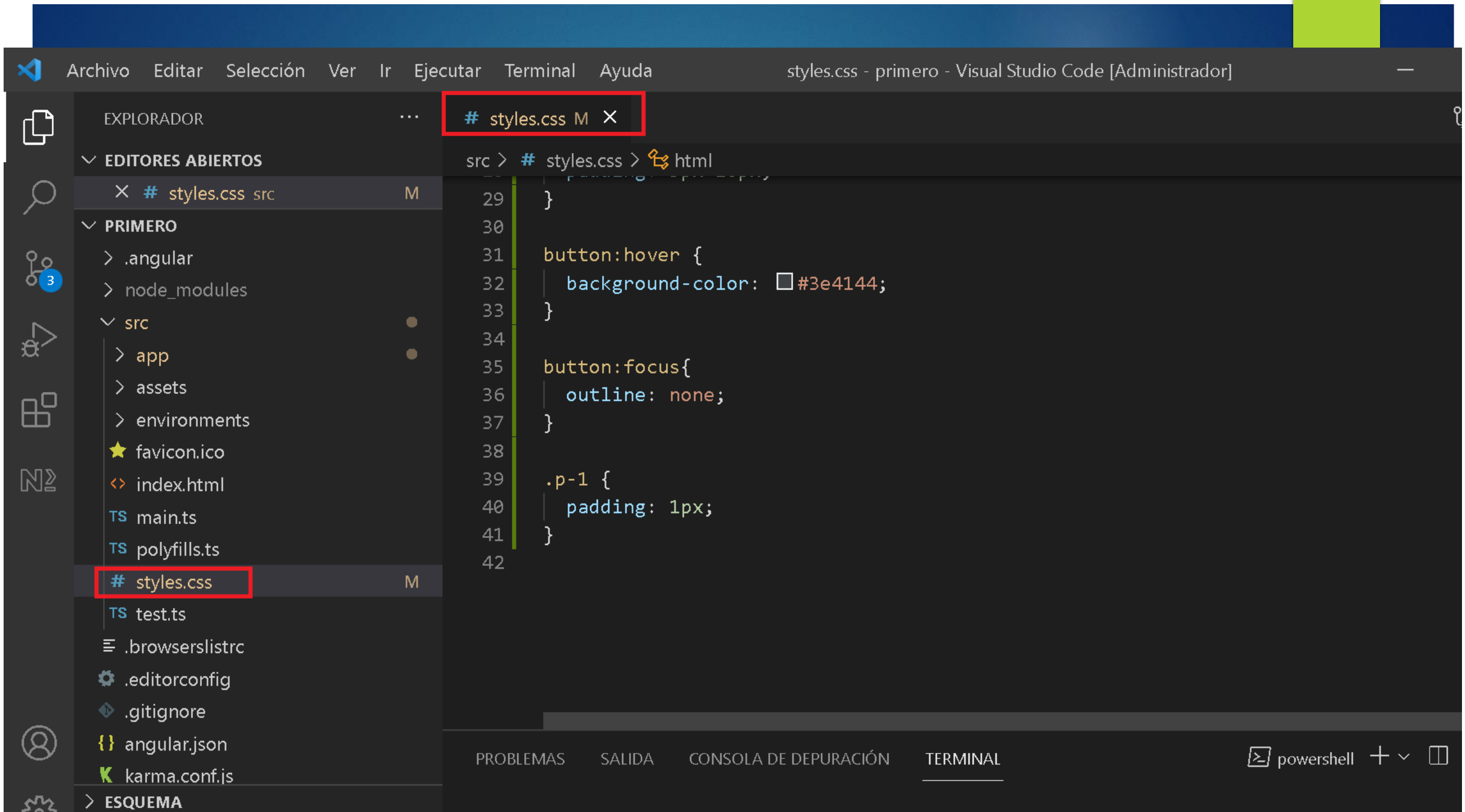
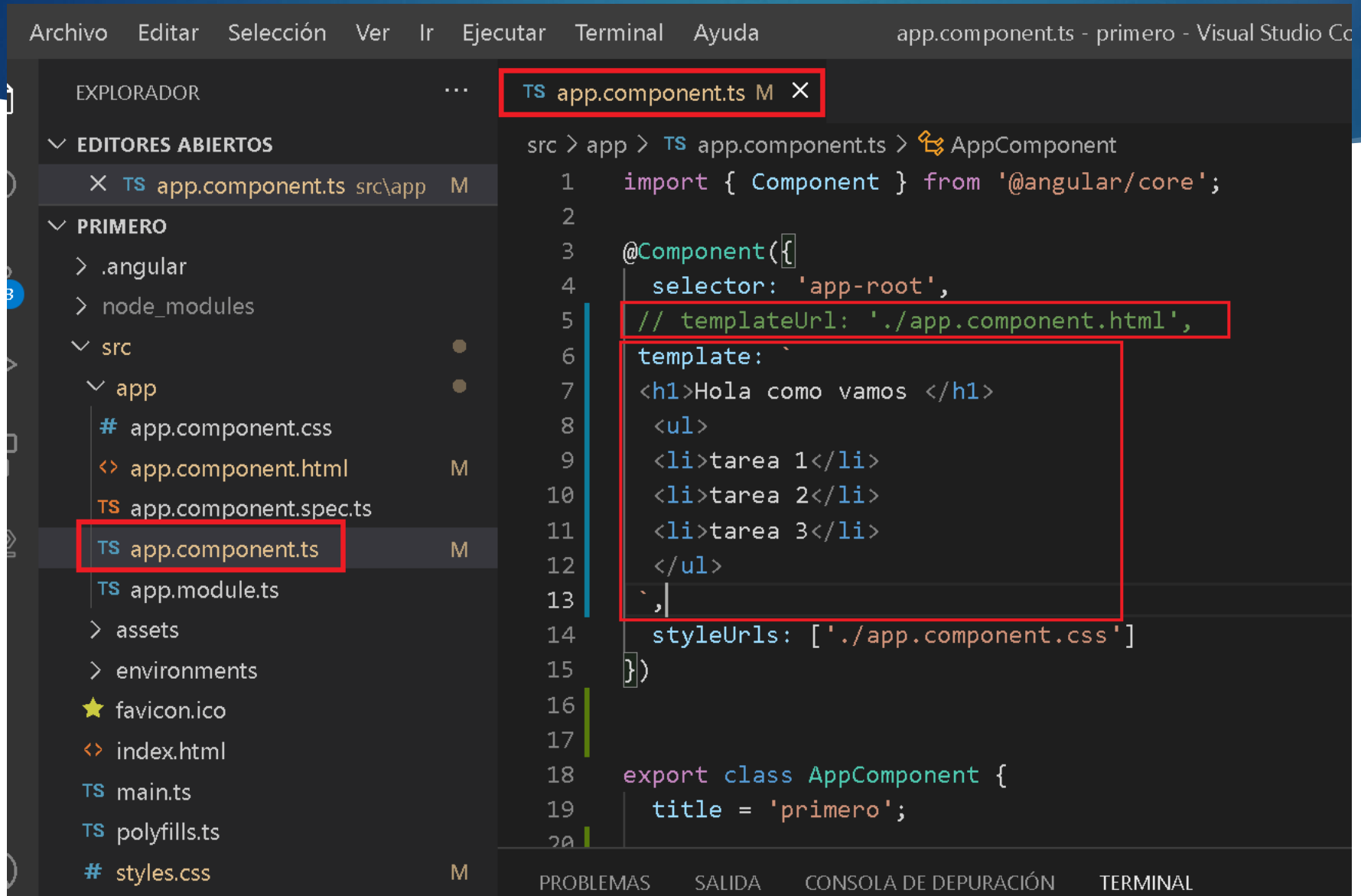
En src/styles.css aplicamos estilos globales



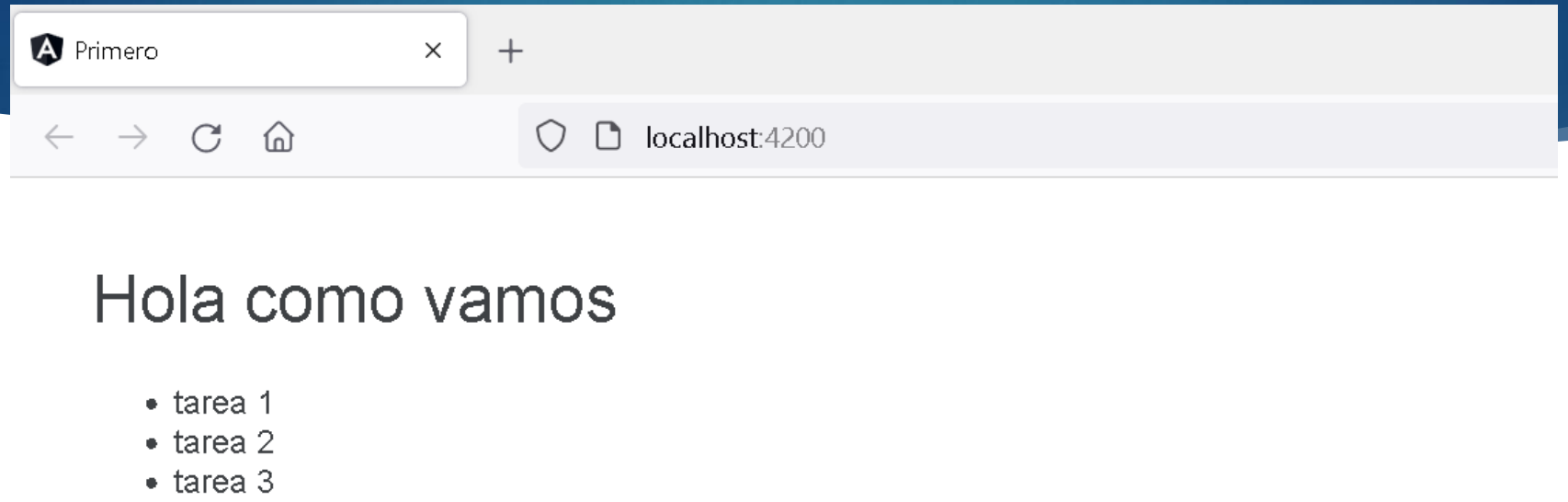

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
styles.css - primero - Visual Studio Code [Administrador]
EXPLORADOR
EDITORES ABIERTOS
styles.css src M
PRIMERO
.angular
node_modules
src
app
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css M
test.ts
.browserslistrc
.editorconfig
.gitignore
angular.json
karma.conf.js
ESQUEMA
styles.css M
src > styles.css > html
button {
background-color: black;
border-radius: 5px;
border: 0px;
color: white;
cursor: pointer;
margin-right: 5px;
margin-left: 5px;
padding: 5px 10px;
}
button:hover {
background-color: #3e4144;
}
button:focus{
outline: none;
}
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
powershell


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
styles.css - primero - Visual Studio Code [Administrador]
EXPLORADOR
EDITORES ABIERTOS
styles.css src M
PRIMERO
.angular
node_modules
src
app
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css M
test.ts
.browserslistrc
.editorconfig
.gitignore
angular.json
karma.conf.js
ESQUEMA
styles.css M
src > styles.css > html
29 }
30
31 button:hover {
32 background-color: #3e4144;
33 }
34
35 button:focus{
36 outline: none;
37 }
38
39 .p-1 {
40 padding: 1px;
41 }
42
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
powershell

En app.component.ts



Comprobamos que podemos tener definida el html dentro del ts



Creo un ejemplo de contador

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda app.component.html - primero - Visual Studio Code [Administrador]

app.component.ts M | app.component.html M

src > app > app.component.ts > AppComponent > titulo

```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})


export class AppComponent {
  titulo = 'Contador';


}

```

app.component.html M

src > app > app.component.html > span

Go to component

```html
<h1>
  {{titulo}}

</h1>

<span> 0 </span>

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda app.component.html - primero - Visual Studio Code [Administrador]

app.component.ts M | app.component.html M

src > app > app.component.ts > AppComponent > numero

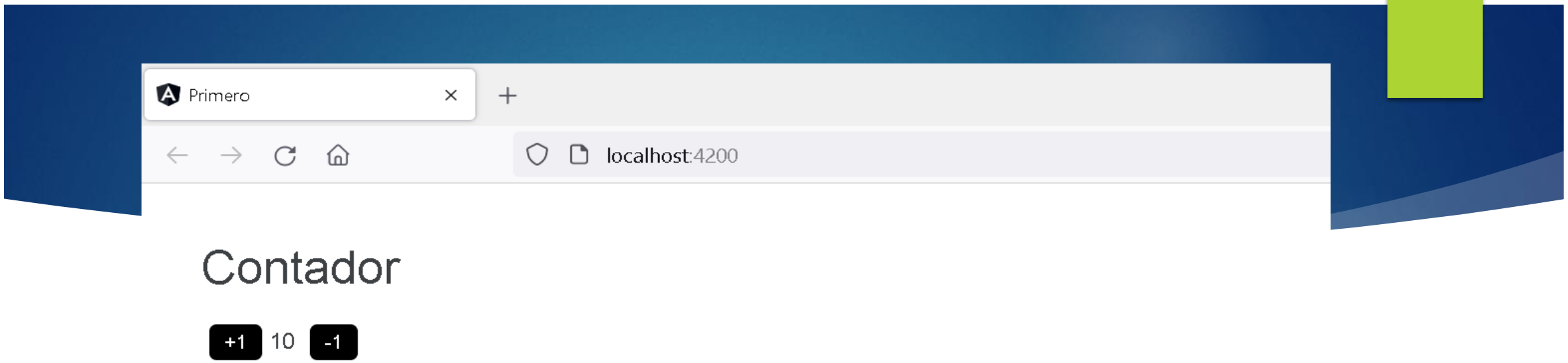
```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})


export class AppComponent {
  titulo:string = 'Contador';
  numero:number = 10;

}

```

app.component.html M

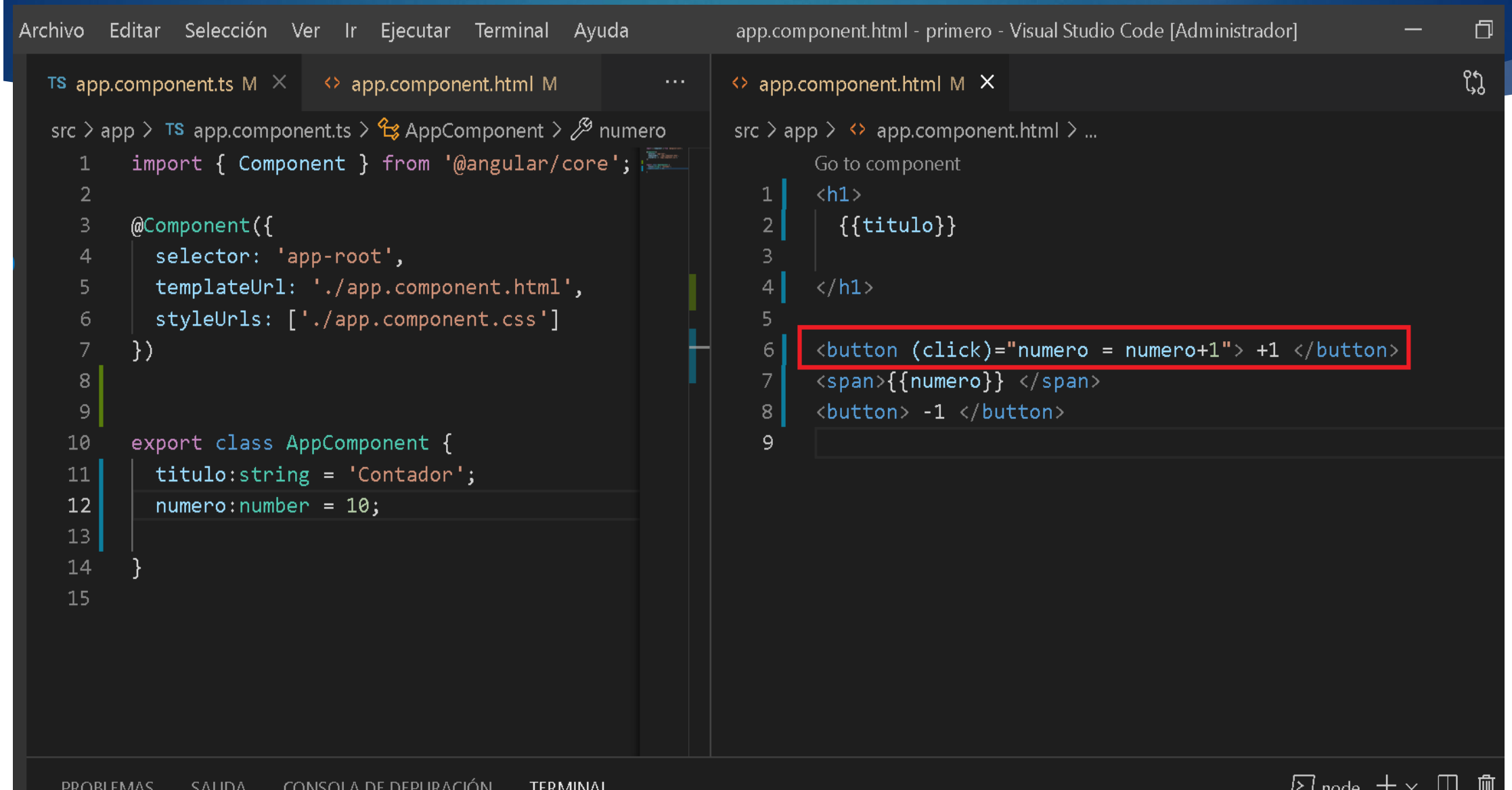
src > app > app.component.html > button

Go to component

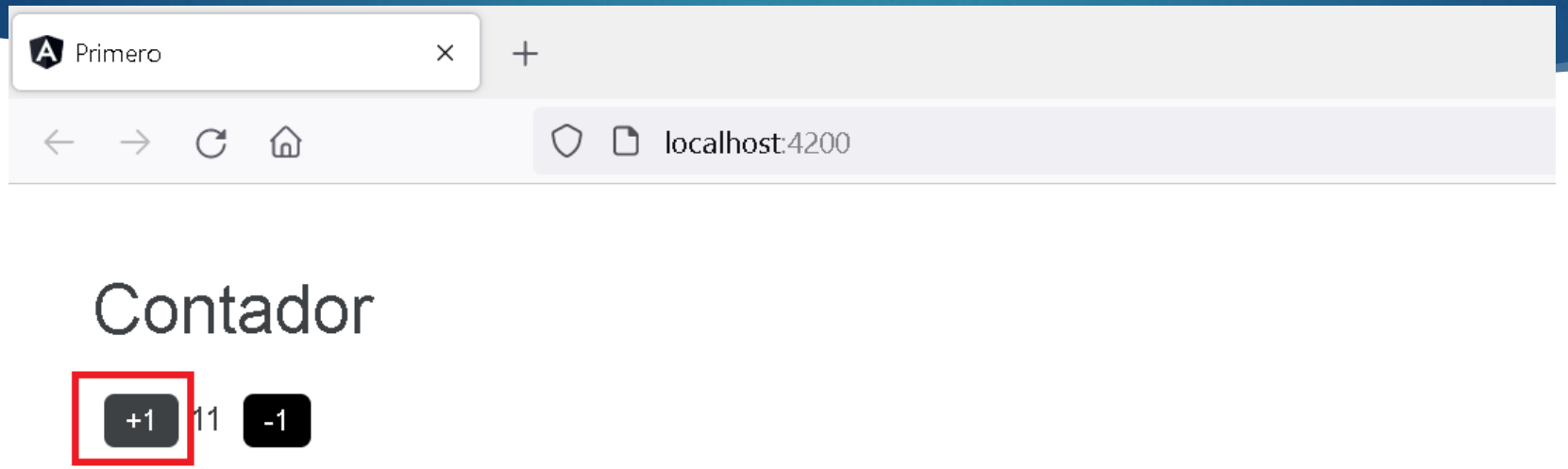
```html
<h1>
  {{titulo}}

</h1>

<button> +1 </button>
<span>{{numero}} </span>
<button> -1 </button>

```


Primero
localhost:4200
Contador
+1
10
-1

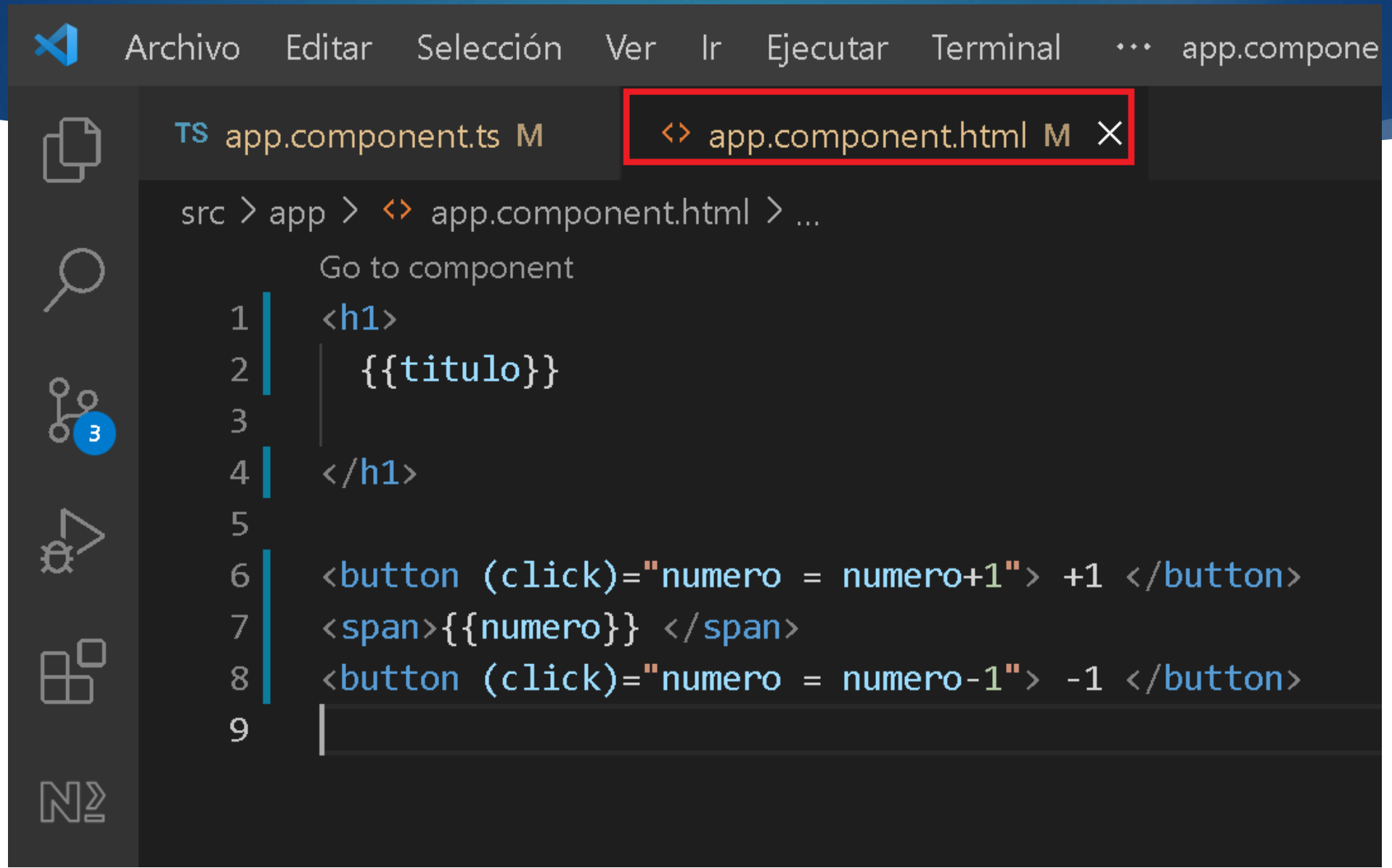
Agregamos un evento al botón



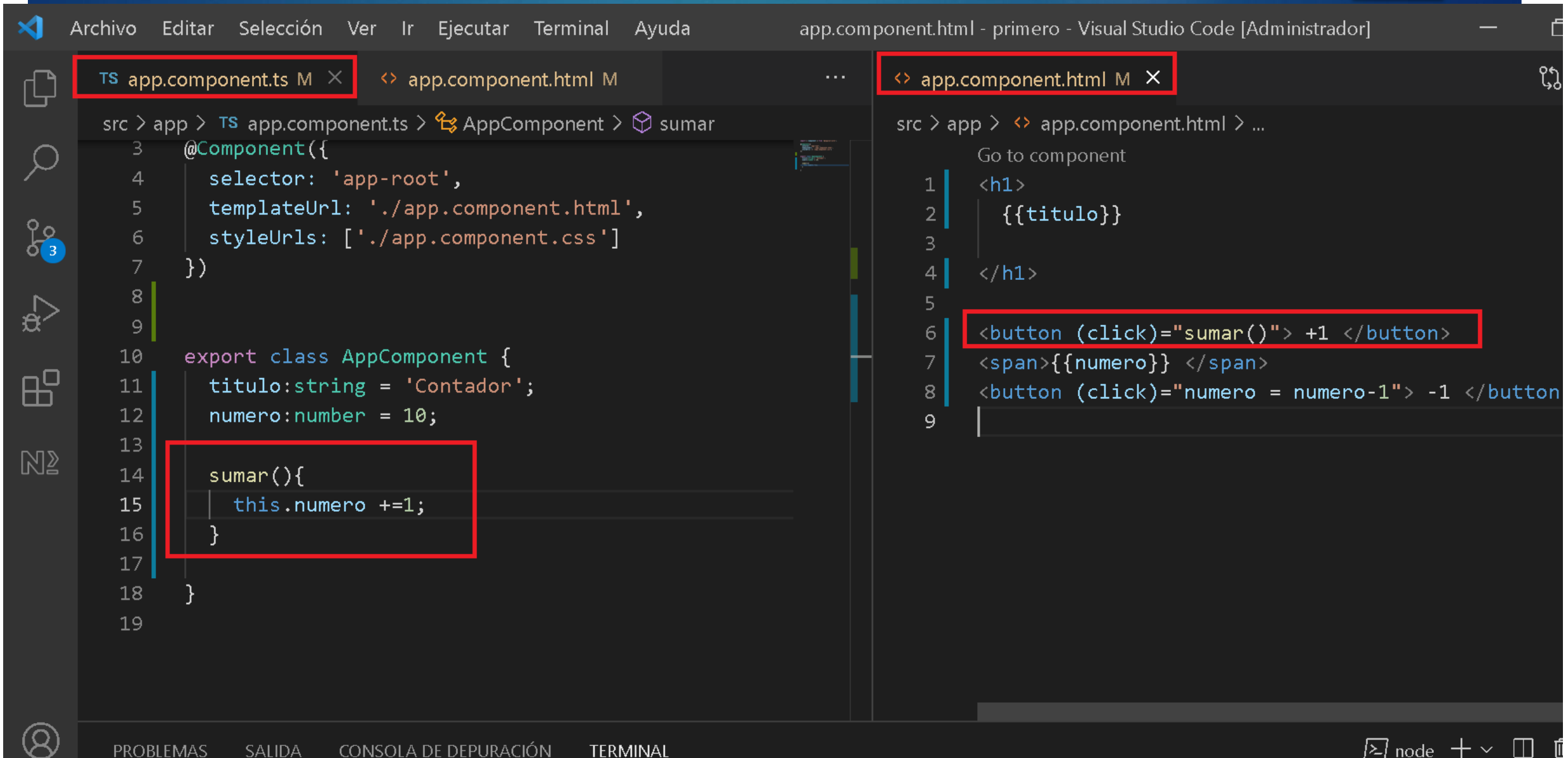
Nos debería de aumentar en 1 el contador

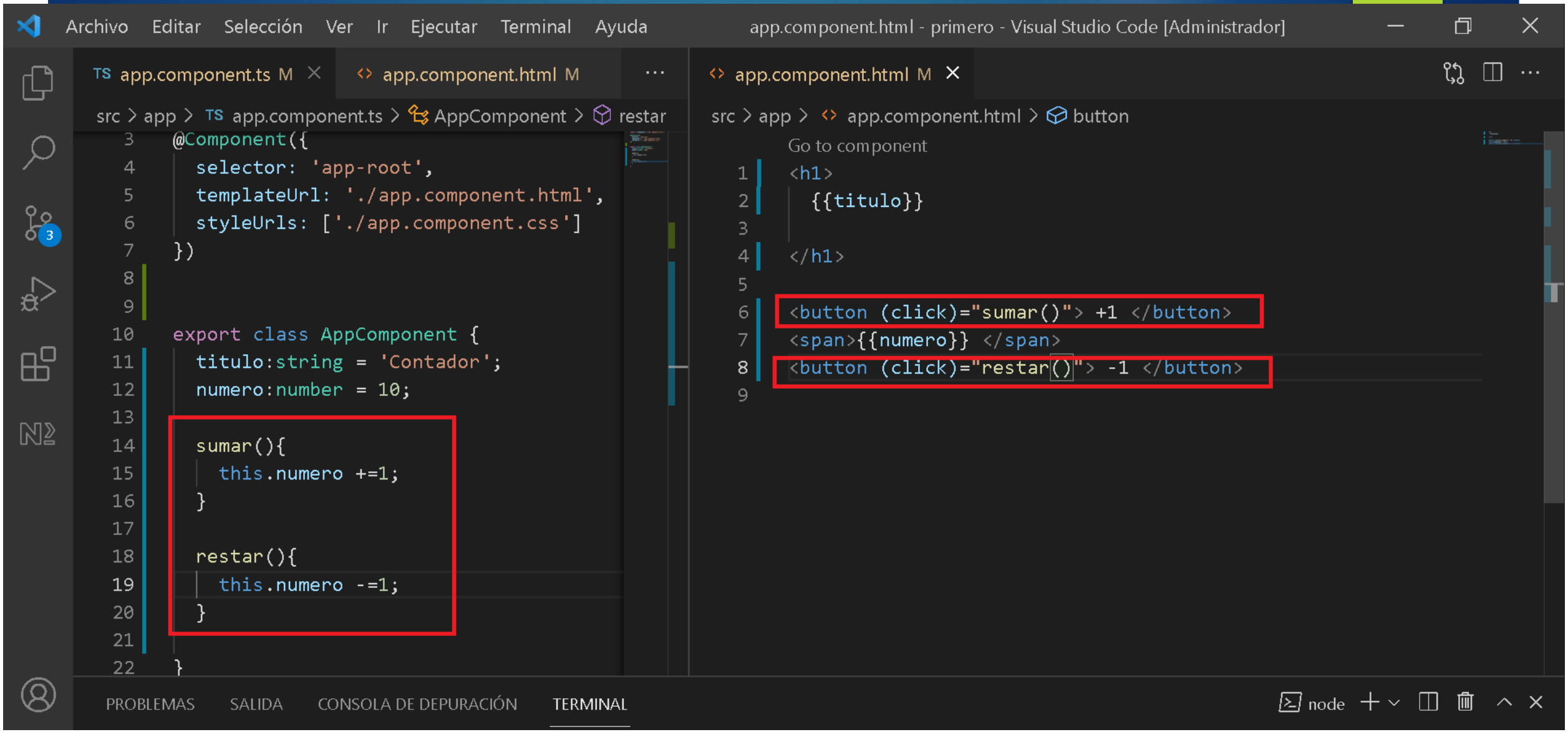


Así quedaría agregando funcionalidad al botón -1

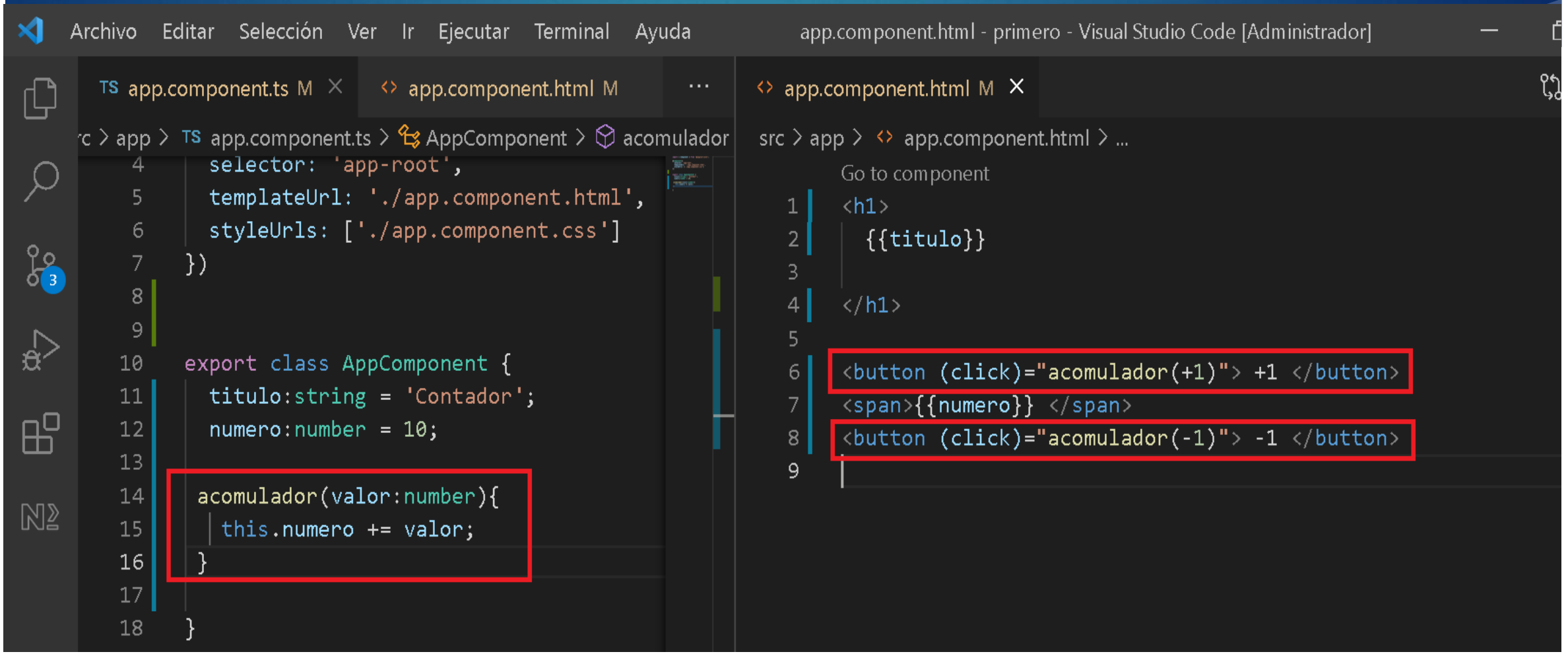


Mejoramos nuestro código




Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
app.component.html - primero - Visual Studio Code [Administrador]
app.component.ts M
app.component.html M
src > app > app.component.ts > AppComponent > restar
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
titulo:string = 'Contador';
numero:number = 10;
sumar(){
this.numero +=1;
}
restar(){
this.numero -=1;
}
}
app.component.html M
src > app > app.component.html > button
Go to component
<h1>
{{titulo}}
</h1>
<button (click)="sumar()"> +1 </button>
{{numero}}
<button (click)="restar()"> -1 </button>
PROBLEMAS
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
node

Optimizamos aun mas



chivo Editar Selección Ver Ir Ejecutar Terminal Ayuda app.component.html - primero - Visual Studio Code [Administrador]

app.component.ts M app.component.html M

src > app > app.component.ts > AppComponent > base

```
4     selector: 'app-root',
5     templateUrl: './app.component.html',
6     styleUrls: ['./app.component.css']
7   })
8
9
10  export class AppComponent {
11    titulo:string = 'Contador';
12    numero:number = 10;
13    base:number = 5;
14
15    acomulador(valor:number){
16      this.numero += valor;
17    }
18
19  }
20
```

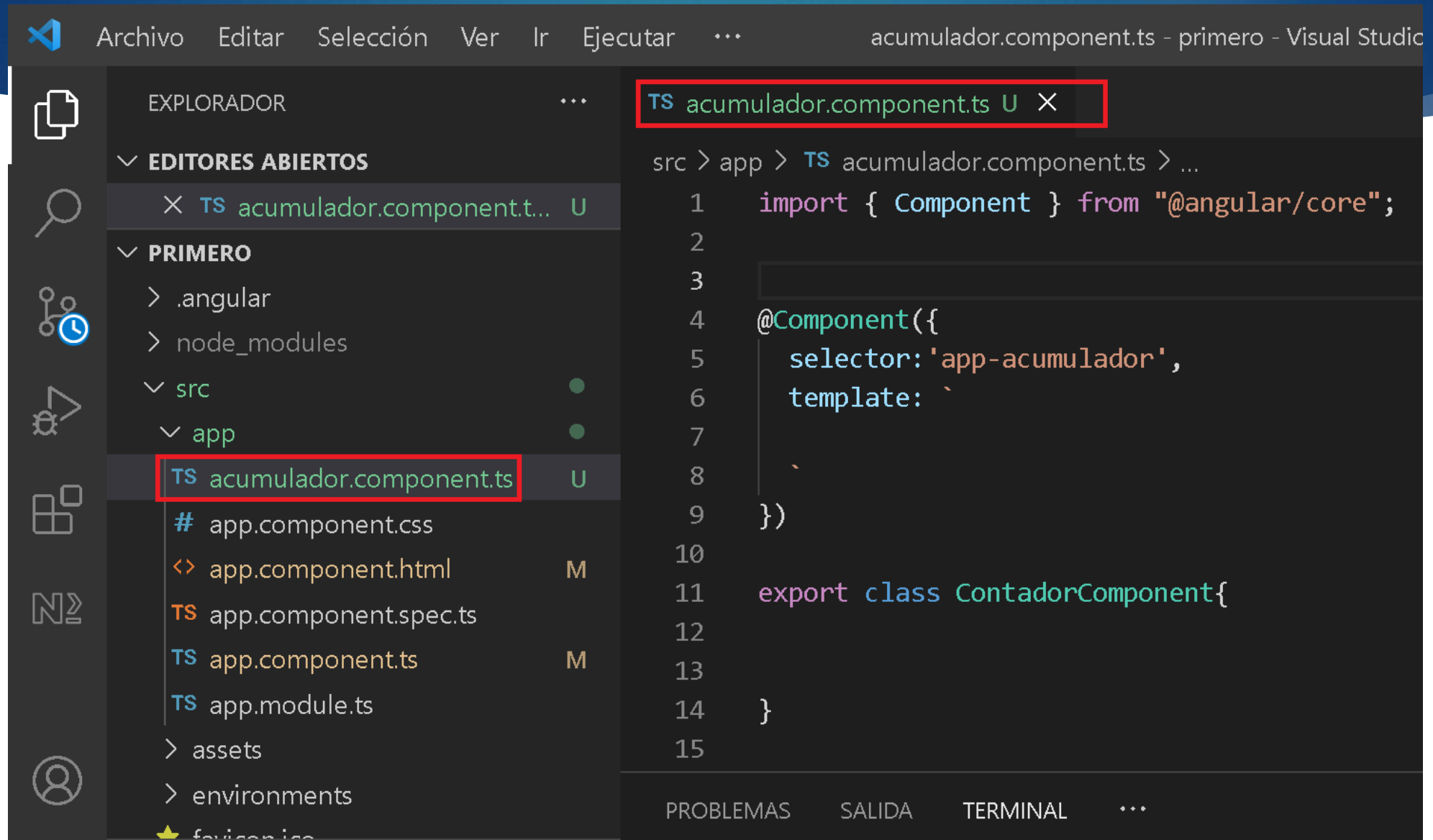
app.component.html M

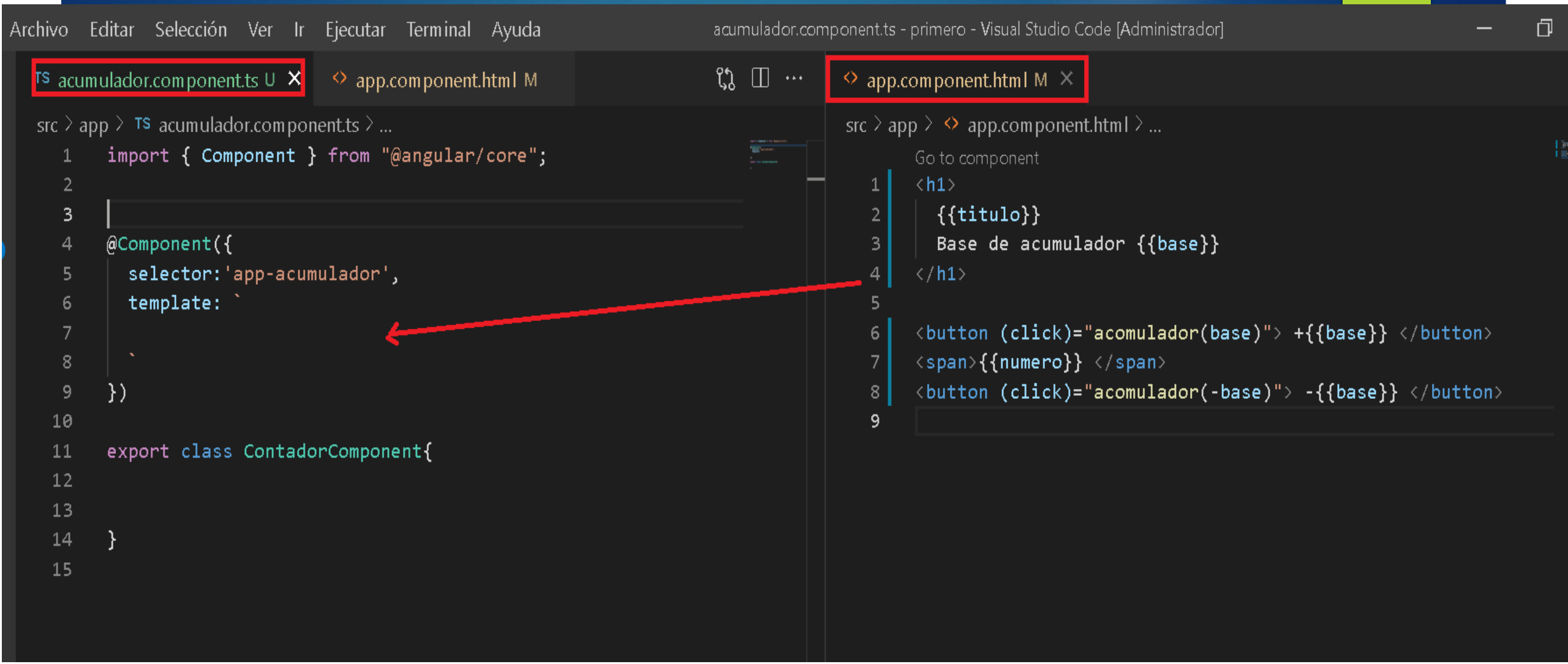
src > app > app.component.html > ...

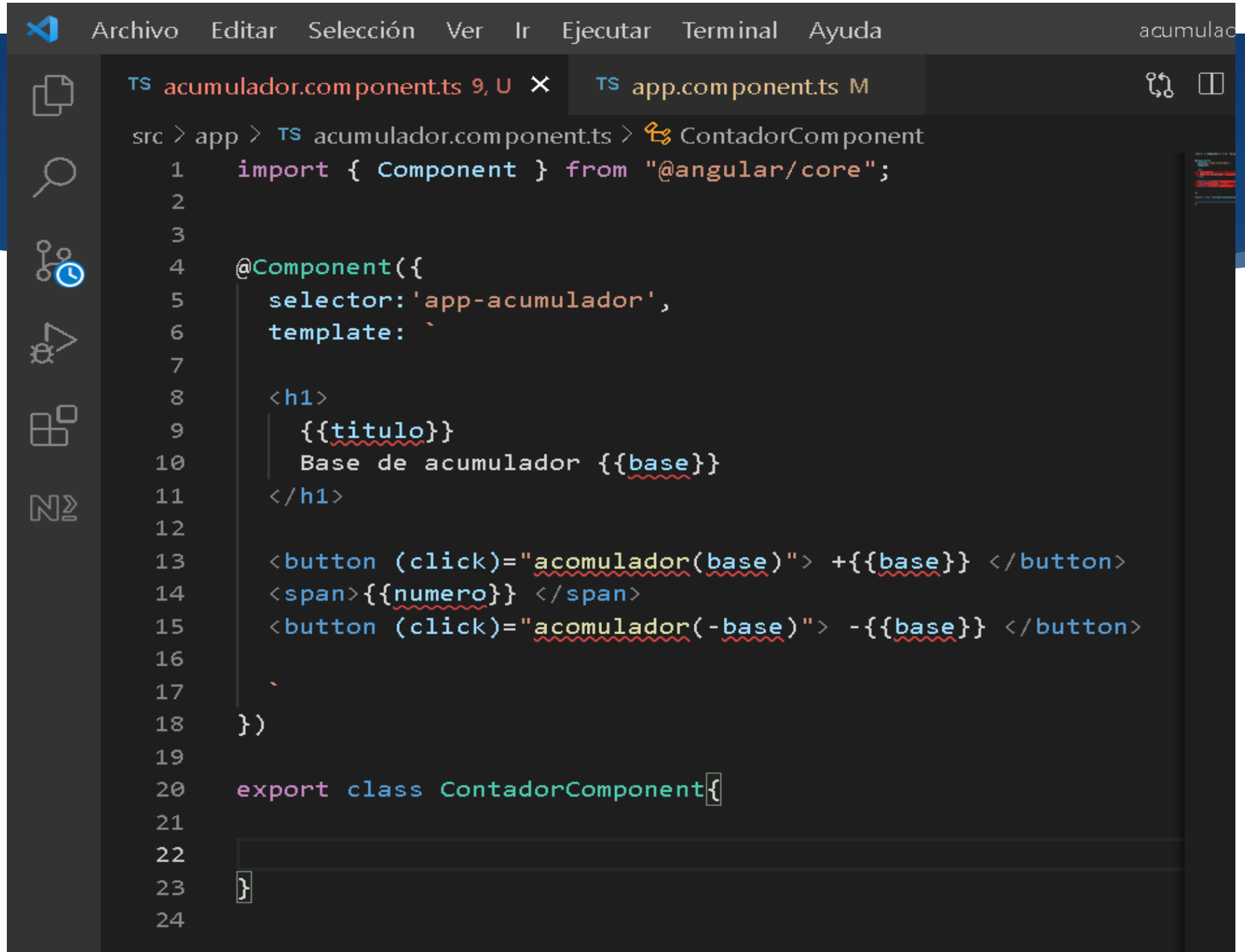
Go to component

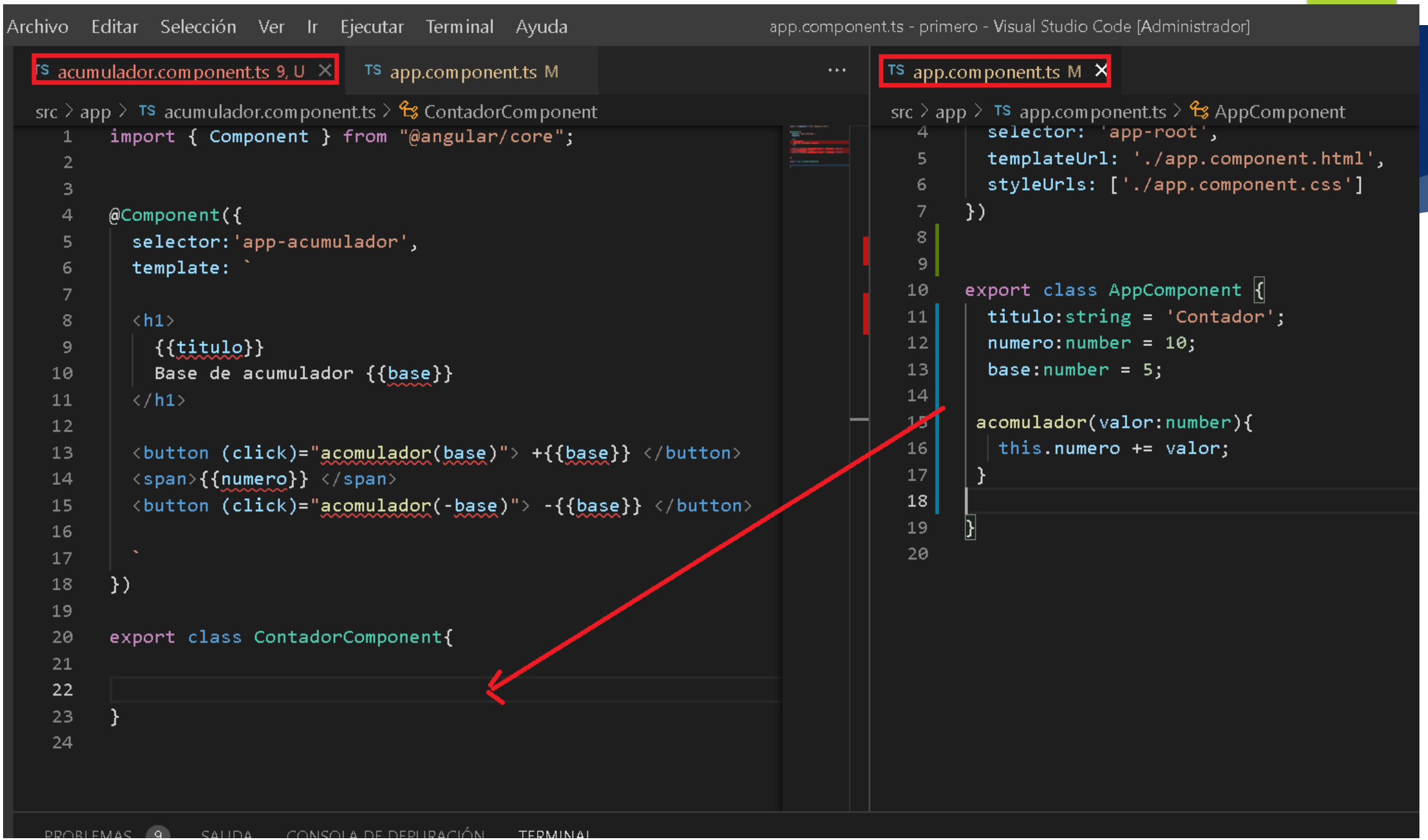
```
1  <h1>
2    {{titulo}}
3    Base de acumulador {{base}}
4  </h1>
5
6  <button (click)="acomulador(base)"> +{{base}} </button>
7  <span>{{numero}} </span>
8  <button (click)="acomulador(-base)"> -{{base}} </button>
9
```

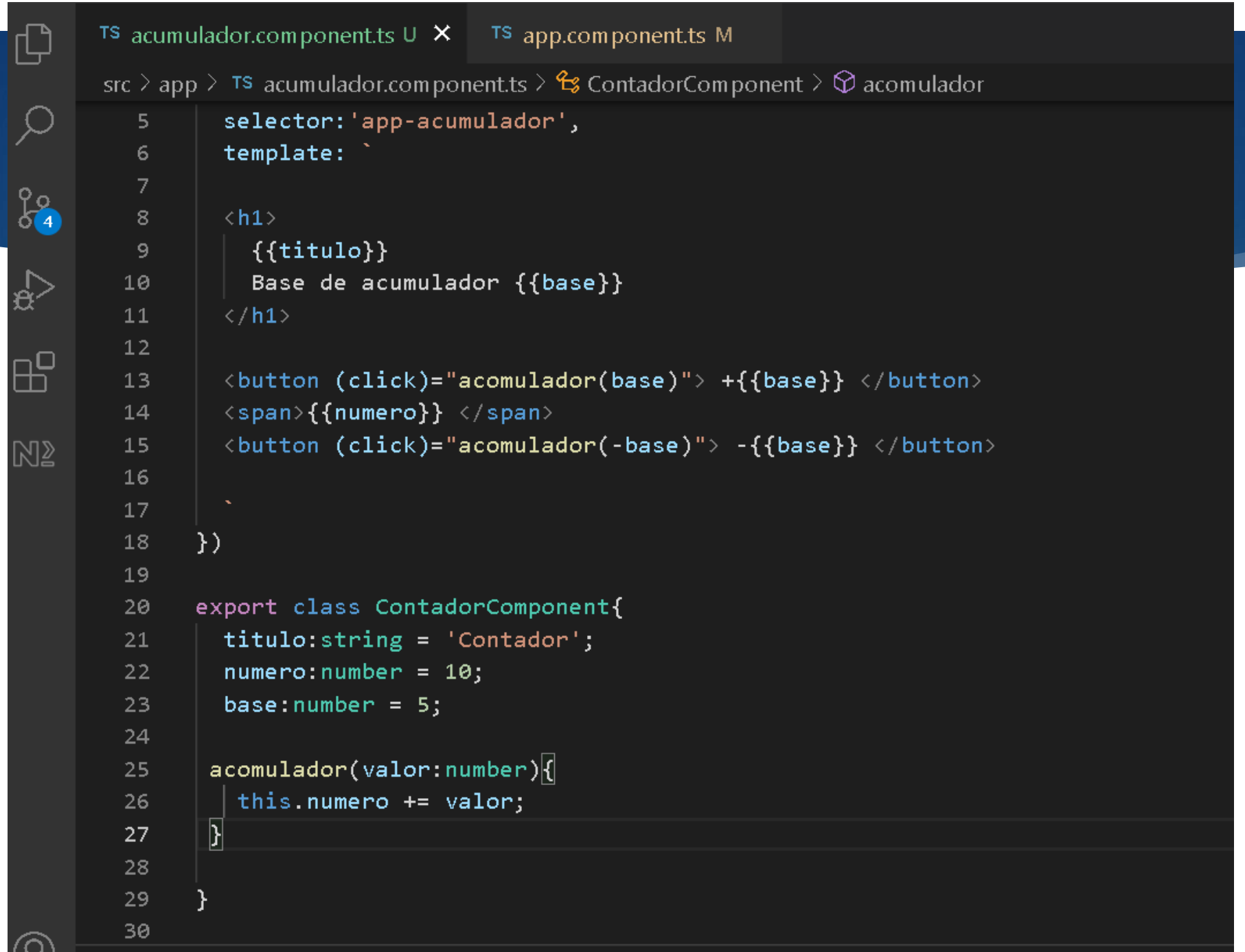
Pasamos el Acumulador a un componente



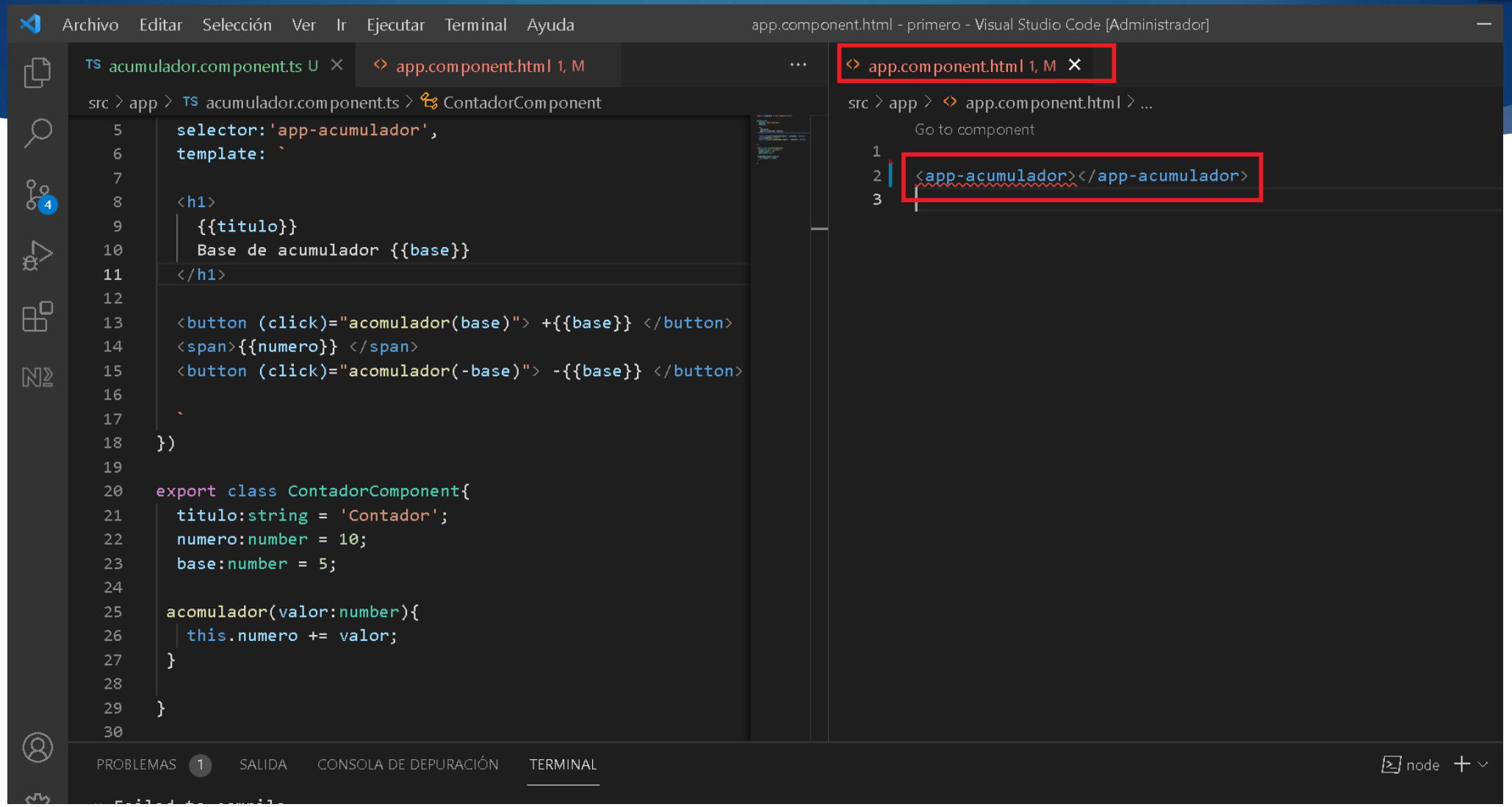

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
acumulador.component.ts - primero - Visual Studio Code [Administrador]
acumulador.component.ts U
app.component.html M
app.component.html M
src > app > acumulador.component.ts > ...
import { Component } from "@angular/core";
@Component({
selector:'app-acumulador',
template: `
`
})
export class ContadorComponent{
}
src > app > app.component.html > ...
Go to component
<h1>
{{titulo}}
Base de acumulador {{base}}
</h1>
<button (click)="acomulador(base)"> +{{base}} </button>
{{numero}}
<button (click)="acomulador(-base)"> -{{base}} </button>


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
acumulador.component.ts 9, U
app.component.ts M
src > app > acumulador.component.ts > ContadorComponent
import { Component } from "@angular/core";
@Component({
selector:'app-acumulador',
template: `
<h1>
{{titulo}}
Base de acumulador {{base}}
</h1>
<button (click)="acomulador(base)"> +{{base}} </button>
{{numero}}
<button (click)="acomulador(-base)"> -{{base}} </button>
`
})
export class ContadorComponent{
}


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
app.component.ts - primero - Visual Studio Code [Administrador]
acumulador.component.ts 9, U
app.component.ts M
src > app > acumulador.component.ts > ContadorComponent
import { Component } from "@angular/core";
@Component({
selector:'app-acumulador',
template: `
<h1>
{{titulo}}
Base de acumulador {{base}}
</h1>
<button (click)="acomulador(base)"> +{{base}} </button>
{{numero}}
<button (click)="acomulador(-base)"> -{{base}} </button>
`
})
export class ContadorComponent{
}
app.component.ts M
src > app > app.component.ts > AppComponent
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
titulo:string = 'Contador';
numero:number = 10;
base:number = 5;
acomulador(valor:number){
this.numero += valor;
}
}
PROBLEMAS 9 SALIDA CONSOLA DE DEPURACIÓN TERMINAL


acumulador.component.ts U
app.component.ts M
src > app > acumulador.component.ts > ContadorComponent > acomulador
selector:'app-acumulador',
template: `
<h1>
{{titulo}}
Base de acumulador {{base}}
</h1>
<button (click)="acomulador(base)"> +{{base}} </button>
{{numero}}
<button (click)="acomulador(-base)"> -{{base}} </button>
`
})
export class ContadorComponent{
titulo:string = 'Contador';
numero:number = 10;
base:number = 5;
acomulador(valor:number){
this.numero += valor;
}
}

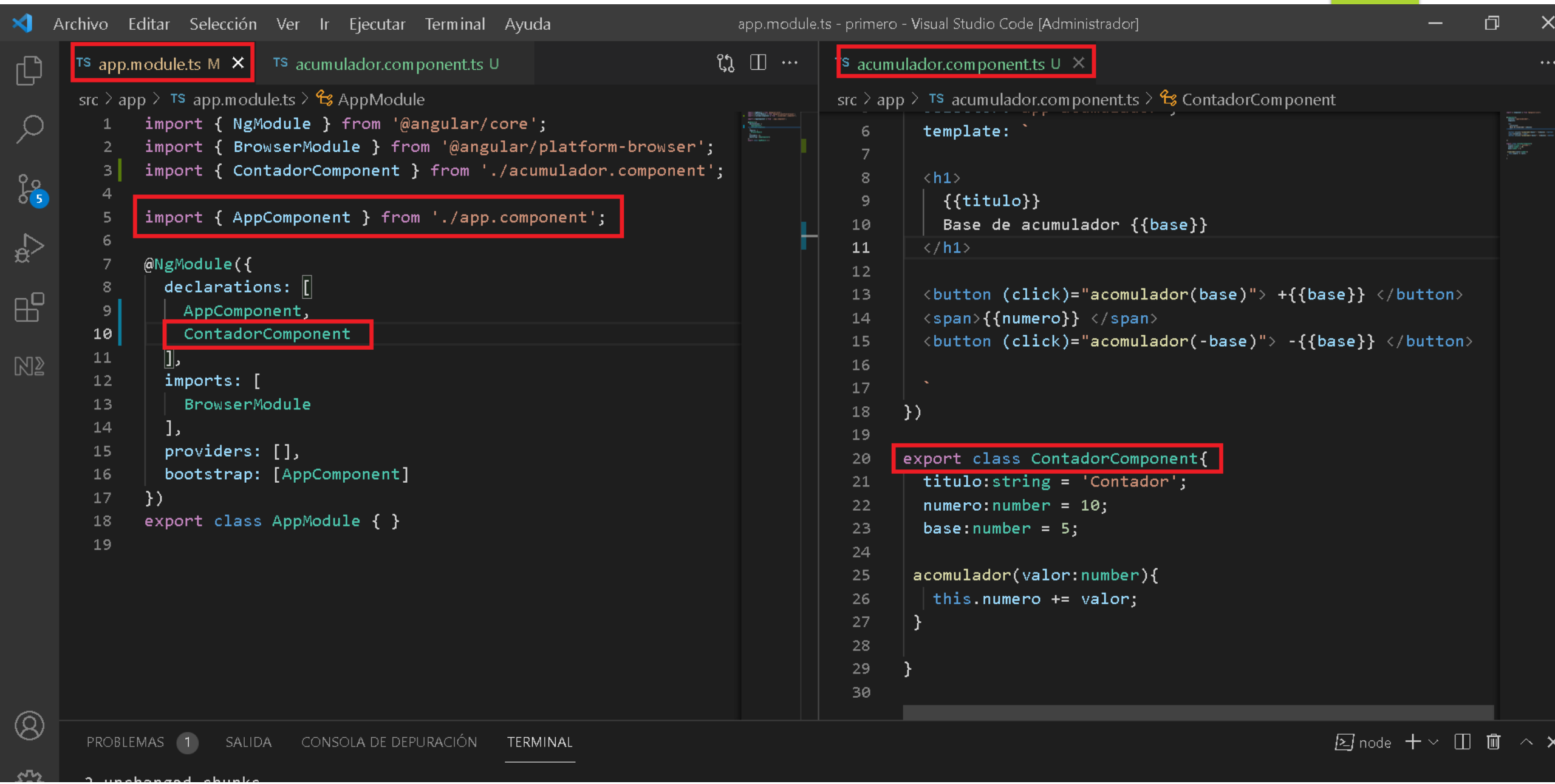
Ahora agregamos este nuevo componente al principal



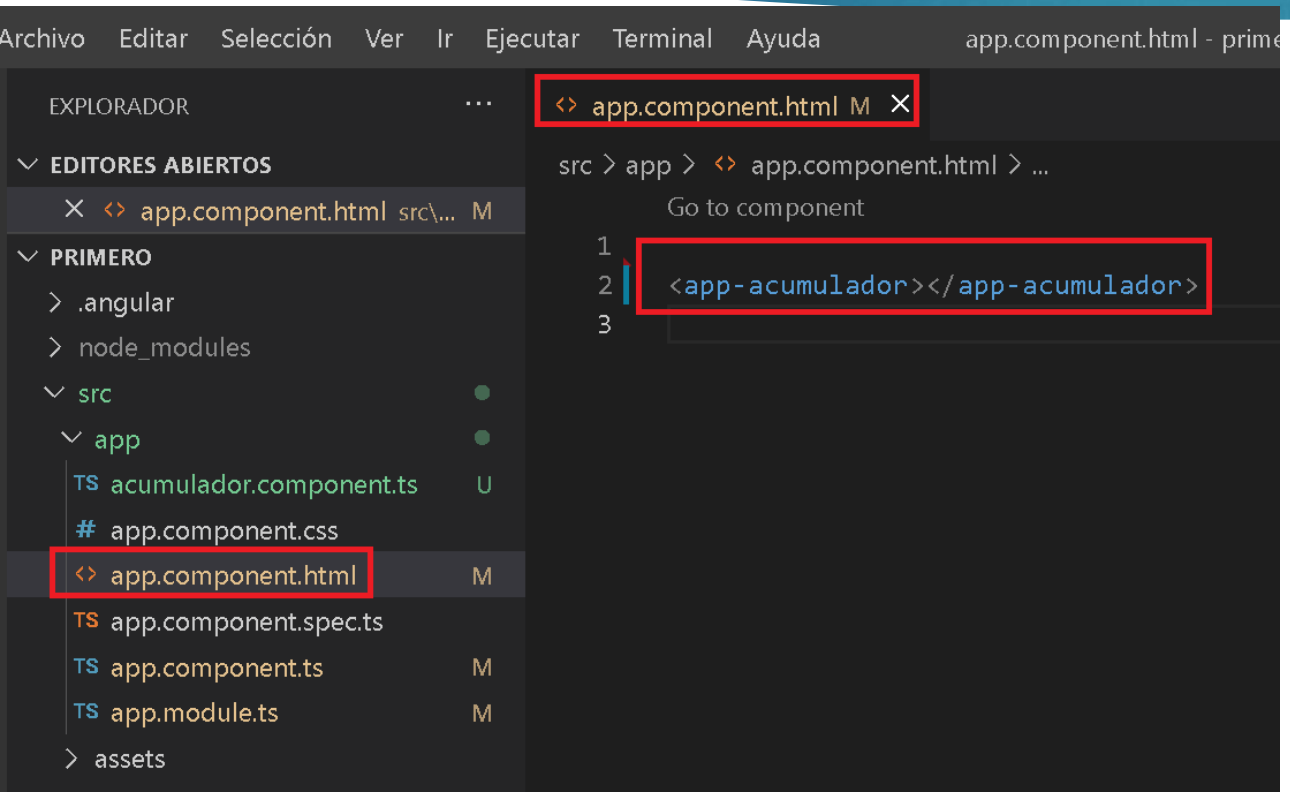
El error nos indica que el componente aun no ha sido agregado el modulo

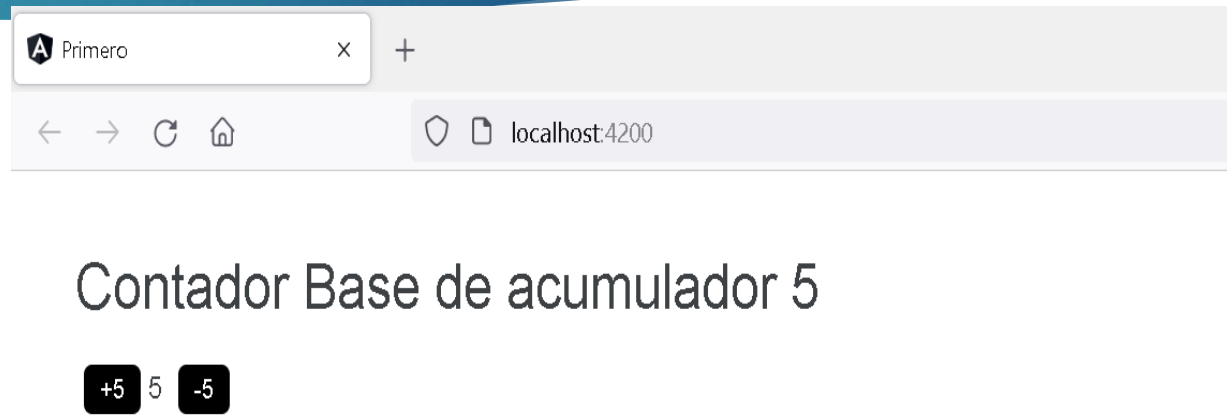
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL node

```
2 unchanged chunks

Build at: 2021-12-07T12:38:54.071Z - Hash: ec56f3aa7b49f626 - Time: 332ms

Error: src/app/app.component.html:2:1 - error NG8001: 'app-acumulador' is not a known element:
1. If 'app-acumulador' is an Angular component, then verify that it is part of this module.
2. If 'app-acumulador' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this compon
ent to suppress this message.

2 <app-acumulador></app-acumulador>
  ~~~~~~~~~~~~~~~~~

  src/app/app.component.ts:5:16
    5   templateUrl: './app.component.html',
                     ~~~~~~~~~~~~~~~~~~~~~~
    Error occurs in the template of component AppComponent.



x Failed to compile.
```

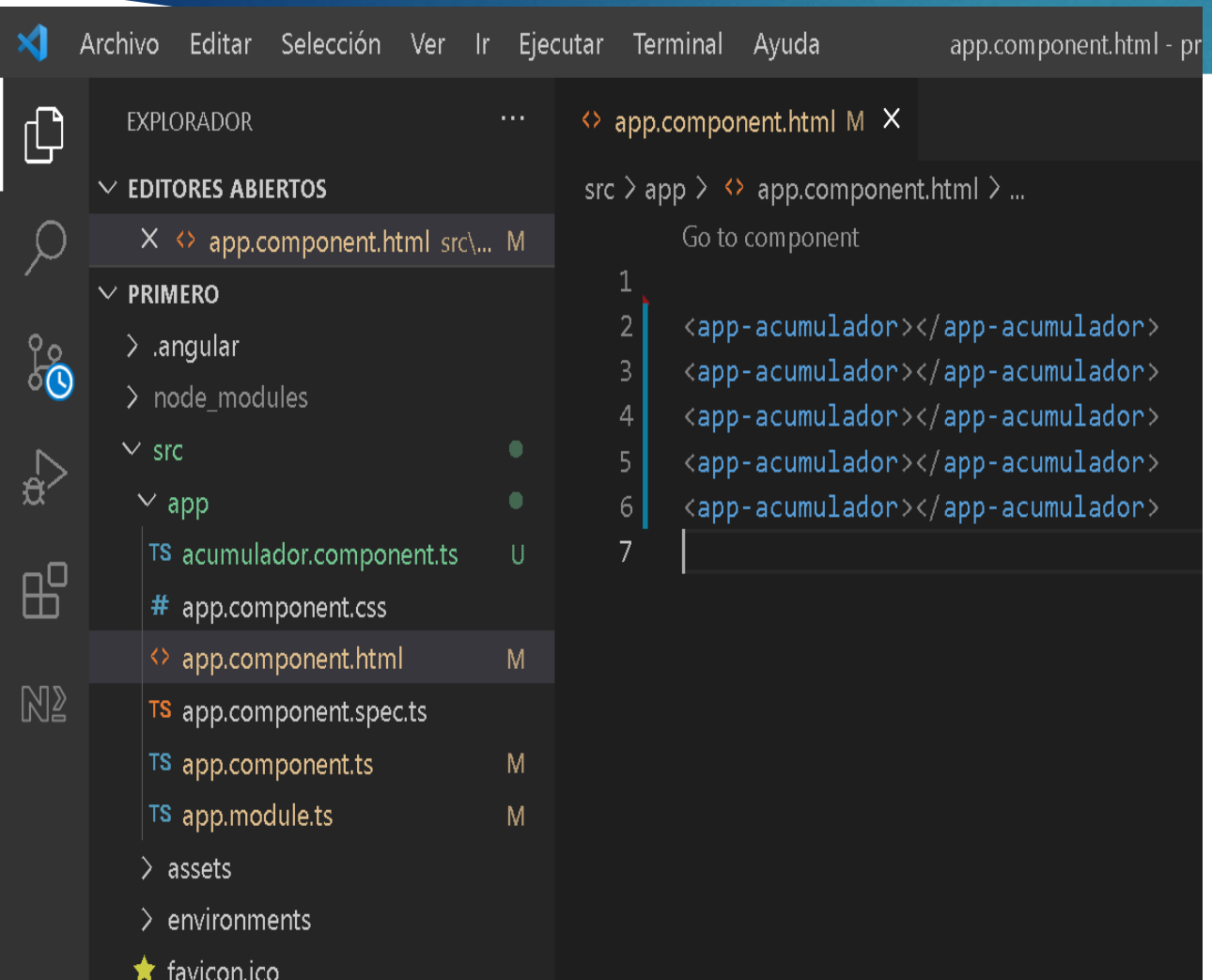

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
app.module.ts - primero - Visual Studio Code [Administrador]
app.module.ts M
acumulador.component.ts U
src > app > app.module.ts > AppModule
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ContadorComponent } from './acumulador.component';
import { AppComponent } from './app.component';
@NgModule({
declarations: [
AppComponent,
ContadorComponent
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
acumulador.component.ts U
src > app > acumulador.component.ts > ContadorComponent
template: `
<h1>
{{titulo}}
Base de acumulador {{base}}
</h1>
<button (click)="acomulador(base)"> +{{base}} </button>
{{numero}}
<button (click)="acomulador(-base)"> -{{base}} </button>
`
})
export class ContadorComponent{
titulo:string = 'Contador';
numero:number = 10;
base:number = 5;
acomulador(valor:number){
this.numero += valor;
}
}
PROBLEMAS 1
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
node

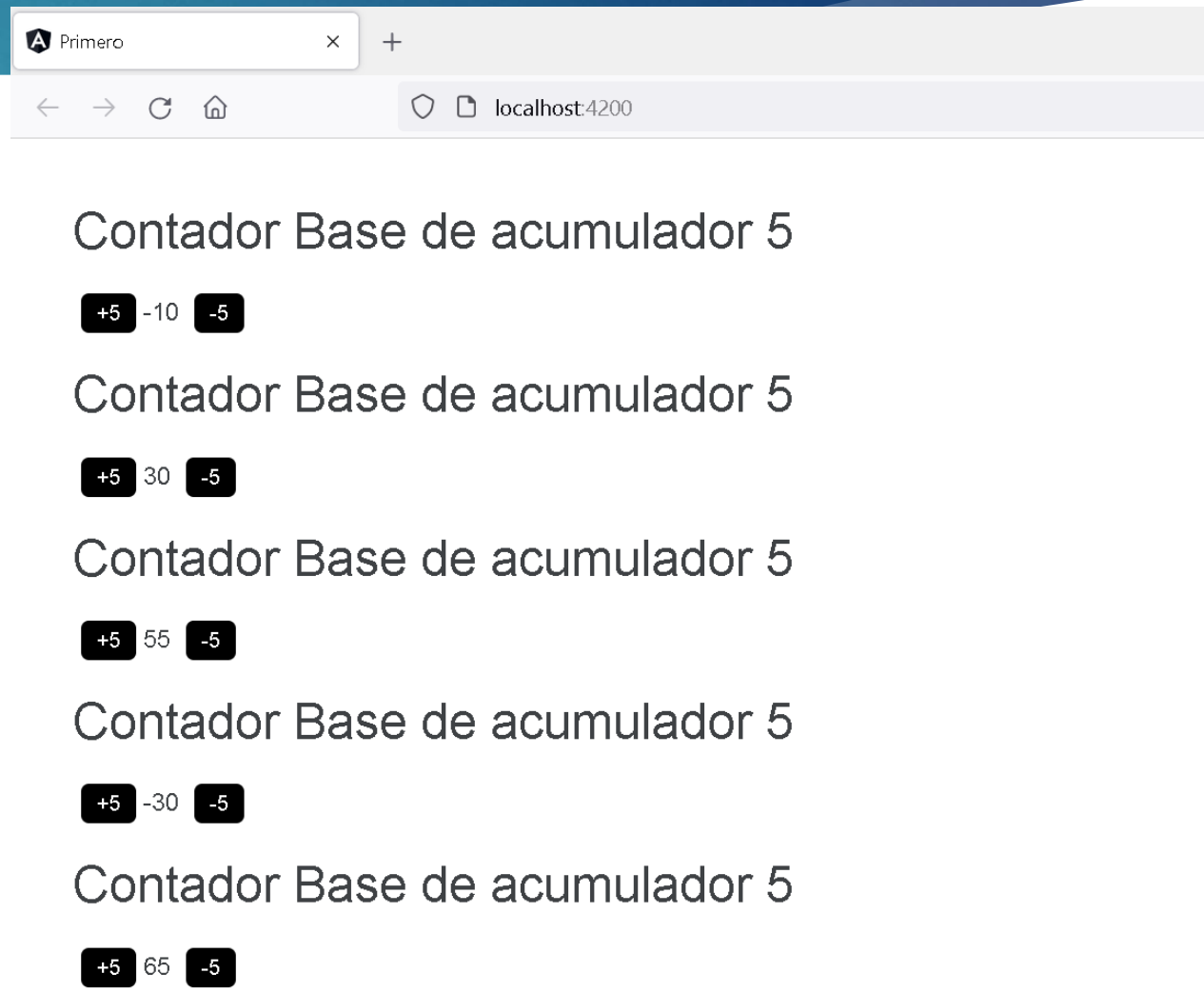
Con esto ya nos funciona el llamado al componente



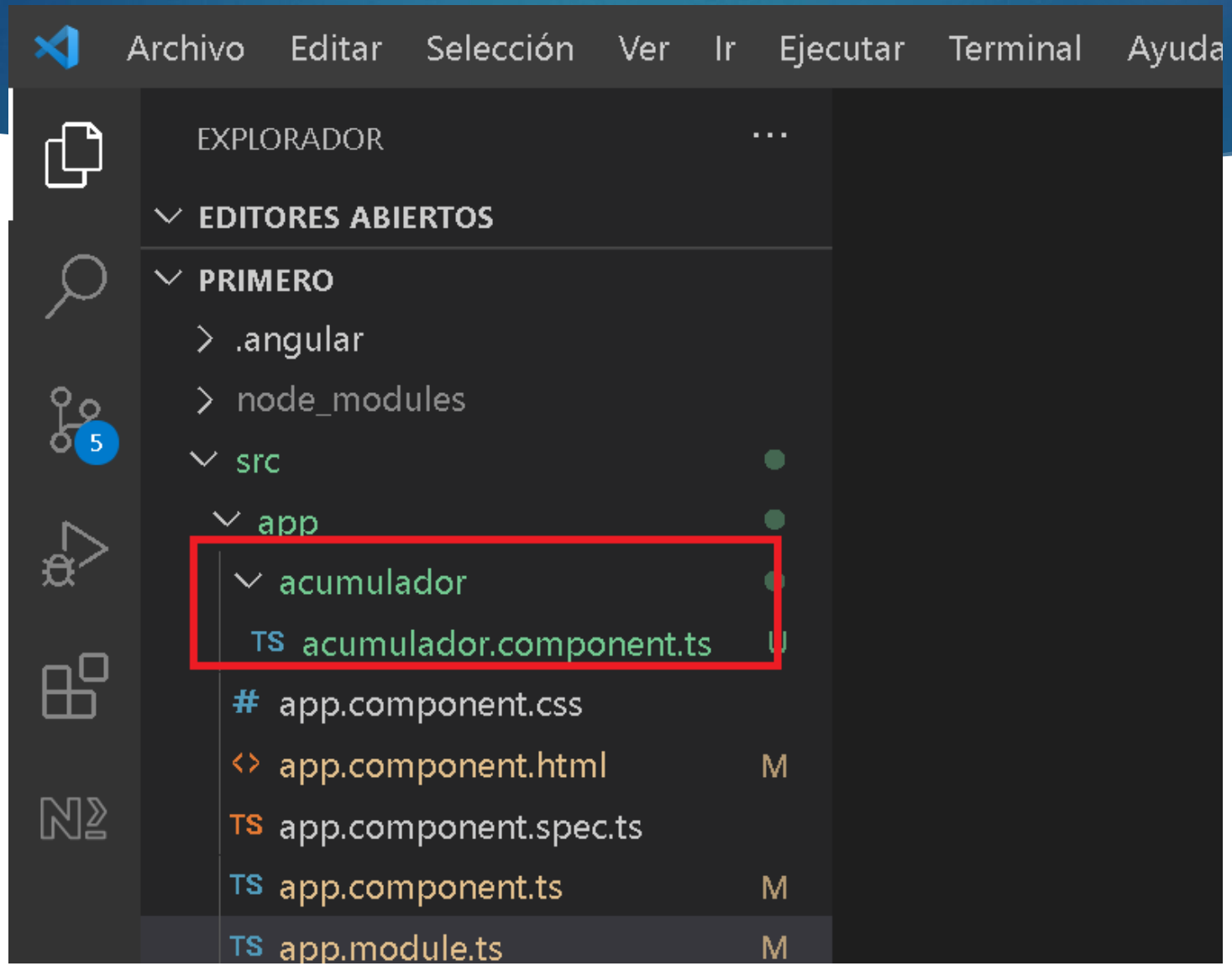


Podemos tener varias instancias del componente y todas se manejan de forma independiente

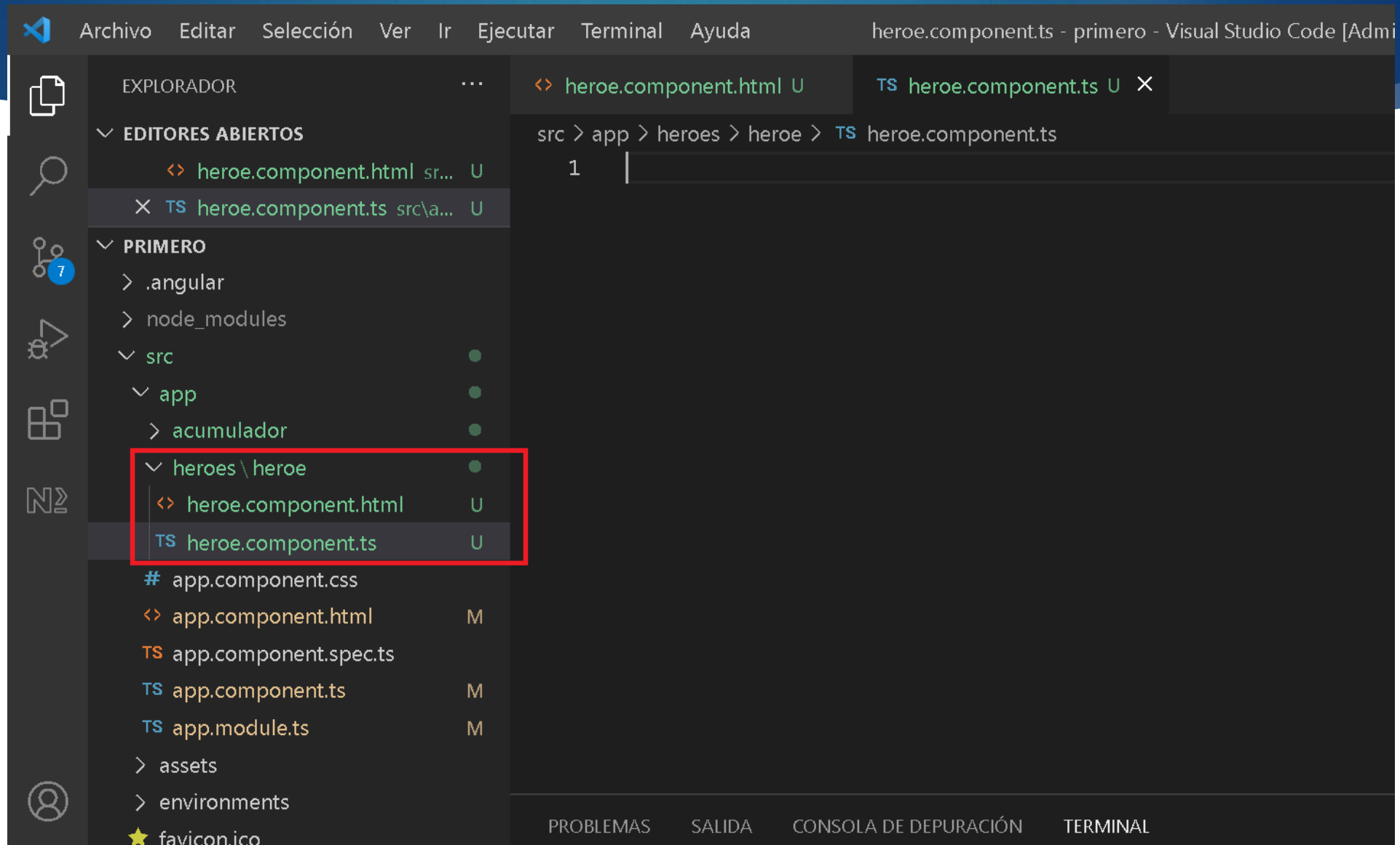


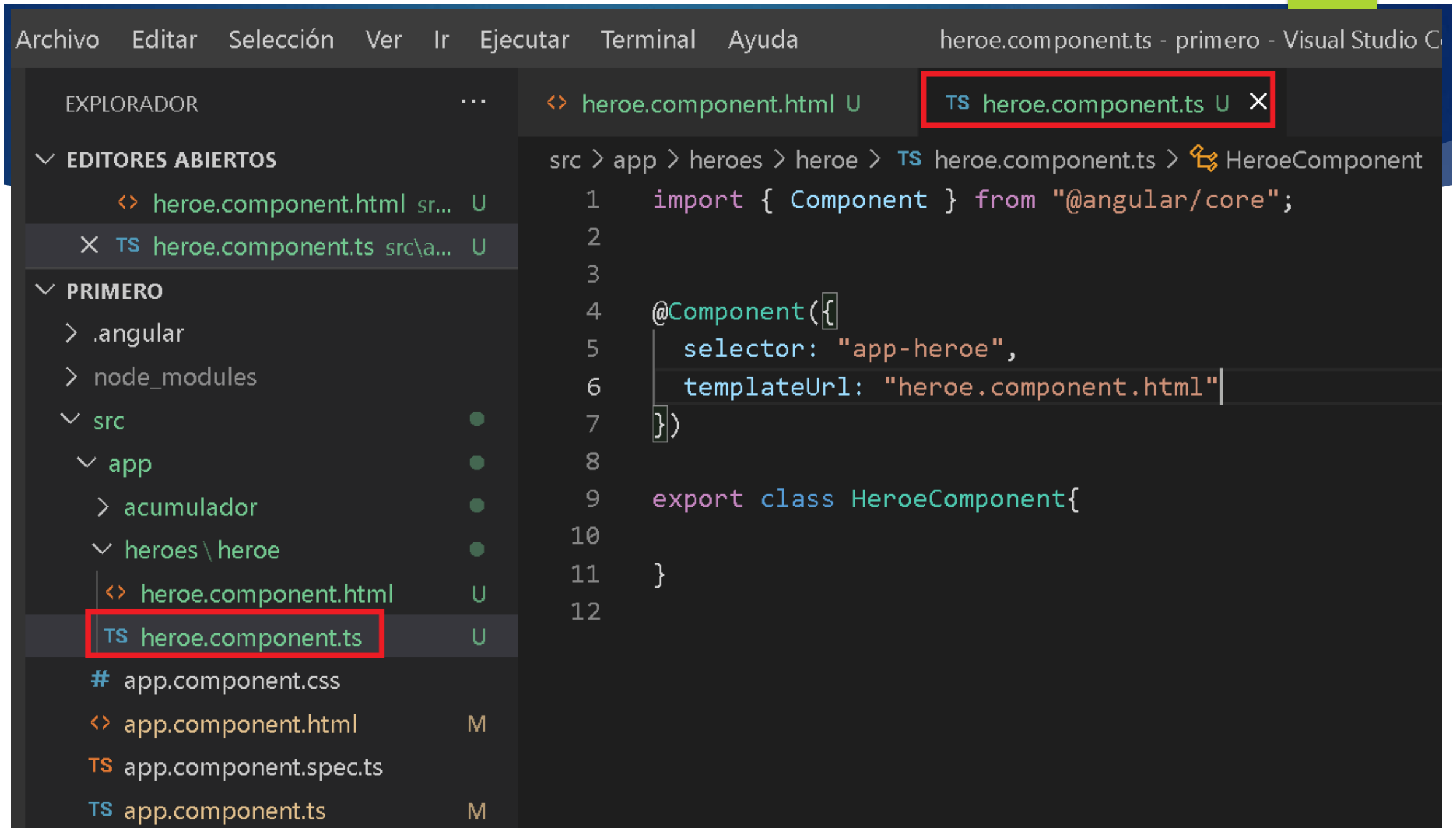


Podemos crear directorios para acomodar mejor los componentes

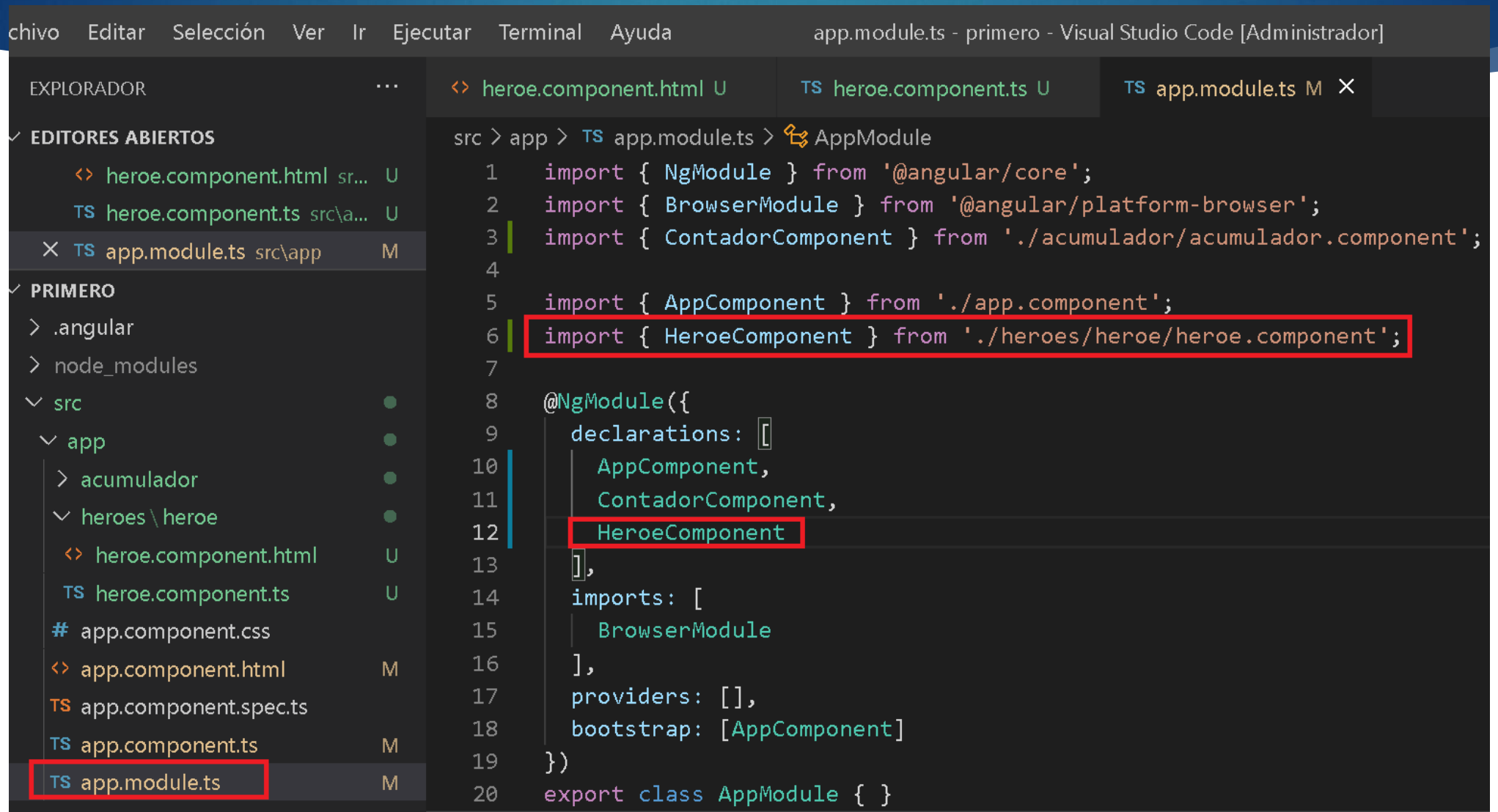


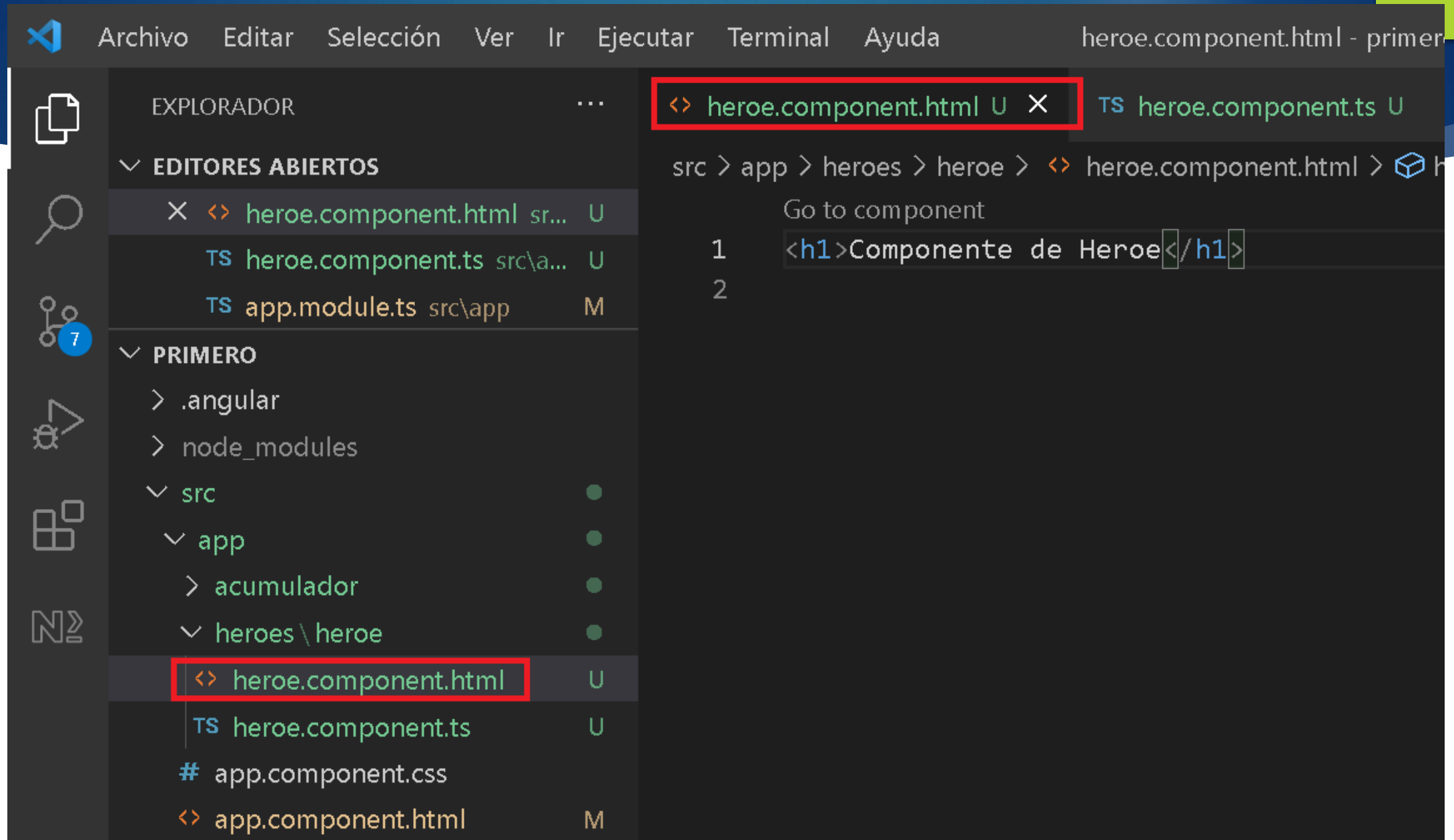
Ahora creamos un nuevo componente

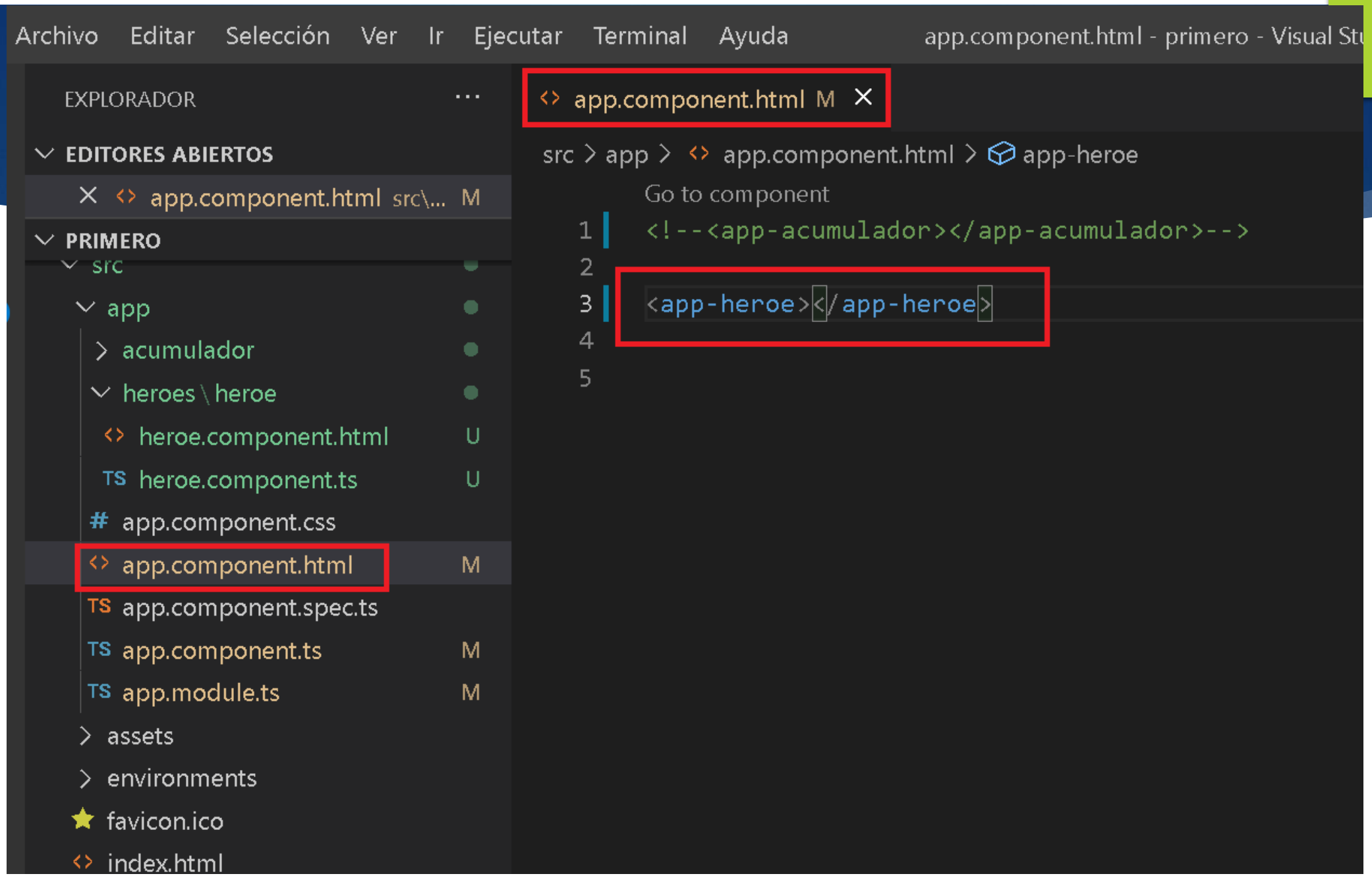


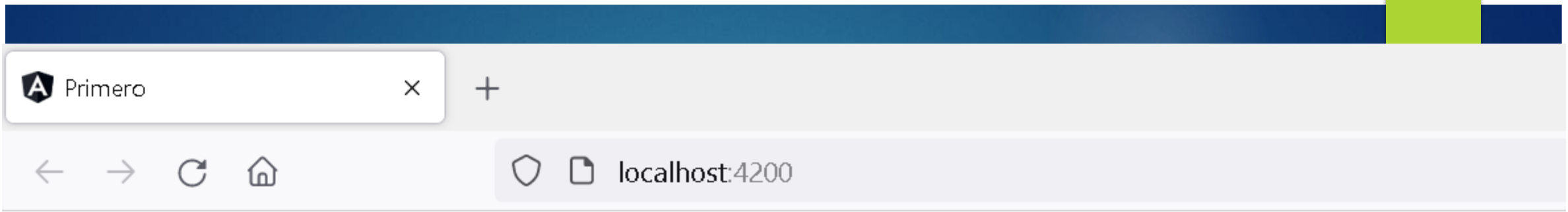

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
heroe.component.ts - primero - Visual Studio C
EXPLORADOR
EDITORES ABIERTOS
heroe.component.html sr... U
heroe.component.ts src\a... U
PRIMERO
.angular
node_modules
src
app
acumulador
heroes \ heroe
heroe.component.html U
heroe.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
heroe.component.html U
heroe.component.ts U
src > app > heroes > heroe > heroe.component.ts > HeroeComponent
1 import { Component } from "@angular/core";
2
3
4 @Component({
5 selector: "app-heroe",
6 templateUrl: "heroe.component.html"
7 })
8
9 export class HeroeComponent{
10
11 }
12

No olvidemos agregar el nuevo componente a app.module.ts




Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
heroe.component.html - primer
EXPLORADOR
EDITORES ABIERTOS
heroe.component.html sr... U
heroe.component.ts src\a... U
app.module.ts src\app M
PRIMERO
.angular
node_modules
src
app
acumulador
heroes \ heroe
heroe.component.html U
heroe.component.ts U
app.component.css
app.component.html M
heroe.component.html U
heroe.component.ts U
src > app > heroes > heroe > heroe.component.html
Go to component
1 <h1>Componente de Heroe</h1>
2


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
app.component.html - primero - Visual St
EXPLORADOR
app.component.html M
EDITORES ABIERTOS
app.component.html src\... M
PRIMERO
src
app
acumulador
heroes \ heroe
heroe.component.html U
heroe.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
index.html
src > app > app.component.html > app-heroe
Go to component
1 <!--<app-acumulador></app-acumulador>-->
2
3 <app-heroe></app-heroe>
4
5



Componente de Heroe

Agregamos lo siguiente

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda heroe.component.ts - primero - Visual Studio Code [Administrador]

TS heroe.component.ts U

src > app > heroes > heroe > TS heroe.component.ts > HeroeComponent

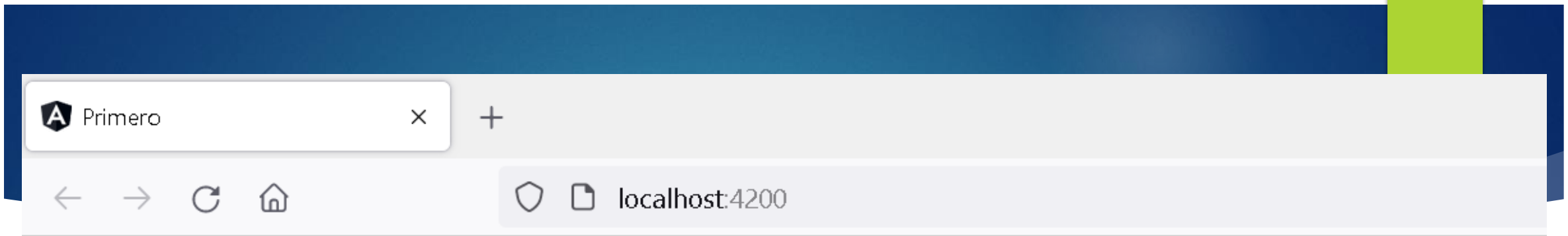
```
3
4   @Component({
5     selector: "app-heroe",
6     templateUrl: "heroe.component.html"
7   })
8
9   export class HeroeComponent{
10    nombre: string = "Ironman";
11    edad: number = 45;
12
13    get nombreCapitalizado(){
14      return this.nombre.toUpperCase();
15    }
16    obtenerNombre():string{
17      return `${this.nombre} - ${this.edad}`;
18    }
19
```

heroe.component.html U

src > app > heroes > heroe > heroe.component.html > dl >

Go to component

```
1   <h1>Componente de Heroe</h1>
2
3   <dl>
4     <td>Nombre:</td>
5     <dd>{{nombre}}</dd>
6
7     <td>Edad:</td>
8     <dd>{{edad}}</dd>
9
10    <td>Funcion:</td>
11    <dd>{{obtenerNombre()}}</dd>
12
13    <td>Capitalizado:</td>
14    <dd>{{nombreCapitalizado}}</dd>
15  </dl>
16
```



Componente de Heroe

Nombre:

Ironman

Edad:

45

Funcion:

Ironman - 45

Capitalizado:

IRONMAN

Agregamos estas funciones y botones

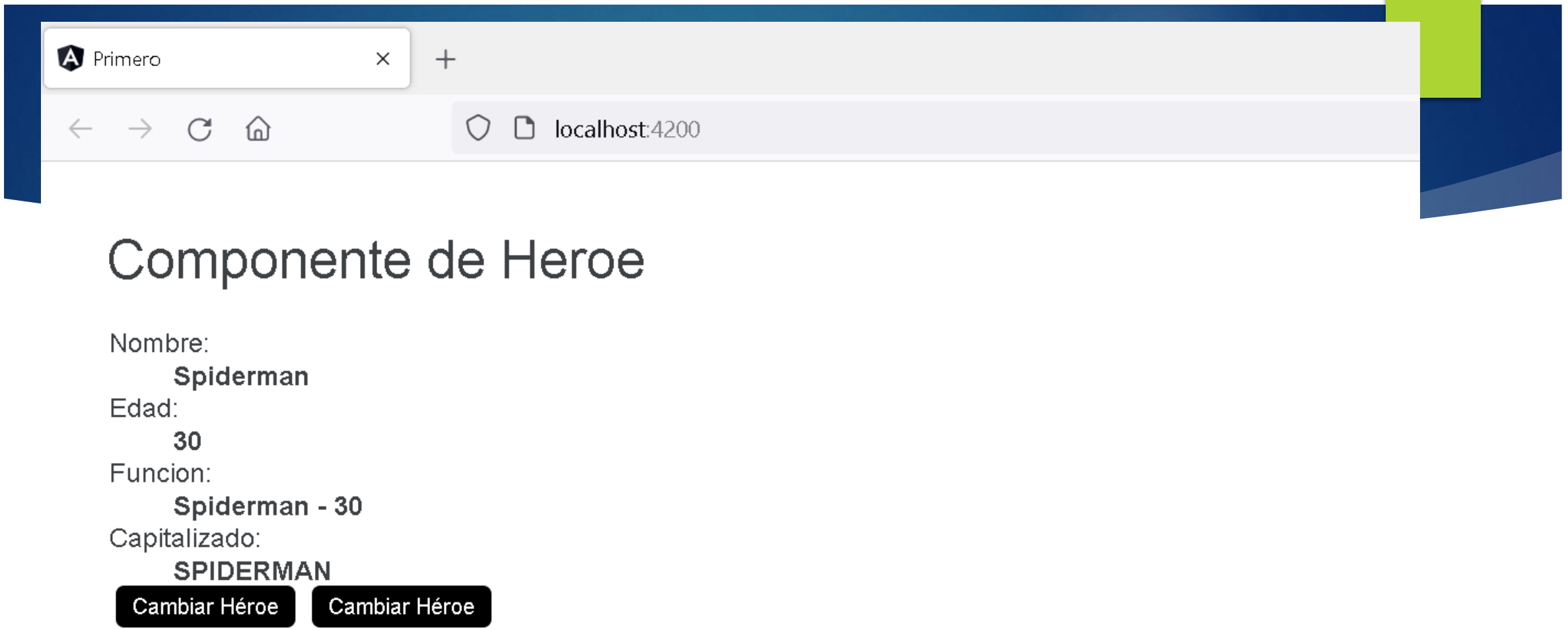
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda — heroe.component.html - primero - Visual Studio Code [Administrador]

heroe.component.ts U

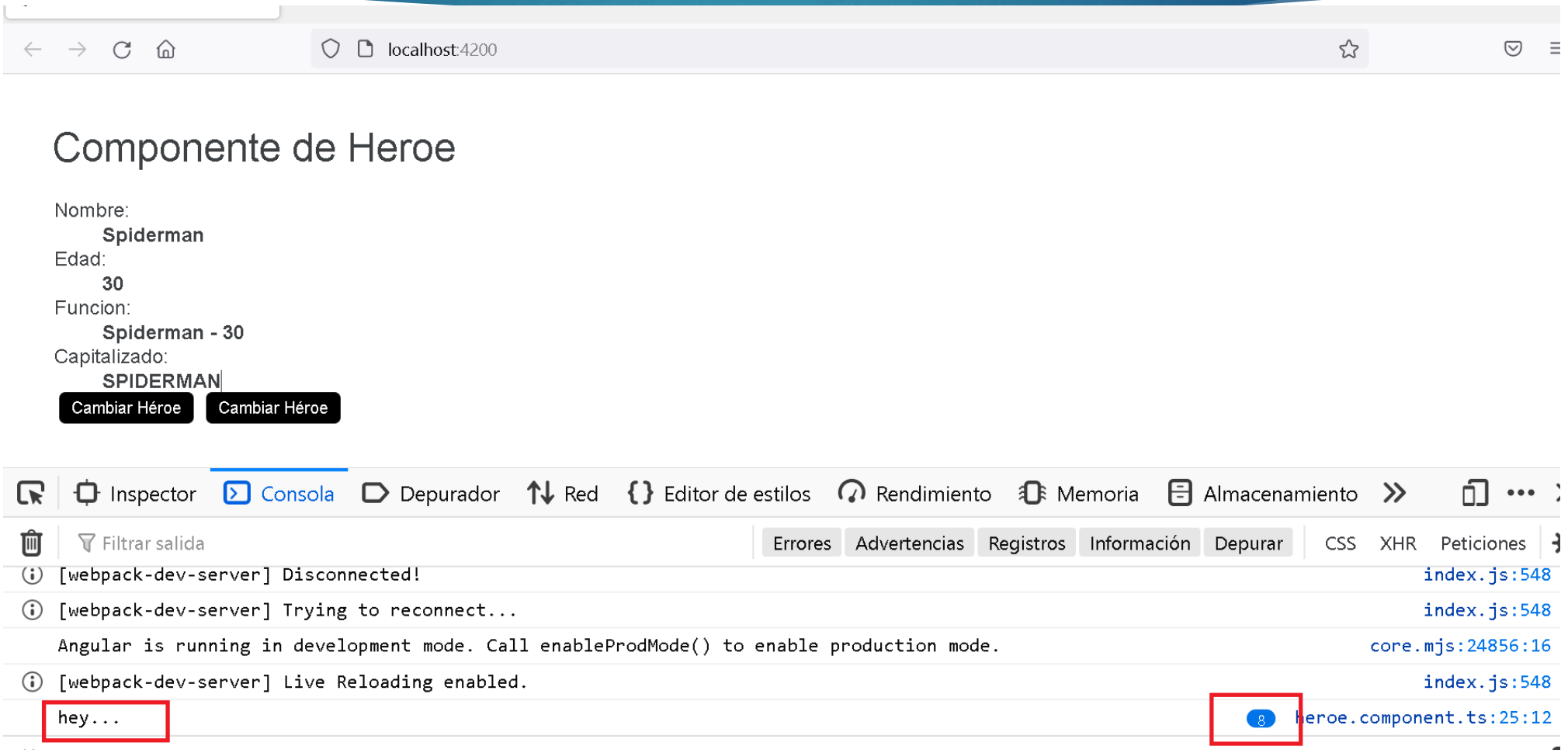
```
10    nombre: string = "Ironman";
11    edad: number = 45;
12
13    get nombreCapitalizado(){
14      return this.nombre.toUpperCase();
15    }
16    obtenerNombre():string{
17      return `${this.nombre} - ${this.edad}`;
18    }
19
20    cambiarNombre():void{
21      this.nombre = "Spiderman";
22    }
23
24    cambiarEdad():void{
25      console.log('hey...');
26      this.edad = 30;
27    }
28
29  }
```

heroe.component.html U

```
1   <h1>Componente de Heroe</h1>
2
3   <dl>
4     <td>Nombre:</td>
5     <dd>{{nombre}}</dd>
6
7     <td>Edad:</td>
8     <dd>{{edad}}</dd>
9
10    <td>Funcion:</td>
11    <dd>{{obtenerNombre()}}</dd>
12
13    <td>Capitalizado:</td>
14    <dd>{{nombreCapitalizado}}</dd>
15
16    <button (click)="cambiarNombre()">Cambiar Héroe</button
17    <button (click)="cambiarEdad()">Cambiar Héroe</button>
18  </dl>
19
```


Primero
localhost:4200
Componente de Heroe
Nombre:
Spiderman
Edad:
30
Funcion:
Spiderman - 30
Capitalizado:
SPIDERMAN
Cambiar Héroe
Cambiar Héroe

One Wey Data Binding – Enlazado en una sola vía

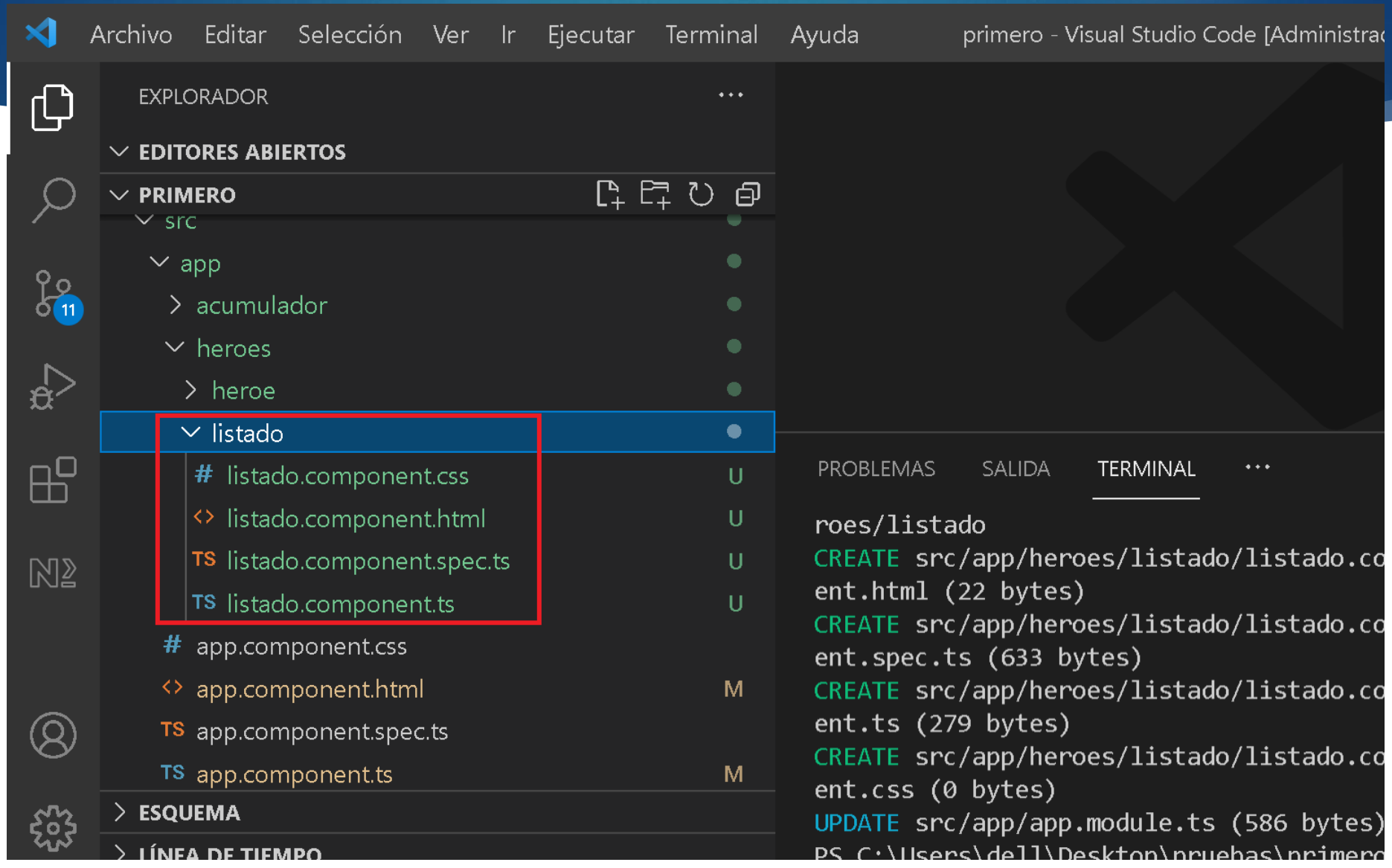


También podemos generar componentes por comando

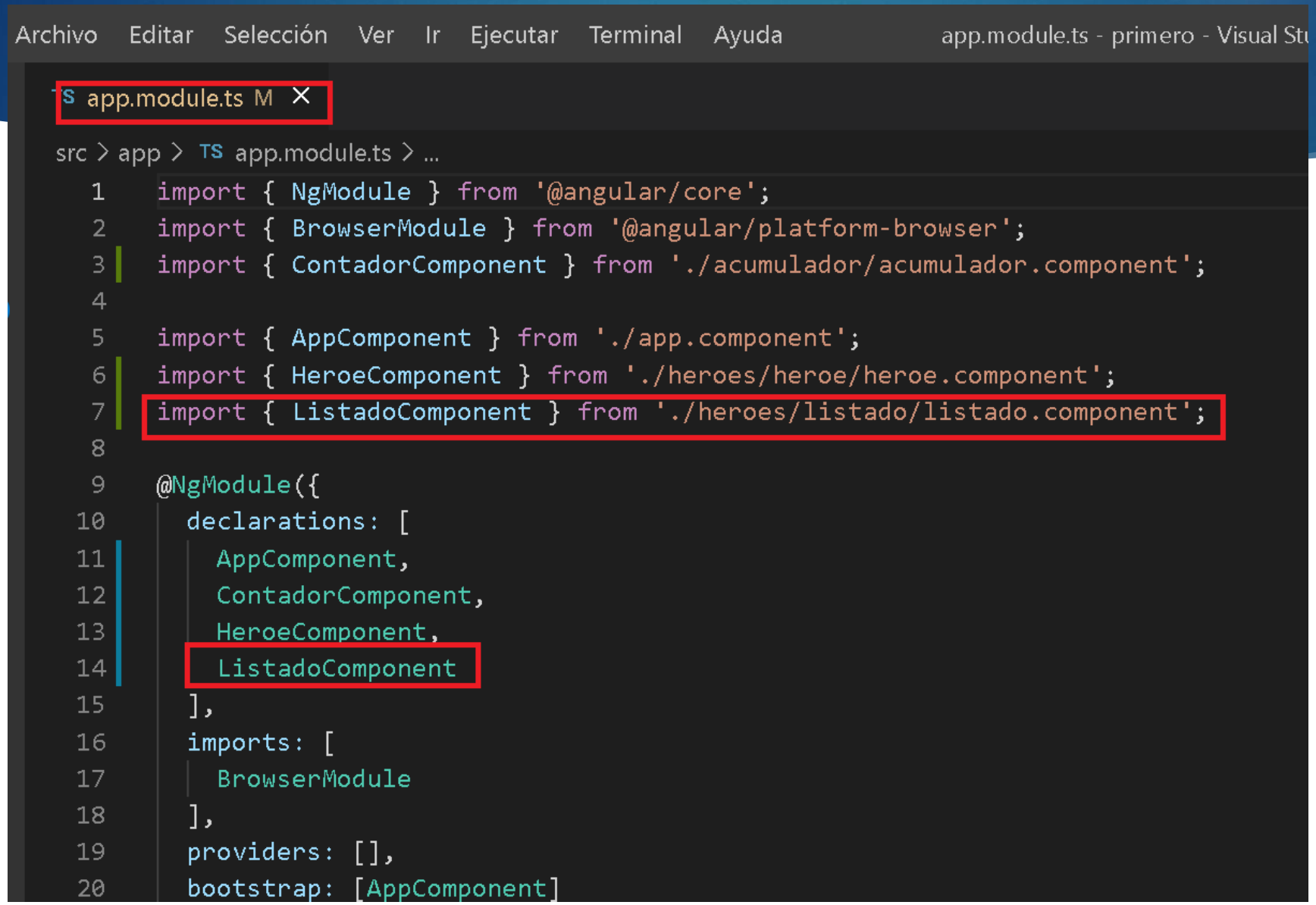
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

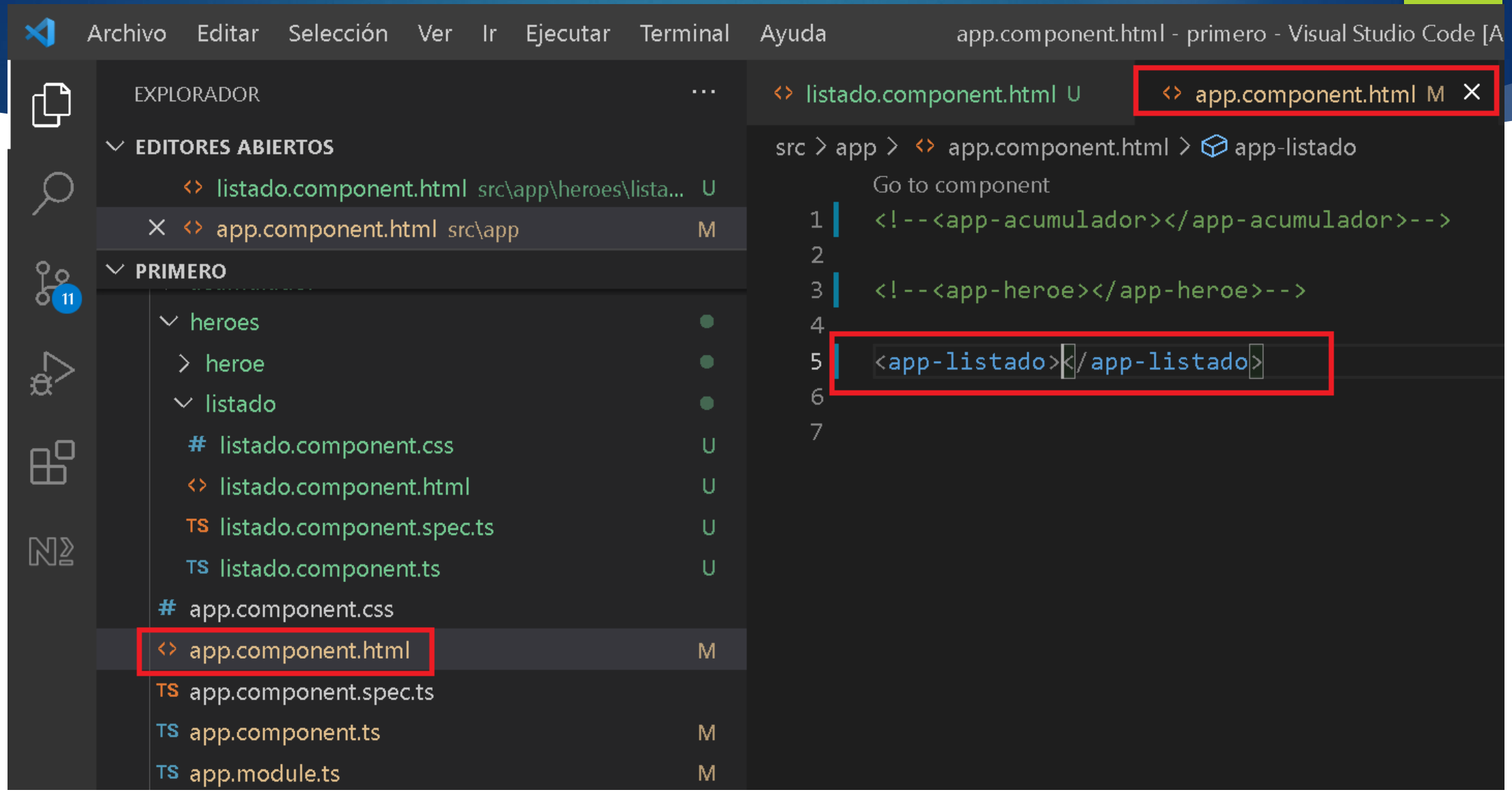
```
PS C:\Users\dell\Desktop\pruebas\primero> ng g c heroes/listado
CREATE src/app/heroes/listado/listado.component.html (22 bytes)
CREATE src/app/heroes/listado/listado.component.spec.ts (633 bytes)
CREATE src/app/heroes/listado/listado.component.ts (279 bytes)
CREATE src/app/heroes/listado/listado.component.css (0 bytes)
UPDATE src/app/app.module.ts (586 bytes)
PS C:\Users\dell\Desktop\pruebas\primero>
```

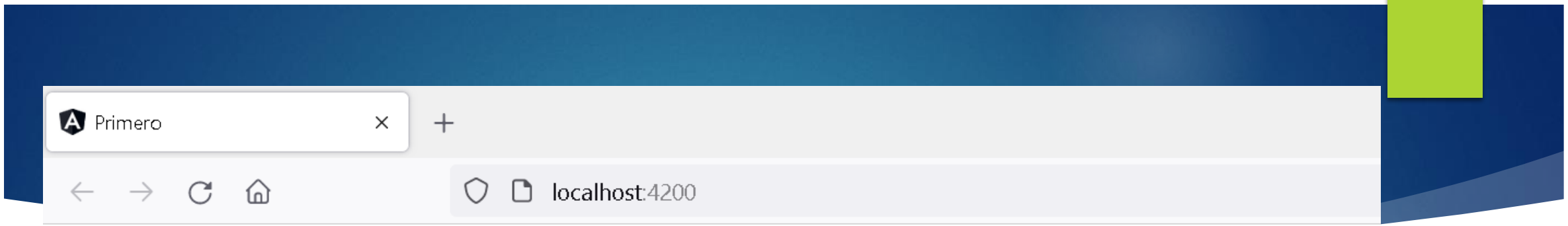
Nos genera de manera automática estos archivos



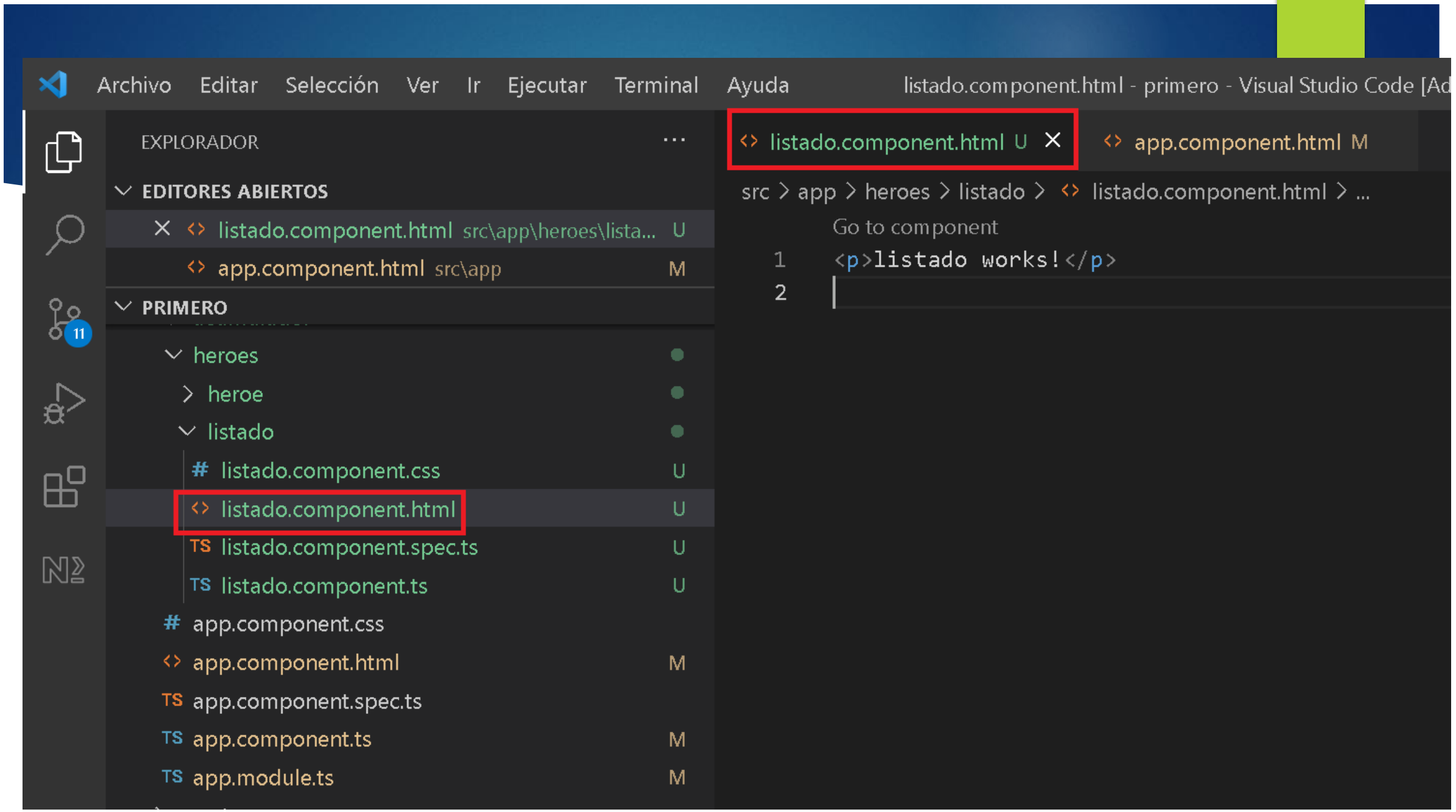
También agrega la función al app.module.ts



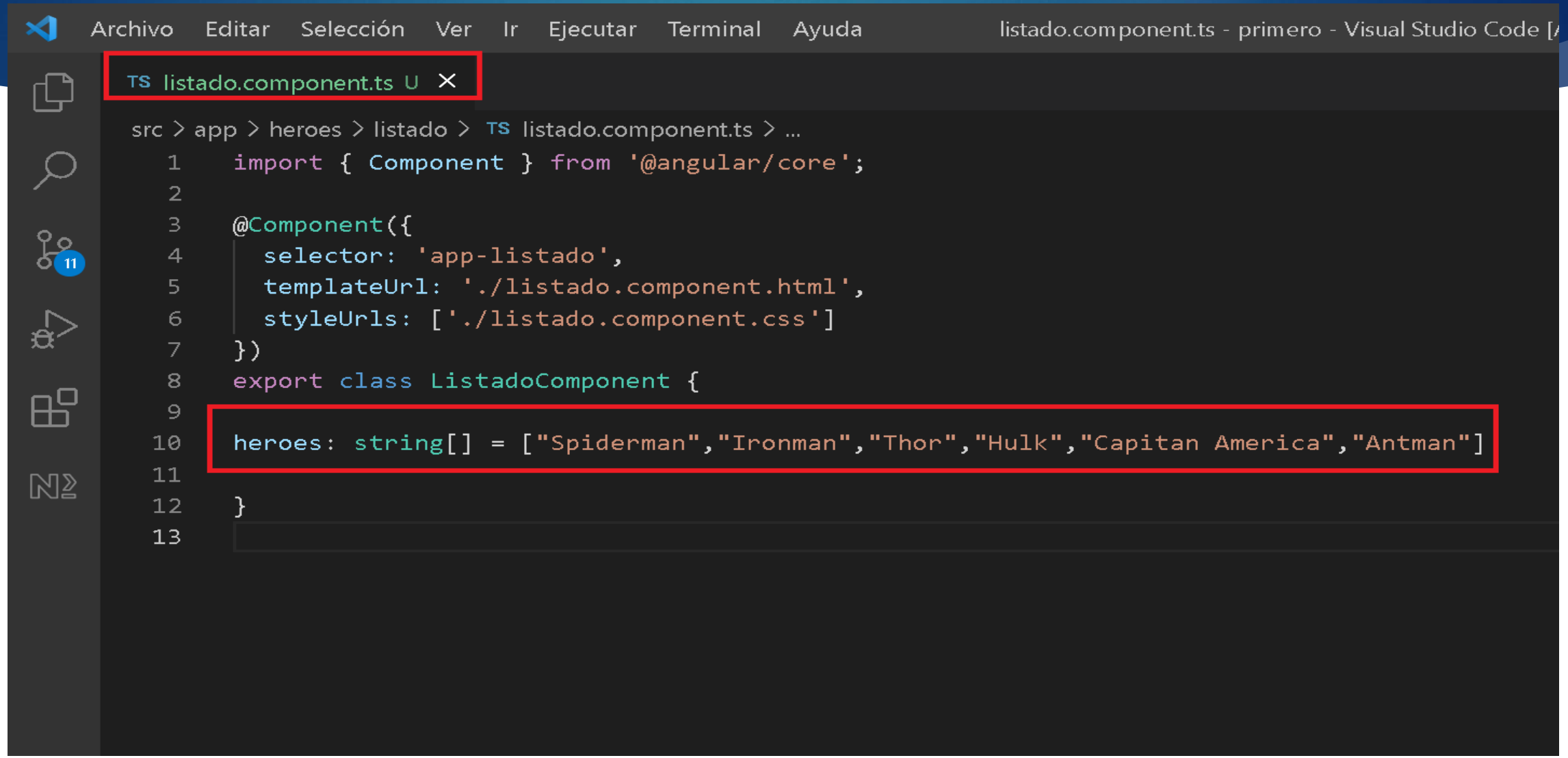

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
app.component.html - primero - Visual Studio Code [A
EXPLORADOR
EDITORES ABIERTOS
listado.component.html src\app\heroes\lista... U
app.component.html src\app M
PRIMERO
heroes
heroe
listado
listado.component.css U
listado.component.html U
listado.component.spec.ts U
listado.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
listado.component.html U
app.component.html M
src > app > app.component.html > app-listado
Go to component
1 <!--<app-acumulador></app-acumulador>-->
2
3 <!--<app-heroe></app-heroe>-->
4
5 <app-listado></app-listado>
6
7

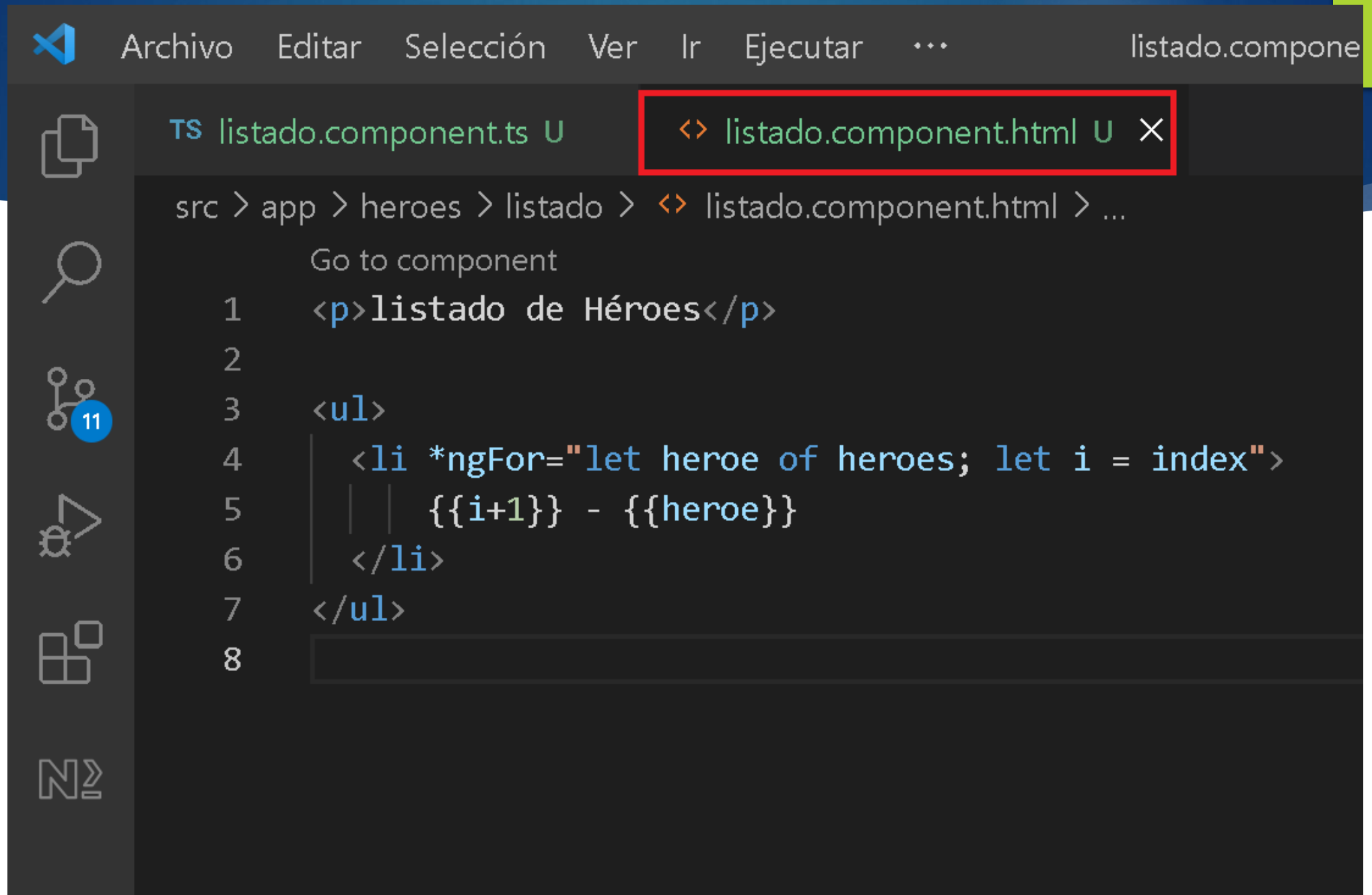


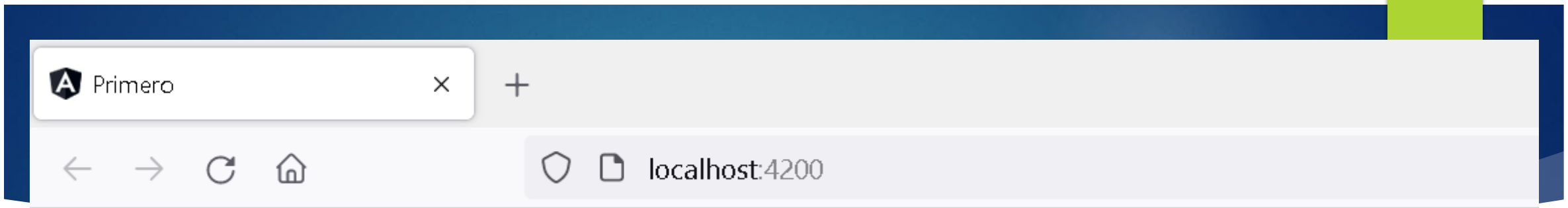
listado works!


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
listado.component.html - primero - Visual Studio Code [Ad
EXPLORADOR
EDITORES ABIERTOS
listado.component.html src\app\heroes\lista... U
app.component.html src\app M
PRIMERO
heroes
heroe
listado
listado.component.css U
listado.component.html U
listado.component.spec.ts U
listado.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
listado.component.html U
app.component.html M
src > app > heroes > listado > listado.component.html > ...
Go to component
1 listado works!
2

Directiva *ngFor




Archivo Editar Selección Ver Ir Ejecutar ··· listado.compone
TS listado.component.ts U
listado.component.html U
src > app > heroes > listado > listado.component.html > ...
Go to component
1 listado de Héroes
2
3
4 <li *ngFor="let heroe of heroes; let i = index">
5 {{i+1}} - {{heroe}}
6
7
8
11



listado de Héroes

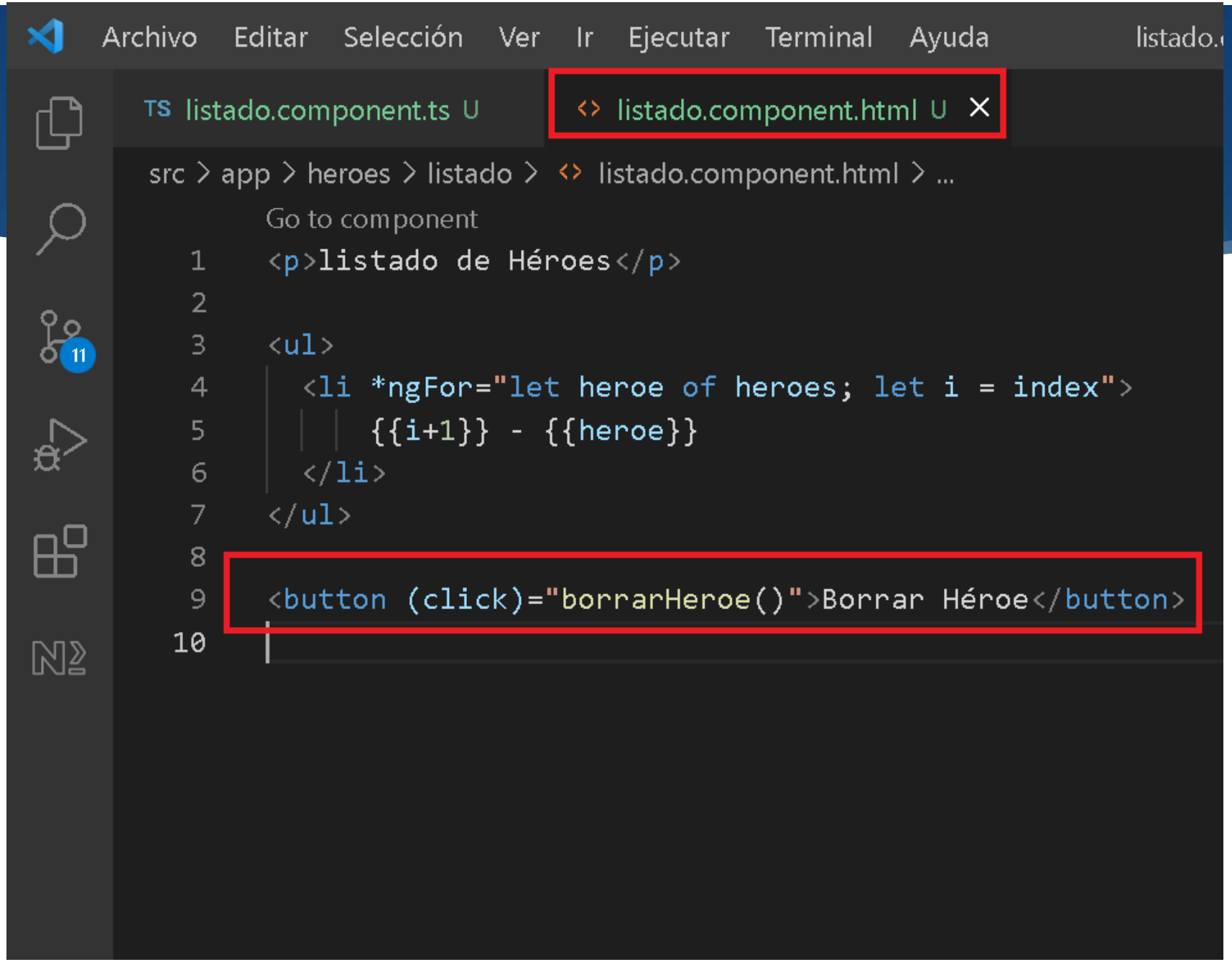
- 1 - Spiderman
- 2 - Ironman
- 3 - Thor
- 4 - Hulk
- 5 - Capitan America
- 6 - Antman

TS listado.component.ts U

<> listado.component.html U

src > app > heroes > listado > TS listado.component.ts > ...

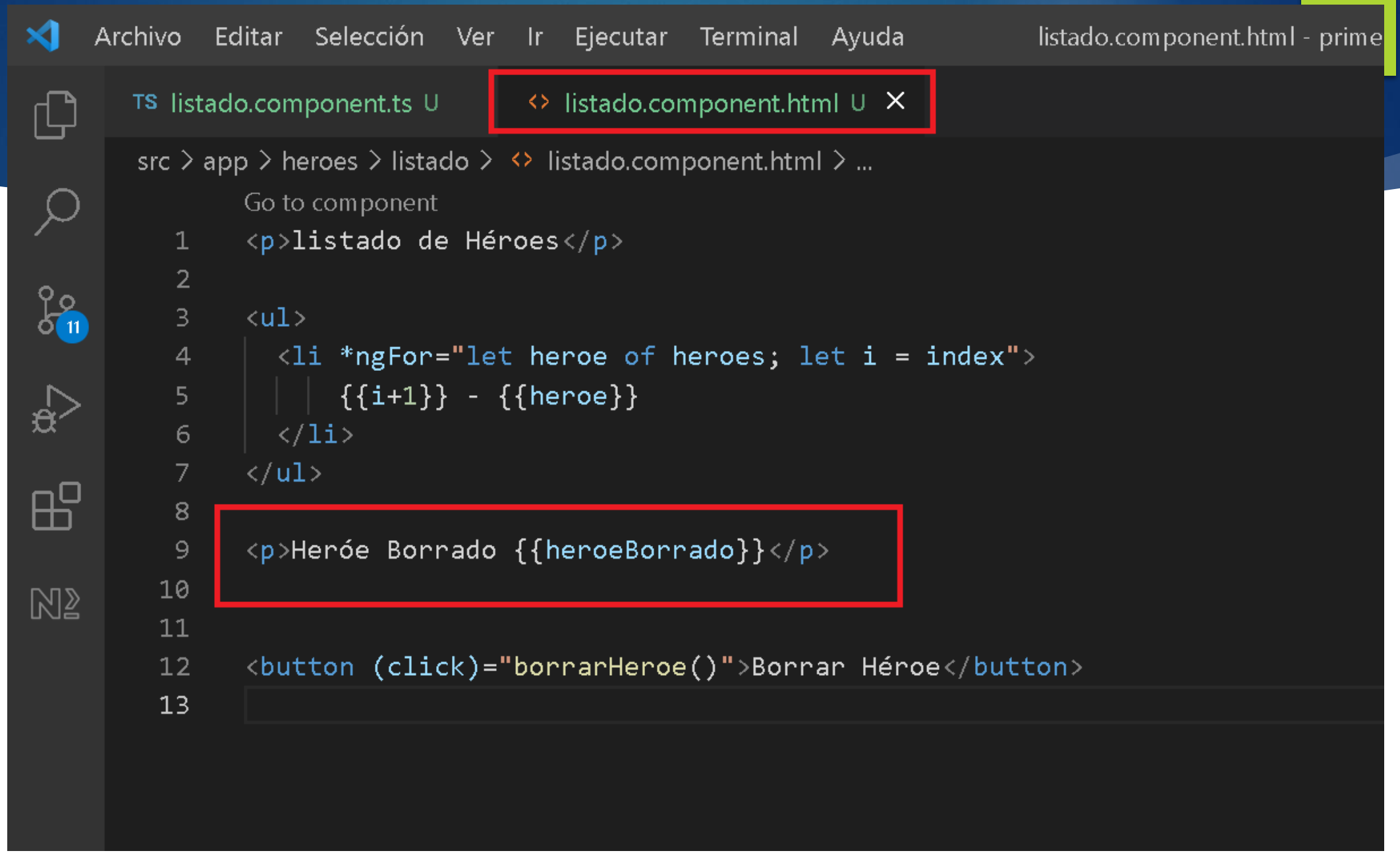
```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'app-listado',
5     templateUrl: './listado.component.html',
6     styleUrls: ['./listado.component.css']
7   })
8   export class ListadoComponent {
9
10  heroes: string[] = ["Spiderman","Ironman","Thor","Hulk","Capitan America","Antman"];
11
12  borrarHeroe(){
13    this.heroes.pop();
14  }
15
16  }
17
```


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
listado.component.ts U
listado.component.html U
src > app > heroes > listado > listado.component.html > ...
Go to component
1 listado de Héroes
2
3
4 <li *ngFor="let heroe of heroes; let i = index">
5 {{i+1}} - {{heroe}}
6
7
8
9 <button (click)="borrarHeroe()">Borrar Héroe</button>
10

TS listado.component.ts U × <> listado.component.html U

src > app > heroes > listado > TS listado.component.ts > ListadoComponent

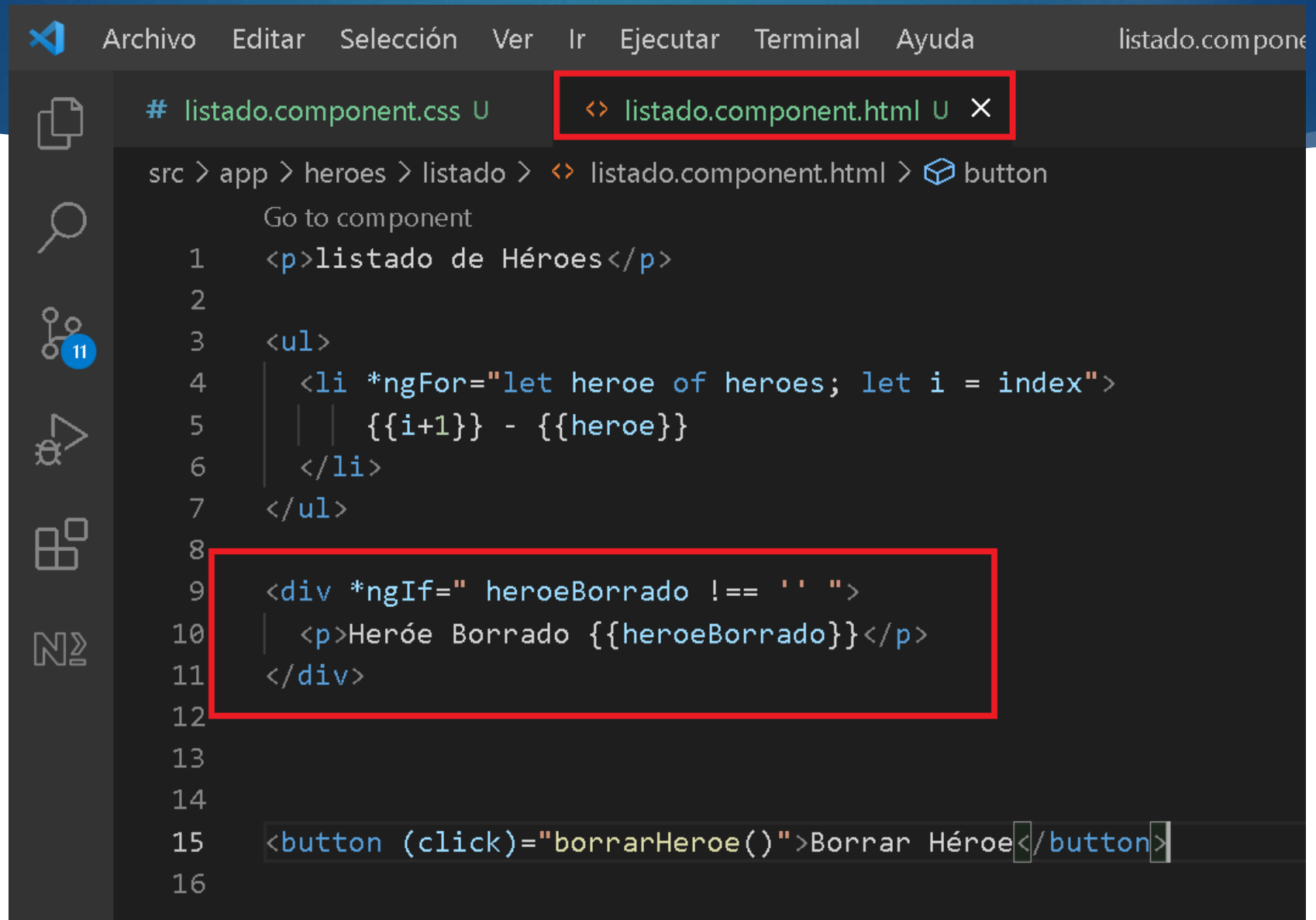
```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-listado',
  templateUrl: './listado.component.html',
  styleUrls: ['./listado.component.css']
})
export class ListadoComponent {

heroes: string[] = ["Spiderman","Ironman","Thor","Hulk","Capitan America","Antman"];
heroeBorrado:string |undefined = "";

  borrarHeroe(){
    this.heroeBorrado = this.heroes.pop();
  }
}

```


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
listado.component.html - prime
TS listado.component.ts U
listado.component.html U
src > app > heroes > listado > listado.component.html > ...
Go to component
listado de Héroes

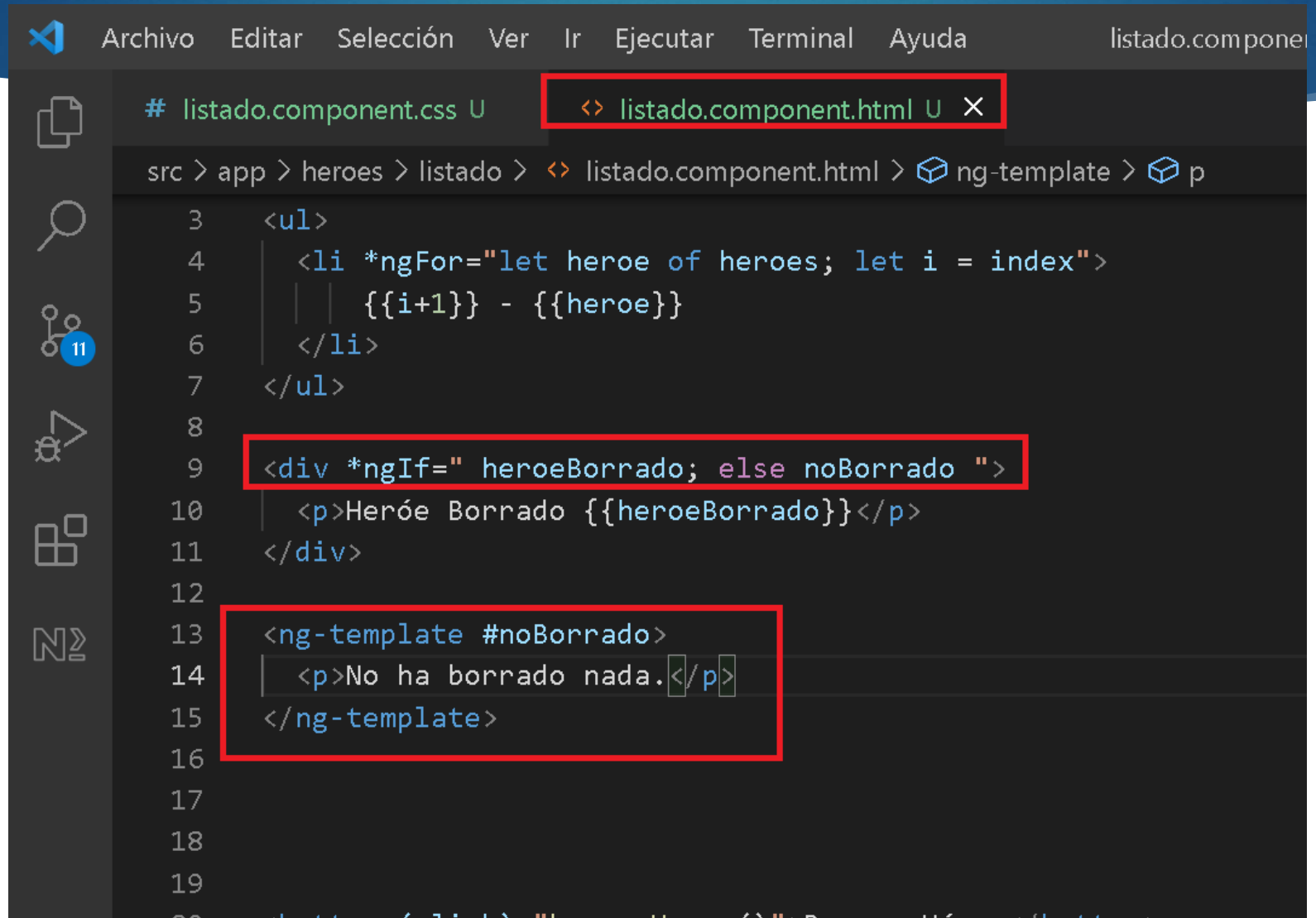
<li *ngFor="let heroe of heroes; let i = index">
{{i+1}} - {{heroe}}

Heróe Borrado {{heroeBorrado}}
<button (click)="borrarHeroe()">Borrar Héroe</button>

Directiva *ngIf



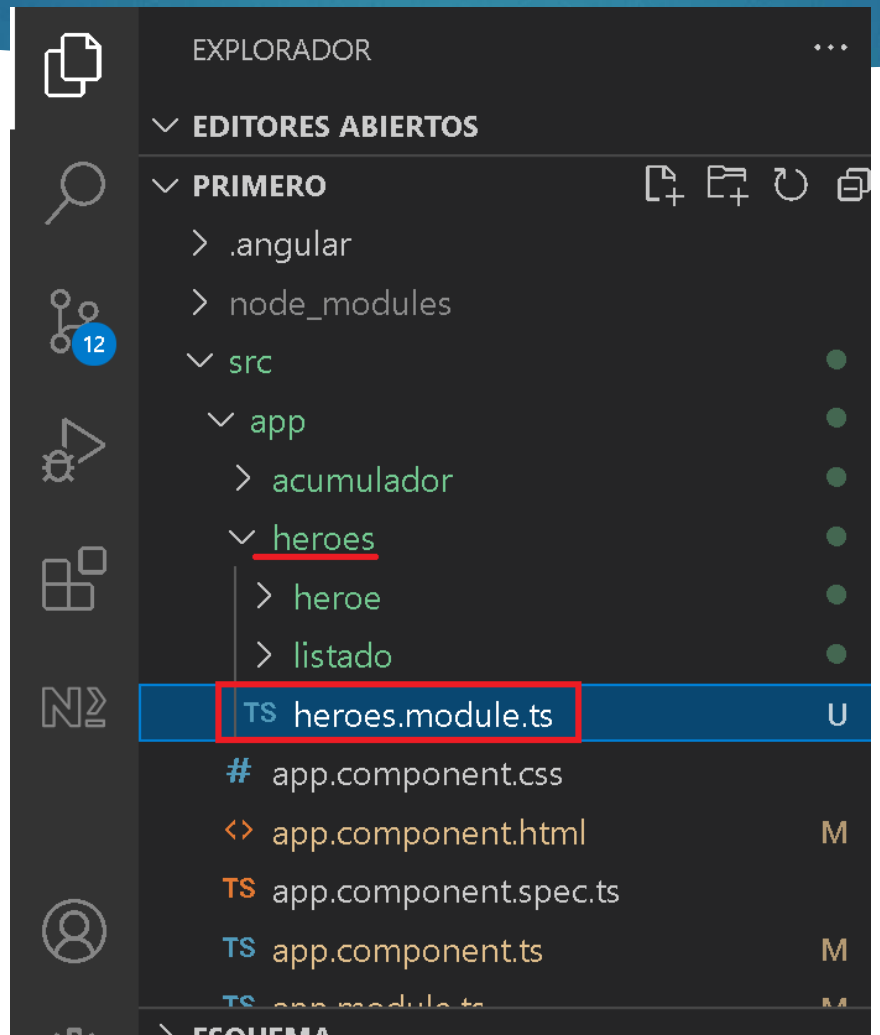
Ng-Template y el ngIf-else, atravez de una referencia local se puede llamar a templates



Módulos

- Los módulos lo podemos tener en varios componentes, esto nos ayuda a encapsular las configuraciones de cada modulo, a no acumular todo el proyecto en un solo modulo y también mejorando el rendimiento no llamando a clases y vistas que no serán utilizadas inicialmente.

En héroes creamos un nuevo modulo

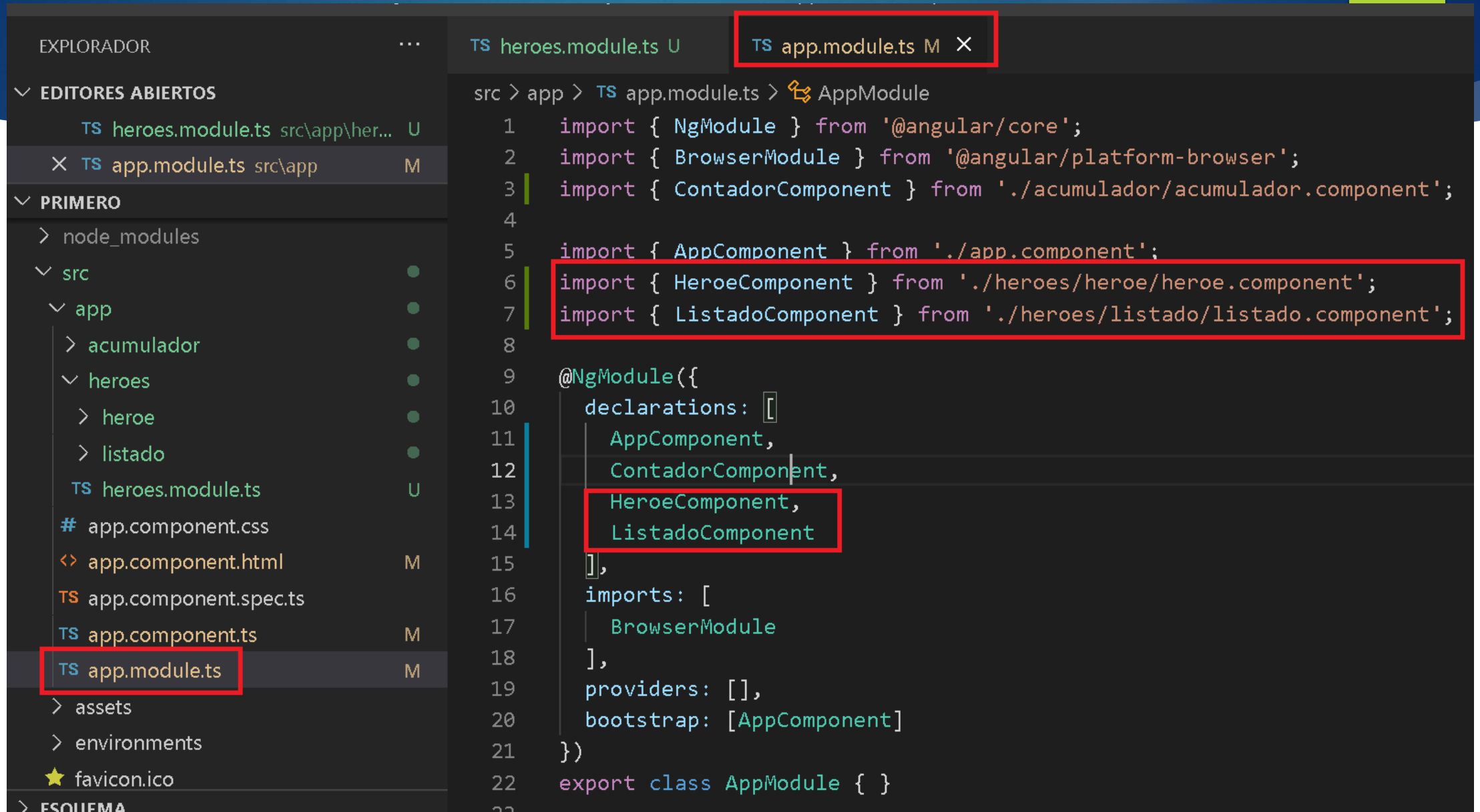


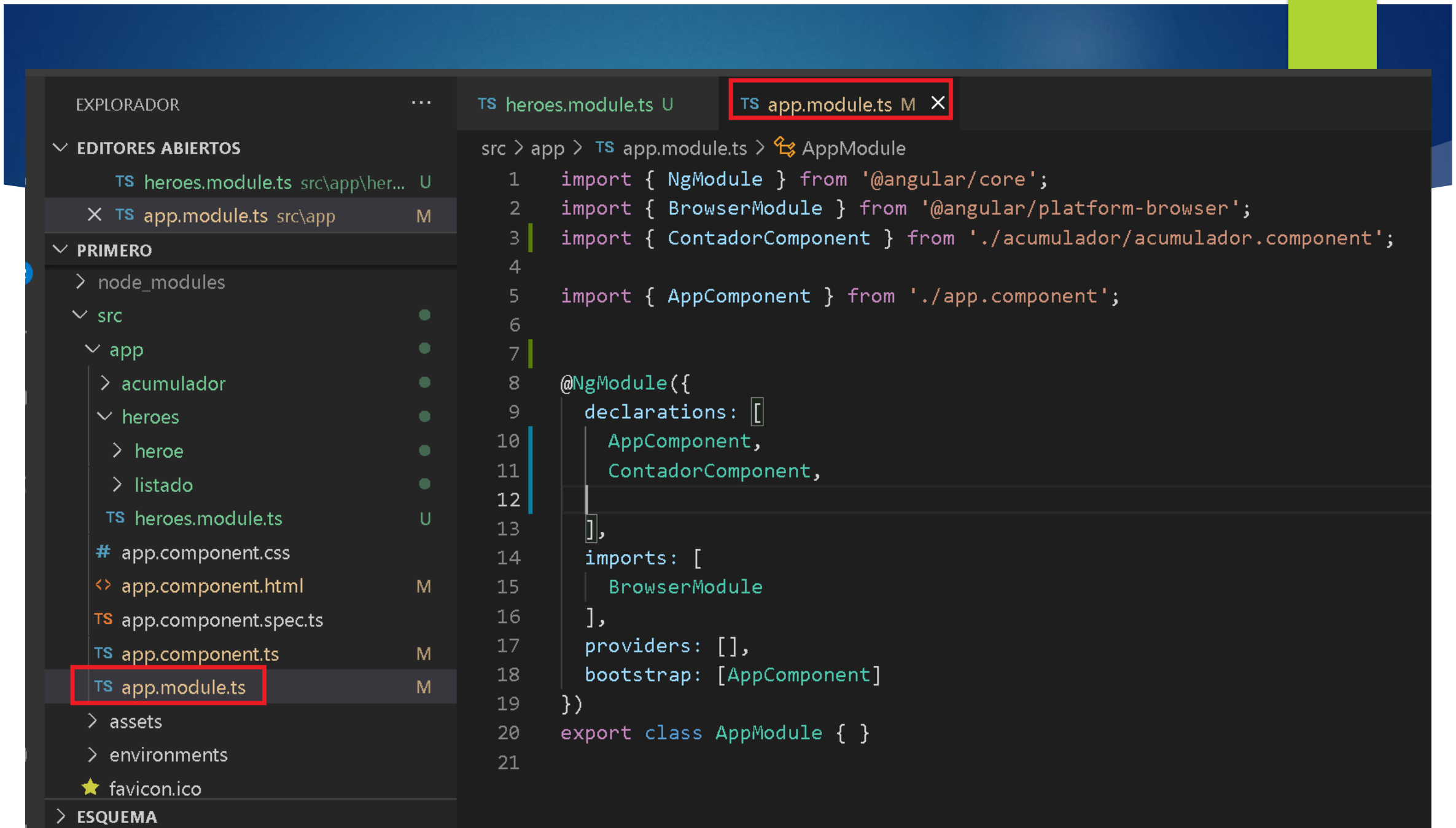
TS heroes.module.ts U

src > app > heroes > TS heroes.module.ts > HeroesModule

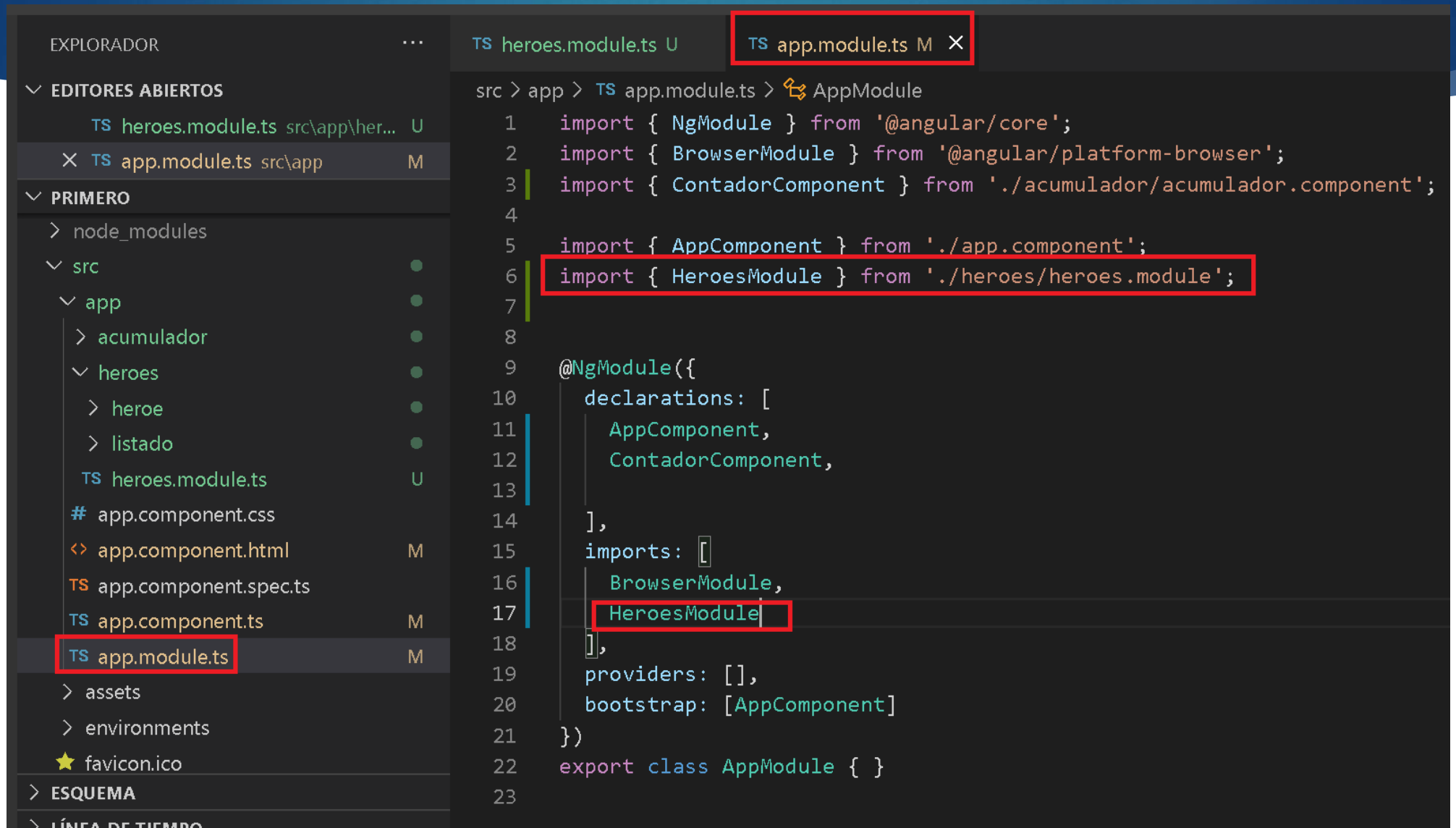
```ts
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { HeroeComponent } from './heroe/heroe.component';
import { ListadoComponent } from './listado/listado.component';

@NgModule({
  declarations: [
    HeroeComponent,
    ListadoComponent
  ],
  exports: [

    ListadoComponent
  ],
  imports: [
    CommonModule
  ]
})

export class HeroesModule{}
```

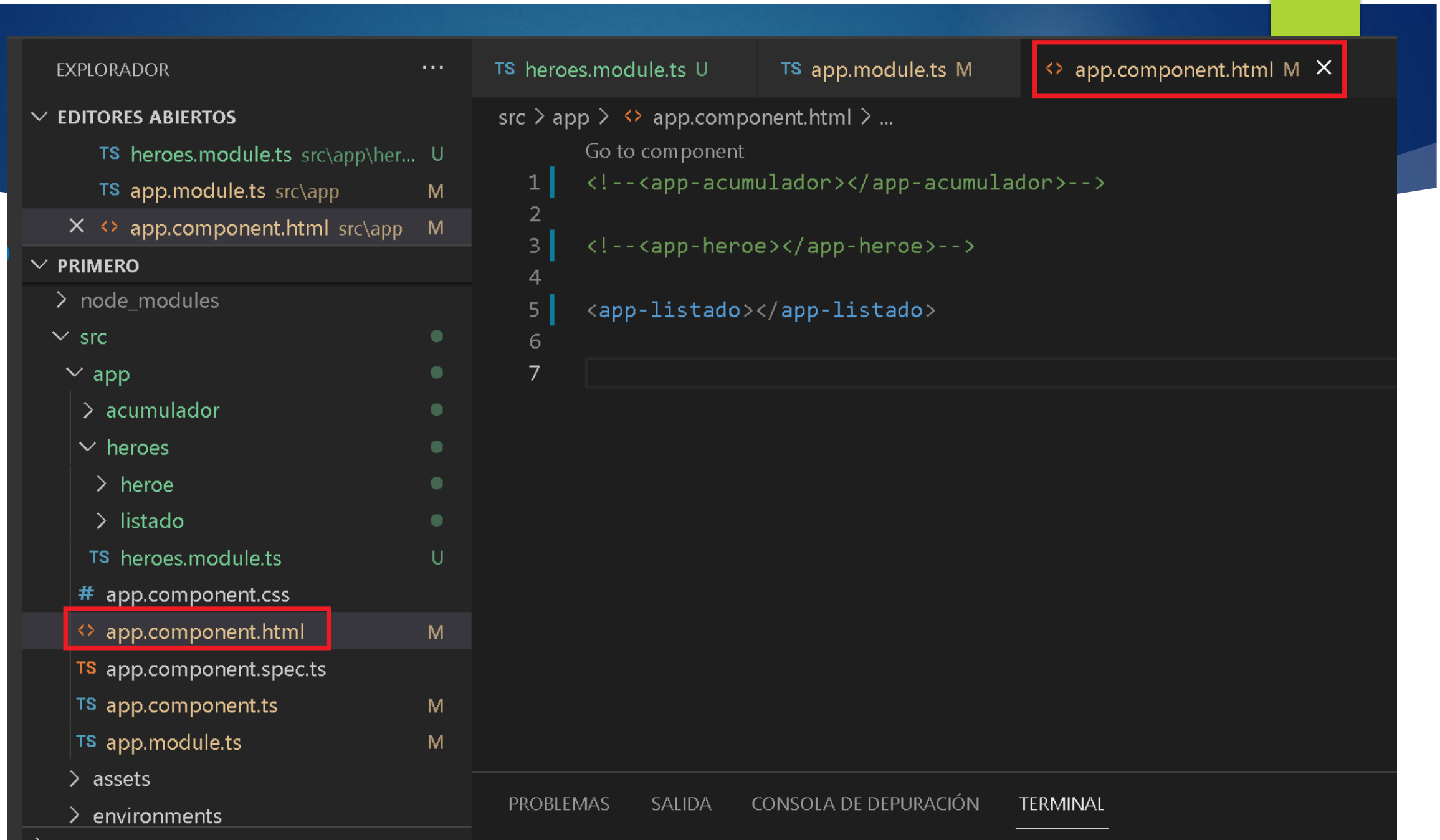
Removemos estas llamadas de app.module.ts

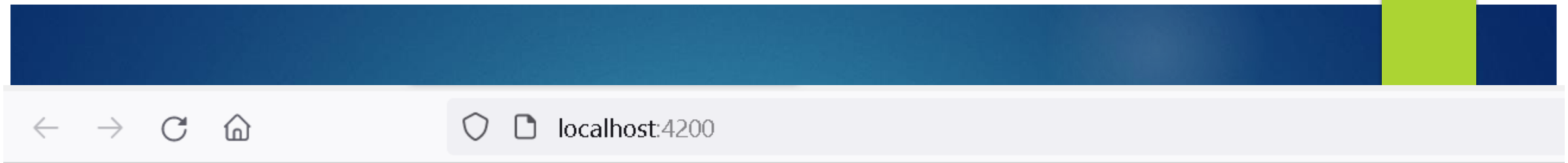



EXPLORADOR
EDITORES ABIERTOS
heroes.module.ts src\app\her... U
app.module.ts src\app M
PRIMERO
node_modules
src
app
acumulador
heroes
heroe
listado
heroes.module.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
ESQUEMA
heroes.module.ts U
app.module.ts M
src > app > app.module.ts > AppModule
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ContadorComponent } from './acumulador/acumulador.component';
import { AppComponent } from './app.component';
@NgModule({
declarations: [
AppComponent,
ContadorComponent,
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

Ahora agregamos al modulo creado




EXPLORADOR
EDITORES ABIERTOS
heroes.module.ts src\app\her... U
app.module.ts src\app M
app.component.html src\app M
PRIMERO
node_modules
src
app
acumulador
heroes
heroe
listado
heroes.module.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
heroes.module.ts U
app.module.ts M
app.component.html M
src > app > app.component.html > ...
Go to component
1 <!--<app-acumulador></app-acumulador>-->
2
3 <!--<app-heroe></app-heroe>-->
4
5 <app-listado></app-listado>
6
7
PROBLEMAS
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL

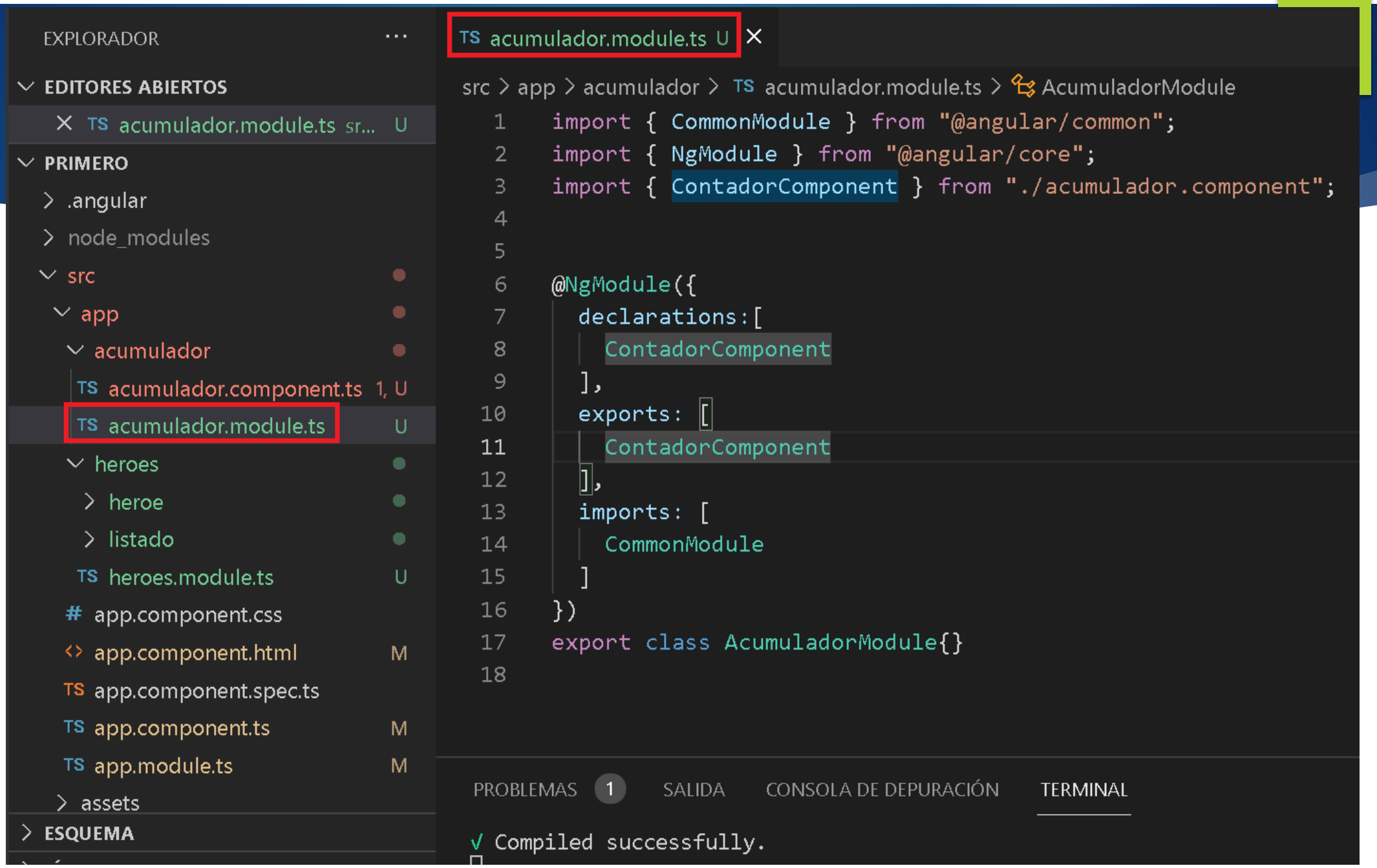


listado de Héroes

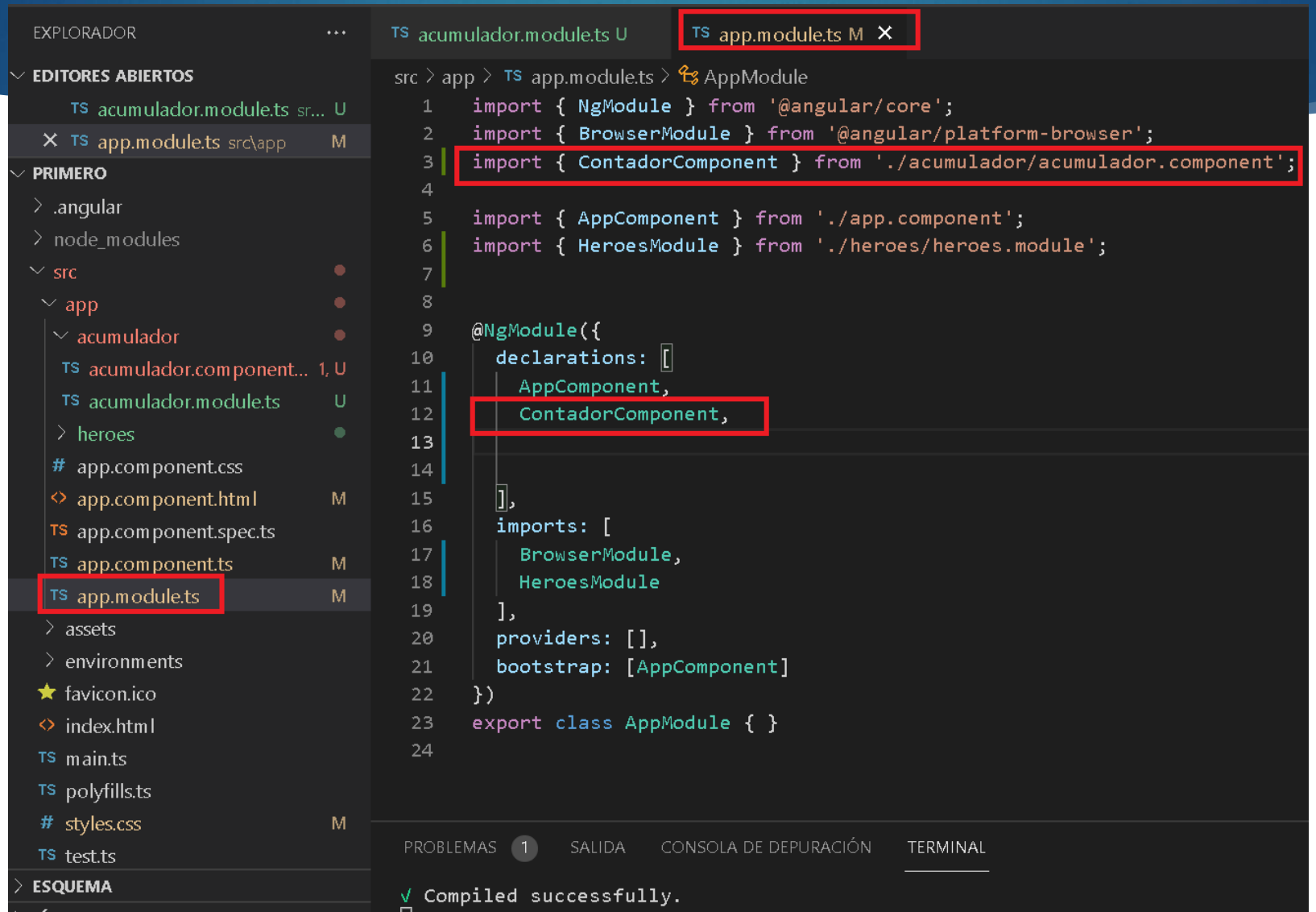
- 1 - Spiderman
- 2 - Ironman
- 3 - Thor
- 4 - Hulk
- 5 - Capitan America
- 6 - Antman

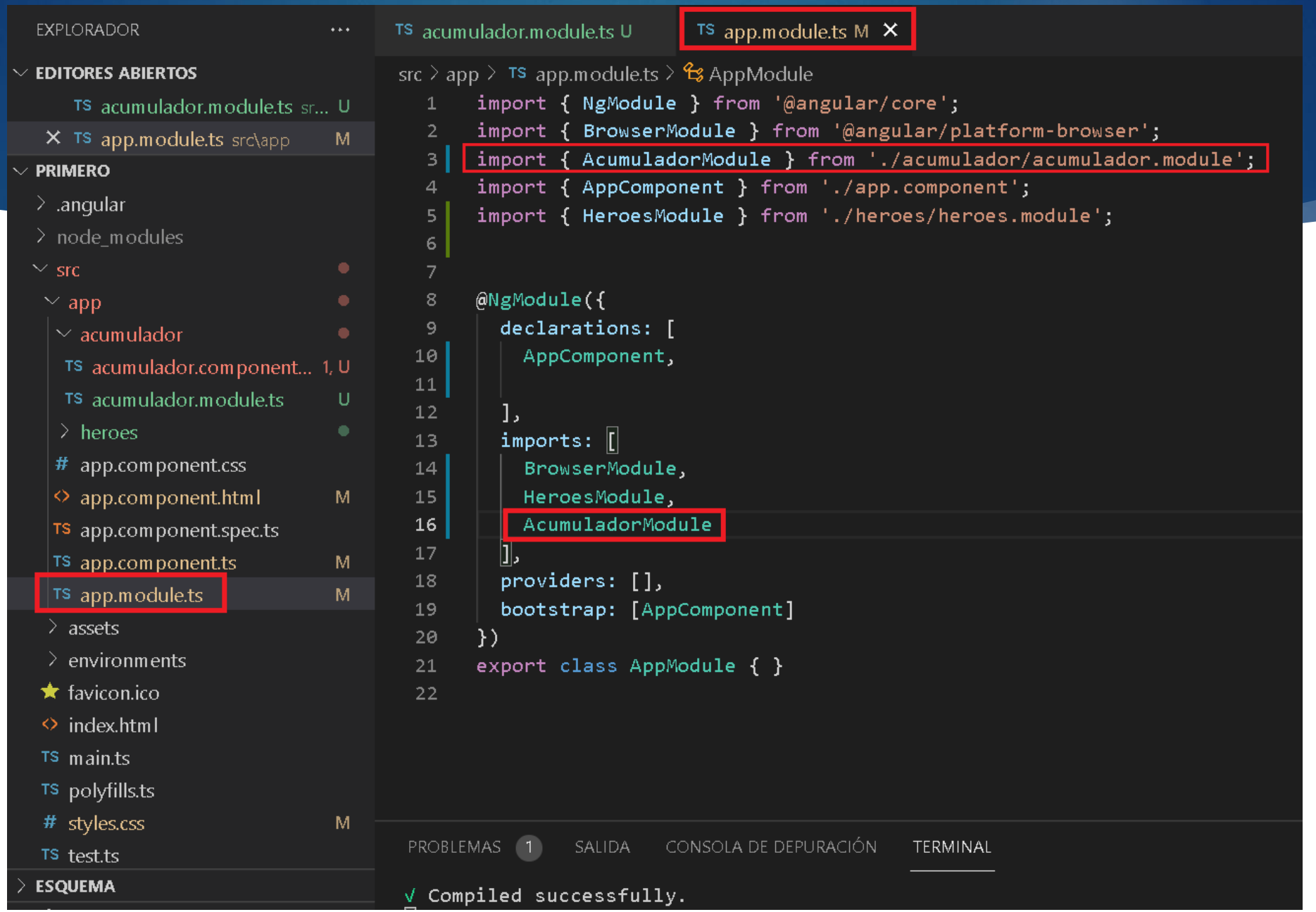
No ha borrado nada.

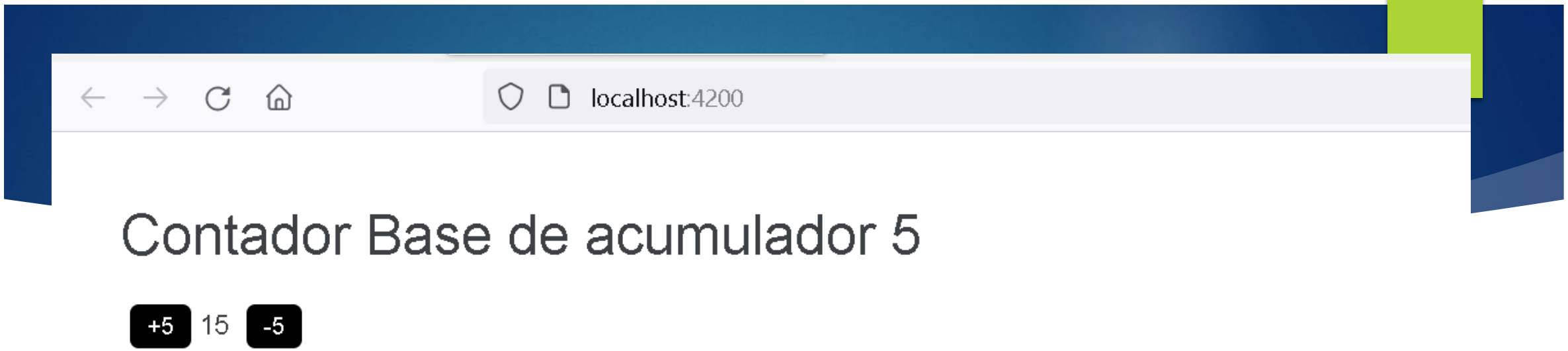
Borrar Héroe


EXPLORADOR
EDITORES ABIERTOS
TS acumulador.module.ts sr... U
PRIMERO
.angular
node_modules
src
app
acumulador
TS acumulador.component.ts 1, U
TS acumulador.module.ts U
heroes
heroe
listado
TS heroes.module.ts U
app.component.css
app.component.html M
app.component.spec.ts
TS app.component.ts M
TS app.module.ts M
assets
ESQUEMA
TS acumulador.module.ts U
src > app > acumulador > TS acumulador.module.ts > AcumuladorModule
1 import { CommonModule } from "@angular/common";
2 import { NgModule } from "@angular/core";
3 import { ContadorComponent } from "./acumulador.component";
4
5
6 @NgModule({
7 declarations:[
8 ContadorComponent
9],
10 exports: [
11 ContadorComponent
12],
13 imports: [
14 CommonModule
15]
16 })
17 export class AcumuladorModule{}
18
PROBLEMAS 1
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
√ Compiled successfully.

Removemos esto




EXPLORADOR
EDITORES ABIERTOS
acumulador.module.ts sr... U
app.module.ts src\app M
PRIMERO
.angular
node_modules
src
app
acumulador
acumulador.component... 1, U
acumulador.module.ts U
heroes
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css M
test.ts
ESQUEMA
acumulador.module.ts U
app.module.ts M
src > app > app.module.ts > AppModule
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { AcumuladorModule } from './acumulador/acumulador.module';
4 import { AppComponent } from './app.component';
5 import { HeroesModule } from './heroes/heroes.module';
6
7
8 @NgModule({
9 declarations: [
10 AppComponent,
11
12],
13 imports: [
14 BrowserModule,
15 HeroesModule,
16 AcumuladorModule
17],
18 providers: [],
19 bootstrap: [AppComponent]
20 })
21 export class AppModule { }
22
PROBLEMAS 1
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
√ Compiled successfully.


localhost:4200
Contador Base de acumulador 5
+5
15
-5

Creamos un nuevo modulo en el mismo proyecto

- Comando
- ng g module nombre_modulo

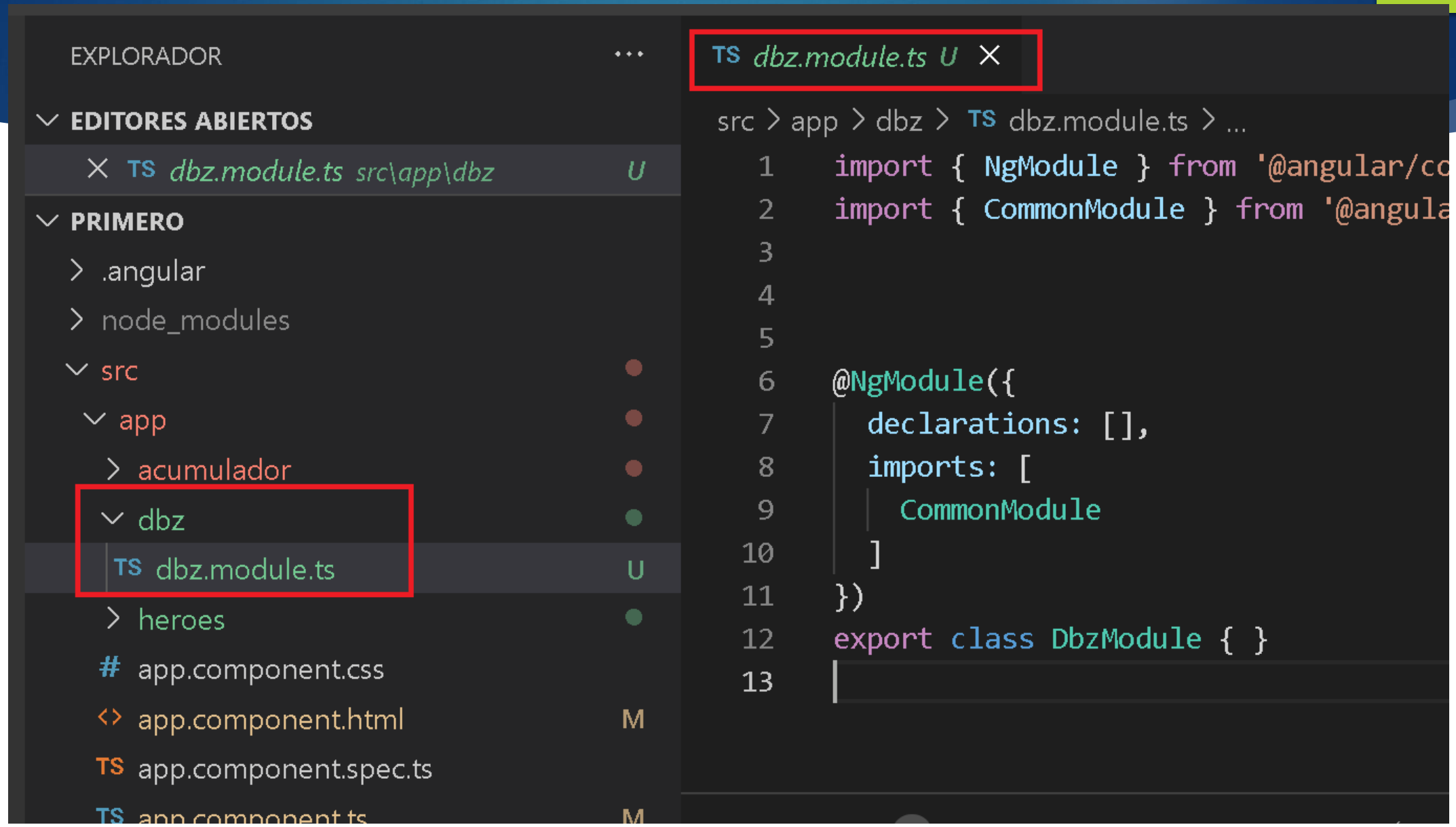
Creamos otro componente en el mismo proyecto

- En este ejemplo vemos como debemos de crear los componentes inicialmente

Generamos el modulo por comando

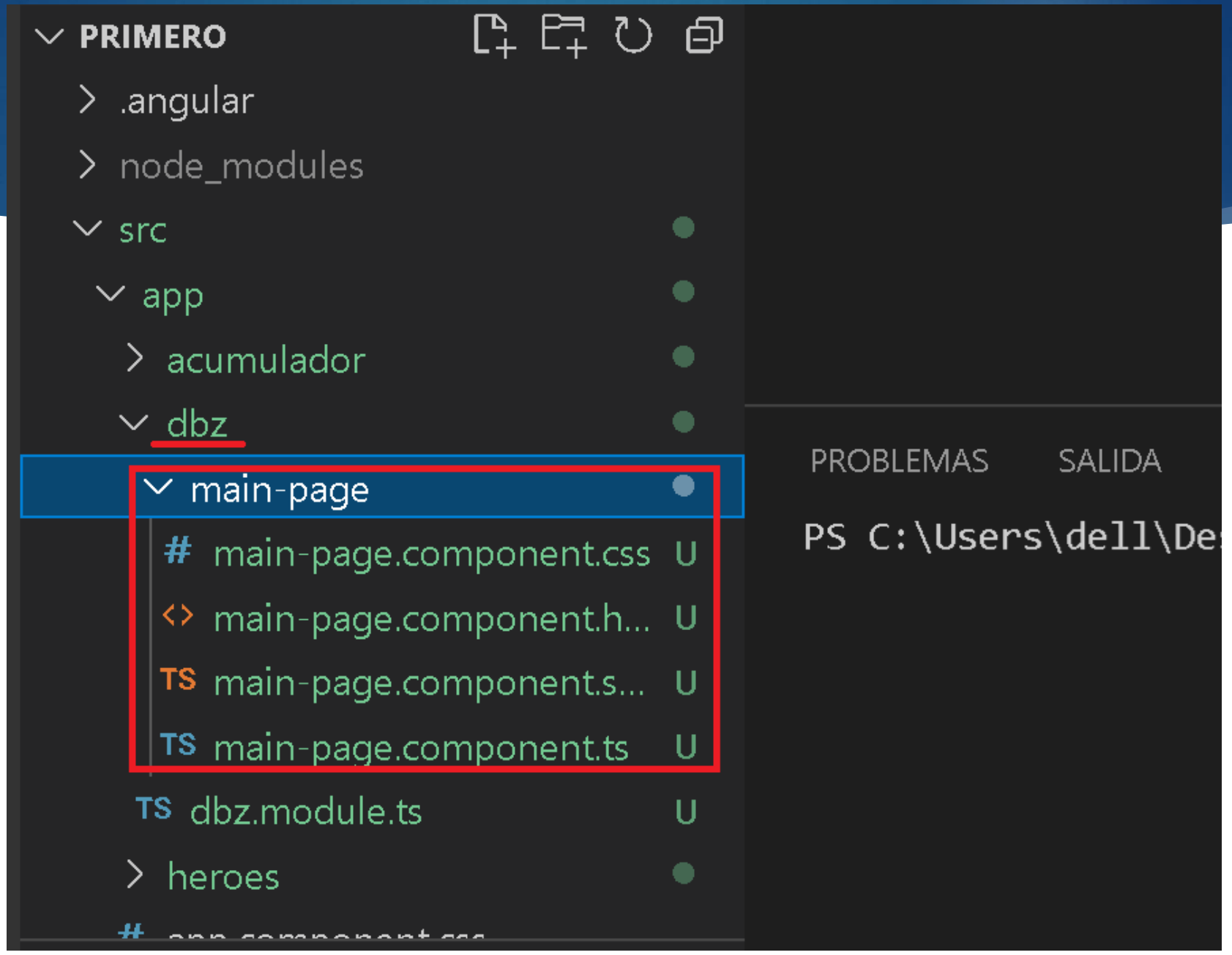
```
      web-worker
PS C:\Users\dell\Desktop\pruebas\primero> ng g module dbz
CREATE src/app/dbz/dbz.module.ts (189 bytes)
PS C:\Users\dell\Desktop\pruebas\primero>
```

Revisamos lo generado

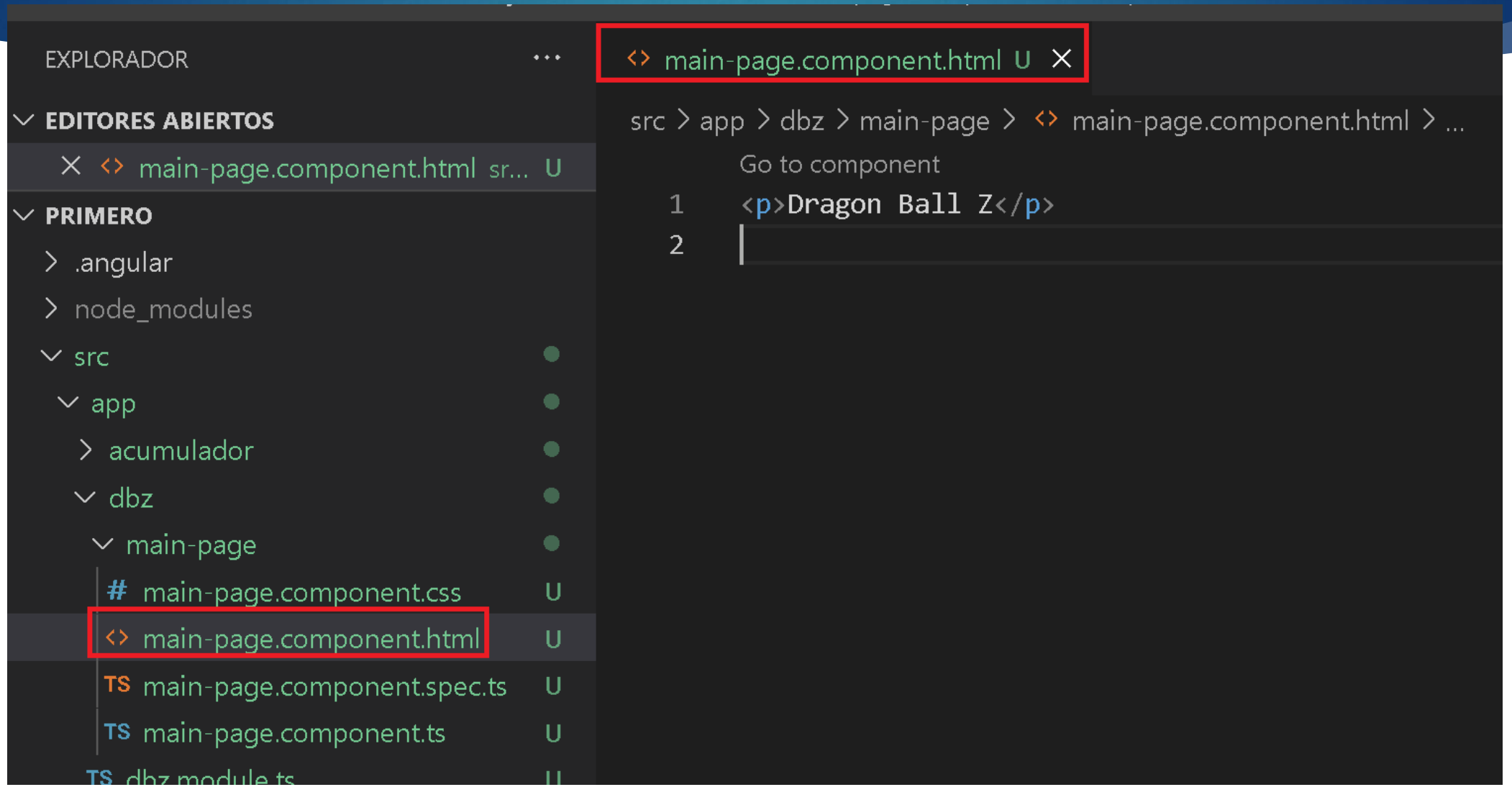


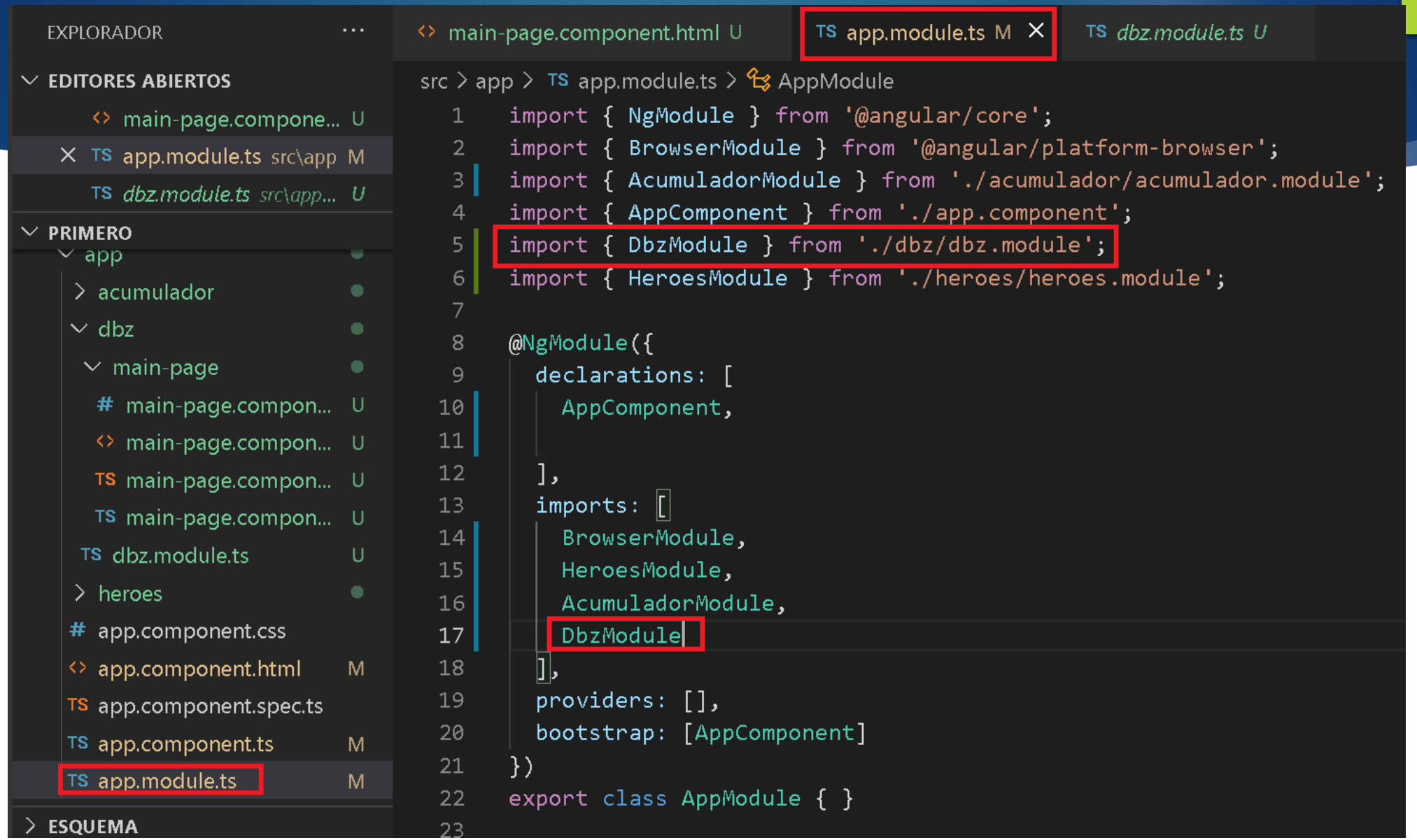
Agrego un componente

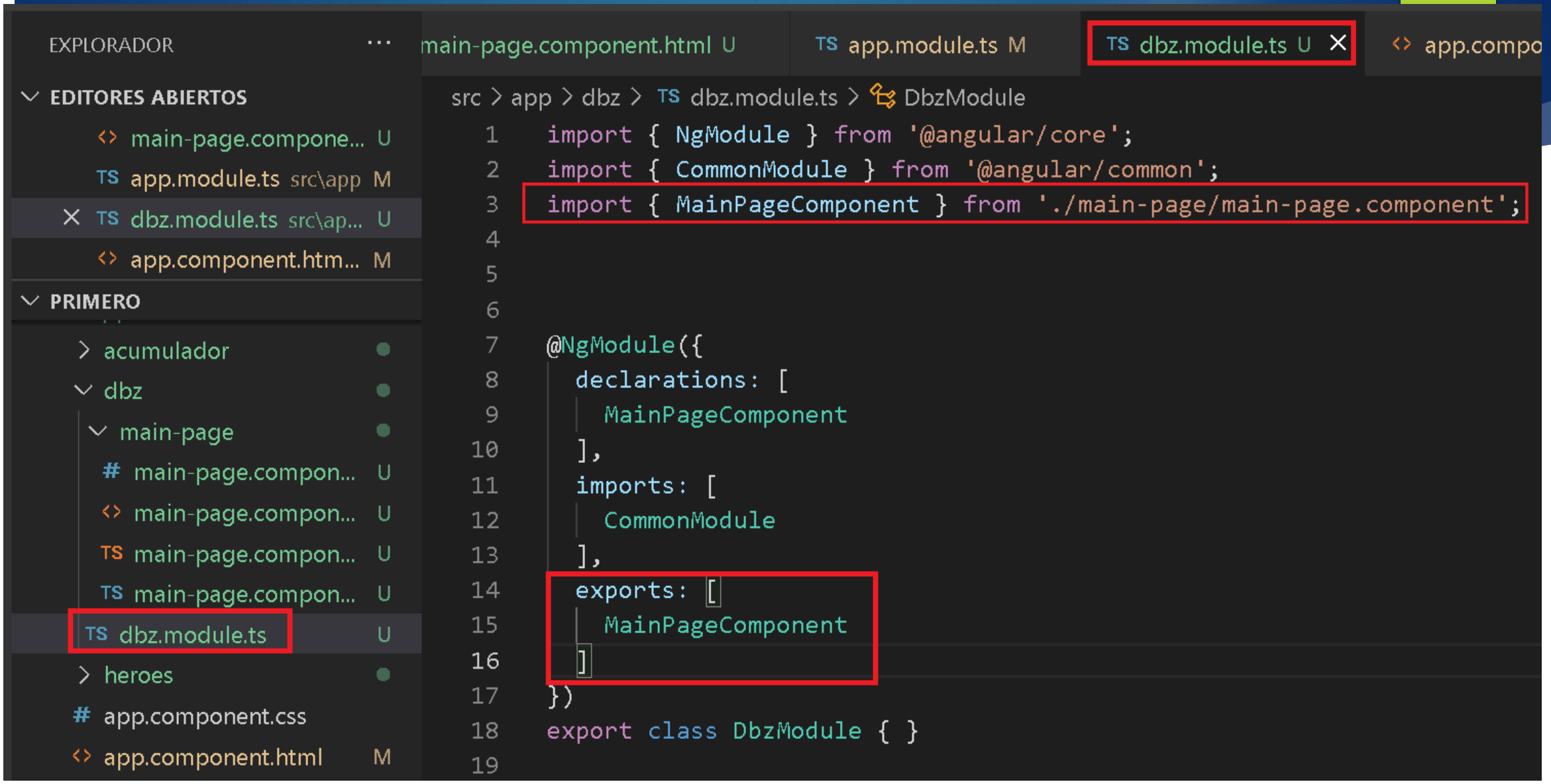
```
PS C:\Users\dell\Desktop\pruebas\primero> ng g c dbz/mainPage

CREATE src/app/dbz/main-page/main-page.component.html (24 byt
es)
CREATE src/app/dbz/main-page/main-page.component.spec.ts (641 bytes)
CREATE src/app/dbz/main-page/main-page.component.ts (286 bytes)
CREATE src/app/dbz/main-page/main-page.component.css (0 bytes)
UPDATE src/app/dbz/dbz.module.ts (283 bytes)
PS C:\Users\dell\Desktop\pruebas\primero>
```


PRIMERO
.angular
node_modules
src
app
acumulador
dbz
main-page
main-page.component.css U
main-page.component.h... U
main-page.component.s... U
main-page.component.ts U
dbz.module.ts U
heroes
PROBLEMAS
SALIDA
PS C:\Users\dell\De

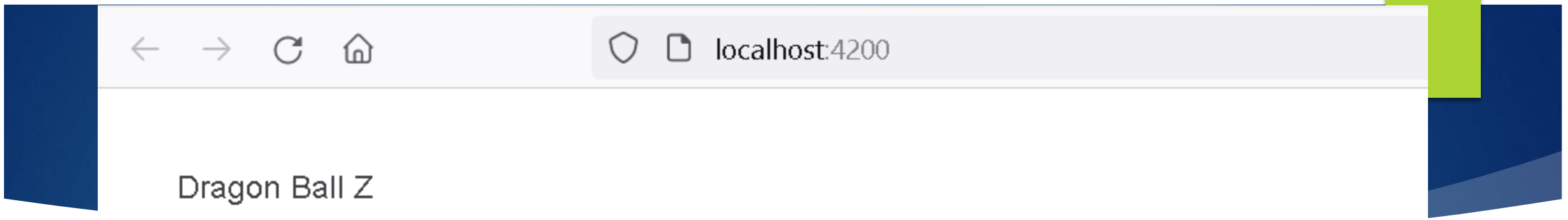
Realizamos los siguientes cambios



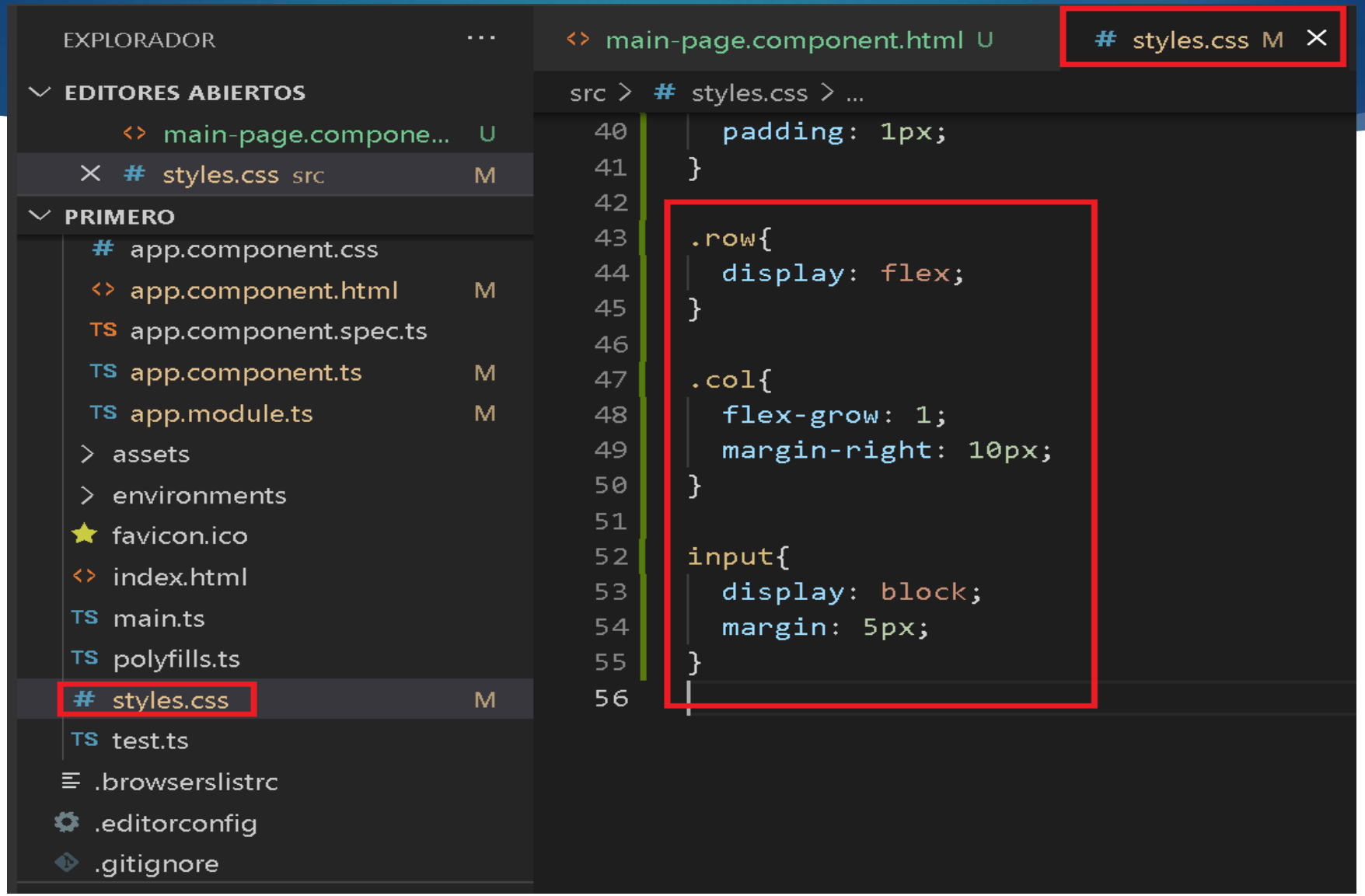

EXPLORADOR
EDITORES ABIERTOS
main-page.compone... U
app.module.ts src\app M
dbz.module.ts src\app... U
PRIMERO
app
acumulador
dbz
main-page
main-page.compon... U
main-page.compon... U
main-page.compon... U
main-page.compon... U
dbz.module.ts U
heroes
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
ESQUEMA
main-page.component.html U
app.module.ts M
dbz.module.ts U
src > app > app.module.ts > AppModule
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { AcumuladorModule } from './acumulador/acumulador.module';
4 import { AppComponent } from './app.component';
5 import { DbzModule } from './dbz/dbz.module';
6 import { HeroesModule } from './heroes/heroes.module';
7
8 @NgModule({
9 declarations: [
10 AppComponent,
11
12],
13 imports: [
14 BrowserModule,
15 HeroesModule,
16 AcumuladorModule,
17 DbzModule
18],
19 providers: [],
20 bootstrap: [AppComponent]
21 })
22 export class AppModule { }
23


EXPLORADOR
EDITORES ABIERTOS
main-page.compone... U
app.module.ts src\app M
dbz.module.ts src\ap... U
app.component.htm... M
PRIMERO
acumulador
dbz
main-page
main-page.compon... U
main-page.compon... U
main-page.compon... U
main-page.compon... U
dbz.module.ts U
heroes
app.component.css
app.component.html M
main-page.component.html U
app.module.ts M
dbz.module.ts U
app.compo
src > app > dbz > dbz.module.ts > DbzModule
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { MainPageComponent } from './main-page/main-page.component';
@NgModule({
declarations: [
MainPageComponent
],
imports: [
CommonModule
],
exports: [
MainPageComponent
]
})
export class DbzModule { }


EXPLORADOR
EDITORES ABIERTOS
main-page.compone... U
app.module.ts src\app M
dbz.module.ts src\ap... U
app.component.htm... M
PRIMERO
acumulador
dbz
main-page
main-page.compon... U
main-page.compon... U
main-page.compon... U
main-page.compon... U
dbz.module.ts U
heroes
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
main-page.component.html U
app.module.ts M
dbz.module.ts U
app.component.html M
src > app > app.component.html > ...
Go to component
1 <!--<app-acumulador></app-acumulador>-->
2
3 <!--<app-heroe></app-heroe>-->
4
5 <!--<app-listado></app-listado>-->
6
7 <app-main-page></app-main-page>
8
9


localhost:4200
Dragon Ball Z

Modificamos de esta manera

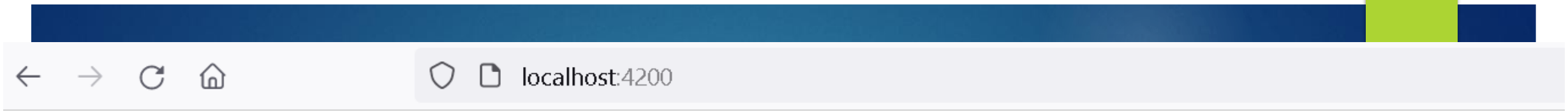


main-page.component.html U

c > app > dbz > main-page > main-page.component.html > ...

Go to component

```
<h1>Dragon Ball Z</h1>
<hr>

<div class="row">

  <div class="col">
   <h3>Personajes</h3>
   <hr>
    <ul>
      <li>Krilin - 1000</li>
      <li>Gohan - 1500</li>
      <li>Goku - 20.000</li>
    </ul>
  </div>

  <div class="col">
      <h3>Agregar</h3>
      <hr>

      <form>
        <input type="text" placeholder="Nombre">
        <input type="text" placeholder="Poder">
        <button>Agregar</button>
      </form>
  </div>
</div>

```



Dragon Ball Z

Personajes

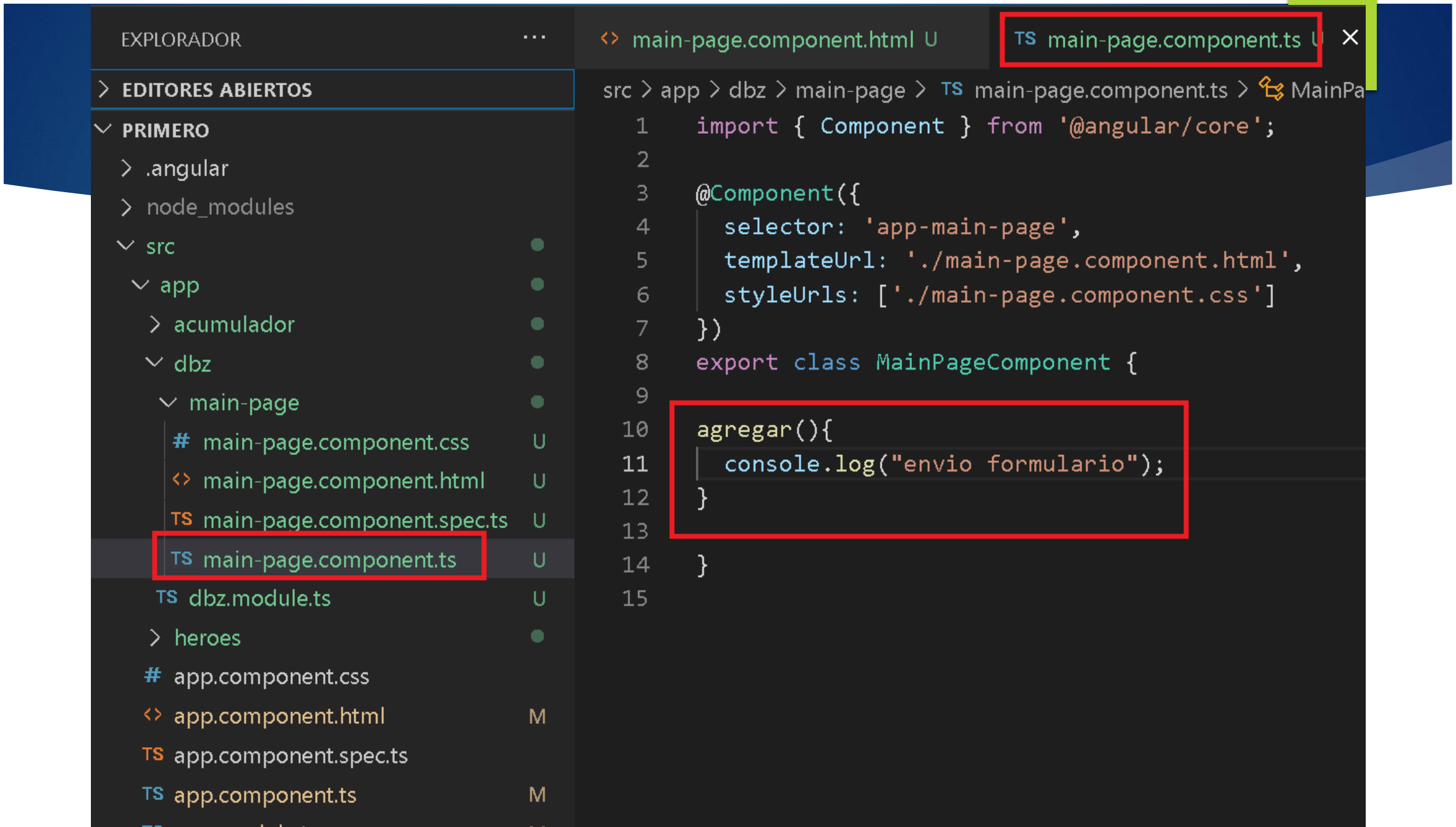
- Krilin - 1000
- Gohan - 1500
- Goku - 20.000

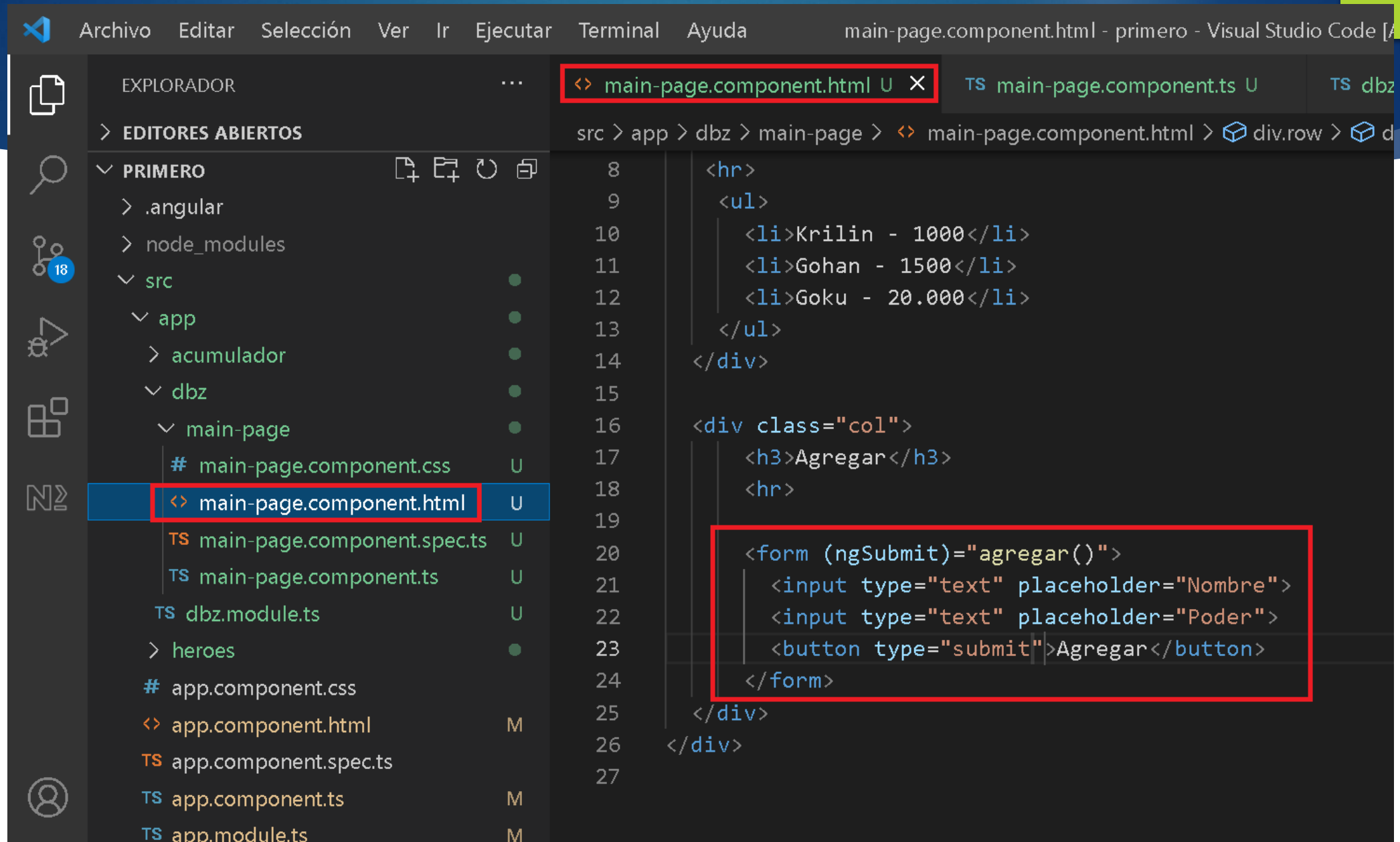
Agregar

Nombre

Poder

Agregar


EXPLORADOR
EDITORES ABIERTOS
PRIMERO
.angular
node_modules
src
app
acumulador
dbz
main-page
main-page.component.css U
main-page.component.html U
main-page.component.spec.ts U
main-page.component.ts U
dbz.module.ts U
heroes
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
main-page.component.html U
main-page.component.ts U
src > app > dbz > main-page > main-page.component.ts > MainPa
1 import { Component } from '@angular/core';
2
3 @Component({
4 selector: 'app-main-page',
5 templateUrl: './main-page.component.html',
6 styleUrls: ['./main-page.component.css']
7 })
8 export class MainPageComponent {
9
10 agregar(){
11 console.log("envio formulario");
12 }
13
14 }
15


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
main-page.component.html - primero - Visual Studio Code
EXPLORADOR
EDITORES ABIERTOS
PRIMERO
.angular
node_modules
src
app
acumulador
dbz
main-page
main-page.component.css U
main-page.component.html U
main-page.component.spec.ts U
main-page.component.ts U
dbz.module.ts U
heroes
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
main-page.component.html U
main-page.component.ts U
src > app > dbz > main-page > main-page.component.html > div.row
<hr>

Krilin - 1000
Gohan - 1500
Goku - 20.000

<h3>Agregar</h3>
<hr>
<form (ngSubmit)="agregar()">
<input type="text" placeholder="Nombre">
<input type="text" placeholder="Poder">
<button type="submit">Agregar</button>
</form>

FormsModule

- Es un modulo para manejar formulario en angular, para utilizarlo debemos invocarlo en el modulo del componente

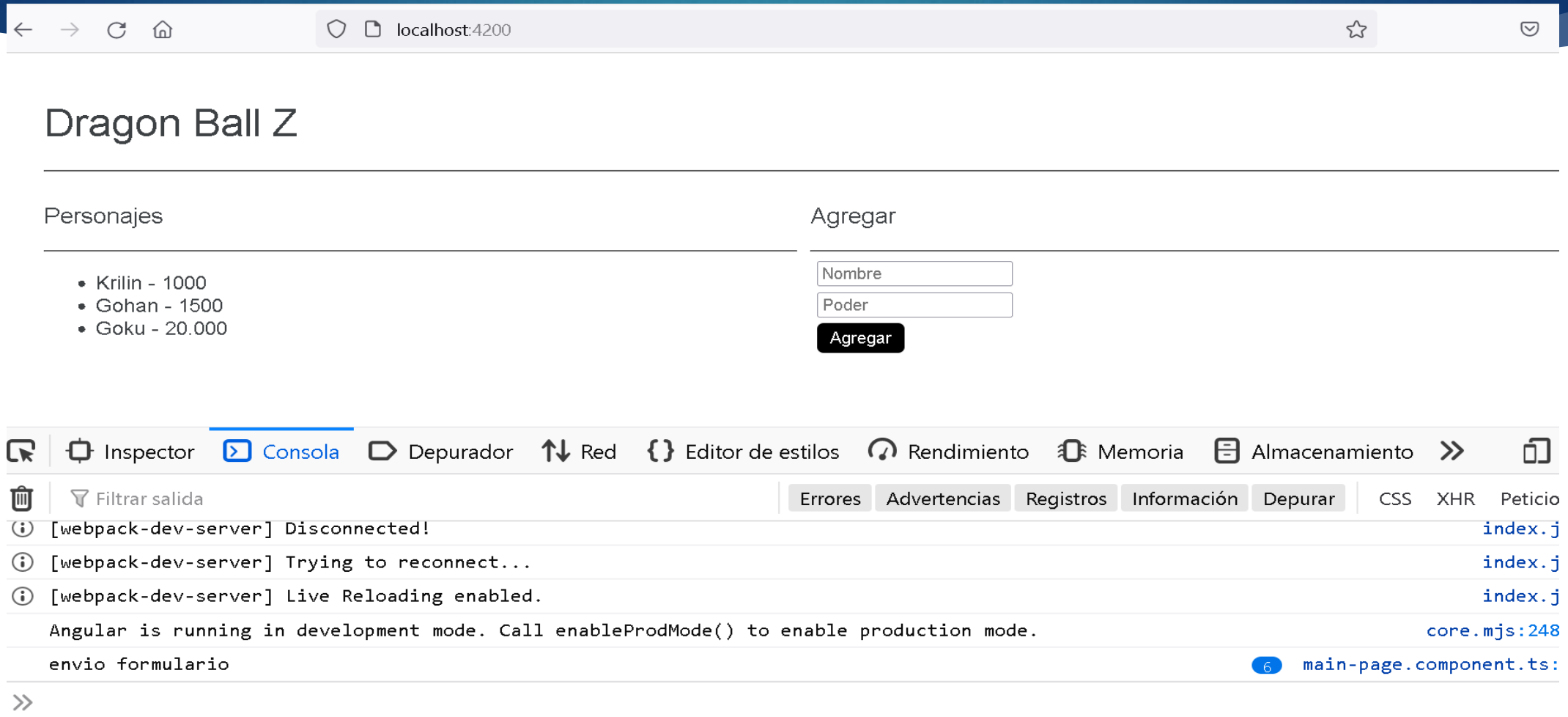
Importamos el FormsModule

main-page.component.html U | dbz.module.ts U

src > app > dbz > dbz.module.ts > DbzModule

```
1   import { NgModule } from '@angular/core';
2   import { CommonModule } from '@angular/common';
3   import { MainPageComponent } from './main-page/main-page.component';
4   import { FormsModule } from '@angular/forms';
5
6
7
8   @NgModule({
9     declarations: [
10      MainPageComponent
11    ],
12    imports: [
13      CommonModule,
14      FormsModule
15    ],
16    exports: [
17      MainPageComponent
18    ]
19  })
20  export class DbzModule { }
21
```

Una vez implementado el modulo FormsModule no debe refrescar al llamar al submit



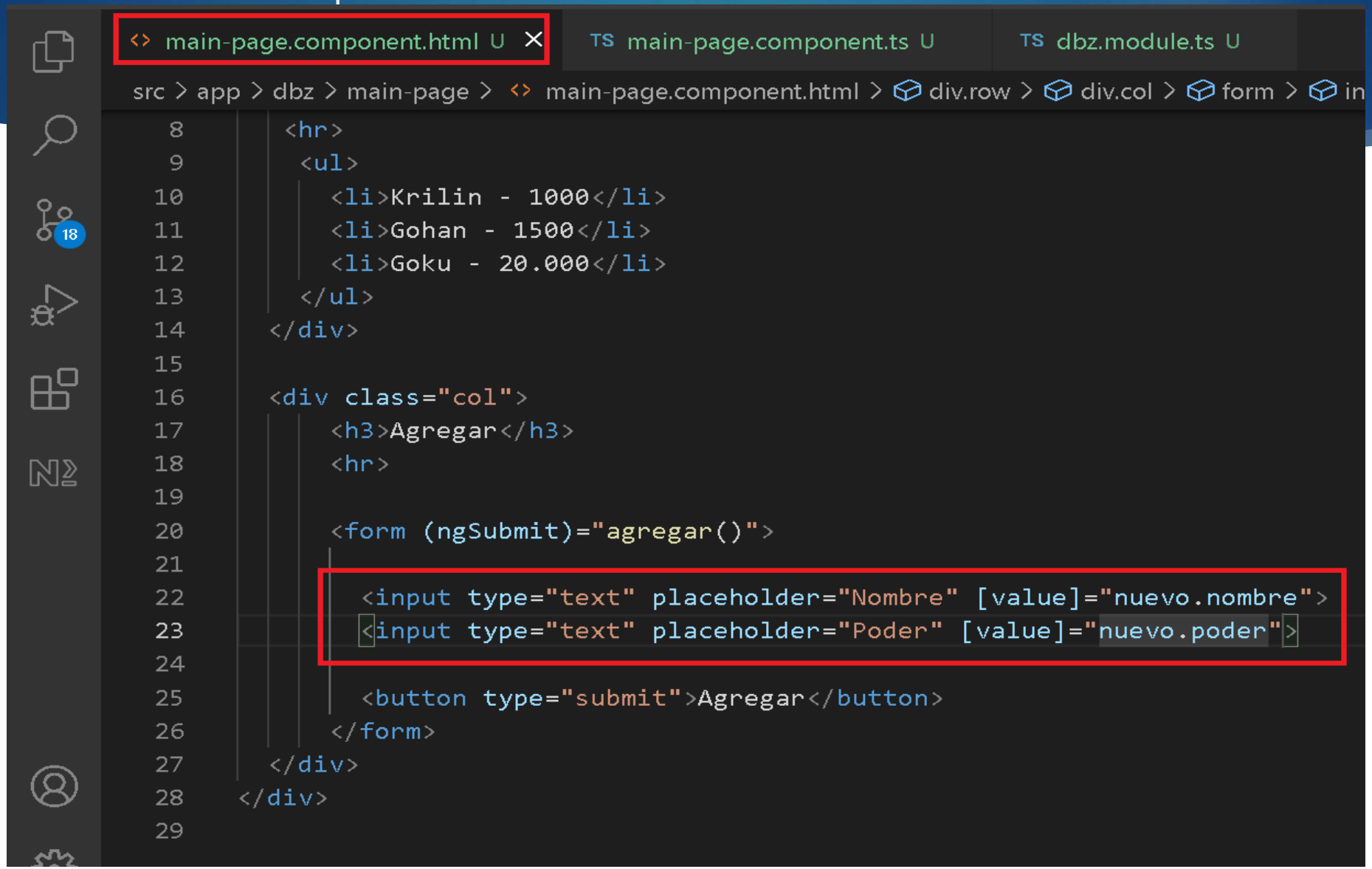
Continuamos con el formulario

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

src > app > dbz > main-page > main-page.component.ts > MainPageComponent > agregar

```ts
import { Component } from '@angular/core';

interface Personaje{
  nombre:string;
  poder:number;
}

@Component({
  selector: 'app-main-page',
  templateUrl: './main-page.component.html',
  styleUrls: ['./main-page.component.css']
})
export class MainPageComponent {

  nuevo: Personaje ={
    nombre:"Vegeta",
    poder: 20000
  }

  agregar(){
    console.log("envio formulario");
  }

}
```

El [value] nos indica que queremos capturar un valor desde nuestra clase componente ts




localhost:4200
Dragon Ball Z
Personajes
Krilin - 1000
Gohan - 1500
Goku - 20.000
Agregar
Vegeta
20000
Agregar

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

src > app > dbz > main-page > main-page.component.ts > MainPageComponent > agregar

```
12    })
13    export class MainPageComponent {
14
15
16    nuevo: Personaje ={
17      nombre:"Vegeta",
18      poder: 20000
19    }
20
21    cambiarNombre( event:any) {
22     console.log(event);
23    }
24
25    agregar(){
26      console.log(this.nuevo);
27    }
28
29    }
30
```

main-page.component.html U

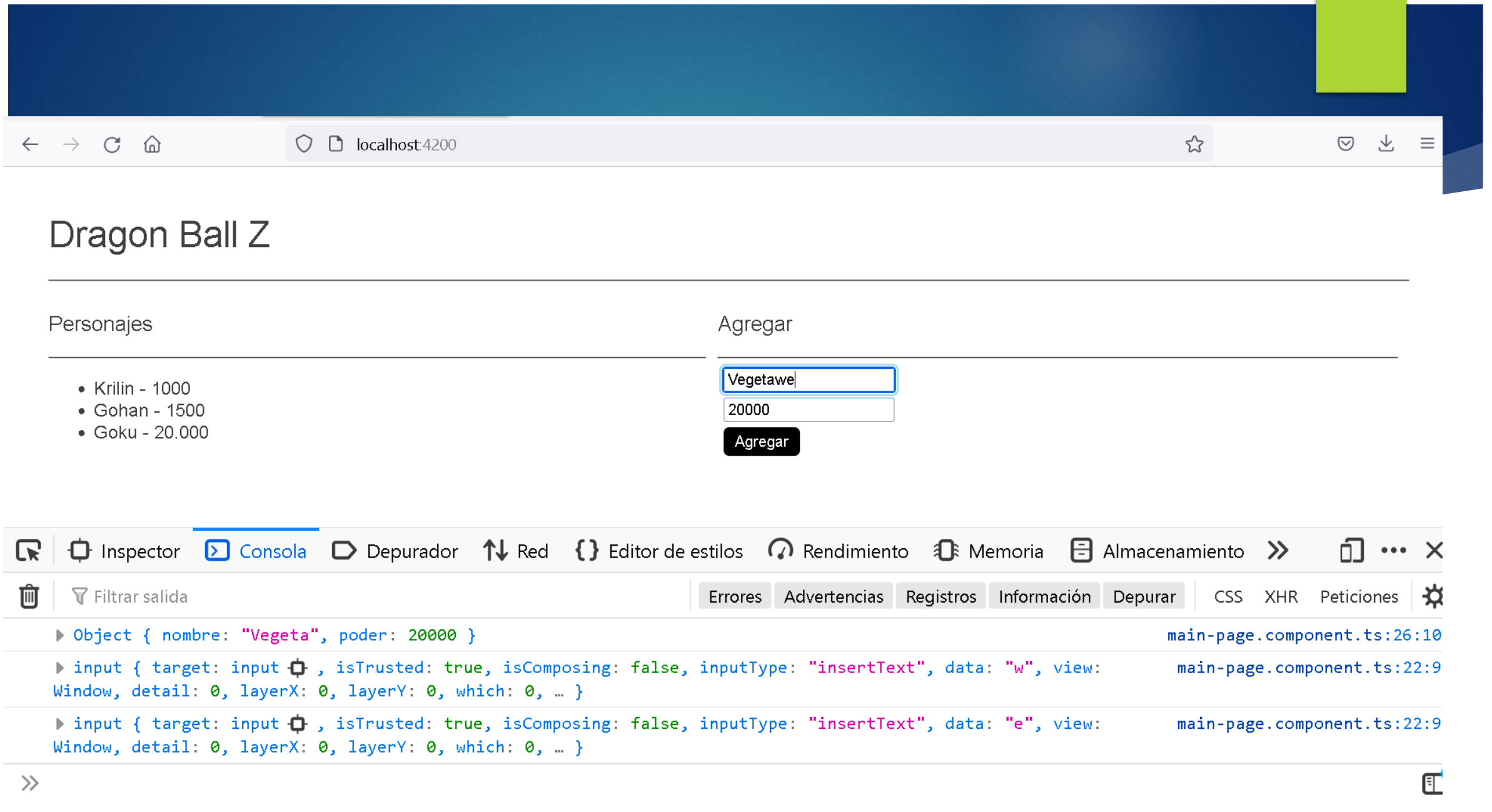
main-page.component.ts U

dbz.module.ts U

src > app > dbz > main-page > main-page.component.html > div.row > div.col > form >

```
16      <div class="col">
17          <h3>Agregar</h3>
18          <hr>
19
20          <form (ngSubmit)="agregar()">
21
22              <input
23                  type="text"
24                  placeholder="Nombre"
25                  [value]="nuevo.nombre"
26                  (input)="cambiarNombre( $event )"
27              >
28
29              <input type="text" placeholder="Poder" [value]="nuevo.poder">
30
31              <button type="submit">Agregar</button>
32          </form>
33      </div>
34  </div>
```

ster* ⊗ 0 ⚠ 0 Lín. 26, col. 42 Espacios: 2 UTF-


localhost:4200
Dragon Ball Z
Personajes
Krilin - 1000
Gohan - 1500
Goku - 20.000
Agregar
Vegetawe
20000
Agregar
Inspector
Consola
Depurador
Red
Editor de estilos
Rendimiento
Memoria
Almacenamiento
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
Object { nombre: "Vegeta", poder: 20000 }
main-page.component.ts:26:10
input { target: input , isTrusted: true, isComposing: false, inputType: "insertText", data: "w", view: Window, detail: 0, layerX: 0, layerY: 0, which: 0, … }
main-page.component.ts:22:9
input { target: input , isTrusted: true, isComposing: false, inputType: "insertText", data: "e", view: Window, detail: 0, layerX: 0, layerY: 0, which: 0, … }
main-page.component.ts:22:9

ng Model : se trata de un enlace, entre algo que tienes en la definición del componente con un campo de formulario del template (vista) del componente.

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

src > app > dbz > main-page > main-page.component.html > div.row > div.col > form

```
26            (input)="cambiarNombre( $event )"
27          >-->
28          <input
29            type="text"
30            placeholder="Nombre"
31            name="nombre"
32            [(ngModel)] = "nuevo.nombre"
33            >
34
35
36          <input type="text" placeholder="Poder" [value]="nuevo.poder">
37
38          <button type="submit">Agregar</button>
39        </form>
40      </div>
41    </div>
42
```

Ya no necesitamos la función cambiarNombre

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

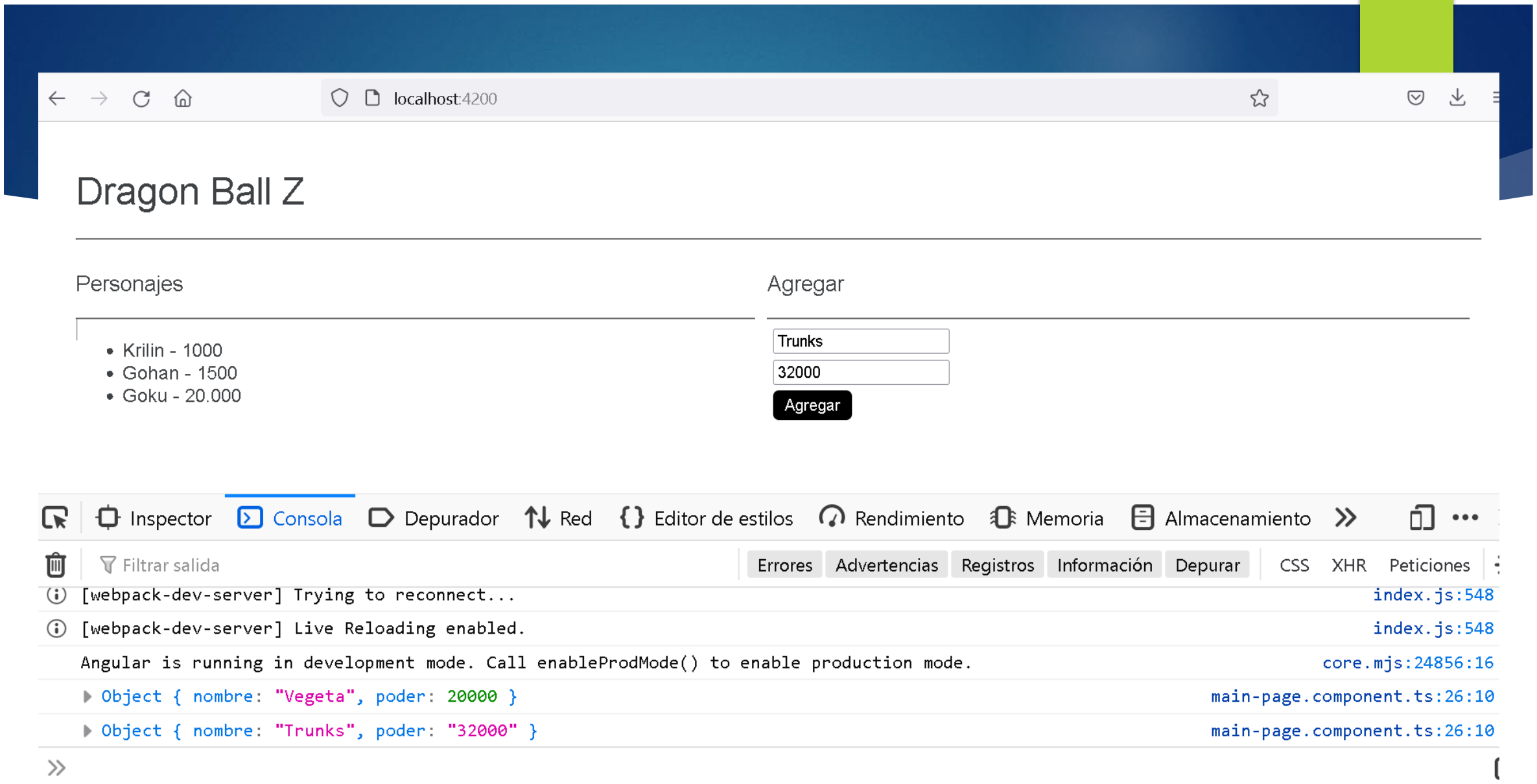
src > app > dbz > main-page > main-page.component.ts > MainPageComponent > agregar

```
12    })
13    export class MainPageComponent {
14
15
16    nuevo: Personaje ={
17      nombre:"Vegeta",
18      poder: 20000
19    }
20
21    /*cambiarNombre( event:any) {
22     console.log(event);
23    }*/
24
25    agregar(){
26      console.log(this.nuevo);
27    }
28
29    }
```

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

src > app > dbz > main-page > main-page.component.html > div.row > div.col > form > input

```
26            (input)="cambiarNombre( $event )"
27          >-->
28          <input
29            type="text"
30            placeholder="Nombre"
31            name="nombre"
32            [(ngModel)] = "nuevo.nombre"
33            >
34            <input
35            type="text"
36            placeholder="Poder"
37            name="poder"
38            [(ngModel)] = "nuevo.poder"
39            >
40
41          <button type="submit">Agregar</button>
42        </form>
43      </div>
44    </div>
```


localhost:4200
Dragon Ball Z
Personajes
Krilin - 1000
Gohan - 1500
Goku - 20.000
Agregar
Trunks
32000
Agregar
Inspector
Consola
Depurador
Red
Editor de estilos
Rendimiento
Memoria
Almacenamiento
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
[webpack-dev-server] Trying to reconnect...
index.js:548
[webpack-dev-server] Live Reloading enabled.
index.js:548
Angular is running in development mode. Call enableProdMode() to enable production mode.
core.mjs:24856:16
Object { nombre: "Vegeta", poder: 20000 }
main-page.component.ts:26:10
Object { nombre: "Trunks", poder: "32000" }
main-page.component.ts:26:10

En main-page.component.ts agregamos las siguientes lineas

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

src > app > dbz > main-page > main-page.component.ts > MainPageComponent > nuevo

```
13  export class MainPageComponent {
14
15
16    personajes: Personaje[] = [
17     {
18      nombre:"Vegeta",
19      poder: 20000
20     },{
21      nombre:"Goku",
22      poder: 25000
23     }
24    ];
25
26    nuevo :Personaje = {
27      nombre: "",
28      poder: 0
29    }
30
```

main-page.component.html U | main-page.component.ts U | dbz.module.ts U

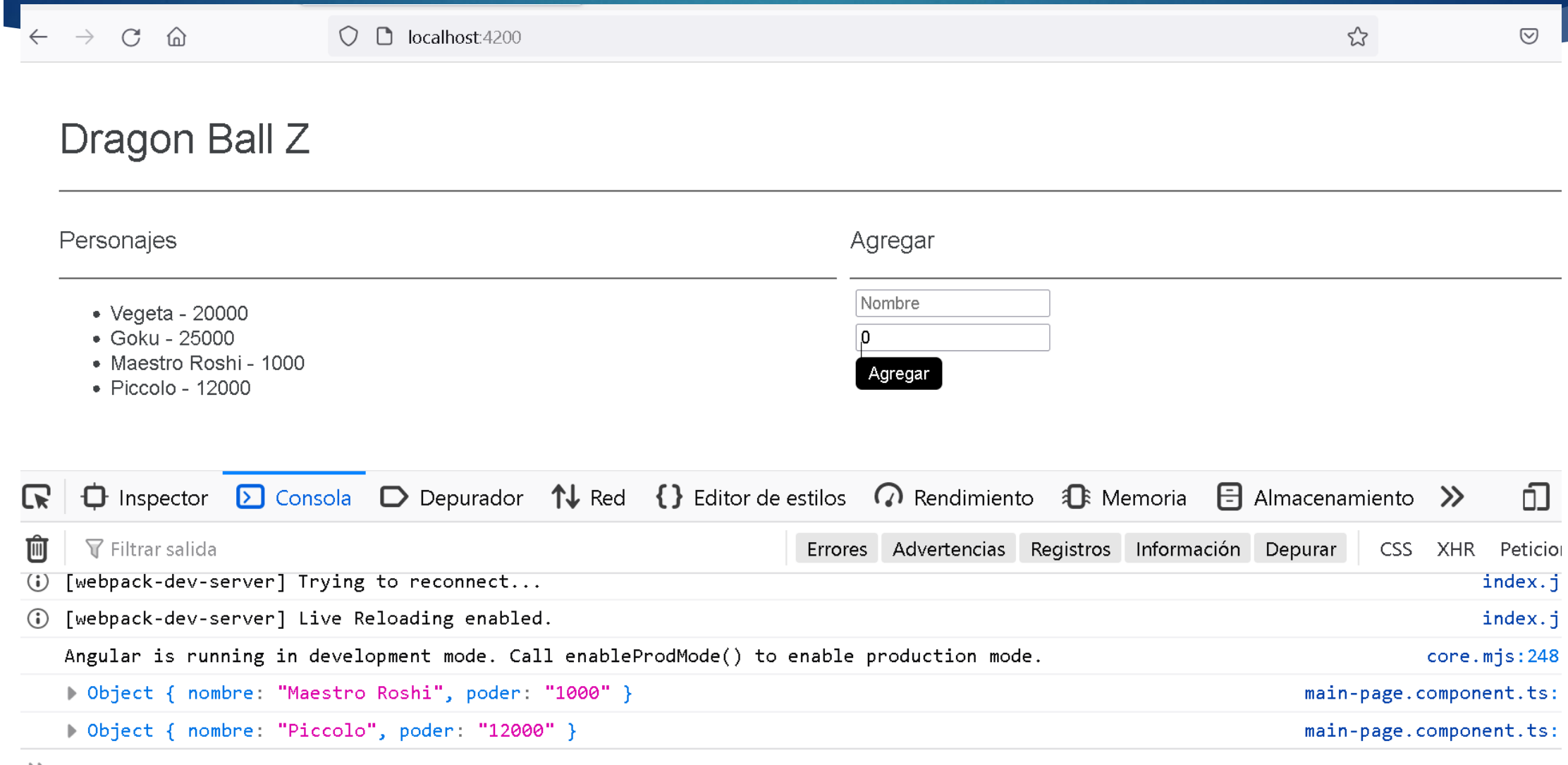
src > app > dbz > main-page > main-page.component.ts > MainPageComponent > nuevo

```
29    }
30
31
32    agregar(){
33      if(this.nuevo.nombre.trim().length === 0){
34        return;
35      }
36      console.log(this.nuevo);
37      this.personajes.push(this.nuevo);
38      this.nuevo = {
39        nombre: '',
40        poder: 0
41      }
42    }
43
44    }
45
```

También agregamos esto en la lista de personajes dentro de main-page.component.html

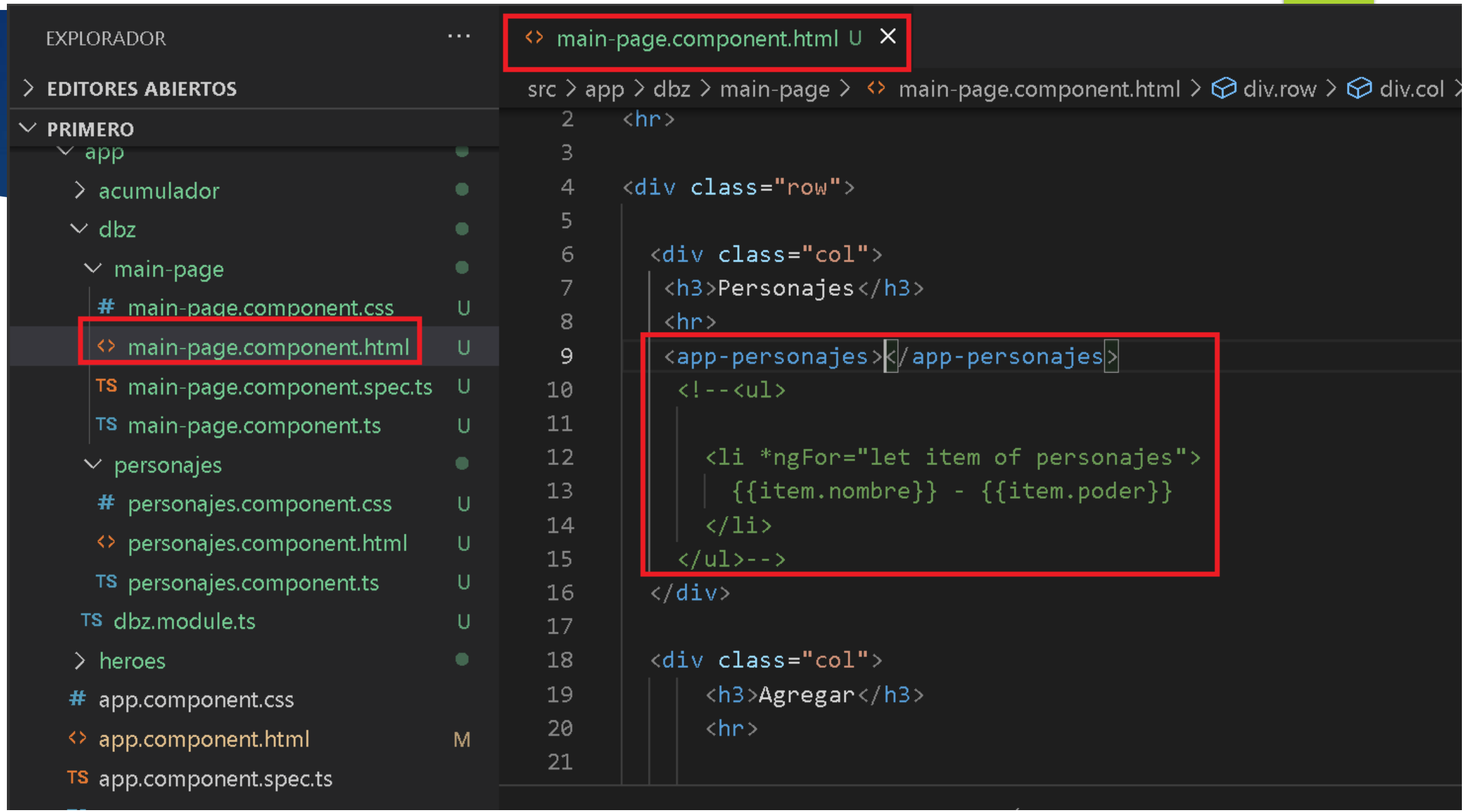
```
<ul>

  <li *ngFor="let item of personajes">
    {{item.nombre}} - {{item.poder}}
  </li>
</ul>
```

Probamos si podemos agregar personajes



Componentes Hijos

```
PS C:\Users\dell\Desktop\pruebas\primero> ng g c dbz/personajes --skipTests
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/dbz/personajes/personajes.component.html (25 bytes)
CREATE src/app/dbz/personajes/personajes.component.ts (291 bytes)
CREATE src/app/dbz/personajes/personajes.component.css (0 bytes)
UPDATE src/app/dbz/dbz.module.ts (484 bytes)
PS C:\Users\dell\Desktop\pruebas\primero>
```


EXPLORADOR
EDITORES ABIERTOS
PRIMERO
app
acumulador
dbz
main-page
main-page.component.css U
main-page.component.html U
main-page.component.spec.ts U
main-page.component.ts U
personajes
personajes.component.css U
personajes.component.html U
personajes.component.ts U
dbz.module.ts U
heroes
app.component.css
app.component.html M
app.component.spec.ts
main-page.component.html U
src > app > dbz > main-page > main-page.component.html > div.row > div.col
2 <hr>
3
4
5
6
7 <h3>Personajes</h3>
8 <hr>
9 <app-personajes></app-personajes>
10 <!--
11
12 <li *ngFor="let item of personajes">
13 {{item.nombre}} - {{item.poder}}
14
15 -->
16
17
18
19 <h3>Agregar</h3>
20 <hr>
21


localhost:4200
Dragon Ball Z
Personajes
personajes works!
Agregar
Nombre
0
Agregar

Realizamos el intercambio

main-page.component.html U

src > app > dbz > main-page > main-page.component.html > div.row

```
2   <hr>
3
4   <div class="row">
5
6     <div class="col">
7      <h3>Personajes</h3>
8      <hr>
9      <app-personajes></app-personajes>
10      <!--<ul>
11
12        <li *ngFor="let item of personajes">
13          {{item.nombre}} - {{item.poder}}
14        </li>
15      </ul>-->
16    </div>
17
18    <div class="col">
19        <h3>Agregar</h3>
20        <hr>
21
```

personajes.component.html U

src > app > dbz > personajes > personajes.component.html >

Go to component

```
1   <p>personajes works!</p>
2
```

main-page.component.html U

app > dbz > main-page > main-page.component.html > div.row > div.col

```
  <hr>

  <div class="row">

    <div class="col">
      <h3>Personajes</h3>
      <hr>
      <app-personajes></app-personajes>

    </div>

    <div class="col">
        <h3>Agregar</h3>
        <hr>

        <form (ngSubmit)="agregar()">

          <input
            type="text"
            placeholder="Nombre"
```

personajes.component.html 1, U

src > app > dbz > personajes > personajes.component.html > ul

Go to component

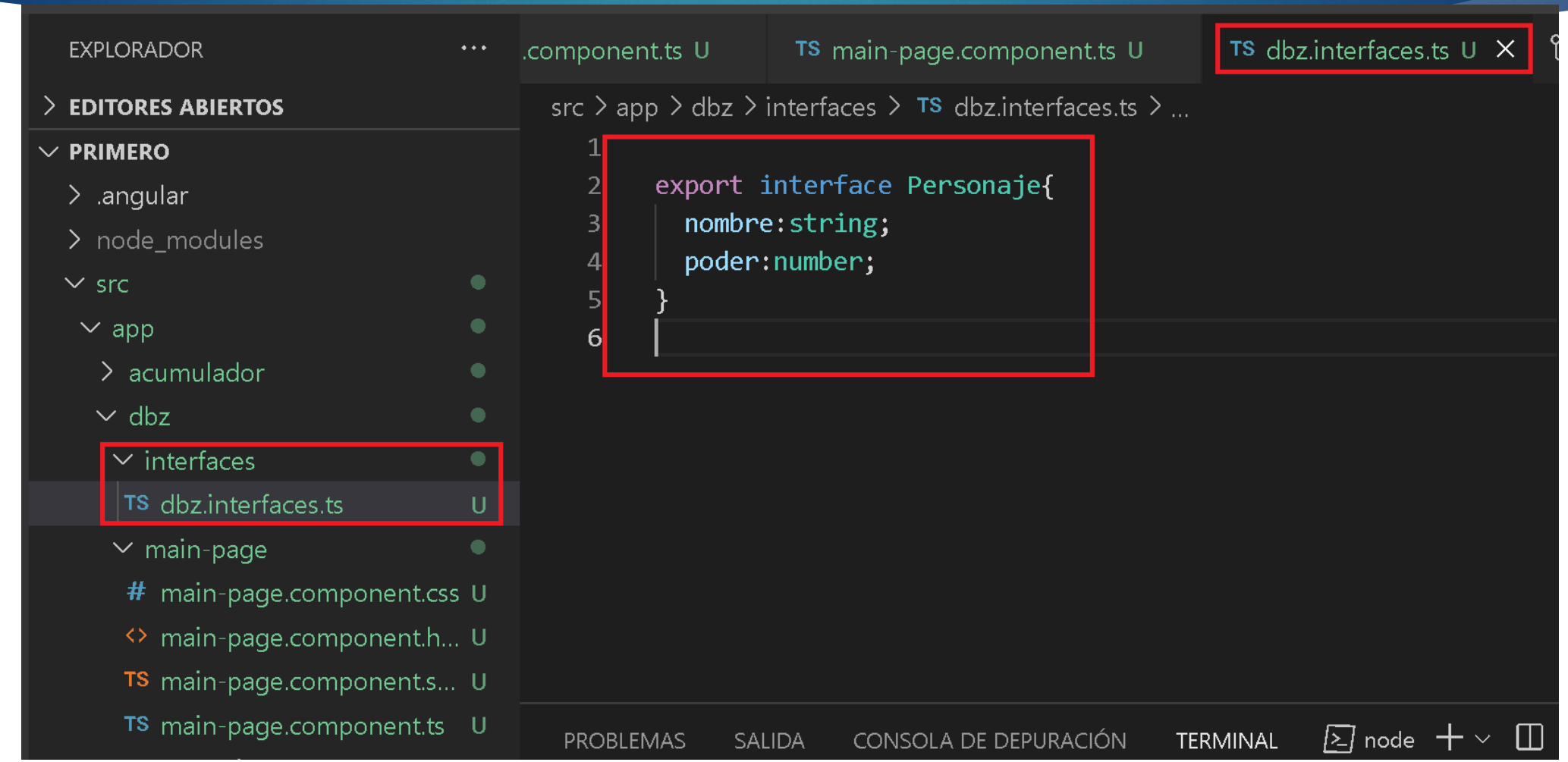
```
<ul>

  <li *ngFor="let item of personajes">
    {{item.nombre}} - {{item.poder}}
  </li>
</ul>

```

TS personajes.component.ts U | main-page.component.html U | TS main-page.component.ts U

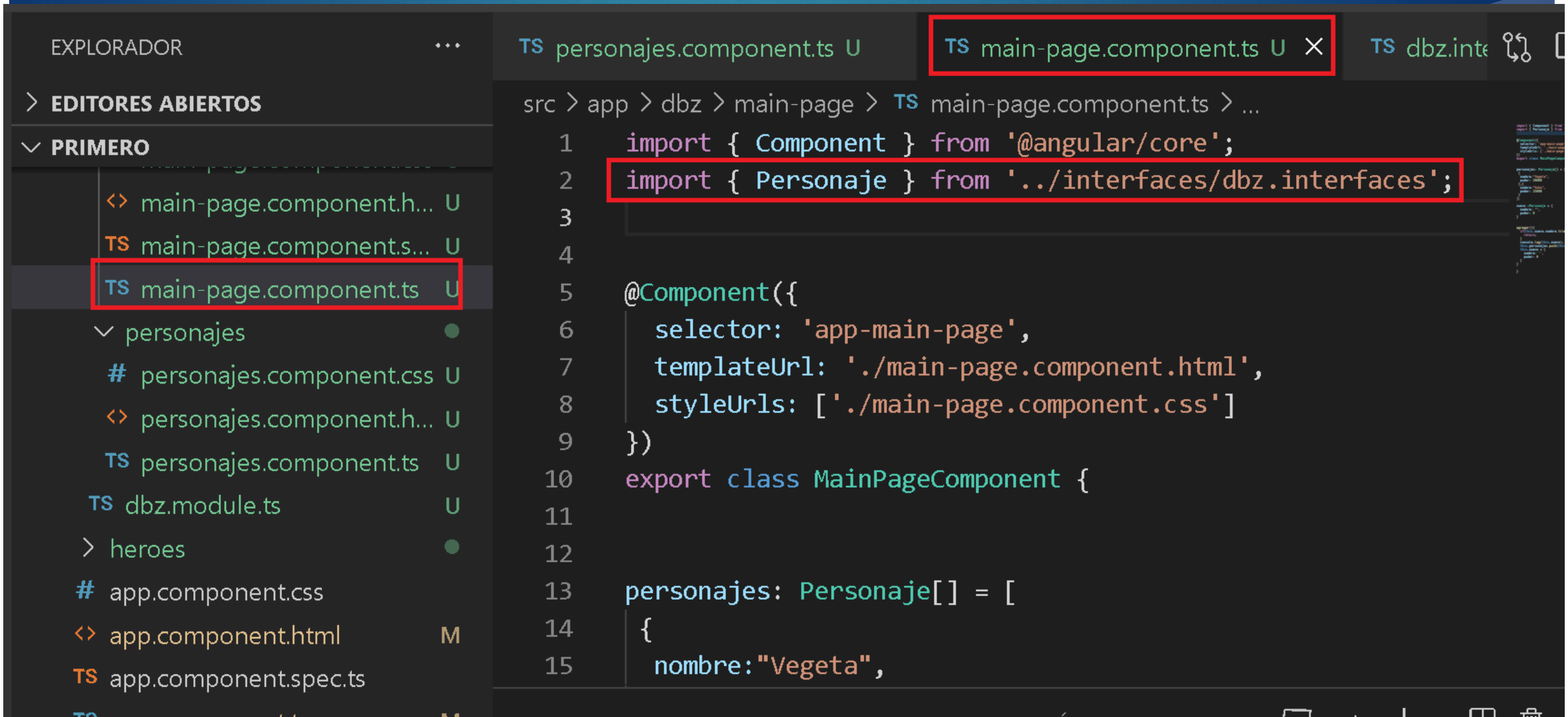
src > app > dbz > main-page > main-page.component.html > div.row

```
4    <div class="row">
5
6      <div class="col">
7       <h3>Personajes</h3>
8       <hr>
9       <app-personajes [personajes]="personajes"></app-personajes>
10
11     </div>
12
13     <div class="col">
14         <h3>Agregar</h3>
15         <hr>
16
17         <form (ngSubmit)="agregar()">
18
```

Cambiamos de lugar nuestra interface



Con este cambio importamos la interfaz en main-page.component.ts



Con @input hacemos referencia a un valor del componente padre

TS personajes.component.ts U × <> main-page.component.html U TS main-page

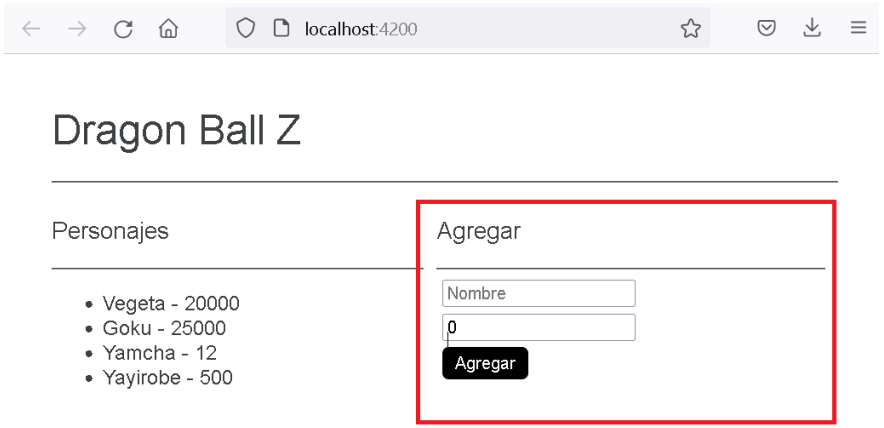
src > app > dbz > personajes > TS personajes.component.ts > ...

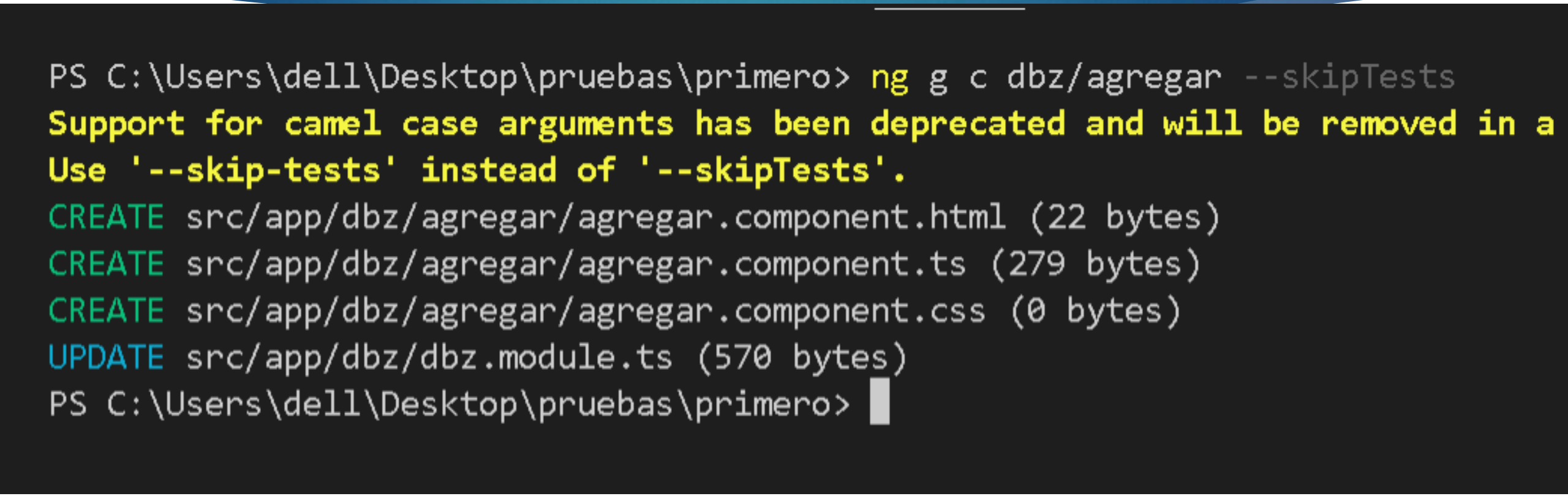
```ts
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-personajes',
  templateUrl: './personajes.component.html',
  styleUrls: ['./personajes.component.css']
})
export class PersonajesComponent {

  @Input() personajes:any[] = [];

}

```

Con esto probamos que funciona todo como debería teniendo diferentes componentes para ello

Ejercicio

- Pasar el formulario a un componente hijo llamado agregar siguiendo los pasos anteriores y probar que todo funcione




PS C:\Users\dell\Desktop\pruebas\primero> ng g c dbz/agregar --skipTests
Support for camel case arguments has been deprecated and will be removed in a
Use '--skip-tests' instead of '--skipTests'.
CREATE src/app/dbz/agregar/agregar.component.html (22 bytes)
CREATE src/app/dbz/agregar/agregar.component.ts (279 bytes)
CREATE src/app/dbz/agregar/agregar.component.css (0 bytes)
UPDATE src/app/dbz/dbz.module.ts (570 bytes)
PS C:\Users\dell\Desktop\pruebas\primero>

main-page.component.html U

src > app > dbz > main-page > main-page.component.html > div.row > div.col > app-ag

Go to component

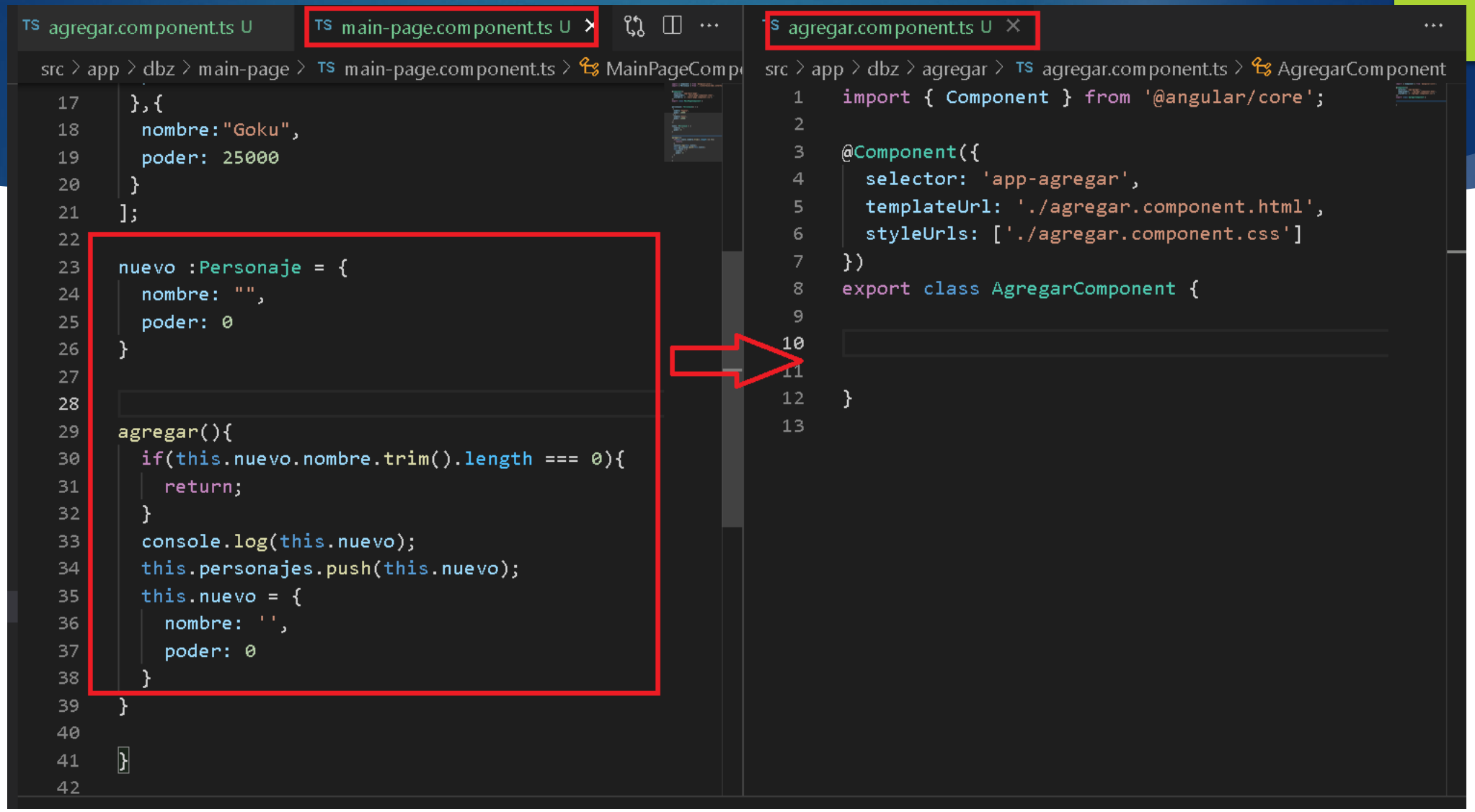
```
1   <h1>Dragon Ball Z</h1>
2   <hr>
3
4   <div class="row">
5
6     <div class="col">
7      <h3>Personajes</h3>
8      <hr>
9      <app-personajes [personajes]="personajes"></app-personajes
10
11    </div>
12
13    <div class="col">
14        <app-agregar></app-agregar>
15    </div>
16  </div>
17
```

agregar.component.html 5, U

src > app > dbz > agregar > agregar.component.html > form

Go to component

```
1   <h3>Agregar</h3>
2   <hr>
3
4   <form (ngSubmit)="agregar()">
5
6     <input
7       type="text"
8       placeholder="Nombre"
9       name="nombre"
10      [(ngModel)] = "nuevo.nombre"
11      >
12      <input
13      type="text"
14      placeholder="Poder"
15      name="poder"
16      [(ngModel)] = "nuevo.poder"
17      >
18
19    <button type="submit">Agregar</button>
20  </form>
21
```


agregar.component.ts U
main-page.component.ts U
src > app > dbz > main-page > main-page.component.ts > MainPageComp
},{
nombre:"Goku",
poder: 25000
}
];
nuevo :Personaje = {
nombre: "",
poder: 0
}
agregar(){
if(this.nuevo.nombre.trim().length === 0){
return;
}
console.log(this.nuevo);
this.personajes.push(this.nuevo);
this.nuevo = {
nombre: '',
poder: 0
}
}
}
agregar.component.ts U
src > app > dbz > agregar > agregar.component.ts > AgregarComponent
import { Component } from '@angular/core';
@Component({
selector: 'app-agregar',
templateUrl: './agregar.component.html',
styleUrls: ['./agregar.component.css']
})
export class AgregarComponent {
}

agregar.component.ts U

src > app > dbz > agregar > agregar.component.ts > AgregarComponent > personajes

```ts
import { Component, Input } from '@angular/core';
import { Personaje } from '../interfaces/dbz.interfaces';

@Component({
  selector: 'app-agregar',
  templateUrl: './agregar.component.html',
  styleUrls: ['./agregar.component.css']
})
export class AgregarComponent {

  @Input() personajes:Personaje[] = [];

  @Input() nuevo :Personaje = {
    nombre: "",
    poder: 0
  }


  agregar(){
    if(this.nuevo.nombre.trim().length === 0){
      return;
    }
    console.log(this.nuevo);
    this.personajes.push(this.nuevo);
    this.nuevo = {
      nombre: '',
      poder: 0
    }
  }


}
```

agregar.component.ts U

main-page.component.html U

app > dbz > main-page > main-page.component.html > div.row > div.col

Go to component

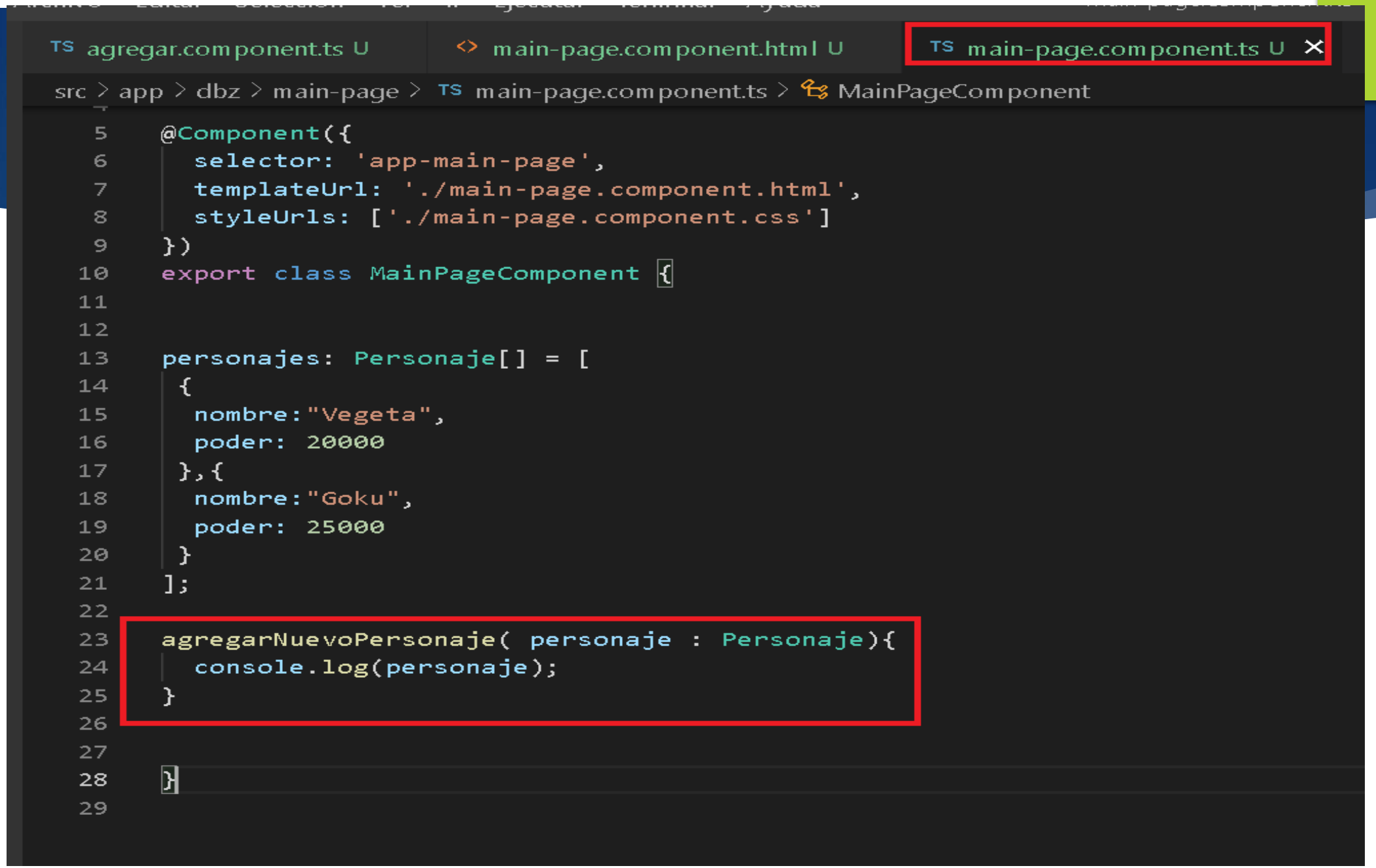
```
1   <h1>Dragon Ball Z</h1>
2   <hr>
3
4   <div class="row">
5
6     <div class="col">
7       <h3>Personajes</h3>
8       <hr>
9       <app-personajes [personajes]="personajes"></app-personajes>
10
11    </div>
12
13    <div class="col">
14      <app-agregar [personajes]="personajes"></app-agregar>
15    </div>
16  </div>
17
```

@Output() nos sirve para emitir eventos desde el componente hijo al componente padre

agregar.component.ts U | main-page.component.html U | main-page.component.ts U

src > app > dbz > agregar > agregar.component.ts > AgregarComponent > personajes

```ts
import { Component, Input, Output, EventEmitter } from '@angular/core';
import { Personaje } from '../interfaces/dbz.interfaces';

@Component({
  selector: 'app-agregar',
  templateUrl: './agregar.component.html',
  styleUrls: ['./agregar.component.css']
})
export class AgregarComponent {

  @Input() personajes:Personaje[] = [];

  @Input() nuevo :Personaje = {
    nombre: "",
    poder: 0
  }

  @Output() onNuevoPersonaje: EventEmitter<Personaje> = new EventEmitter();

  agregar(){
    if(this.nuevo.nombre.trim().length === 0){
      return;
    }
    console.log(this.nuevo);
   this.personajes.push(this.nuevo);
   this.onNuevoPersonaje.emit(this.nuevo);

    this.nuevo = {
      nombre: '',
      poder: 0
```


agregar.component.ts U
main-page.component.html U
main-page.component.ts U
src > app > dbz > main-page > main-page.component.ts > MainPageComponent
@Component({
selector: 'app-main-page',
templateUrl: './main-page.component.html',
styleUrls: ['./main-page.component.css']
})
export class MainPageComponent {
personajes: Personaje[] = [
{
nombre:"Vegeta",
poder: 20000
},{
nombre:"Goku",
poder: 25000
}
];
agregarNuevoPersonaje(personaje : Personaje){
console.log(personaje);
}
}

TS agregar.component.ts U | <> main-page.component.html U | TS main-page.component.ts U

src > app > dbz > main-page > <> main-page.component.html > div.row > div.col > app-agregar

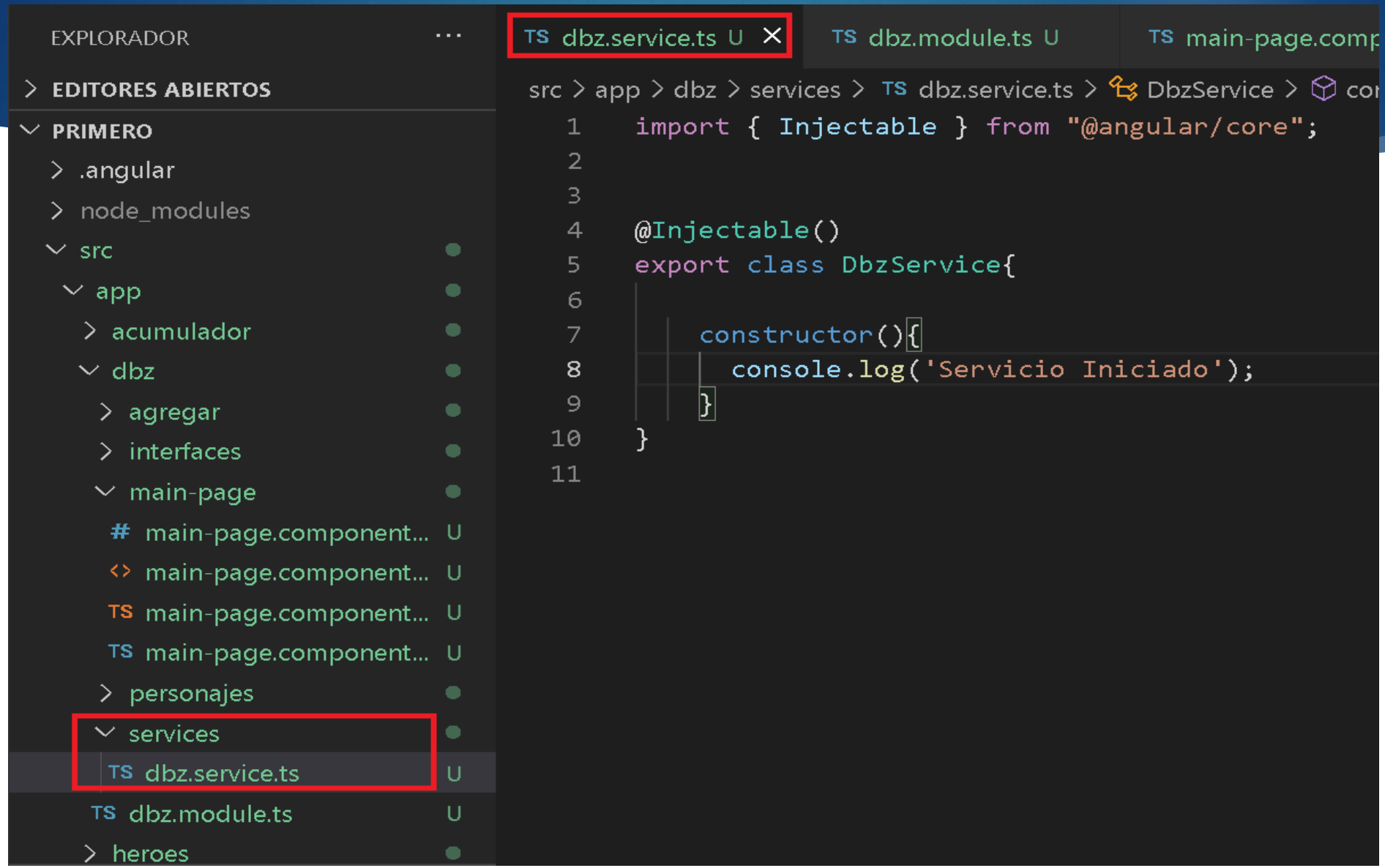
Go to component

```
1   <h1>Dragon Ball Z</h1>
2   <hr>
3
4   <div class="row">
5
6     <div class="col">
7       <h3>Personajes</h3>
8       <hr>
9       <app-personajes [personajes]="personajes"></app-personajes>
10
11    </div>
12
13    <div class="col">
14
15      <app-agregar
16        [personajes]="personajes"
17        (onNuevoPersonaje) = "agregarNuevoPersonaje( $event )">
18      </app-agregar>
19
20    </div>
21  </div>
22
```

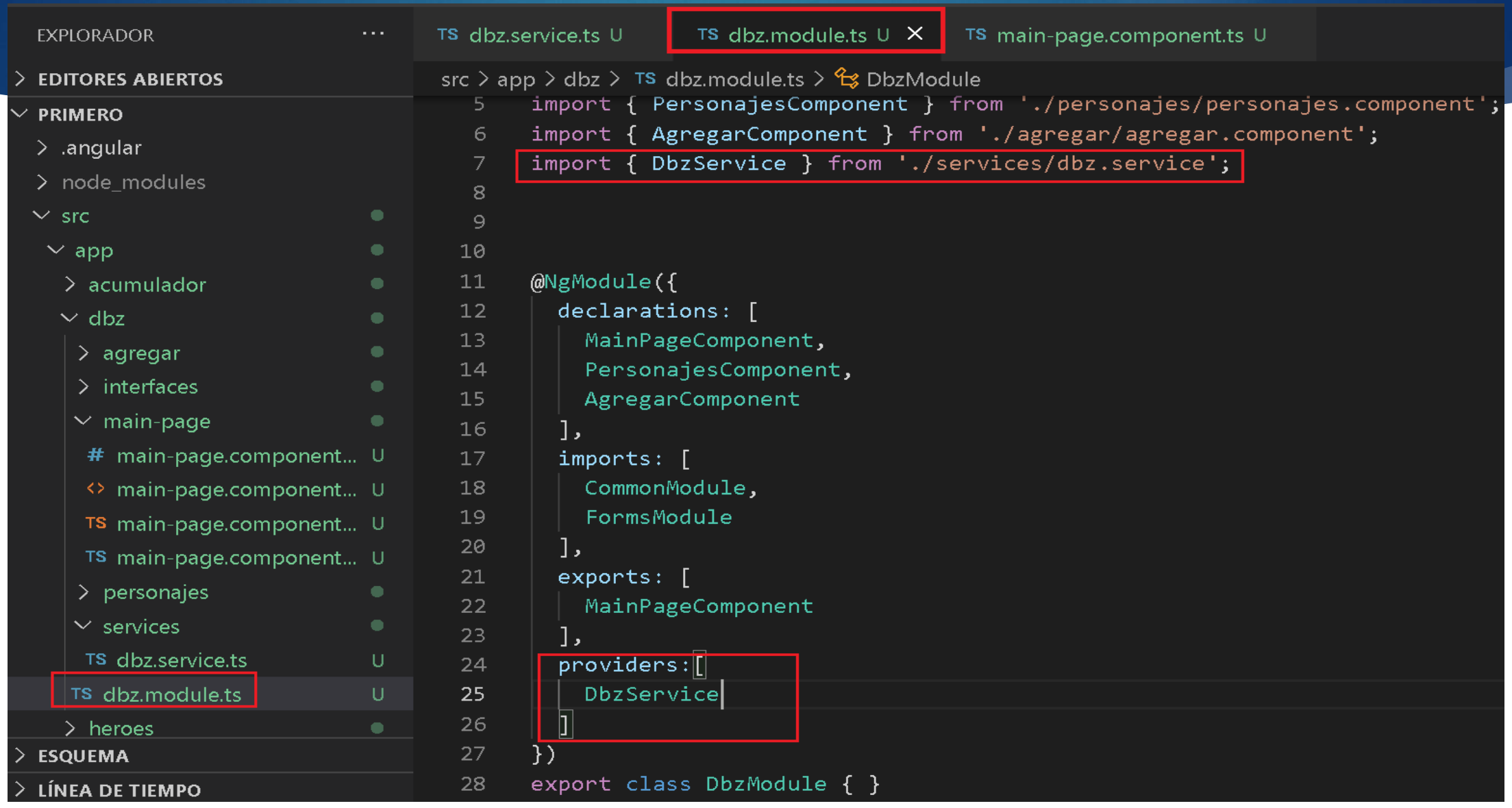
Servicios

- Los servicios en angular son manejados de manera muy eficiente, es muy parecida a la manera de trabajar con un patrón de diseño singleton. Es como tener una clase que este de manera global en nuestra aplicación.

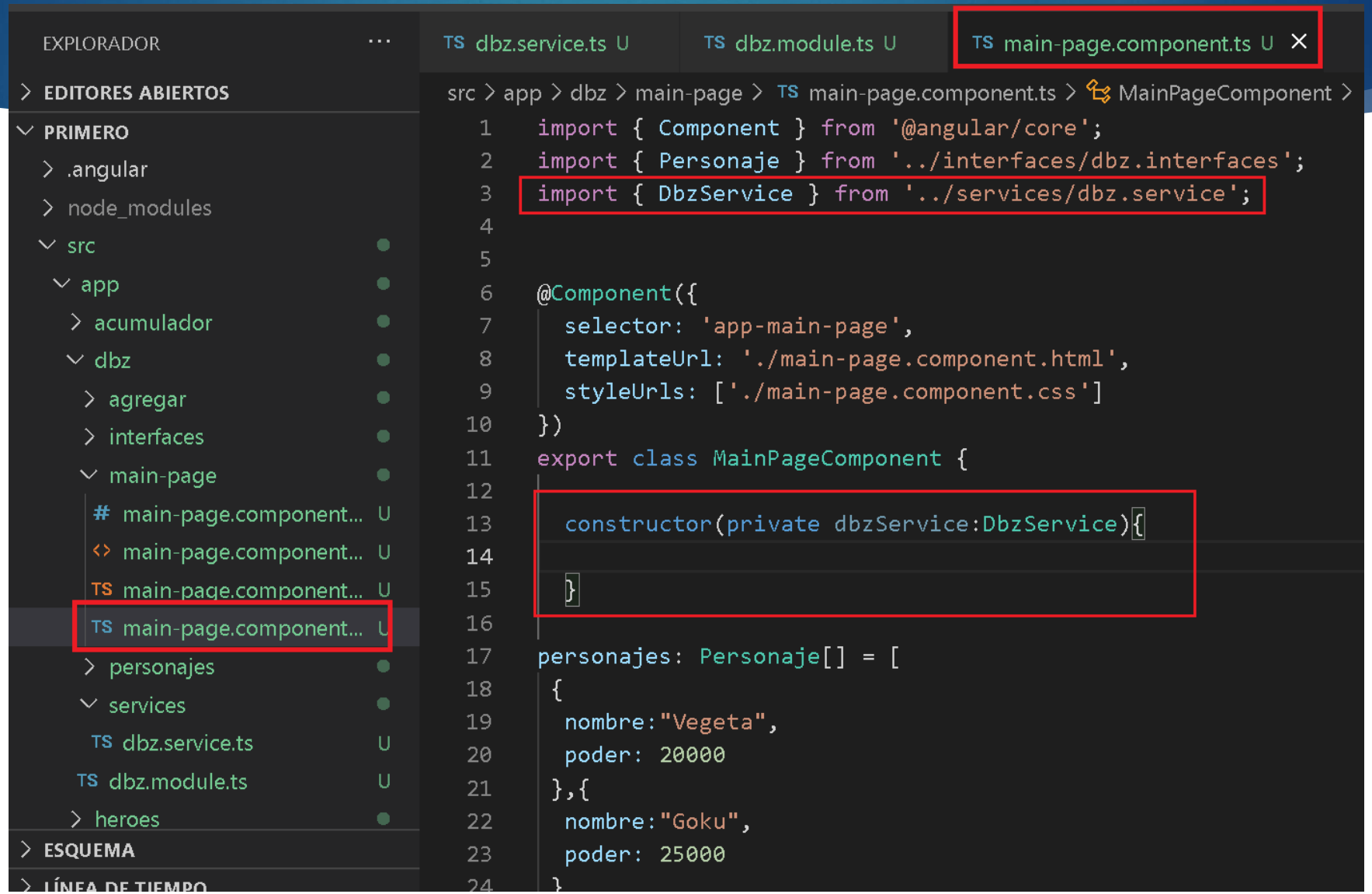
Creamos un directorio y un archivo para nuestro servicio

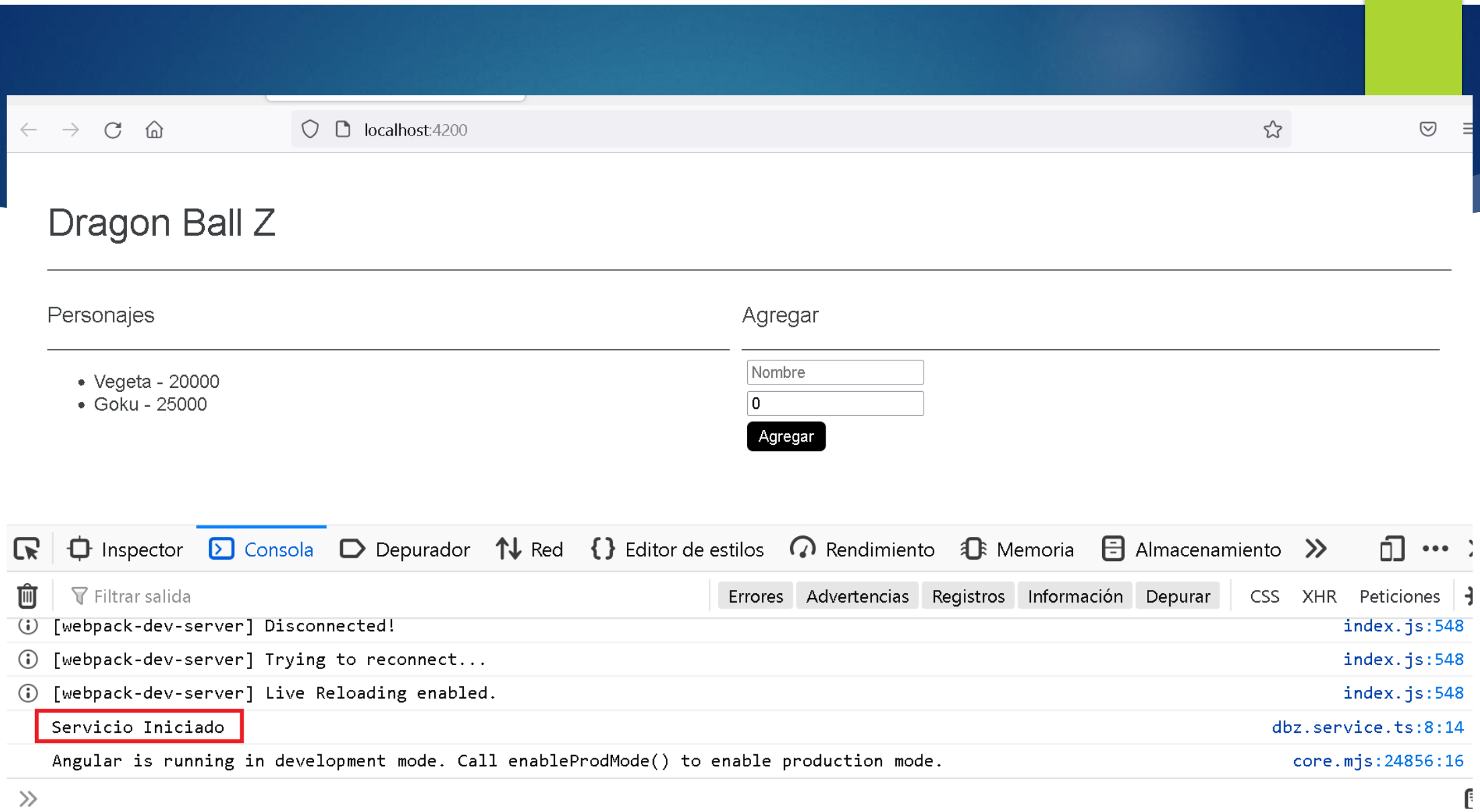


Se añade el servicio al modulo

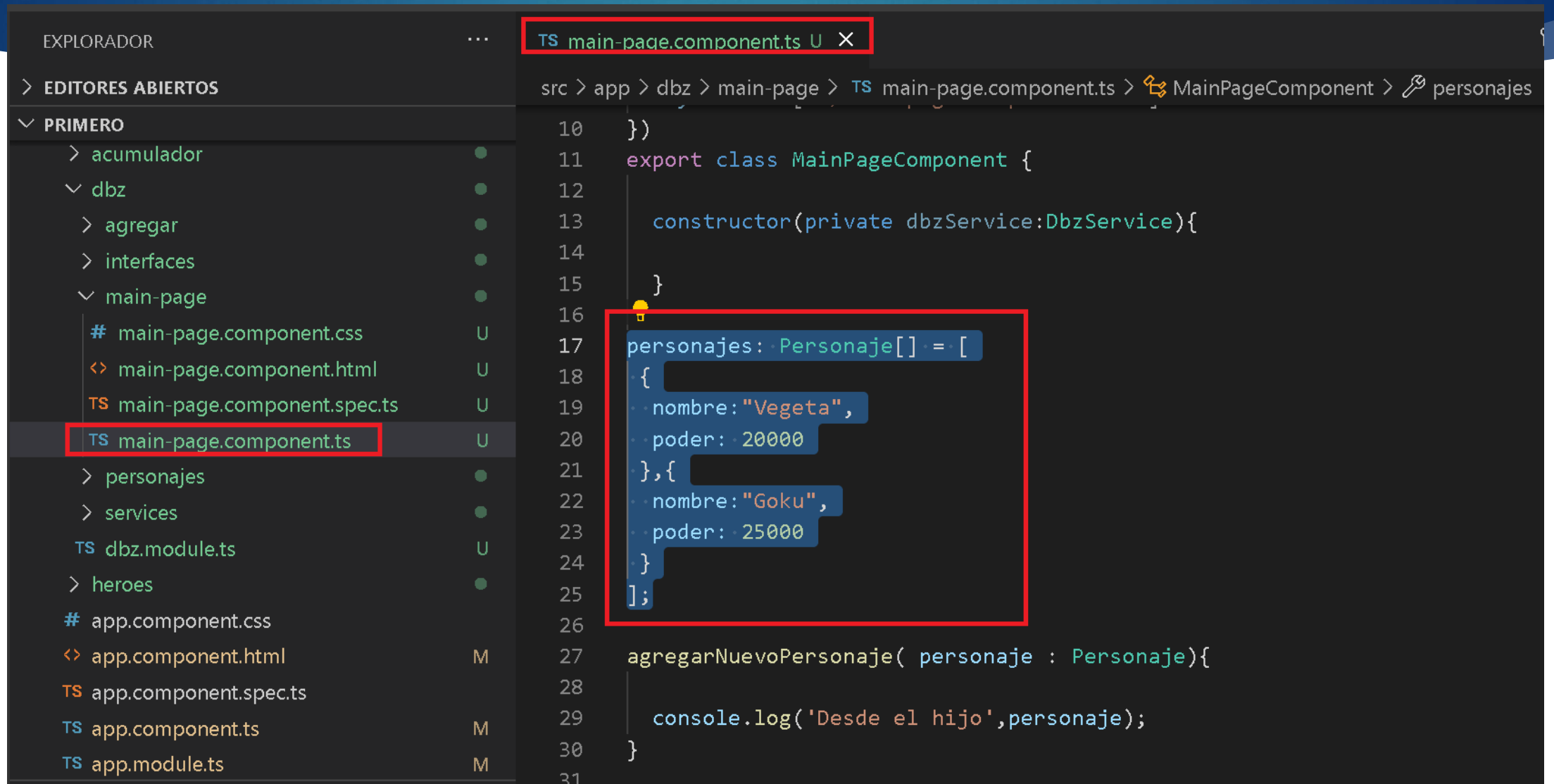


El servicio es ejecutado cuando se es solicitado o creado una instancia, por medio de inyección de dependencia

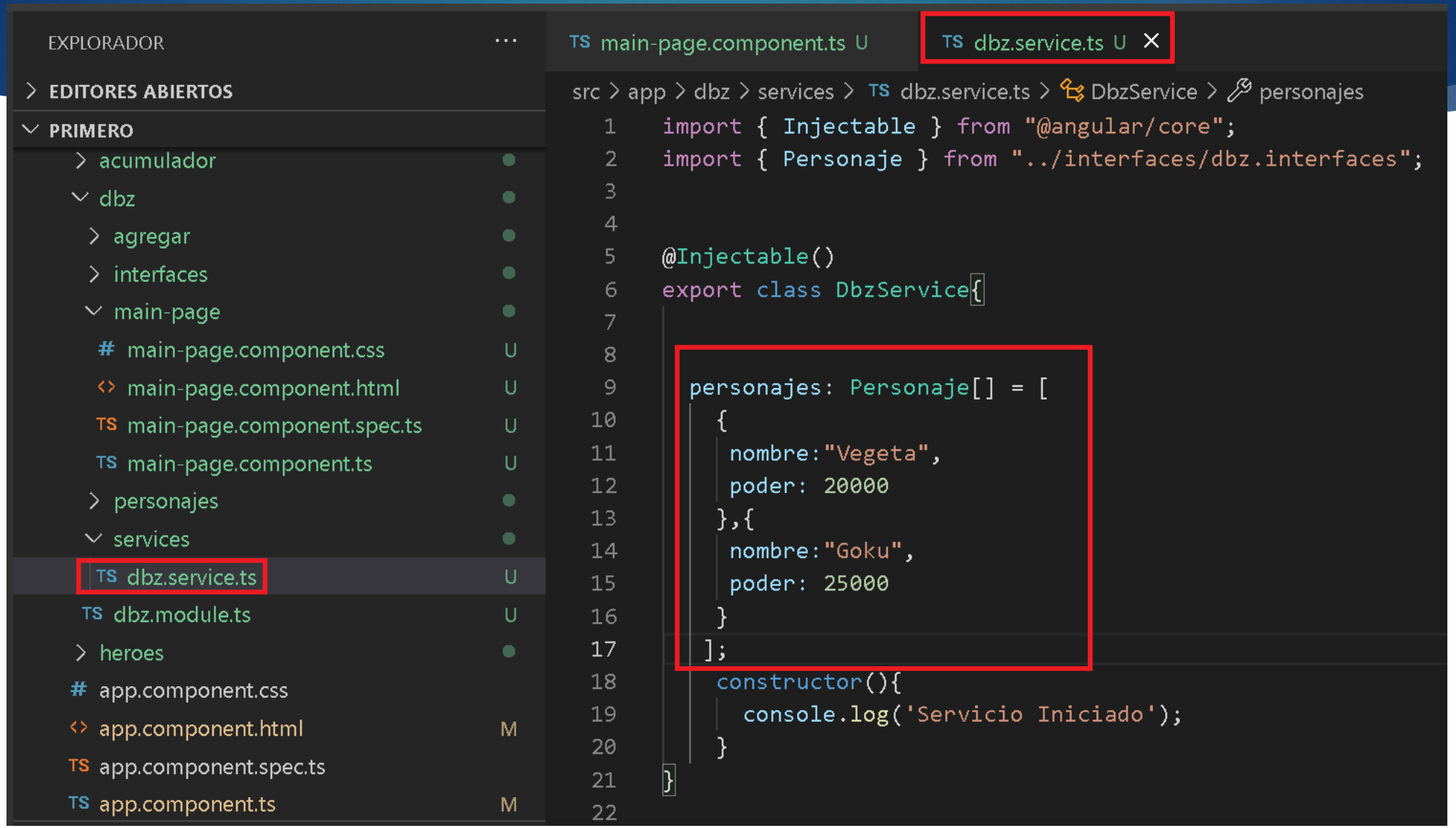



localhost:4200
Dragon Ball Z
Personajes
Vegeta - 20000
Goku - 25000
Agregar
Nombre
0
Agregar
Inspector
Consola
Depurador
Red
Editor de estilos
Rendimiento
Memoria
Almacenamiento
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
[webpack-dev-server] Disconnected!
index.js:548
[webpack-dev-server] Trying to reconnect...
index.js:548
[webpack-dev-server] Live Reloading enabled.
index.js:548
Servicio Iniciado
dbz.service.ts:8:14
Angular is running in development mode. Call enableProdMode() to enable production mode.
core.mjs:24856:16

Centralizamos los datos en los servicios, pasamos nuestro array de personajes a nuestro servicio

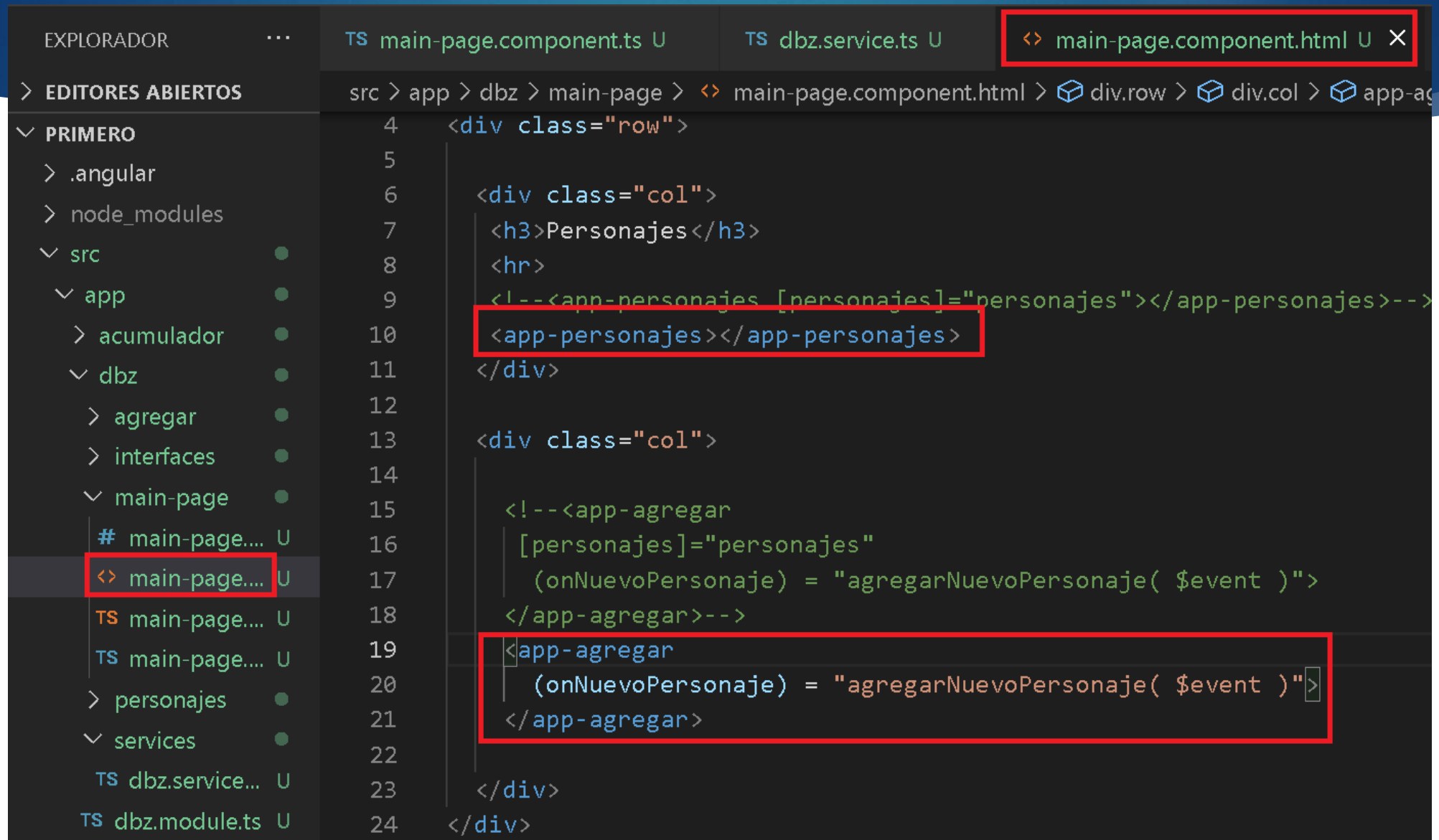


Lo pasamos al servicio creado

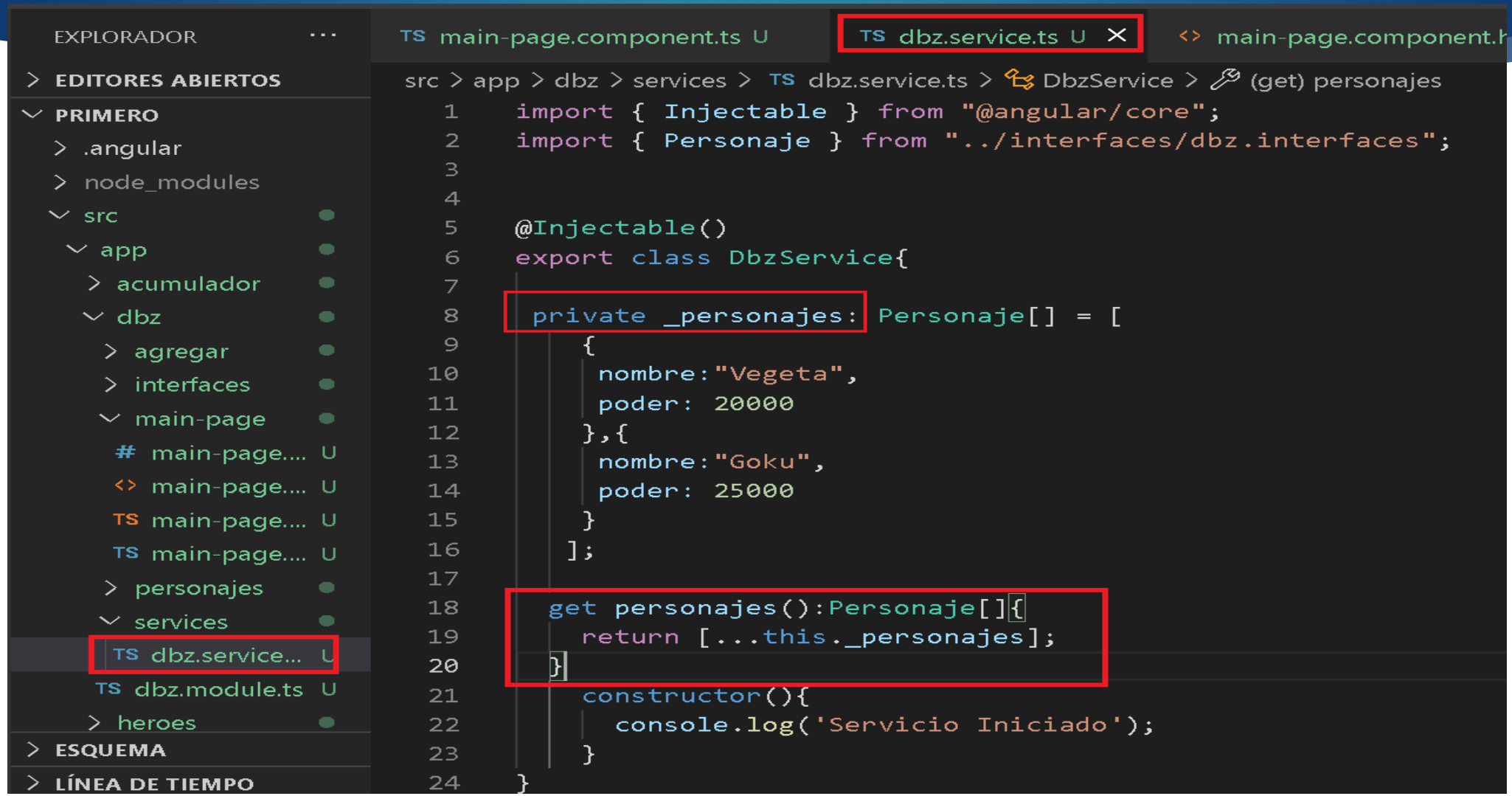


Tendremos estos errores en main-page.component.html quitamos o comentamos el código que ya no utilizaremos

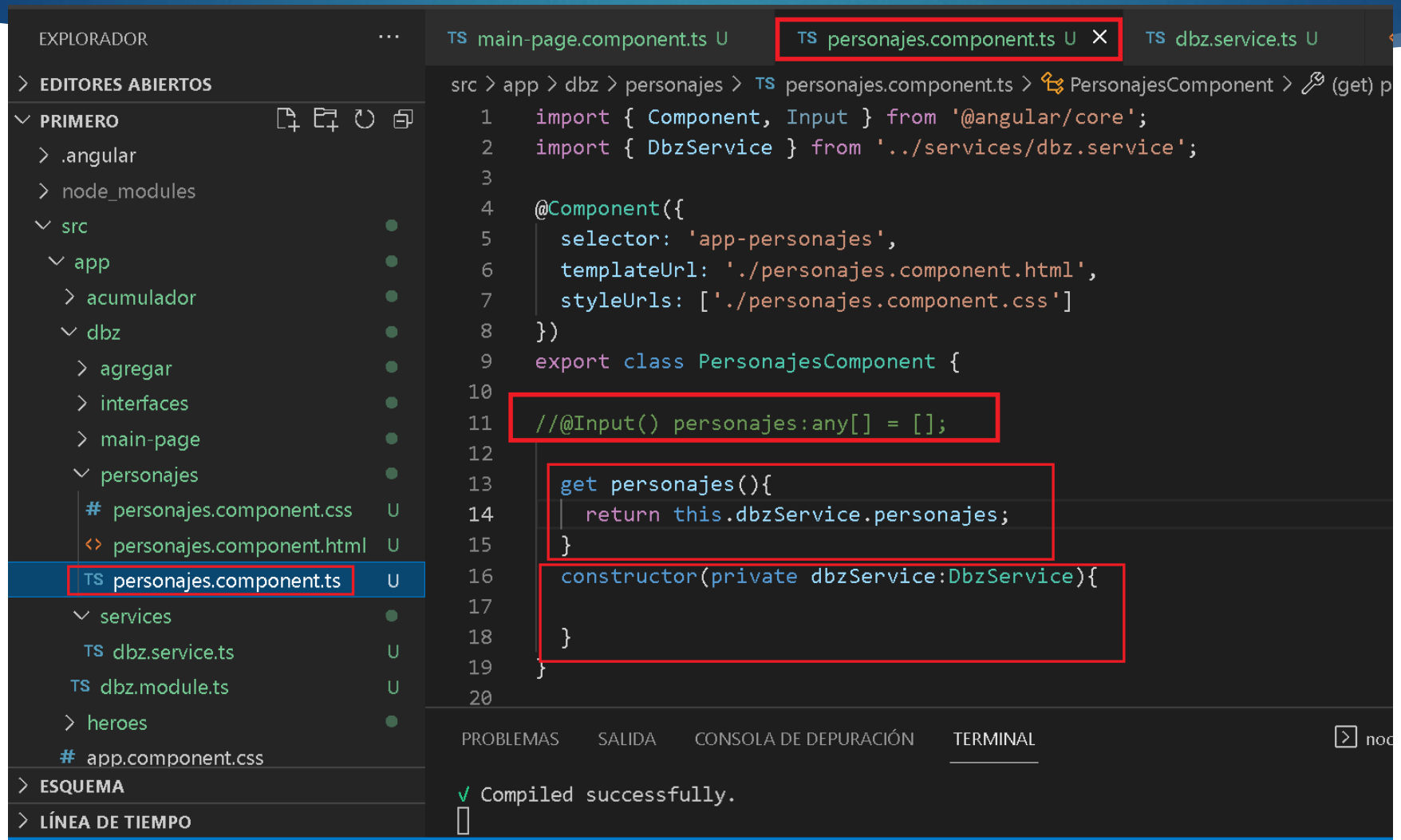
Quedaría así nuestro main-page.component.html



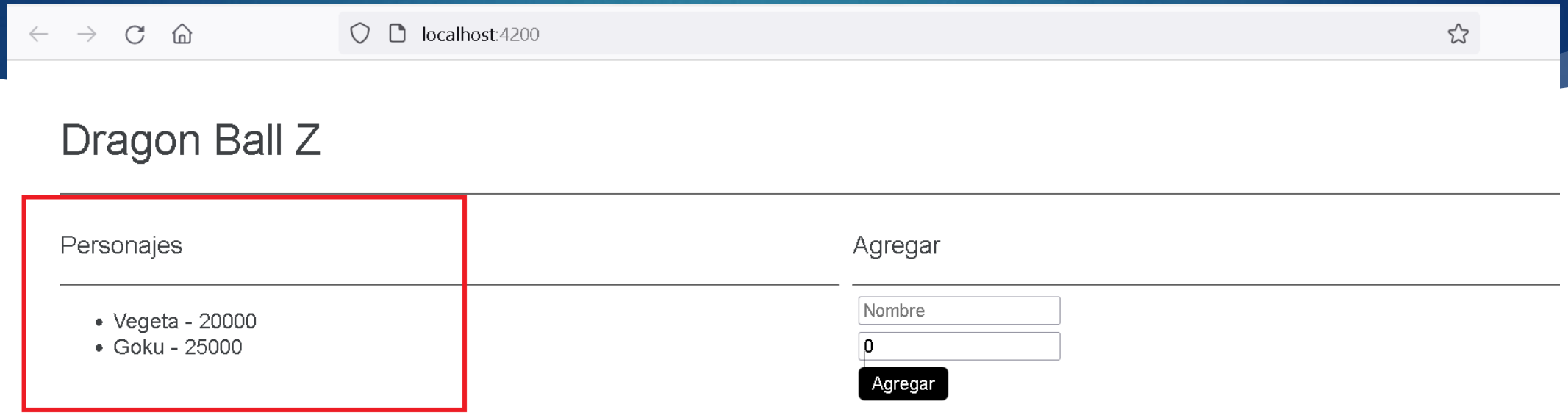
Modificamos el acceso a nuestros datos personajes a privado, también agregamos un get para poder acceder a ellos, los …(tres puntos en el return) significa operador spread este separa cada uno de los elementos independiente que tiene este arreglo y crear uno nuevo, esto lo hacemos por default el get se hace por referencia y con esto evitamos la manipulación



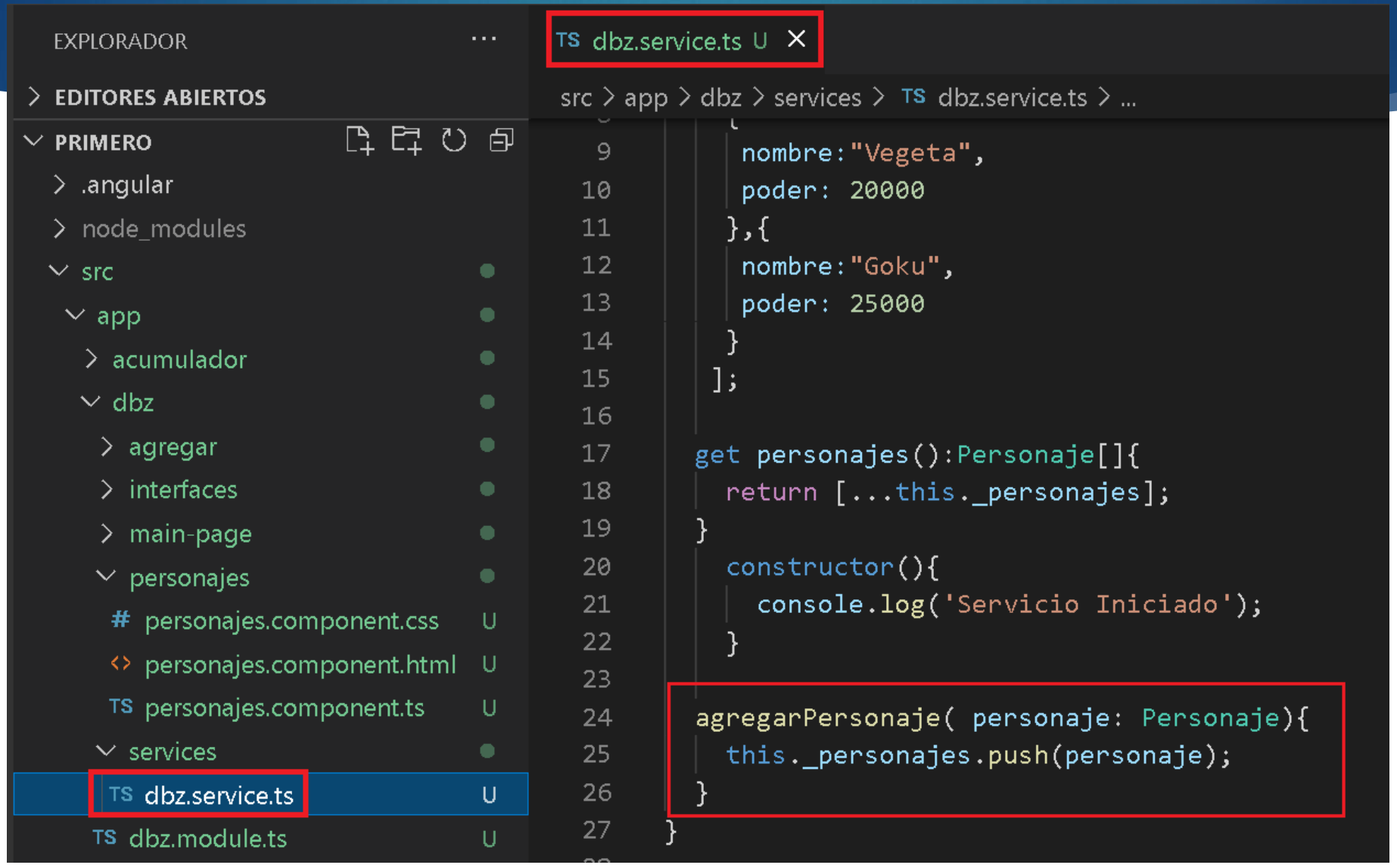
En personajes.component.ts llamamos al servicio y devolvemos un get personajes

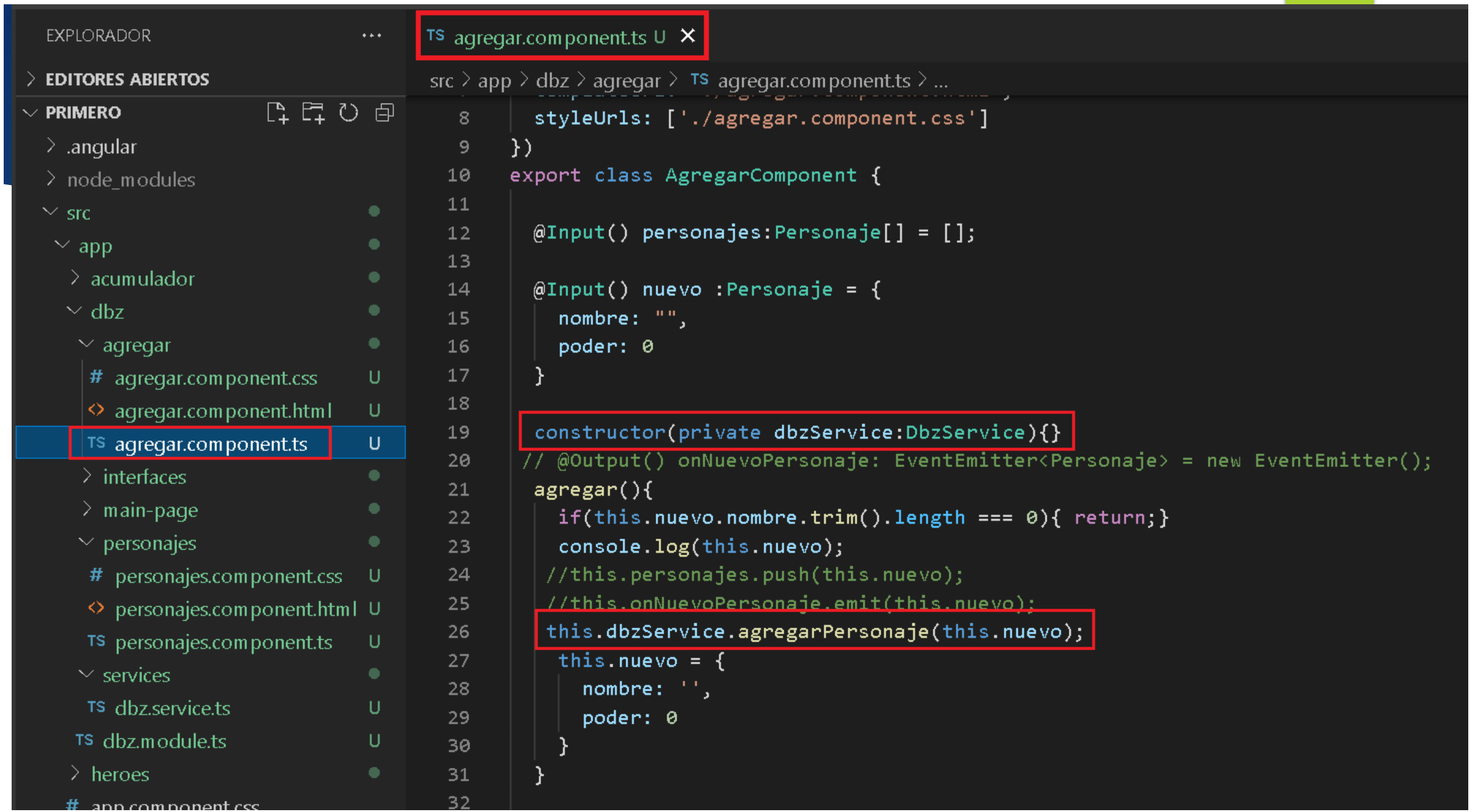


Deberíamos de poder ver el listado desde el servicio

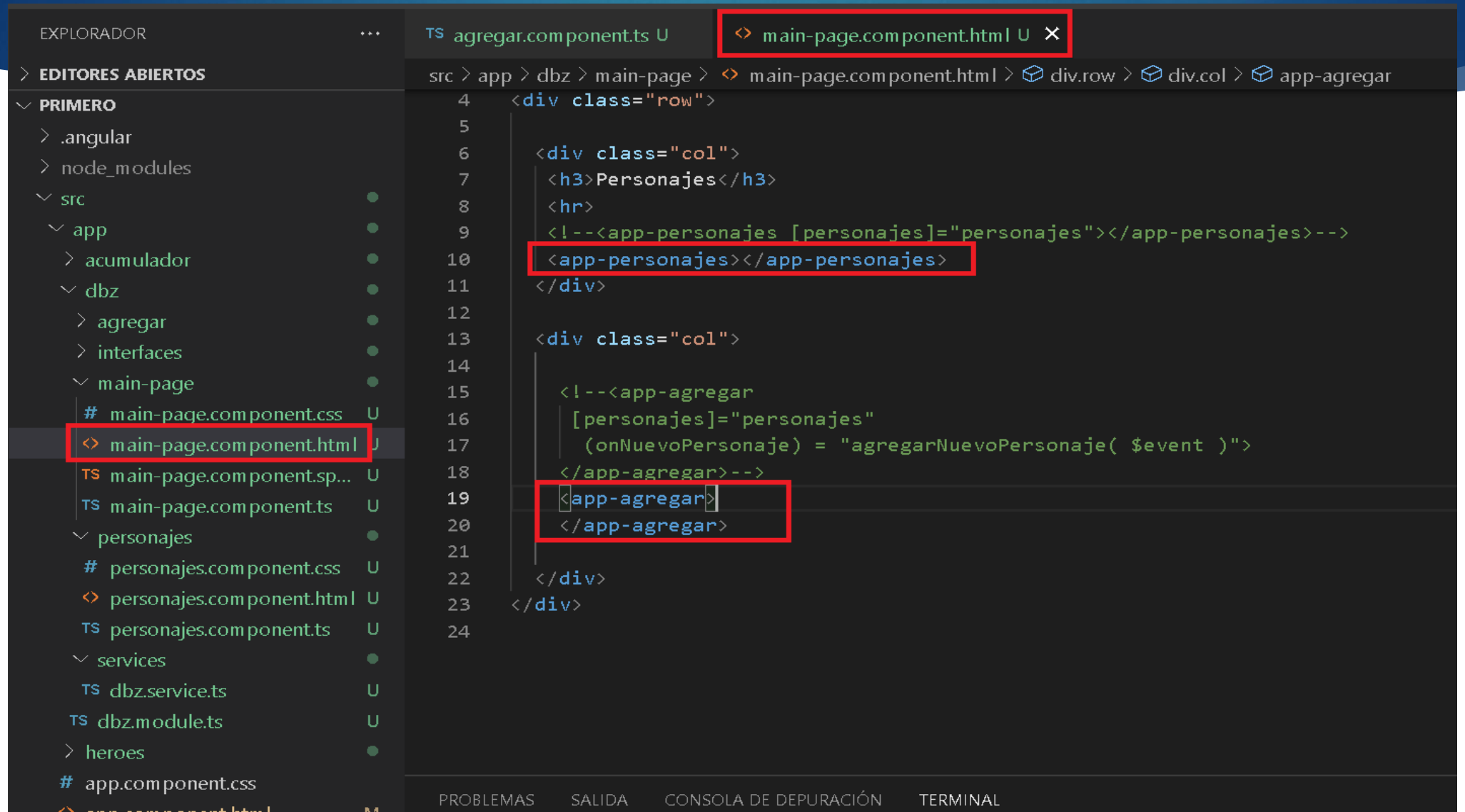


Ahora agregamos al servicio la funcionalidad de crear personaje



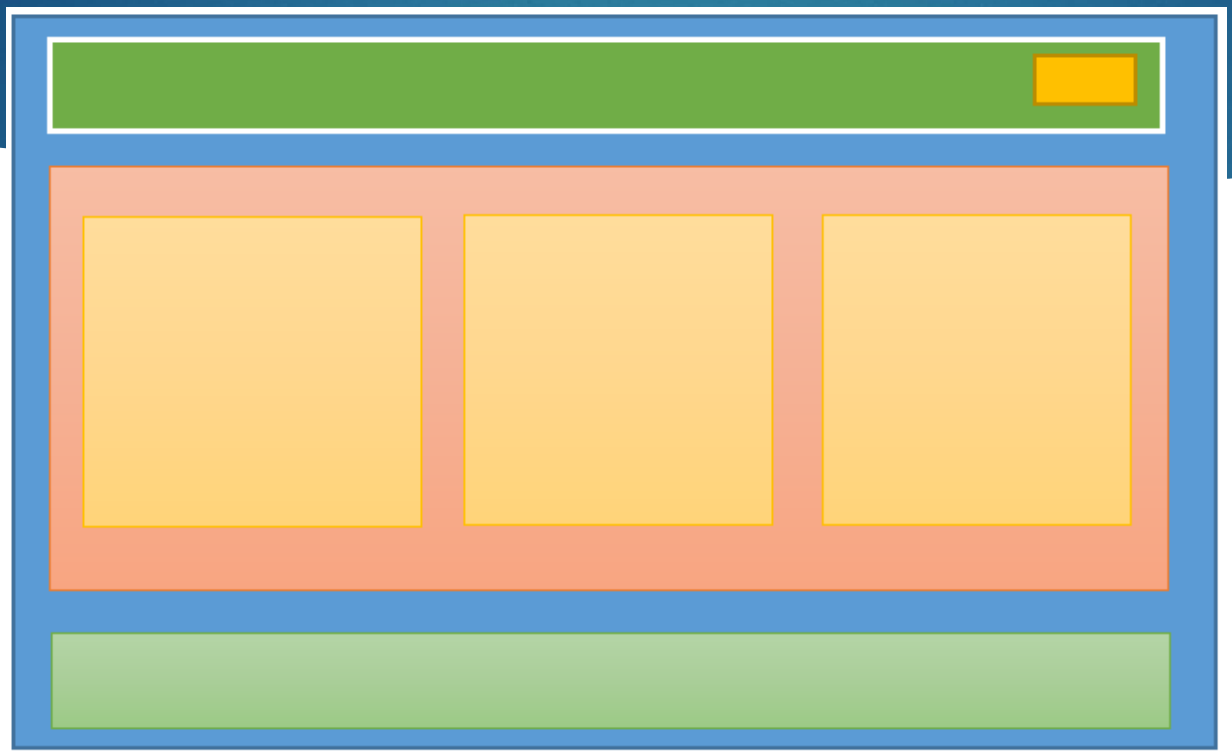

EXPLORADOR
EDITORES ABIERTOS
PRIMERO
.angular
node_modules
src
app
acumulador
dbz
agregar
agregar.component.css U
agregar.component.html U
agregar.component.ts U
interfaces
main-page
personajes
personajes.component.css U
personajes.component.html U
personajes.component.ts U
services
dbz.service.ts U
dbz.module.ts U
heroes
app.component.css
agregar.component.ts U
src > app > dbz > agregar > agregar.component.ts > ...
8 styleUrls: ['./agregar.component.css']
9 })
10 export class AgregarComponent {
11
12 @Input() personajes:Personaje[] = [];
13
14 @Input() nuevo :Personaje = {
15 nombre: "",
16 poder: 0
17 }
18
19 constructor(private dbzService:DbzService){}
20 // @Output() onNuevoPersonaje: EventEmitter<Personaje> = new EventEmitter();
21 agregar(){
22 if(this.nuevo.nombre.trim().length === 0){ return;}
23 console.log(this.nuevo);
24 //this.personajes.push(this.nuevo);
25 //this.onNuevoPersonaje.emit(this.nuevo);
26 this.dbzService.agregarPersonaje(this.nuevo);
27 this.nuevo = {
28 nombre: '',
29 poder: 0
30 }
31 }
32

El main-page.component.html queda así



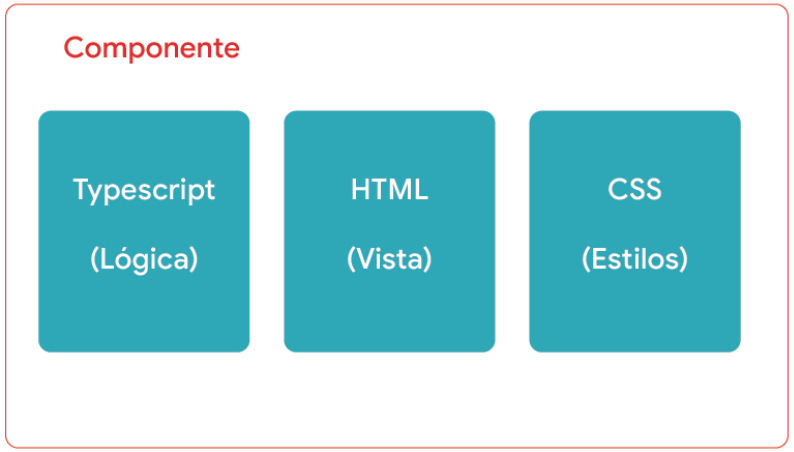
Finalmente debería mostrar y agregar los datos desde servicios

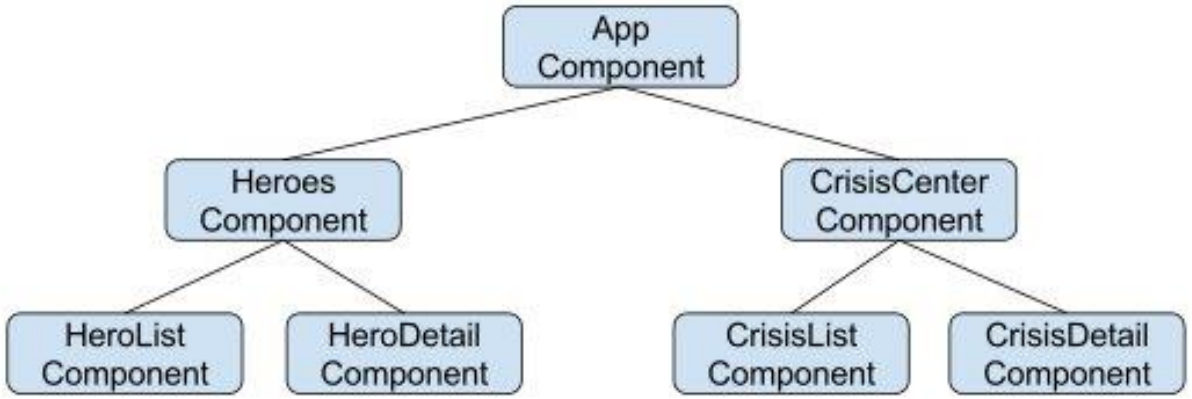
Resumen

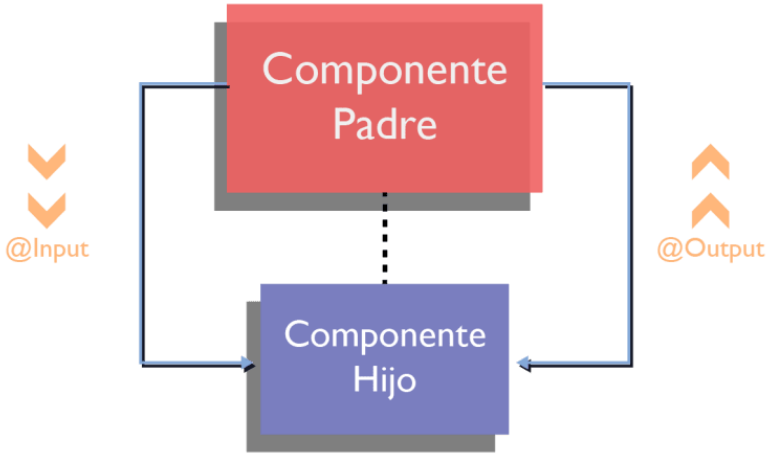


Aplicación = Componente + Componente + Componente

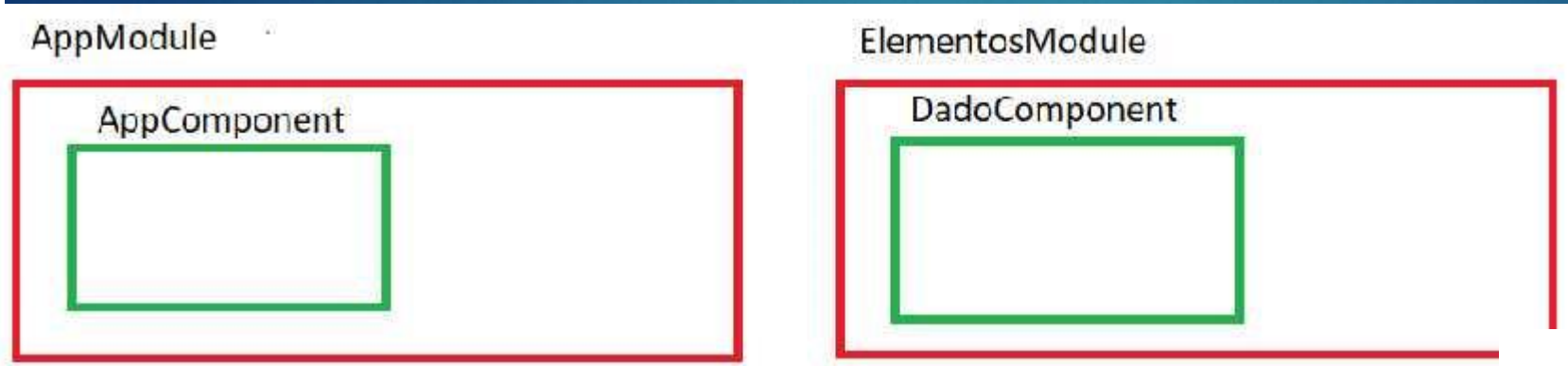
Resumen Componente

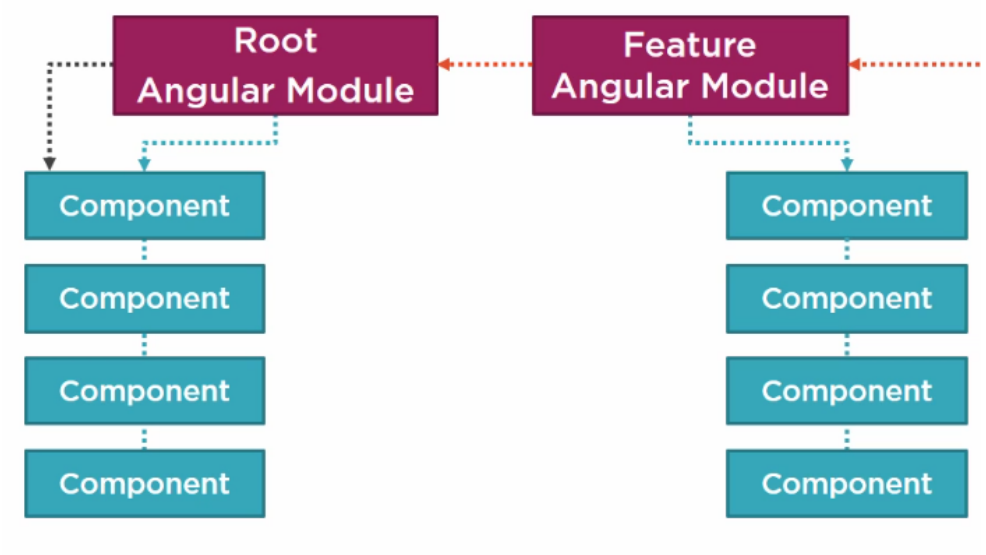


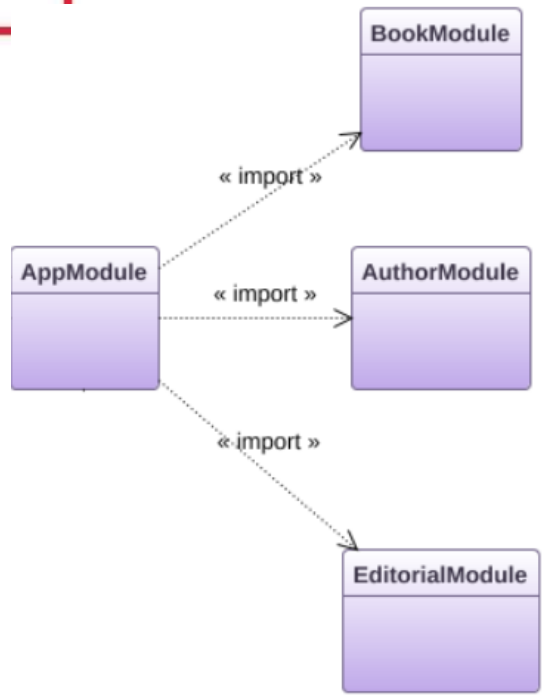




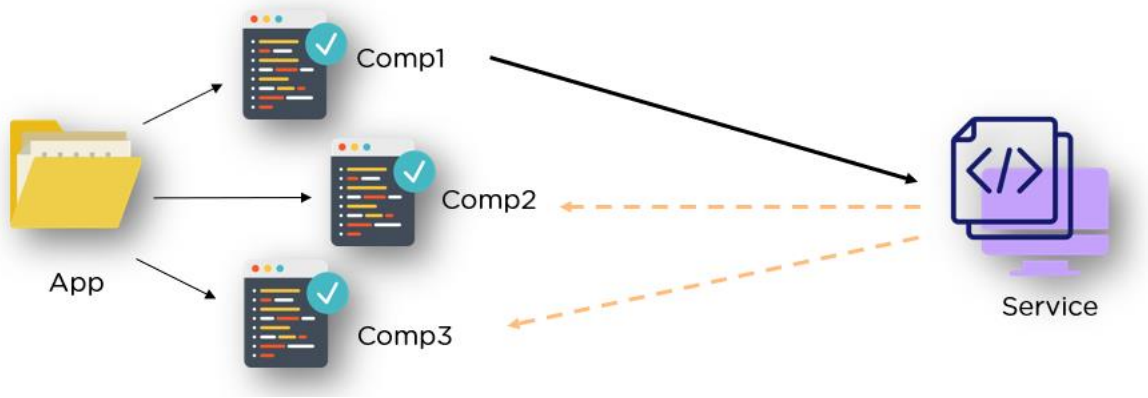
Resumen Modulo

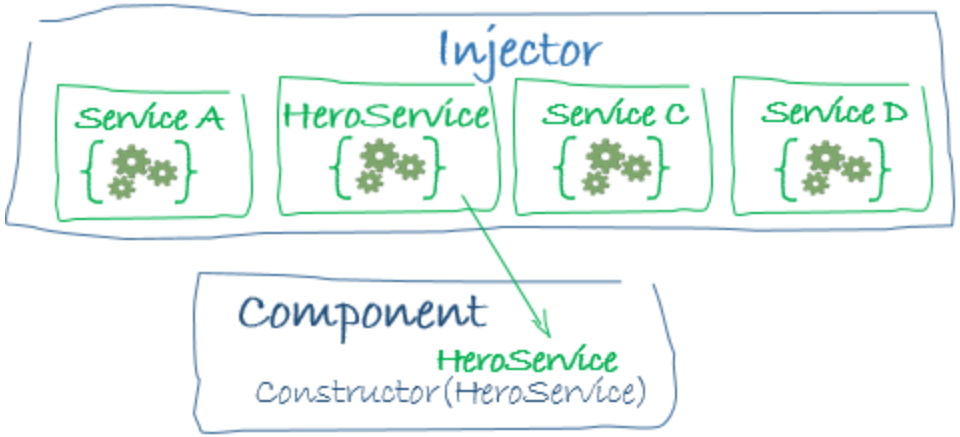






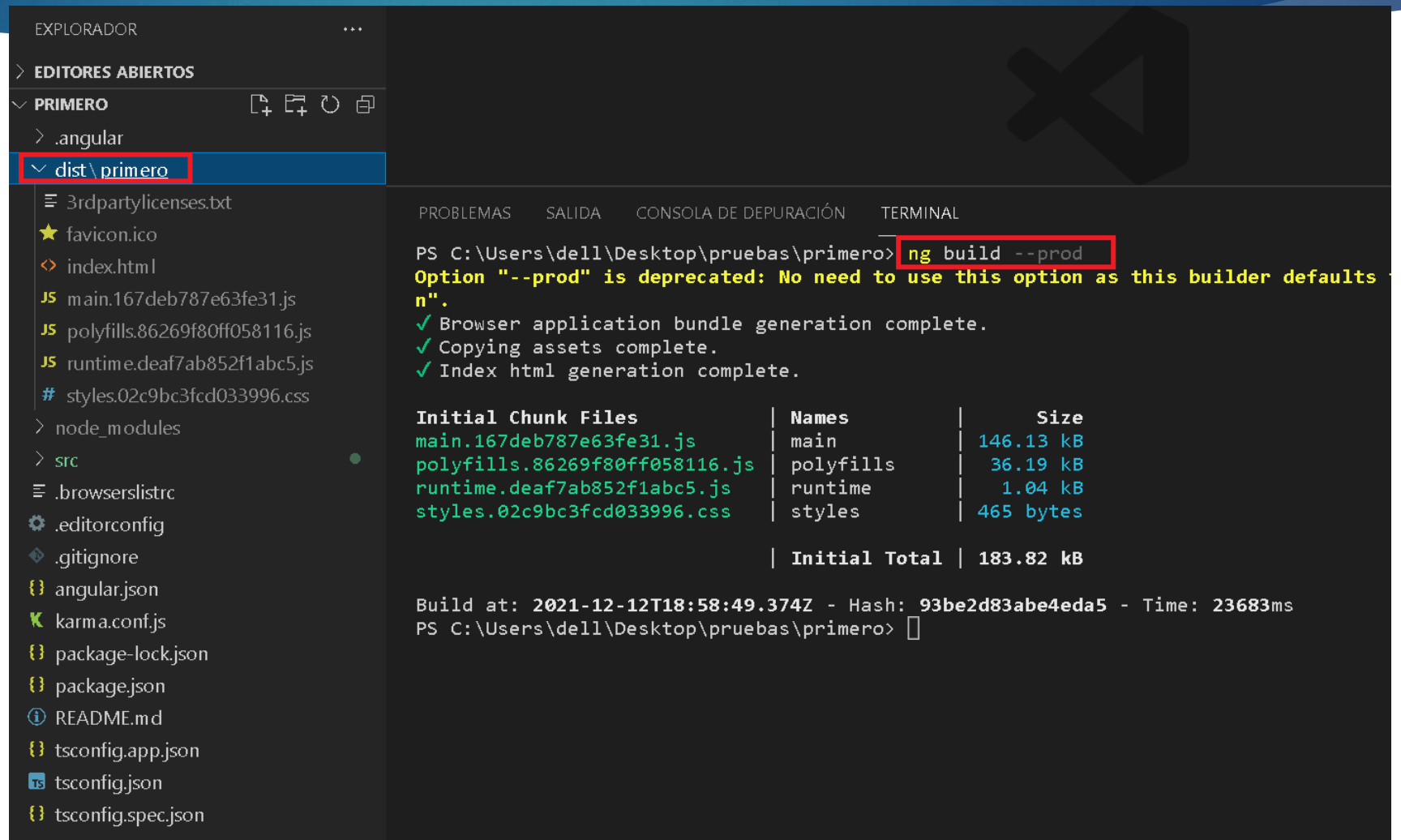
Resumen Servicios

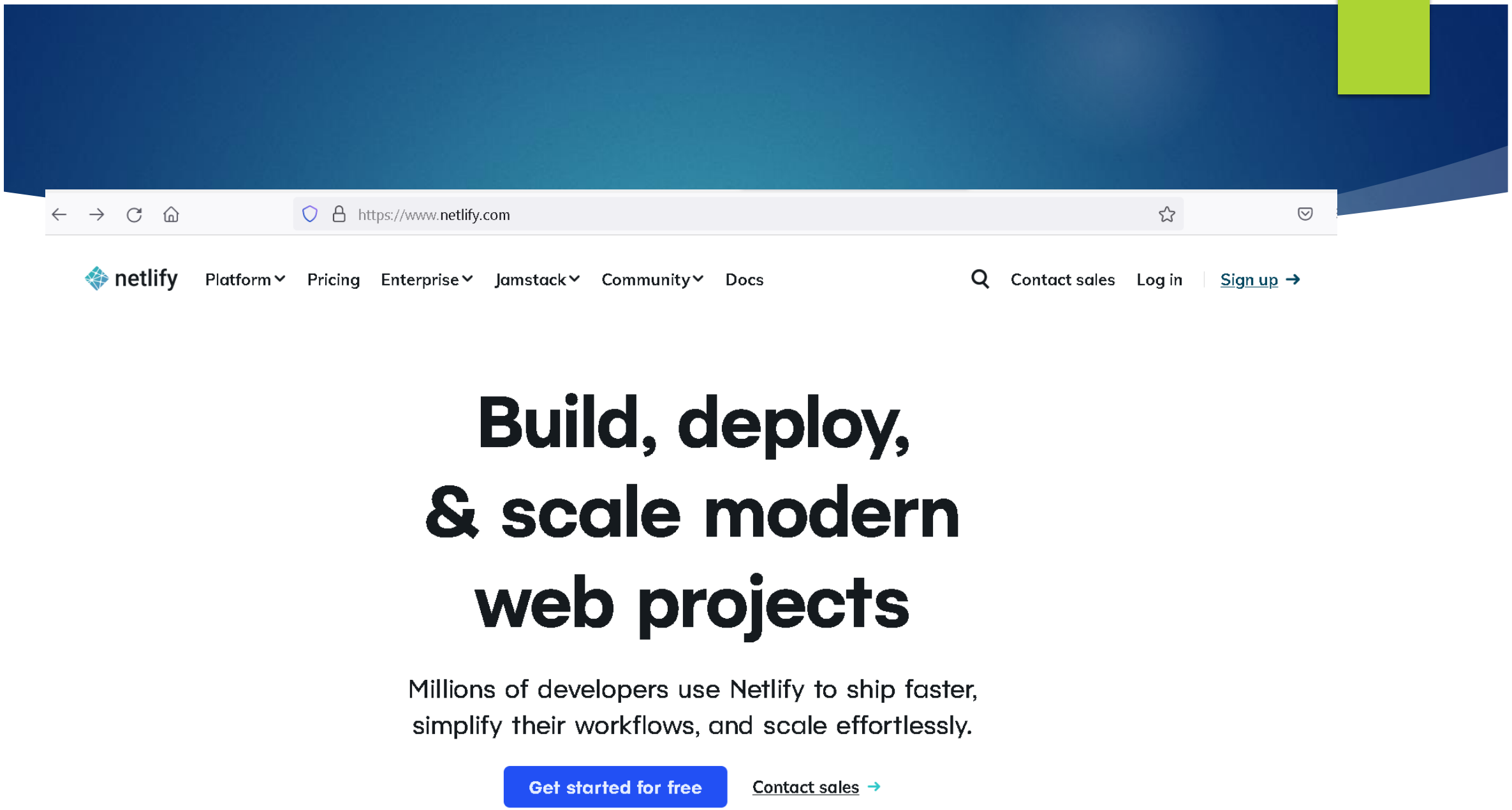




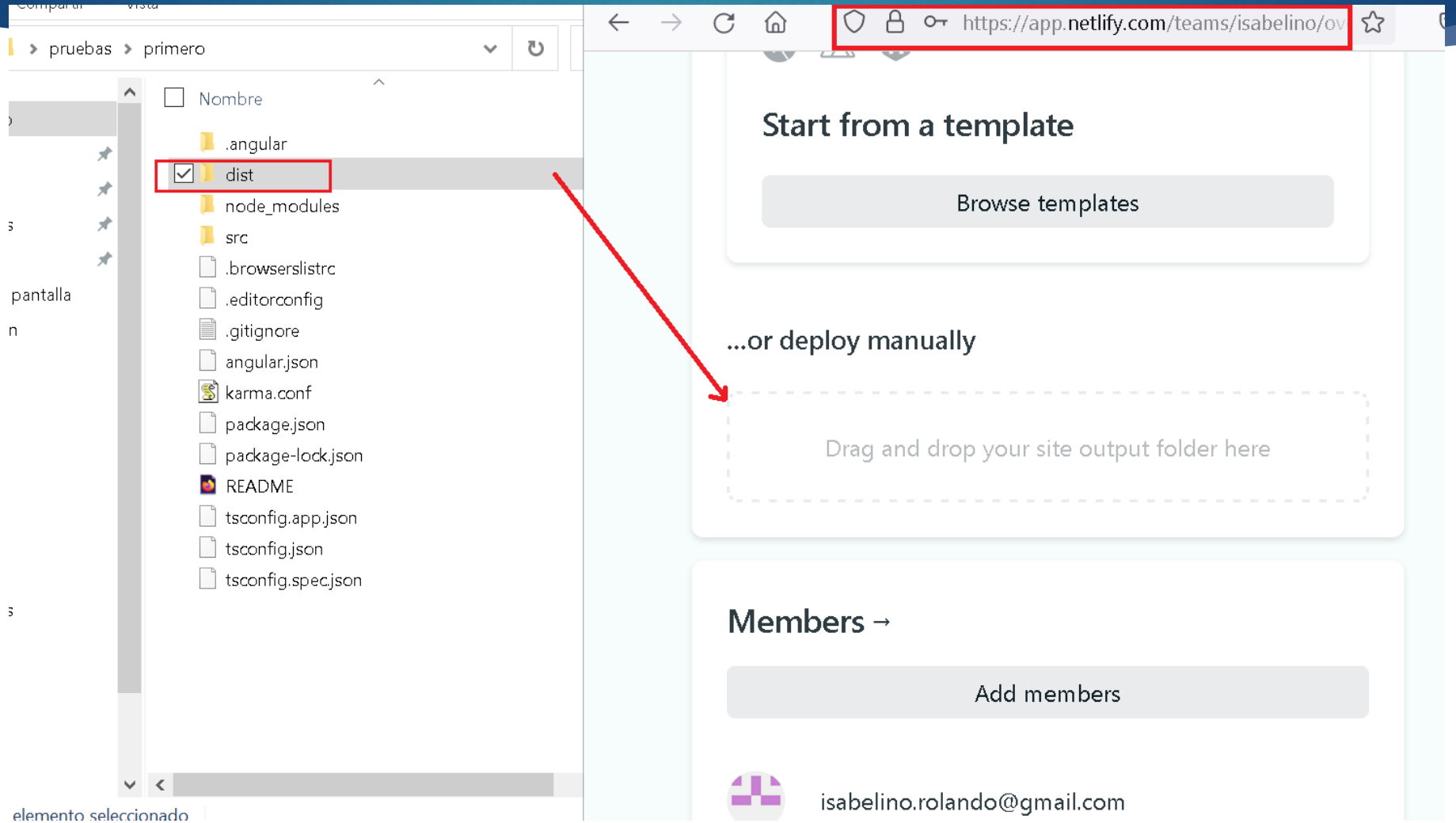
Lanzar nuestra app a producción

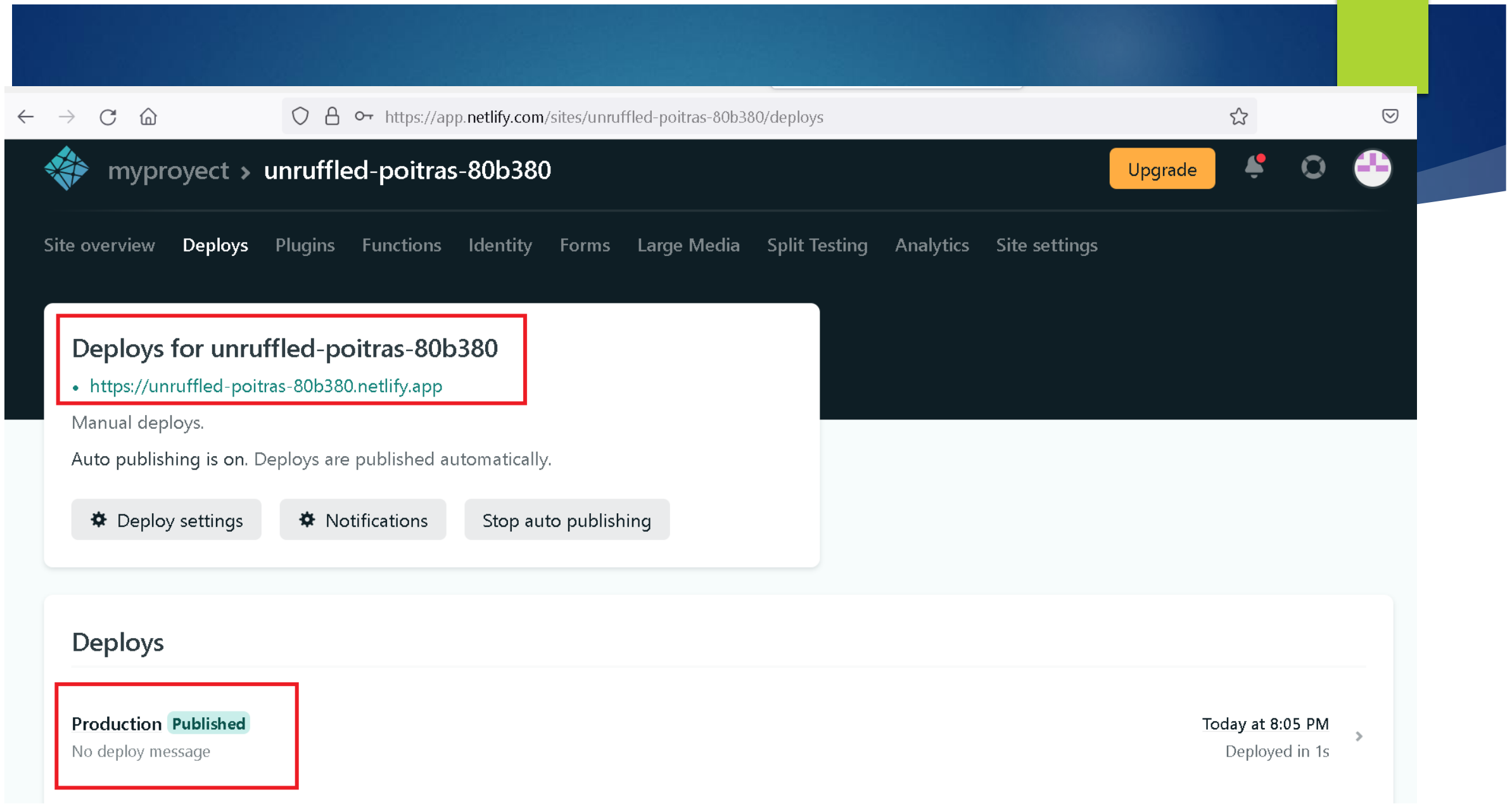
- ng build --prod

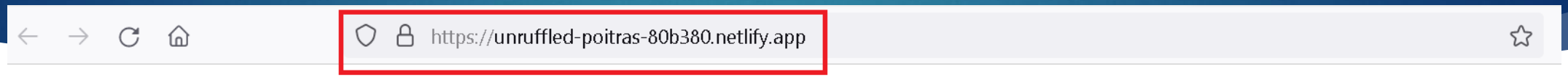



https://www.netlify.com
netlify
Platform
Pricing
Enterprise
Jamstack
Community
Docs
Contact sales
Log in
Sign up
Build, deploy, & scale modern web projects
Millions of developers use Netlify to ship faster, simplify their workflows, and scale effortlessly.
Get started for free
Contact sales

Arrastramos el archivo dist de esta manera




https://app.netlify.com/sites/unruffled-poitras-80b380/deploys
myproyect › unruffled-poitras-80b380
Upgrade
Site overview
Deploys
Plugins
Functions
Identity
Forms
Large Media
Split Testing
Analytics
Site settings
Deploys for unruffled-poitras-80b380
https://unruffled-poitras-80b380.netlify.app
Manual deploys.
Auto publishing is on. Deploys are published automatically.
Deploy settings
Notifications
Stop auto publishing
Deploys
Production Published
No deploy message
Today at 8:05 PM
Deployed in 1s



Dragon Ball Z

Personajes

- Vegeta - 20000
- Goku - 25000
- Mister Satan - 34

Agregar

Nombre

0

Agregar

Lifecycle hook

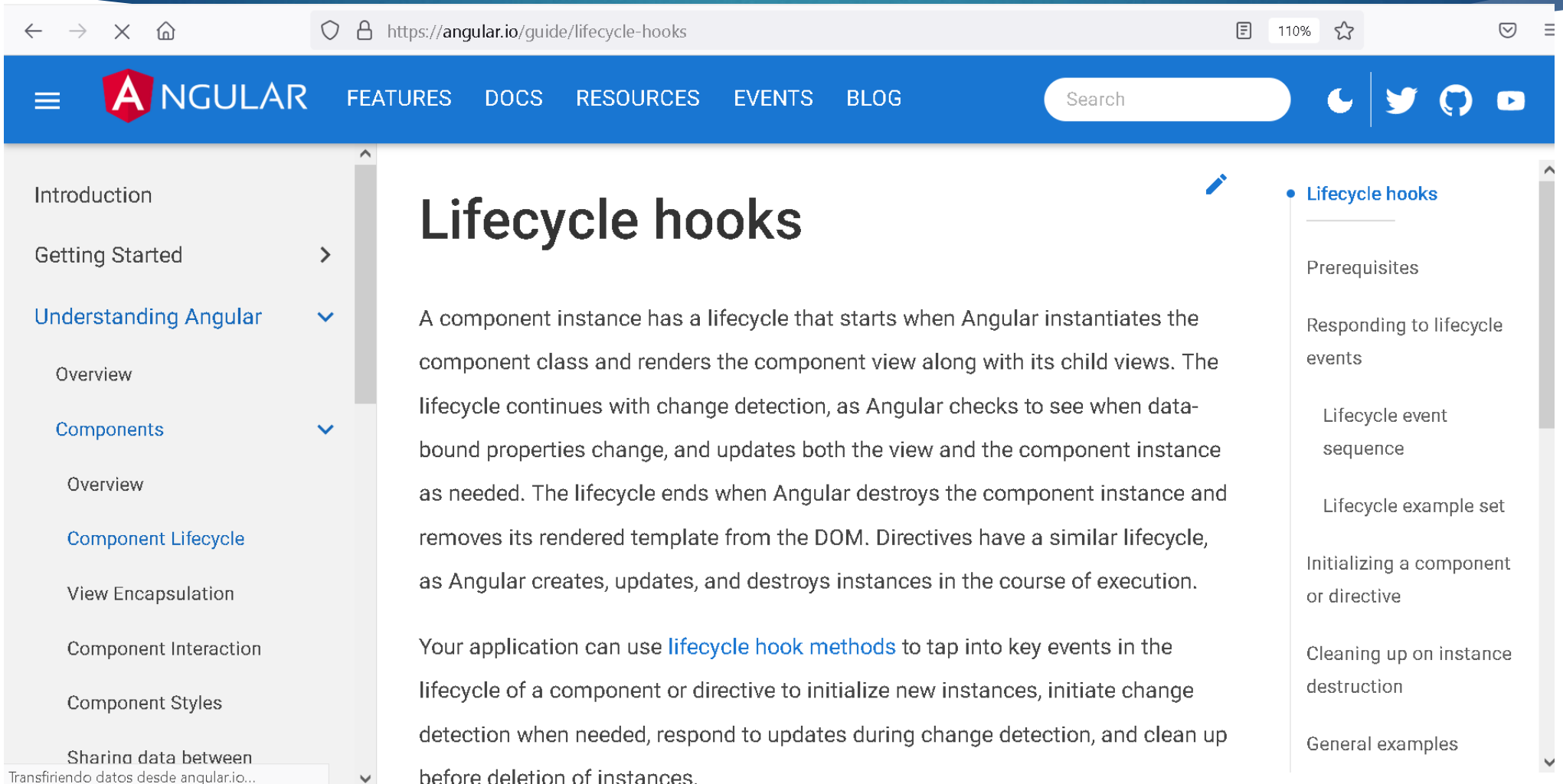
- Son métodos que podemos llamar cuando angular ejecute ciertos momentos como montaje de componentes, cuando se inicializa, cuando se actualiza el dom entre otros.

Creamos un nuevo proyecto angular

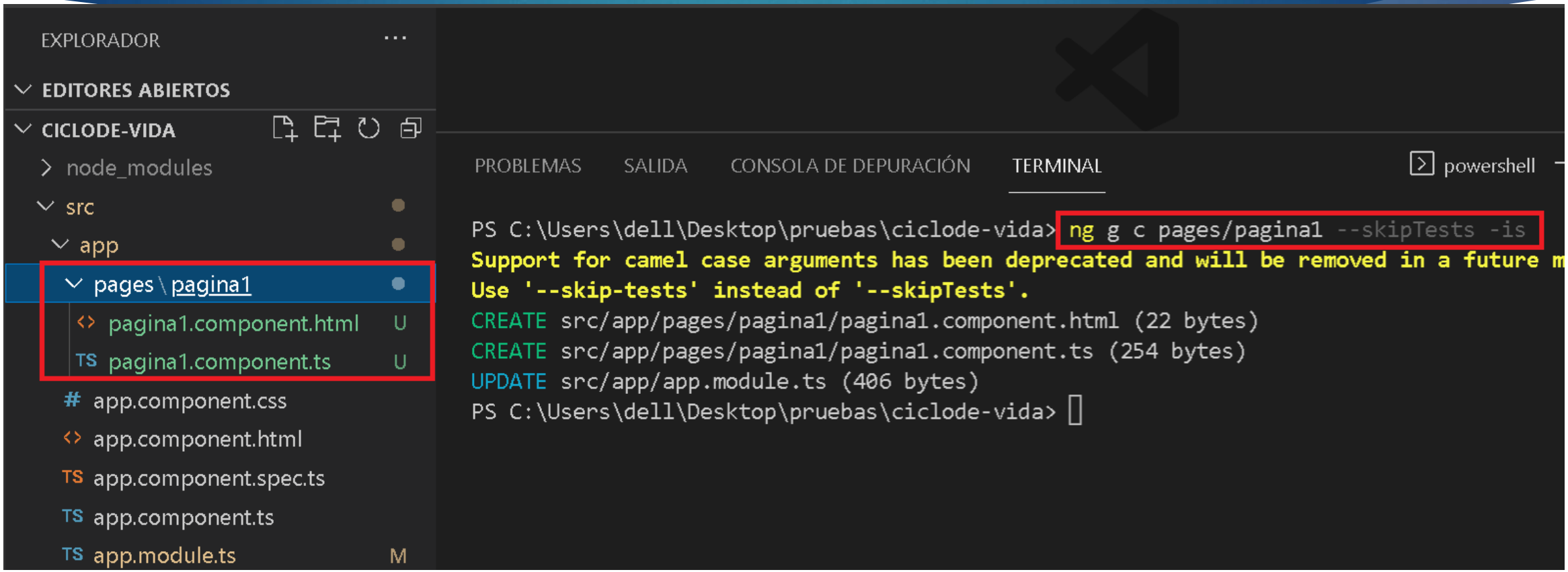
npm

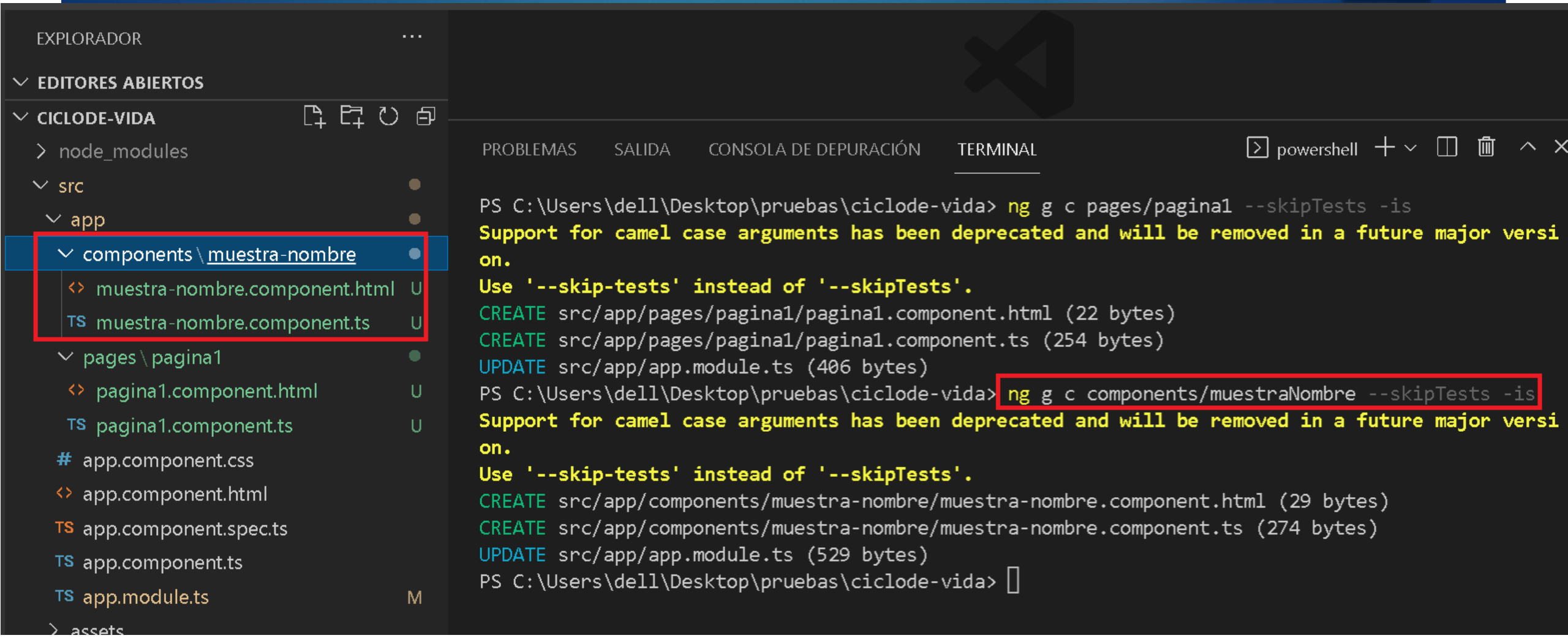
```
C:\Users\dell\Desktop\pruebas> ng new ciclode-vida
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? CSS
CREATE ciclode-vida/angular.json (3075 bytes)
CREATE ciclode-vida/package.json (1076 bytes)
CREATE ciclode-vida/README.md (1057 bytes)
CREATE ciclode-vida/tsconfig.json (863 bytes)
CREATE ciclode-vida/.editorconfig (274 bytes)
CREATE ciclode-vida/.gitignore (620 bytes)
```

La guía oficial

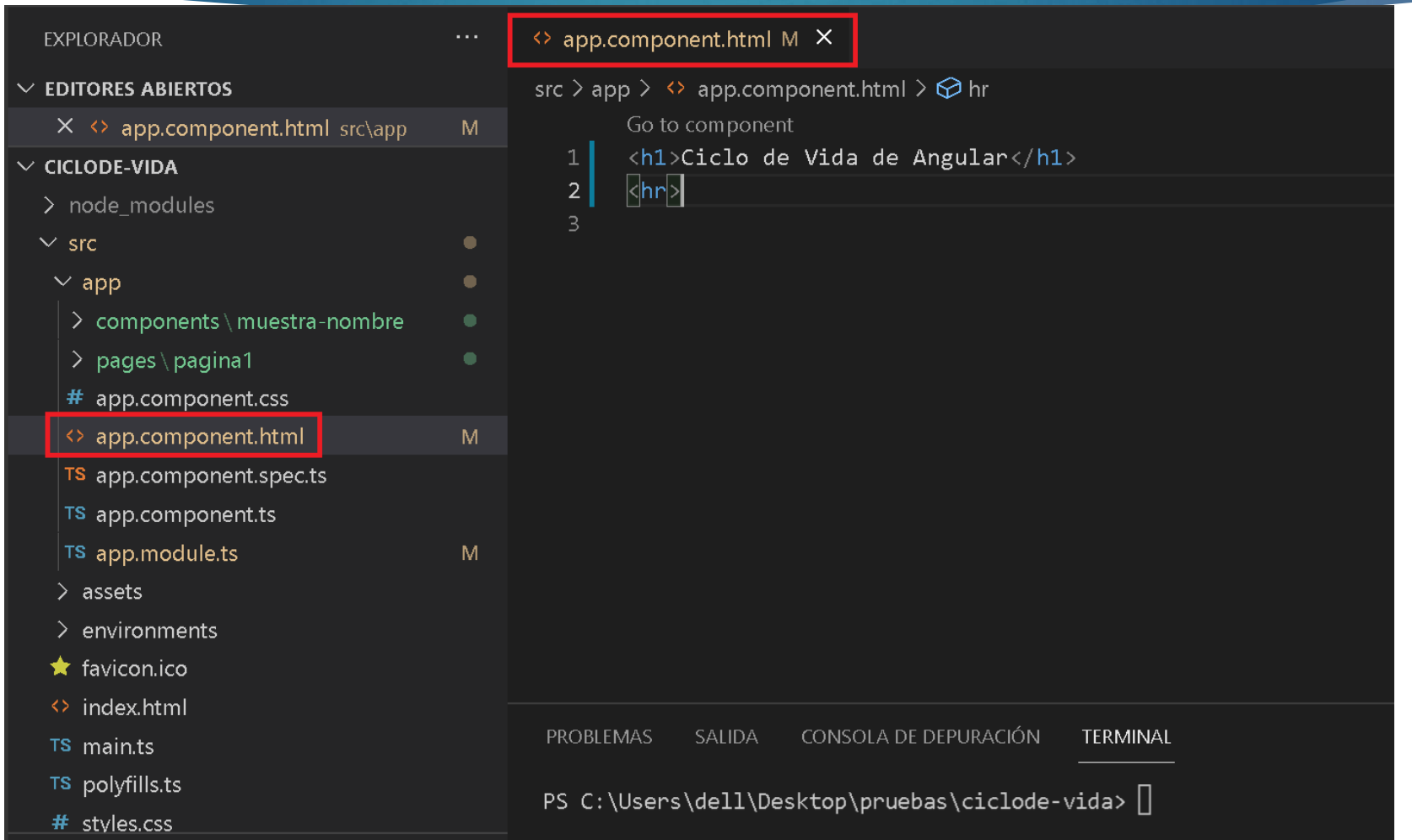


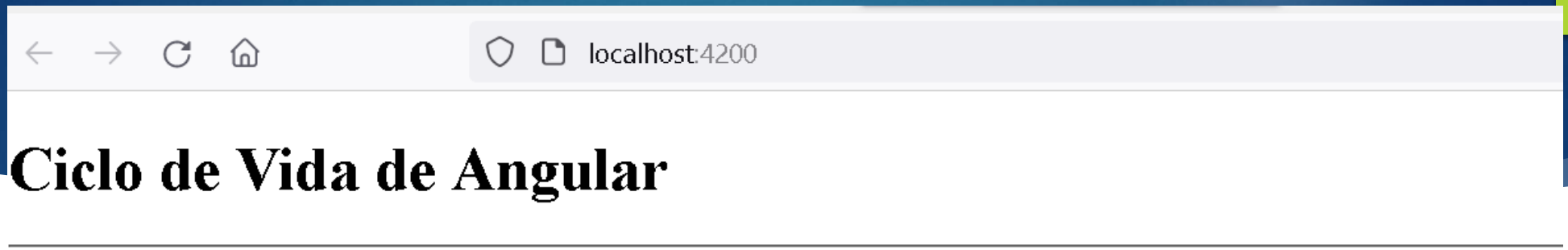
Creamos los siguientes componentes




EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
node_modules
src
app
components \ muestra-nombre
muestra-nombre.component.html U
muestra-nombre.component.ts U
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts M
assets
PROBLEMAS
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
powershell
PS C:\Users\dell\Desktop\pruebas\ciclode-vida> ng g c pages/pagina1 --skipTests -is
Support for camel case arguments has been deprecated and will be removed in a future major versi
on.
Use '--skip-tests' instead of '--skipTests'.
CREATE src/app/pages/pagina1/pagina1.component.html (22 bytes)
CREATE src/app/pages/pagina1/pagina1.component.ts (254 bytes)
UPDATE src/app/app.module.ts (406 bytes)
PS C:\Users\dell\Desktop\pruebas\ciclode-vida> ng g c components/muestraNombre --skipTests -is
Support for camel case arguments has been deprecated and will be removed in a future major versi
on.
Use '--skip-tests' instead of '--skipTests'.
CREATE src/app/components/muestra-nombre/muestra-nombre.component.html (29 bytes)
CREATE src/app/components/muestra-nombre/muestra-nombre.component.ts (274 bytes)
UPDATE src/app/app.module.ts (529 bytes)
PS C:\Users\dell\Desktop\pruebas\ciclode-vida>

El app.component.html lo dejamos así




localhost:4200
Ciclo de Vida de Angular

Método de gancho	Propósito	Momento
`ngOnChanges()`	Responder cuando Angular establece o restablece propiedades de entrada vinculadas a datos. El método recibe un `SimpleChanges` objeto de valores de propiedad actuales y anteriores. Tenga en cuenta que esto sucede con mucha frecuencia, por lo que cualquier operación que realice aquí tendrá un impacto significativo en el rendimiento. Consulte los detalles en Uso de ganchos de detección de cambios en este documento.	Se llama antes `ngOnInit()` (si el componente tiene entradas vinculadas) y cada vez que cambian una o más propiedades de entrada vinculadas a datos. Tenga en cuenta que si su componente no tiene entradas o lo usa sin proporcionar ninguna entrada, el marco no llamará `ngOnChanges()` .

Método de gancho	Propósito	Momento
`ngOnInit()`	Inicialice la directiva o el componente después de que Angular muestre primero las propiedades enlazadas a datos y establezca las propiedades de entrada de la directiva o del componente. Consulte los detalles en Inicialización de un componente o directiva en este documento.	Llamado una vez, después de la primera `ngOnChanges()`. `ngOnInit()` todavía se llama incluso cuando `ngOnChanges()` no lo es (que es el caso cuando no hay entradas vinculadas a la plantilla).

Método de gancho	Propósito	Momento
`ngDoCheck()`	Detecte y actúe sobre los cambios que Angular no puede o no detectará por sí solo. Consulte los detalles y el ejemplo en Definición de la detección de cambios personalizada en este documento.	Se llama inmediatamente después `ngOnChanges()` de cada ejecución de detección de cambios e inmediatamente después `ngOnInit()` de la primera ejecución.

Método de gancho	Propósito	Momento
`ngAfterContentInit()`	Responda después de que Angular proyecte contenido externo en la vista del componente o en la vista en la que se encuentra una directiva. Consulte los detalles y el ejemplo en Respuesta a cambios en el contenido de este documento.	Llamado *una vez* después del primero `ngDoCheck()`.

Método de gancho	Propósito	Momento
`ngAfterContentChecked()`	Responda después de que Angular verifique el contenido proyectado en la directiva o el componente. Consulte los detalles y el ejemplo en Respuesta a los cambios de contenido proyectados en este documento.	Llamado después `ngAfterContentInit()` y todos los posteriores `ngDoCheck()` .

Método de gancho	Propósito	Momento
`ngAfterViewInit()`	Responda después de que Angular inicialice las vistas del componente y las vistas secundarias, o la vista que contiene la directiva. Consulte los detalles y el ejemplo en Responder para ver los cambios en este documento.	Llamado *una vez* después del primero `ngAfterContentChecked()`.

Método de gancho	Propósito	Momento
`ngAfterViewChecked()`	Responda después de que Angular verifique las vistas del componente y las vistas secundarias, o la vista que contiene la directiva.	Llamado después del `ngAfterViewInit()` y todos los siguientes `ngAfterContentChecked()`.

Método de gancho	Propósito	Momento
`ngOnDestroy()`	Limpieza justo antes de que Angular destruya la directiva o el componente. Cancele la suscripción a Observables y desconecte los controladores de eventos para evitar pérdidas de memoria. Consulte los detalles en Limpieza de la destrucción de instancias en este documento.	Se llama inmediatamente antes de que Angular destruya la directiva o el componente.

En pagina1.component.ts agregamos esto

TS pagina1.component.ts U

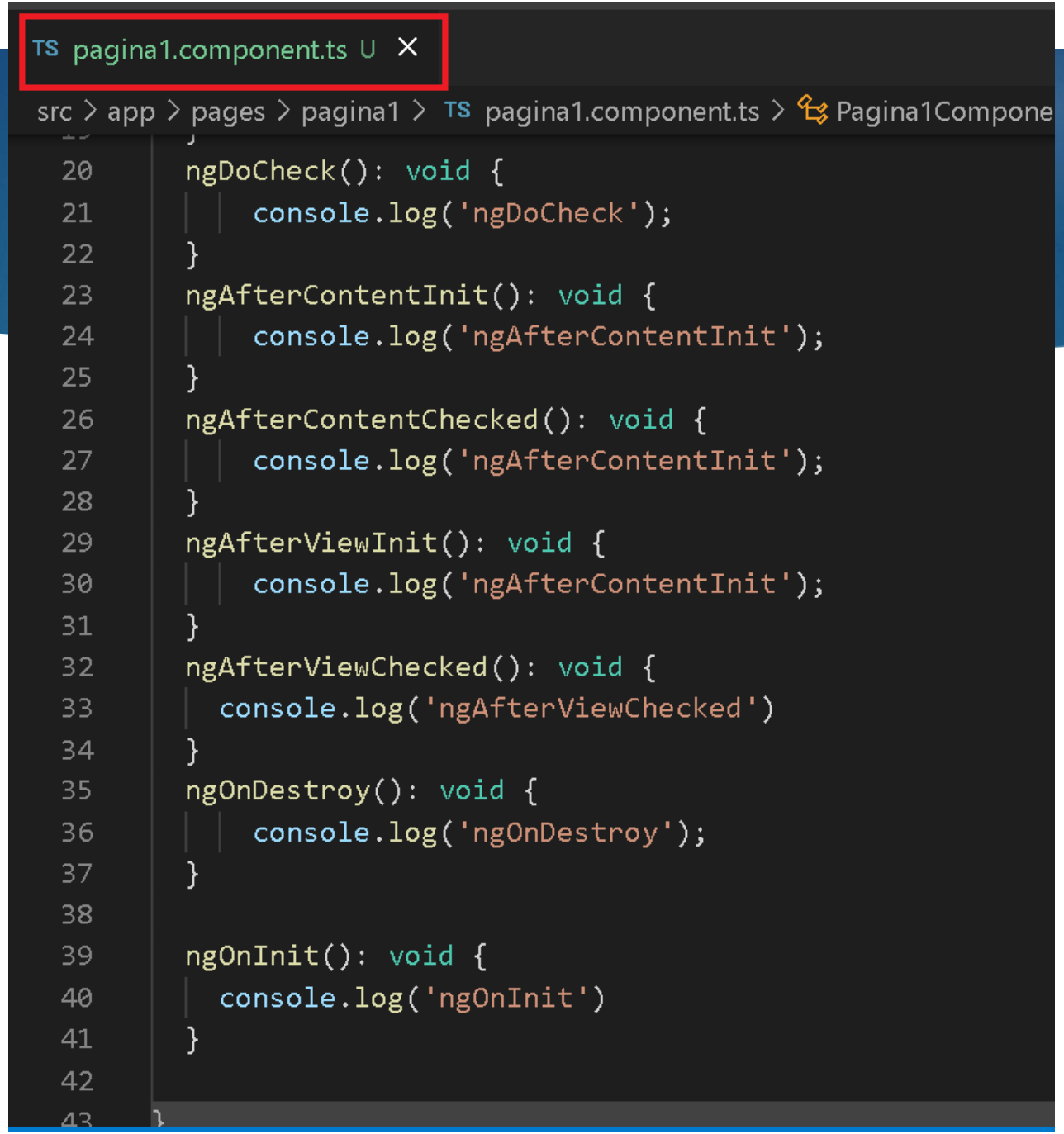
src > app > pages > pagina1 > TS pagina1.component.ts > Pagina1Component

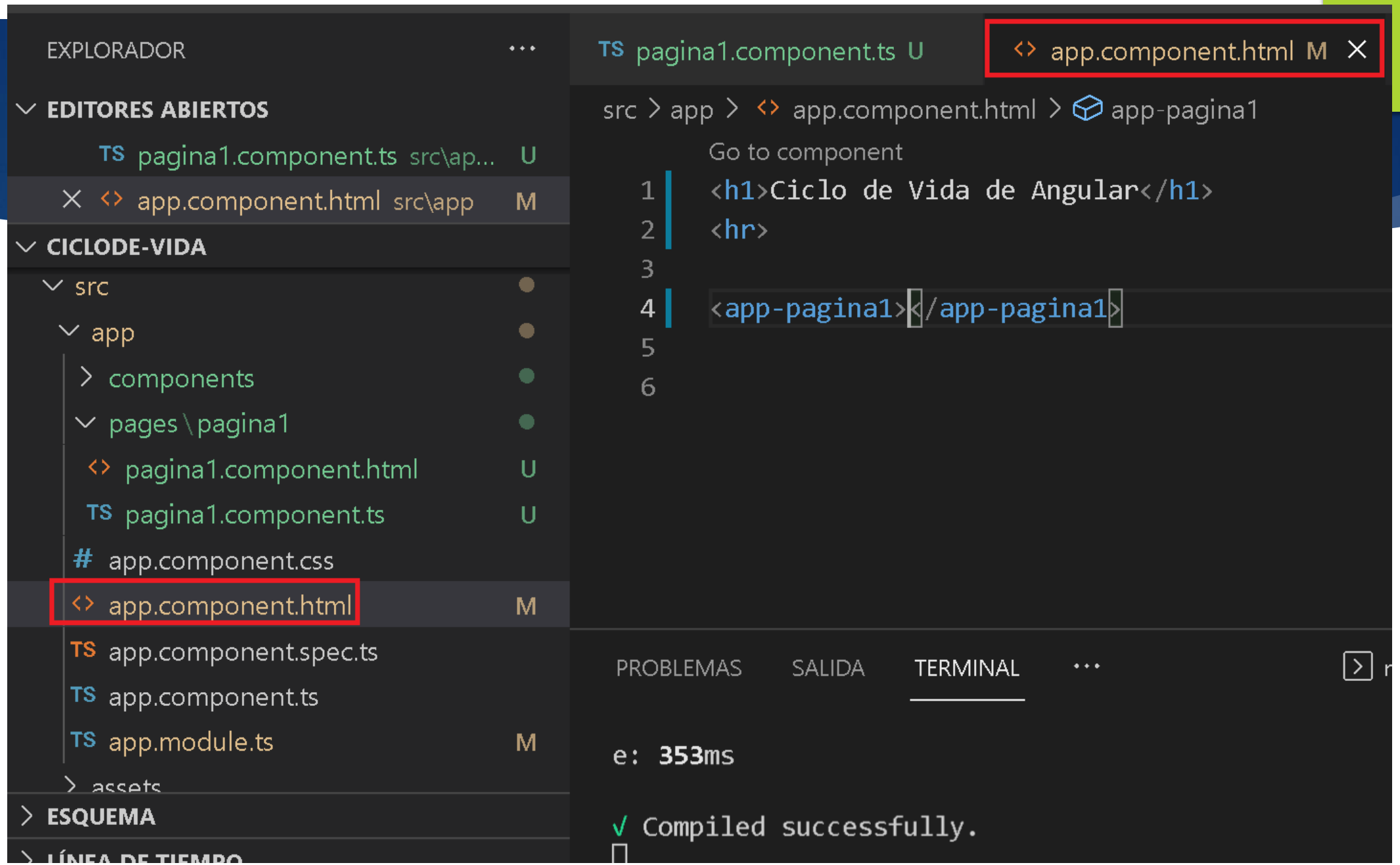
```ts
import { AfterContentChecked, AfterContentInit,
  AfterViewChecked, AfterViewInit,
   Component, DoCheck, OnChanges, OnDestroy,
    OnInit, SimpleChanges } from '@angular/core';

@Component({
  selector: 'app-pagina1',
  templateUrl: './pagina1.component.html',
  styles: [
  ]
})

export class Pagina1Component implements OnInit, OnChanges, DoCheck, AfterContentInit,
AfterContentChecked, AfterViewInit, AfterViewChecked, OnDestroy  {

  constructor() { }
  ngOnChanges(changes: SimpleChanges): void {
      console.log('ngOnChanges');
  }
  ngDoCheck(): void {
      console.log('ngDoCheck');
  }
  ngAfterContentInit(): void {
```

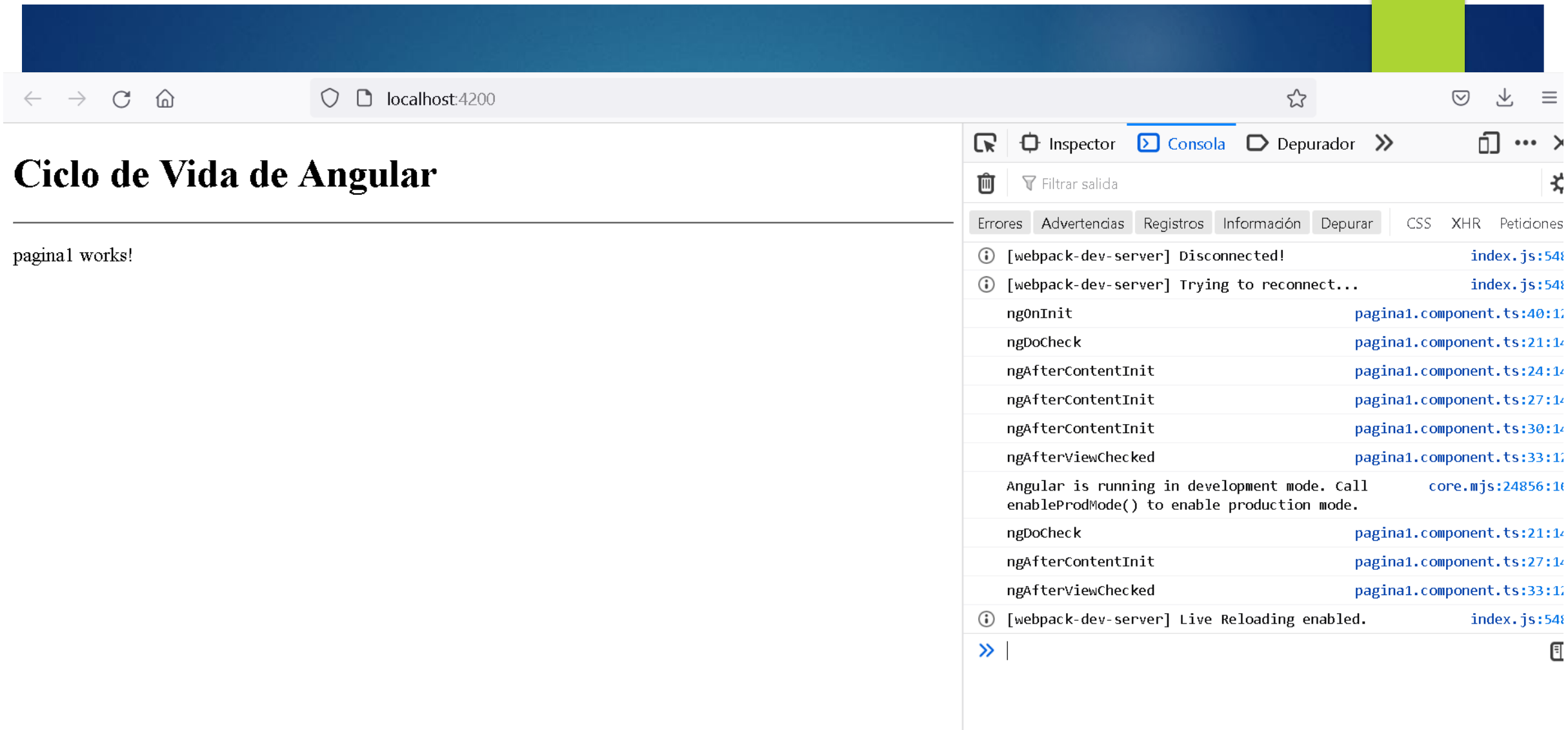
TS pagina1.component.ts U

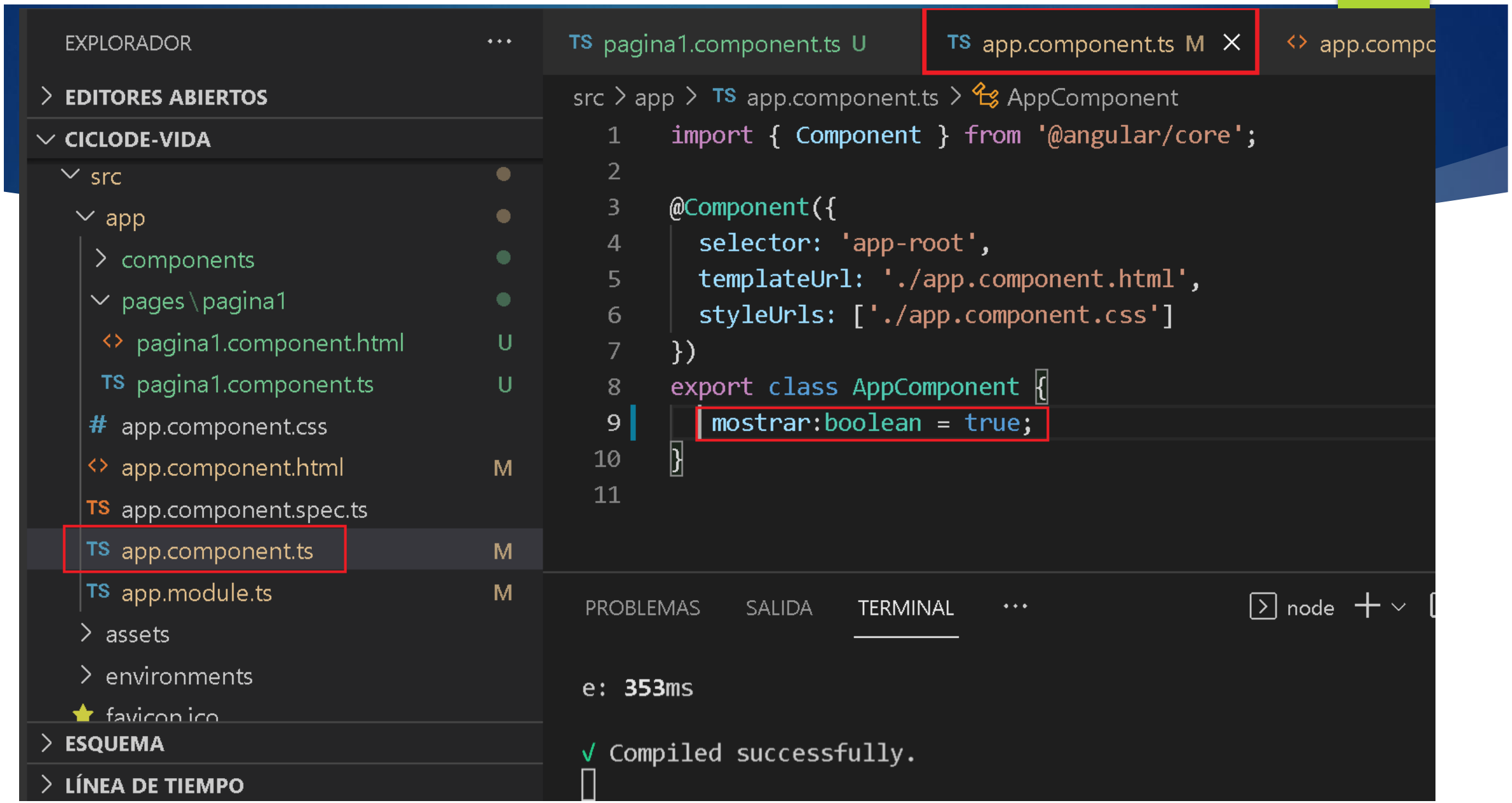
src > app > pages > pagina1 > TS pagina1.component.ts > Pagina1Component

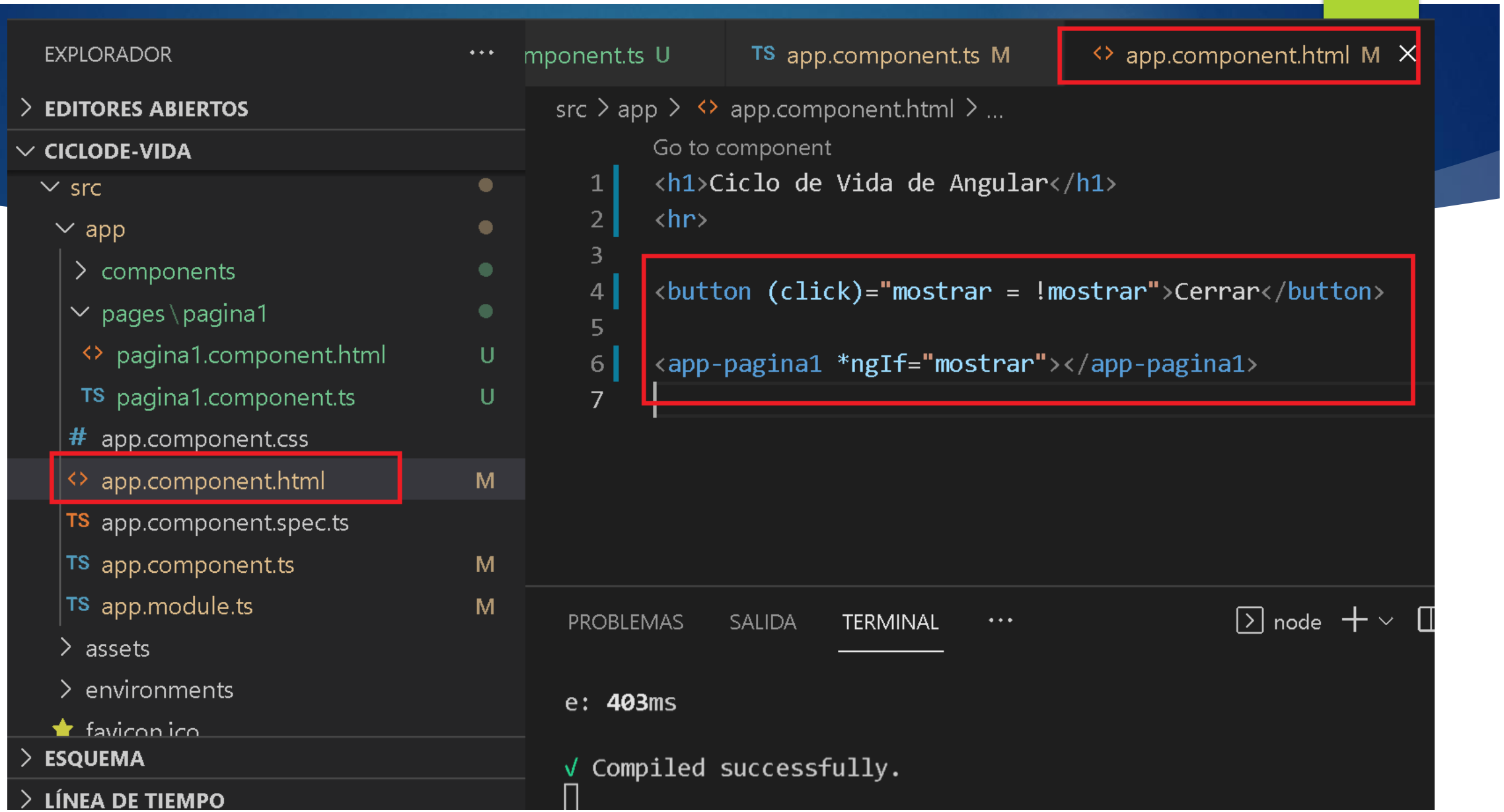
```
15
16    constructor() { }
17    ngOnChanges(changes: SimpleChanges): void {
18        console.log('ngOnChanges');
19    }
20    ngDoCheck(): void {
21        console.log('ngDoCheck');
22    }
23    ngAfterContentInit(): void {
24        console.log('ngAfterContentInit');
25    }
26    ngAfterContentChecked(): void {
27        console.log('ngAfterContentInit');
28    }
29    ngAfterViewInit(): void {
30        console.log('ngAfterContentInit');
31    }
32    ngAfterViewChecked(): void {
33      console.log('ngAfterViewChecked')
34    }
35    ngOnDestroy(): void {
36        console.log('ngOnDestroy');
37    }
38
```

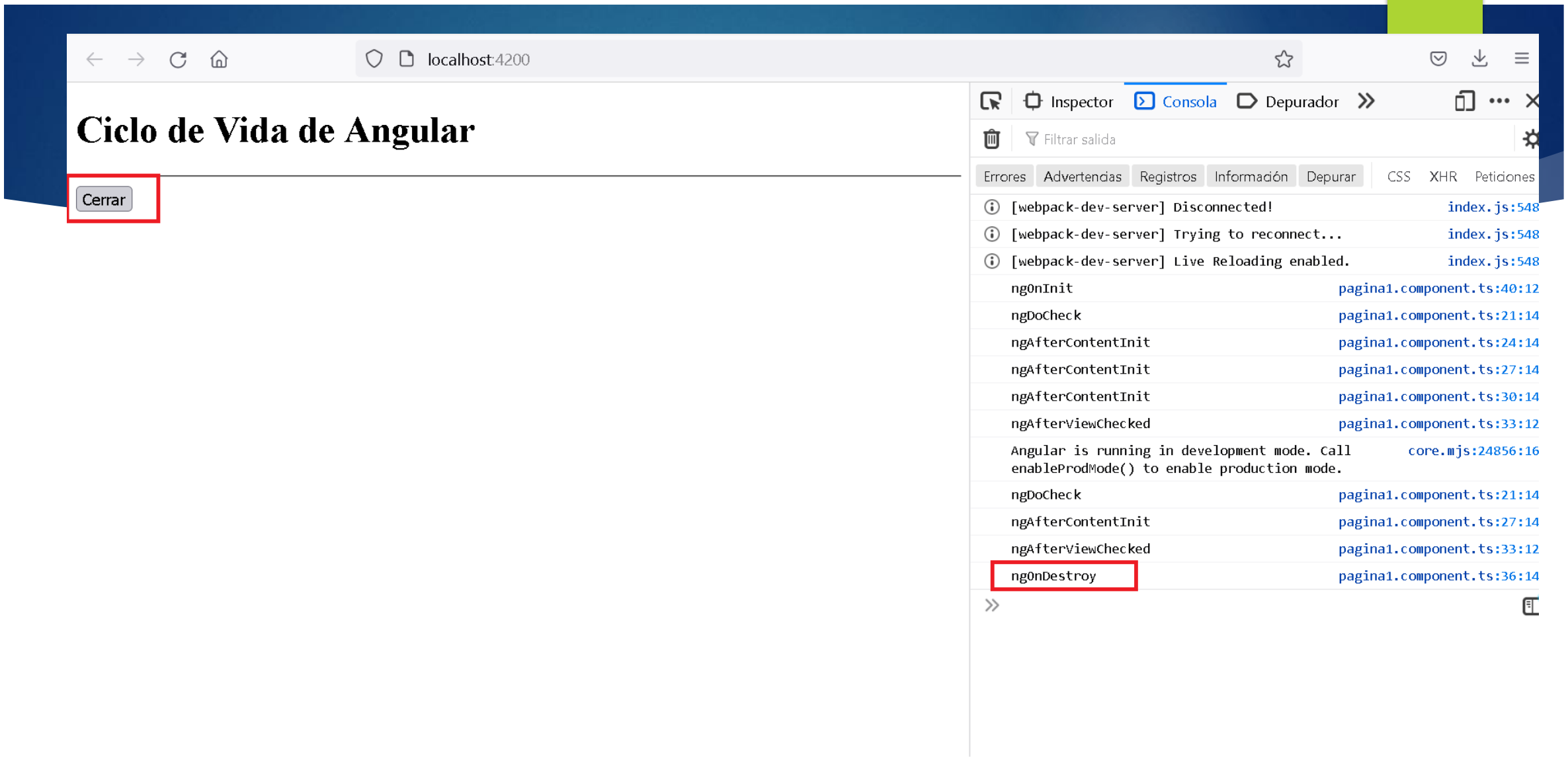

TS pagina1.component.ts U
src > app > pages > pagina1 > TS pagina1.component.ts > Pagina1Compone
ngDoCheck(): void {
console.log('ngDoCheck');
}
ngAfterContentInit(): void {
console.log('ngAfterContentInit');
}
ngAfterContentChecked(): void {
console.log('ngAfterContentInit');
}
ngAfterViewInit(): void {
console.log('ngAfterContentInit');
}
ngAfterViewChecked(): void {
console.log('ngAfterViewChecked')
}
ngOnDestroy(): void {
console.log('ngOnDestroy');
}
ngOnInit(): void {
console.log('ngOnInit')
}

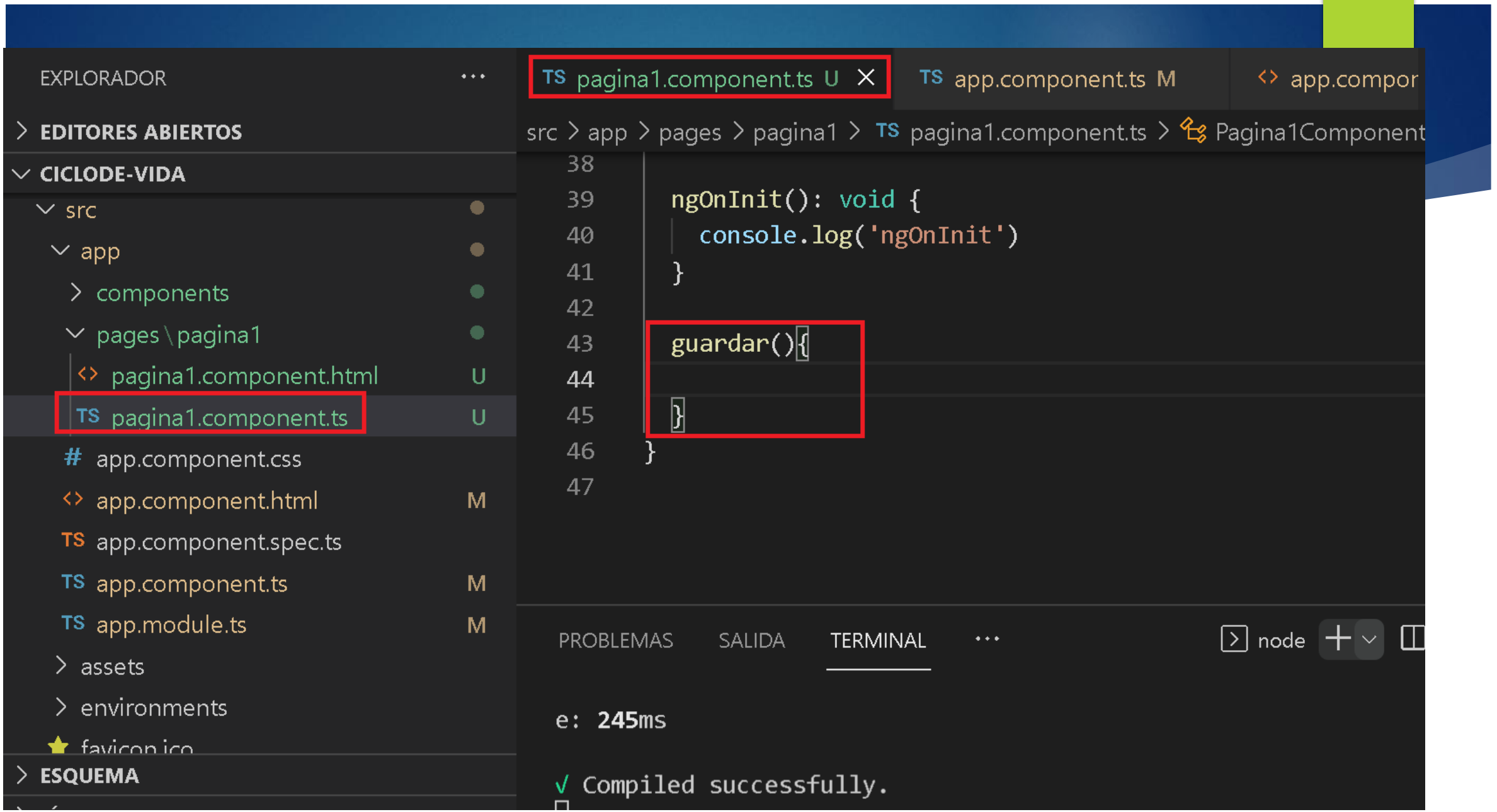

EXPLORADOR
EDITORES ABIERTOS
pagina1.component.ts src\ap... U
app.component.html src\app M
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts
app.module.ts M
ESQUEMA
pagina1.component.ts U
app.component.html M
src > app > app.component.html > app-pagina1
Go to component
1 <h1>Ciclo de Vida de Angular</h1>
2 <hr>
3
4 <app-pagina1></app-pagina1>
5
6
PROBLEMAS
SALIDA
TERMINAL
e: 353ms
√ Compiled successfully.

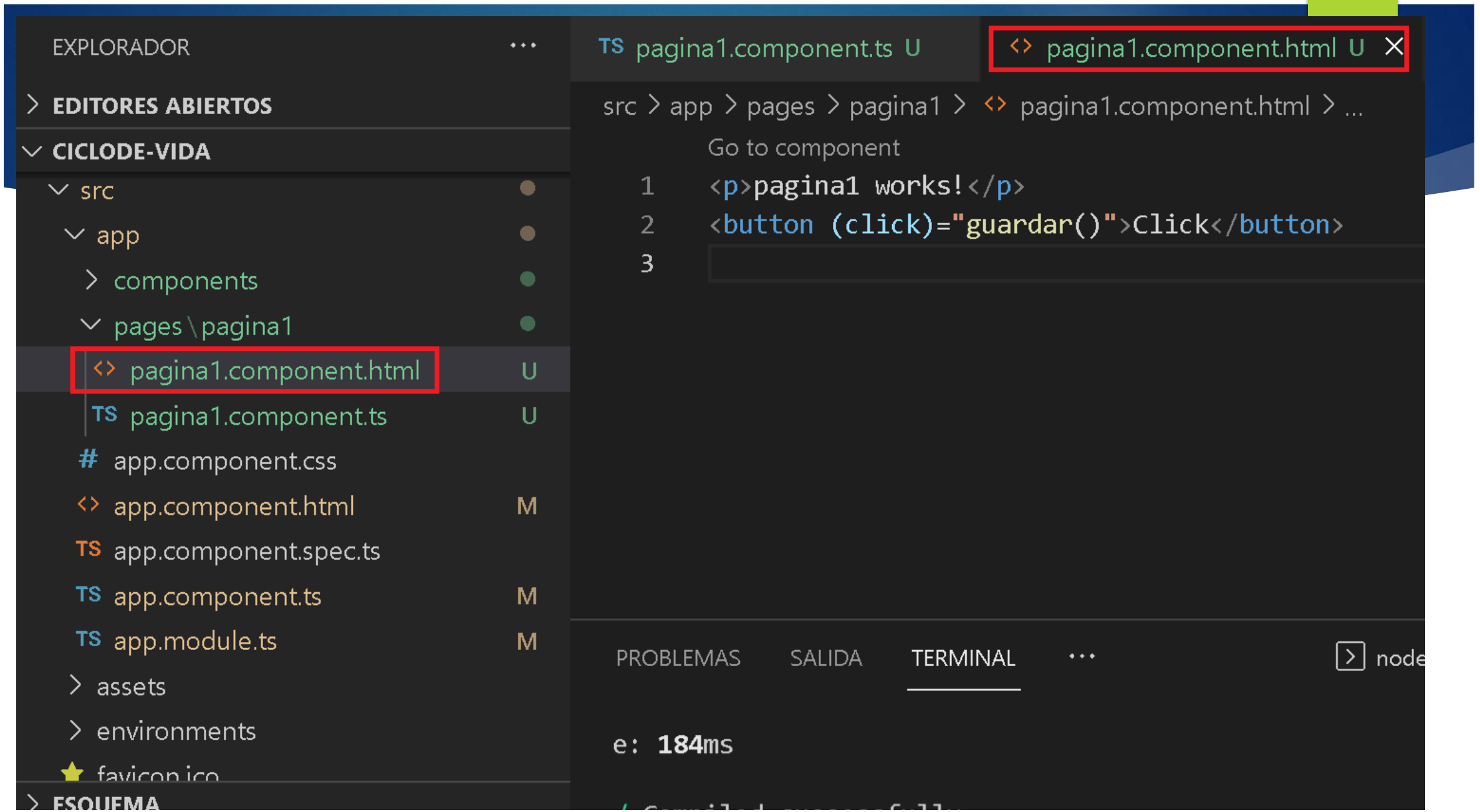

localhost:4200
Ciclo de Vida de Angular
pagina1 works!
Inspector
Consola
Depurador
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
[webpack-dev-server] Disconnected!
index.js:548
[webpack-dev-server] Trying to reconnect...
index.js:548
ngOnInit
pagina1.component.ts:40:17
ngDoCheck
pagina1.component.ts:21:14
ngAfterContentInit
pagina1.component.ts:24:14
ngAfterContentInit
pagina1.component.ts:27:14
ngAfterContentInit
pagina1.component.ts:30:14
ngAfterViewChecked
pagina1.component.ts:33:17
Angular is running in development mode. Call enableProdMode() to enable production mode.
core.mjs:24856:16
ngDoCheck
pagina1.component.ts:21:14
ngAfterContentInit
pagina1.component.ts:27:14
ngAfterViewChecked
pagina1.component.ts:33:17
[webpack-dev-server] Live Reloading enabled.
index.js:548

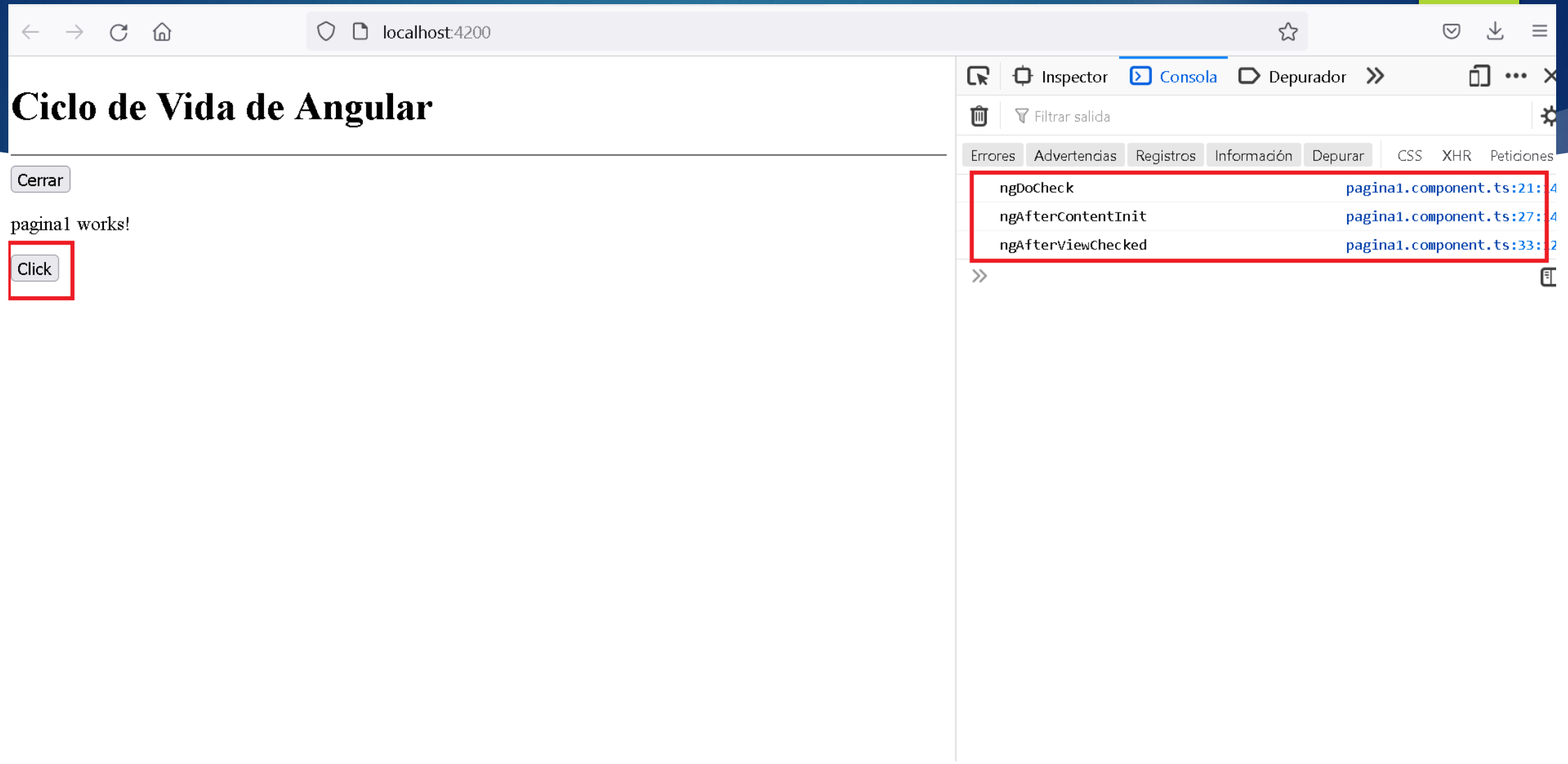

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
ESQUEMA
LÍNEA DE TIEMPO
pagina1.component.ts U
app.component.ts M
app.compc
src > app > app.component.ts > AppComponent
1 import { Component } from '@angular/core';
2
3 @Component({
4 selector: 'app-root',
5 templateUrl: './app.component.html',
6 styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9 mostrar:boolean = true;
10 }
11
PROBLEMAS
SALIDA
TERMINAL
node
e: 353ms
√ Compiled successfully.

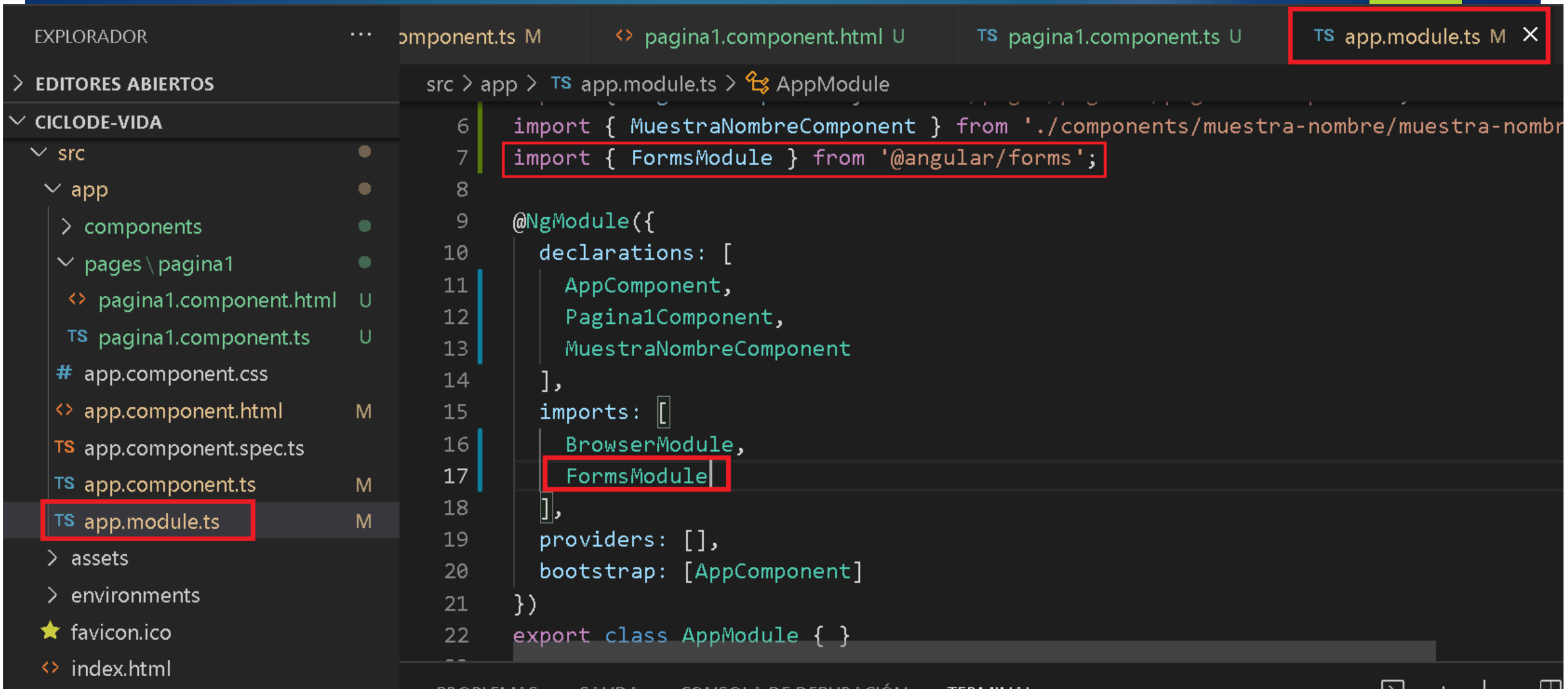

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
ESQUEMA
LÍNEA DE TIEMPO
mponent.ts U
app.component.ts M
app.component.html M
src > app > app.component.html > ...
Go to component
1 <h1>Ciclo de Vida de Angular</h1>
2 <hr>
3
4 <button (click)="mostrar = !mostrar">Cerrar</button>
5
6 <app-pagina1 *ngIf="mostrar"></app-pagina1>
7
PROBLEMAS
SALIDA
TERMINAL
node
e: 403ms
√ Compiled successfully.

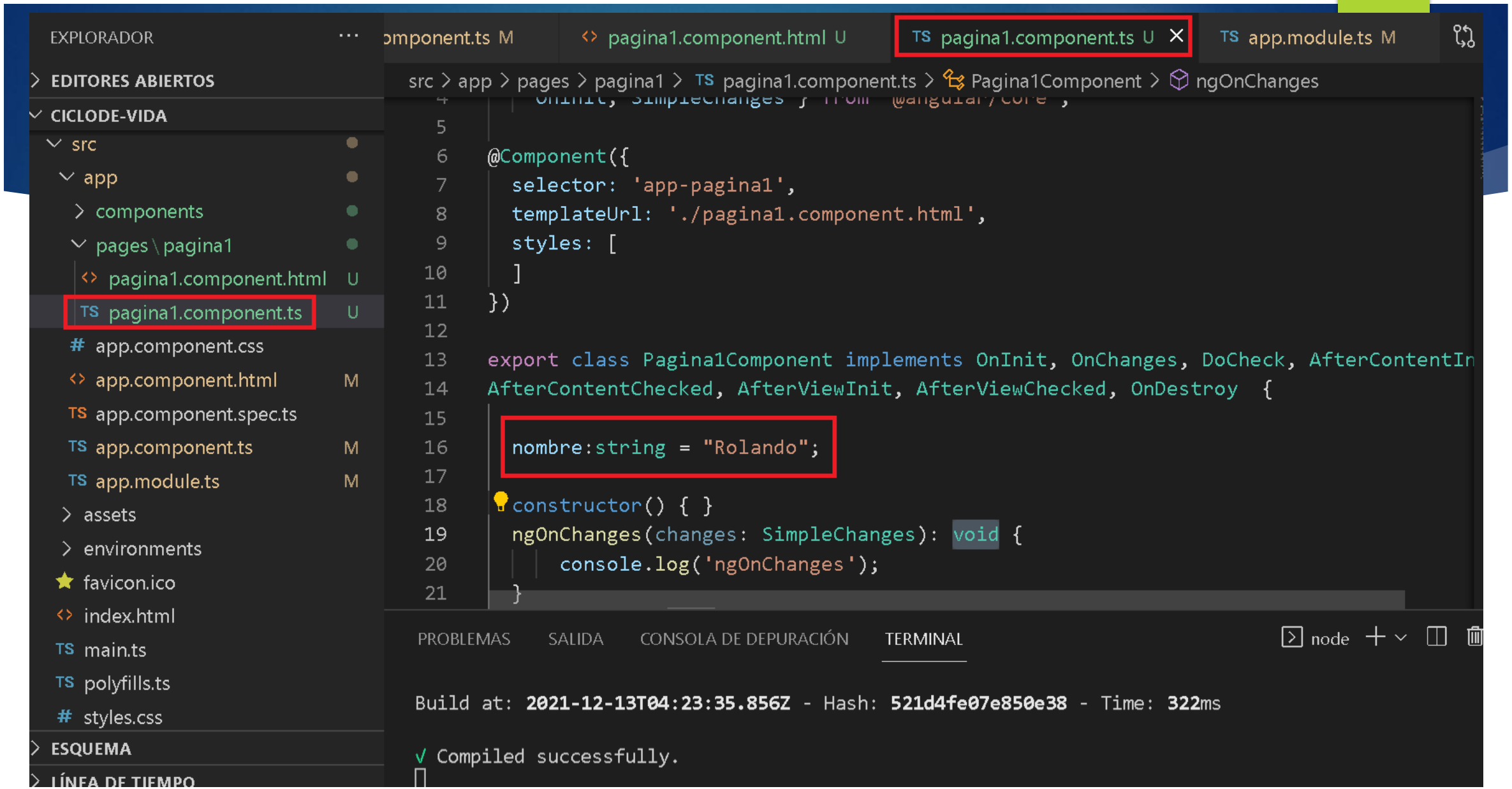

localhost:4200
Ciclo de Vida de Angular
Cerrar
Inspector
Consola
Depurador
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
[webpack-dev-server] Disconnected! index.js:548
[webpack-dev-server] Trying to reconnect... index.js:548
[webpack-dev-server] Live Reloading enabled. index.js:548
ngOnInit pagina1.component.ts:40:12
ngDoCheck pagina1.component.ts:21:14
ngAfterContentInit pagina1.component.ts:24:14
ngAfterContentInit pagina1.component.ts:27:14
ngAfterContentInit pagina1.component.ts:30:14
ngAfterViewChecked pagina1.component.ts:33:12
Angular is running in development mode. Call enableProdMode() to enable production mode. core.mjs:24856:16
ngDoCheck pagina1.component.ts:21:14
ngAfterContentInit pagina1.component.ts:27:14
ngAfterViewChecked pagina1.component.ts:33:12
ngOnDestroy pagina1.component.ts:36:14

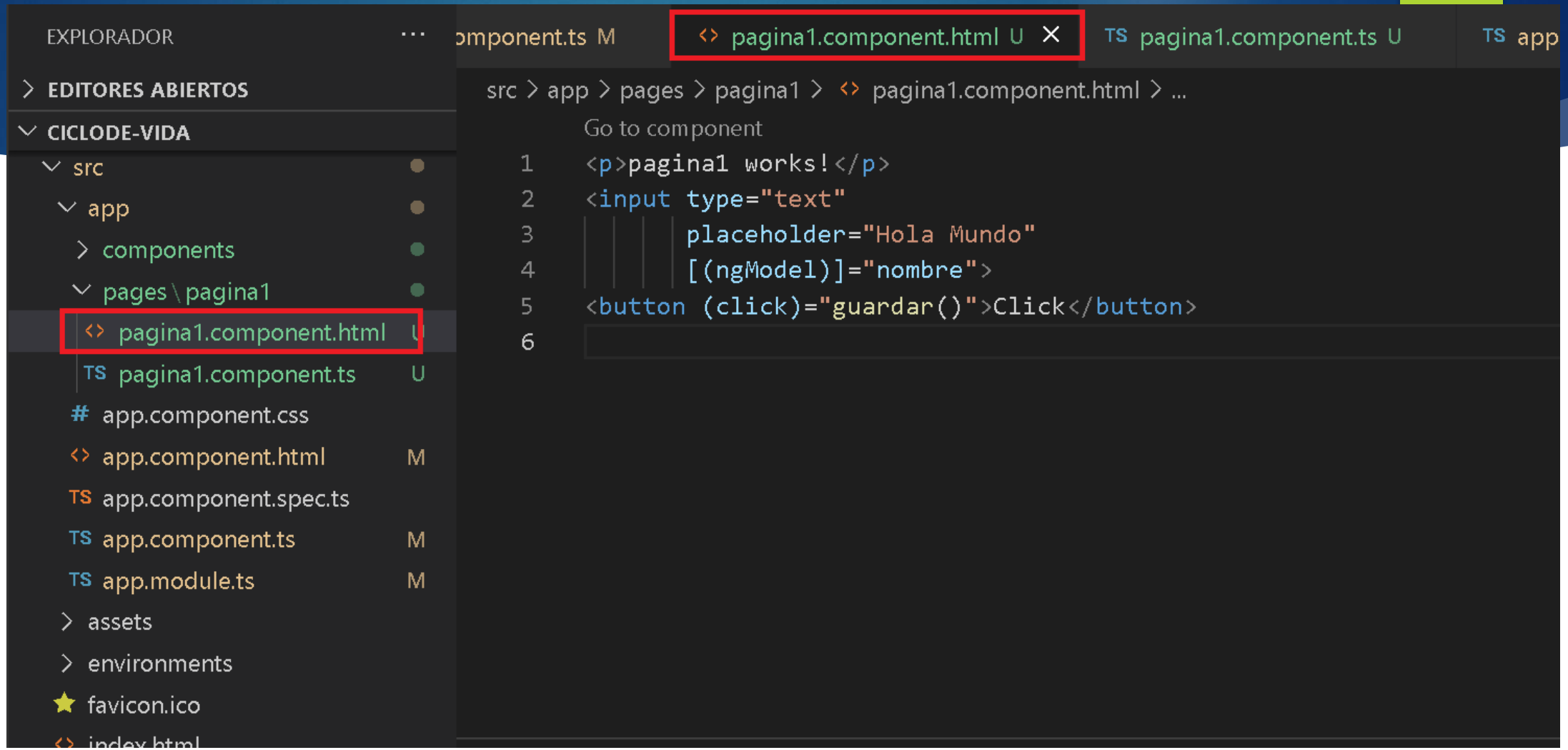

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
ESQUEMA
pagina1.component.ts U
app.component.ts M
app.compor
src > app > pages > pagina1 > pagina1.component.ts > Pagina1Component
38
39 ngOnInit(): void {
40 console.log('ngOnInit')
41 }
42
43 guardar(){
44
45 }
46 }
47
PROBLEMAS
SALIDA
TERMINAL
node
e: 245ms
√ Compiled successfully.

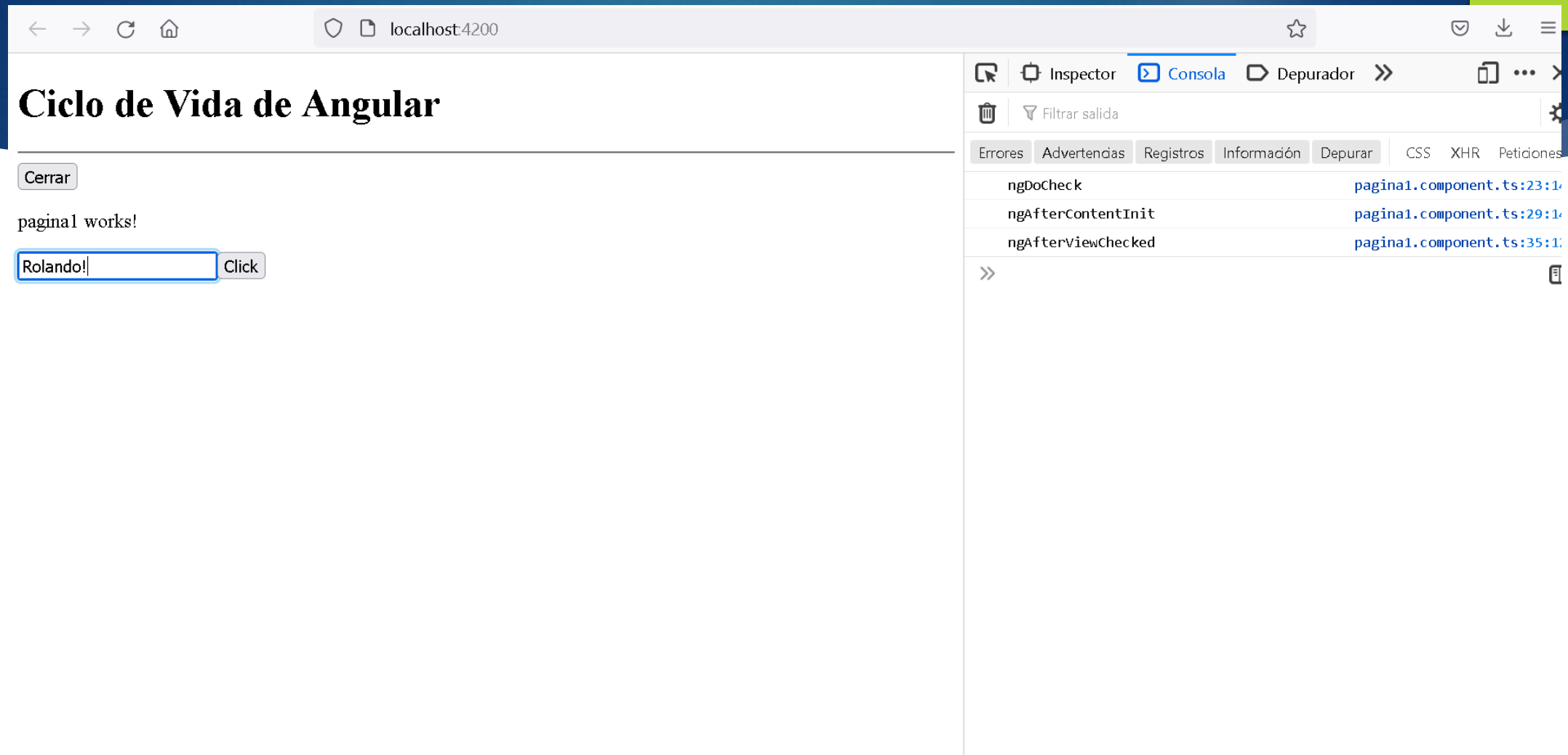

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages\pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
ESQUEMA
pagina1.component.ts U
pagina1.component.html U
src > app > pages > pagina1 > pagina1.component.html > ...
Go to component
1 pagina1 works!
2 <button (click)="guardar()">Click</button>
3
PROBLEMAS
SALIDA
TERMINAL
node
e: 184ms


localhost:4200
Ciclo de Vida de Angular
Cerrar
pagina1 works!
Click
Inspector
Consola
Depurador
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
ngDoCheck
pagina1.component.ts:21:
ngAfterContentInit
pagina1.component.ts:27:
ngAfterViewChecked
pagina1.component.ts:33:

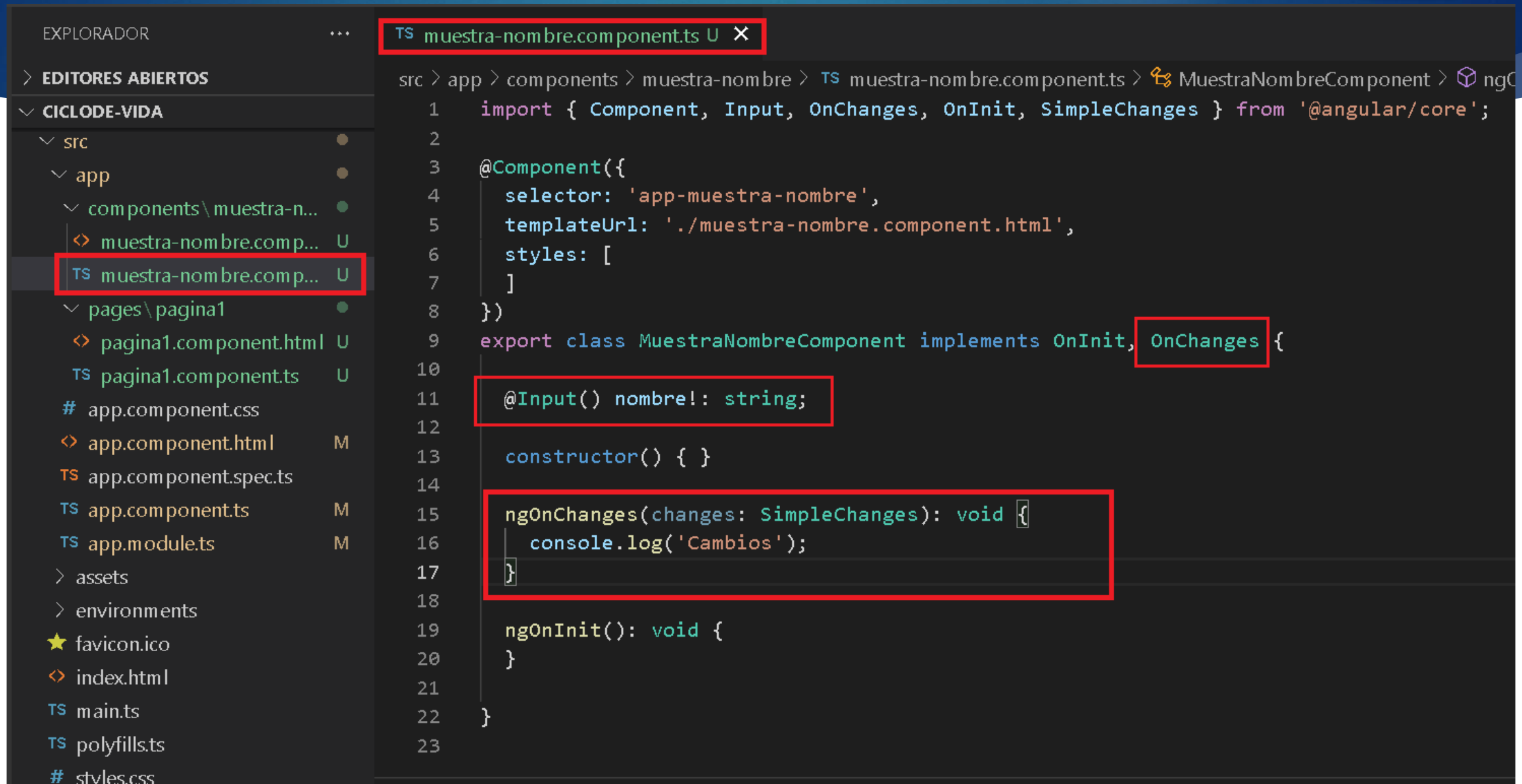

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
index.html
omponent.ts M
pagina1.component.html U
pagina1.component.ts U
app.module.ts M
src > app > app.module.ts > AppModule
import { MuestraNombreComponent } from './components/muestra-nombre/muestra-nombr
import { FormsModule } from '@angular/forms';
@NgModule({
declarations: [
AppComponent,
Pagina1Component,
MuestraNombreComponent
],
imports: [
BrowserModule,
FormsModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

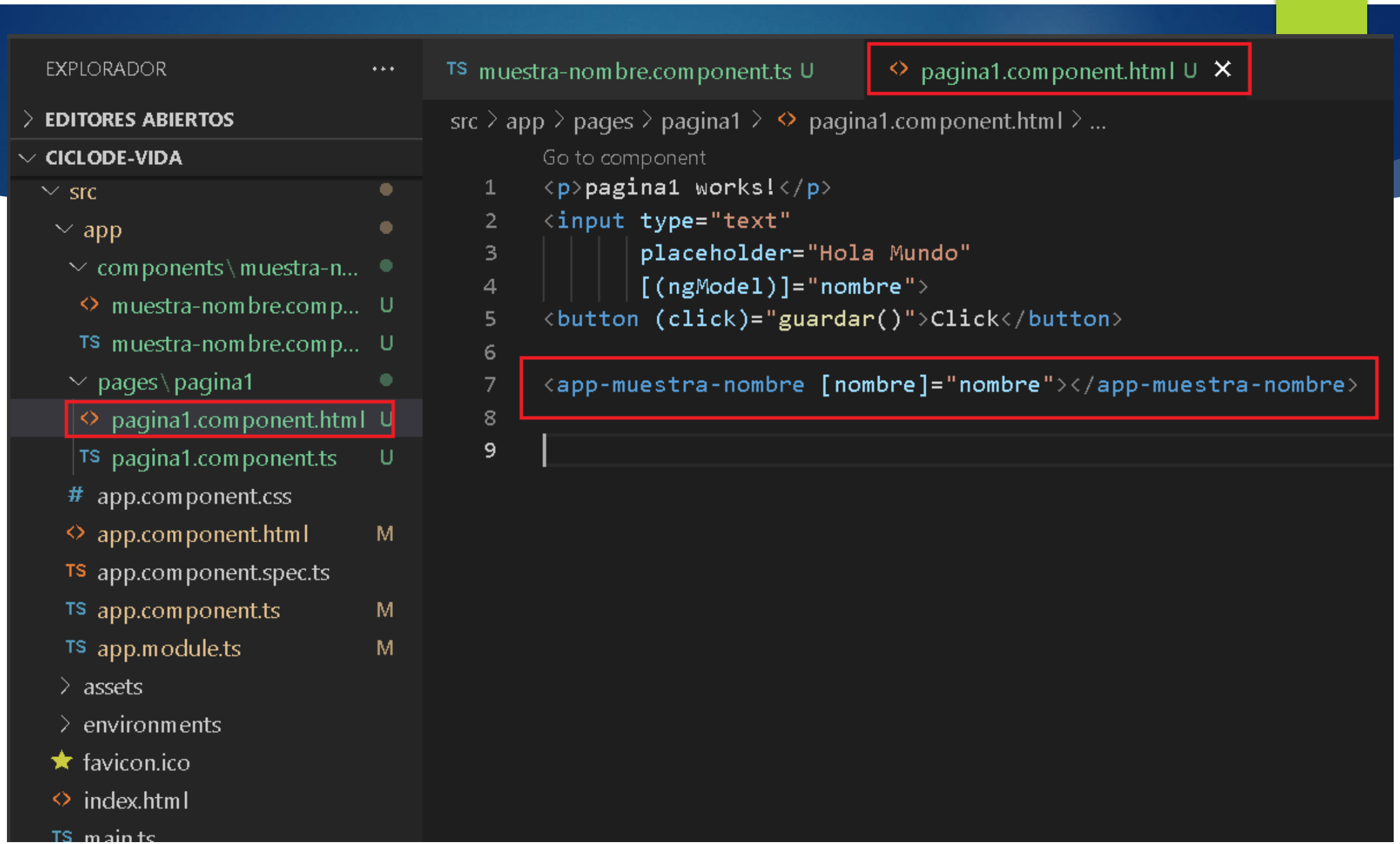

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css
ESQUEMA
LÍNEA DE TIEMPO
omponent.ts M
pagina1.component.html U
pagina1.component.ts U
app.module.ts M
src > app > pages > pagina1 > pagina1.component.ts > Pagina1Component > ngOnChanges
5
6 @Component({
7 selector: 'app-pagina1',
8 templateUrl: './pagina1.component.html',
9 styles: [
10]
11 })
12
13 export class Pagina1Component implements OnInit, OnChanges, DoCheck, AfterContentIn
14 AfterContentChecked, AfterViewInit, AfterViewChecked, OnDestroy {
15
16 nombre:string = "Rolando";
17
18 constructor() { }
19 ngOnChanges(changes: SimpleChanges): void {
20 console.log('ngOnChanges');
21 }
PROBLEMAS
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
node
Build at: 2021-12-13T04:23:35.856Z - Hash: 521d4fe07e850e38 - Time: 322ms
√ Compiled successfully.

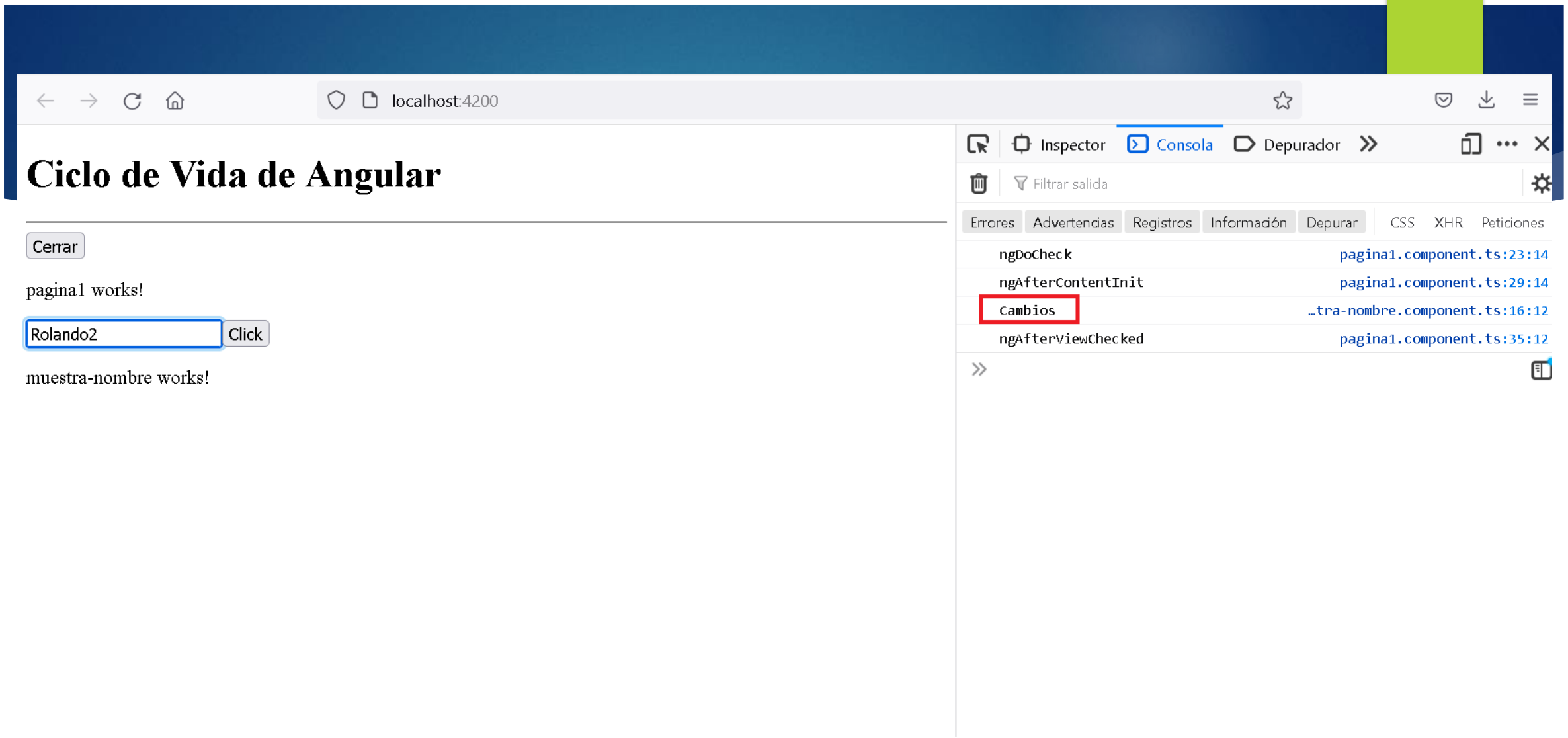

EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components
pages \ pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
index.html
omponent.ts M
pagina1.component.html U
pagina1.component.ts U
app
src > app > pages > pagina1 > pagina1.component.html > ...
Go to component
1 pagina1 works!
2 <input type="text"
3 placeholder="Hola Mundo"
4 [(ngModel)]="nombre">
5 <button (click)="guardar()">Click</button>
6


localhost:4200
Ciclo de Vida de Angular
Cerrar
pagina1 works!
Rolando!
Click
Inspector
Consola
Depurador
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
ngDoCheck
pagina1.component.ts:23:1
ngAfterContentInit
pagina1.component.ts:29:1
ngAfterViewChecked
pagina1.component.ts:35:1

Ejemplo de onChange




EXPLORADOR
EDITORES ABIERTOS
CICLODE-VIDA
src
app
components\muestra-n...
muestra-nombre.comp... U
muestra-nombre.comp... U
pages\pagina1
pagina1.component.html U
pagina1.component.ts U
app.component.css
app.component.html M
app.component.spec.ts
app.component.ts M
app.module.ts M
assets
environments
favicon.ico
index.html
main.ts
muestra-nombre.component.ts U
pagina1.component.html U
src > app > pages > pagina1 > pagina1.component.html > ...
Go to component
1 pagina1 works!
2 <input type="text"
3 placeholder="Hola Mundo"
4 [(ngModel)]="nombre">
5 <button (click)="guardar()">Click</button>
6
7 <app-muestra-nombre [nombre]="nombre"></app-muestra-nombre>
8
9


localhost:4200
Ciclo de Vida de Angular
Cerrar
pagina1 works!
Rolando2
Click
muestra-nombre works!
Inspector
Consola
Depurador
Filtrar salida
Errores
Advertencias
Registros
Información
Depurar
CSS
XHR
Peticiones
ngDoCheck
pagina1.component.ts:23:14
ngAfterContentInit
pagina1.component.ts:29:14
Cambios
…tra-nombre.component.ts:16:12
ngAfterViewChecked
pagina1.component.ts:35:12