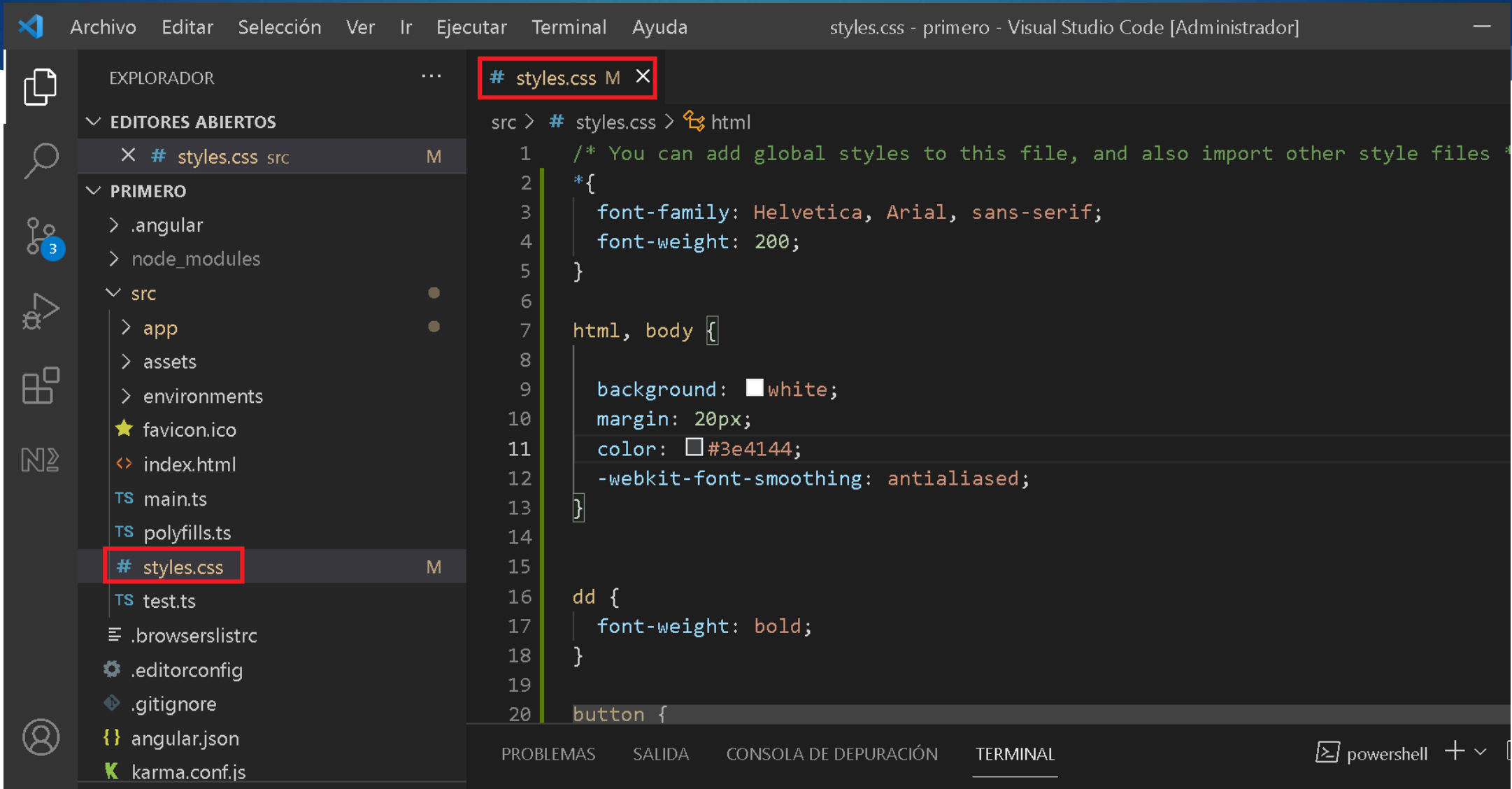


En src/styles.css aplicamos estilos globales





EXPLORADOR

EDITORES ABIERTOS

styles.css src M

PRIMERO

> .angular

> node_modules

src

> app

> assets

> environments

★ favicon.ico

<> index.html

TS main.ts

TS polyfills.ts

styles.css M

TS test.ts

≡ .browserslistrc

⚙ .editorconfig

📄 .gitignore

{} angular.json

🟢 karma.conf.js

> ESQUEMA

styles.css M X

src > # styles.css > html

```

19
20 button {
21     background-color: black;
22     border-radius: 5px;
23     border: 0px;
24     color: white;
25     cursor: pointer;
26     margin-right: 5px;
27     margin-left: 5px;
28     padding: 5px 10px;
29 }
30
31 button:hover {
32     background-color: #3e4144;
33 }
34
35 button:focus{
36     outline: none;
37 }
38
    
```

PROBLEMAS

SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL

📄 powershell + -



EXPLORADOR



styles.css M X



EDITORES ABIERTOS

X # styles.css src M



PRIMERO

> .angular

> node_modules



src

> app

> assets

> environments

★ favicon.ico

<> index.html

TS main.ts

TS polyfills.ts

styles.css M

TS test.ts

≡ .browserslistrc

⚙ .editorconfig

🔒 .gitignore

{ } angular.json

⚡ karma.conf.js



> ESQUEMA



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

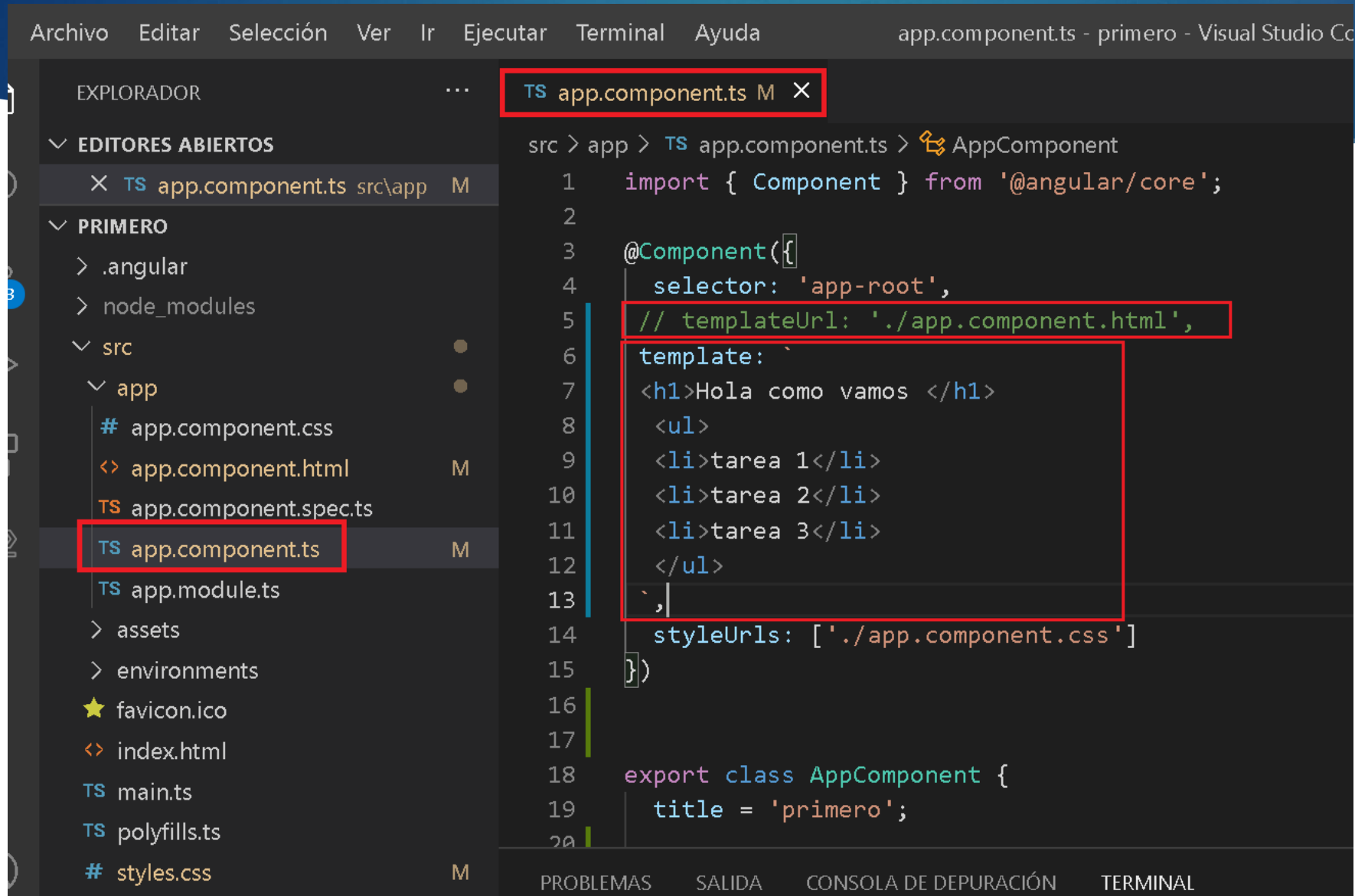
SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL

📄 powershell + ▢

En app.component.ts



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda app.component.ts - primero - Visual Studio Co

EXPLORADOR

EDITORES ABIERTOS

- TS app.component.ts src\app M

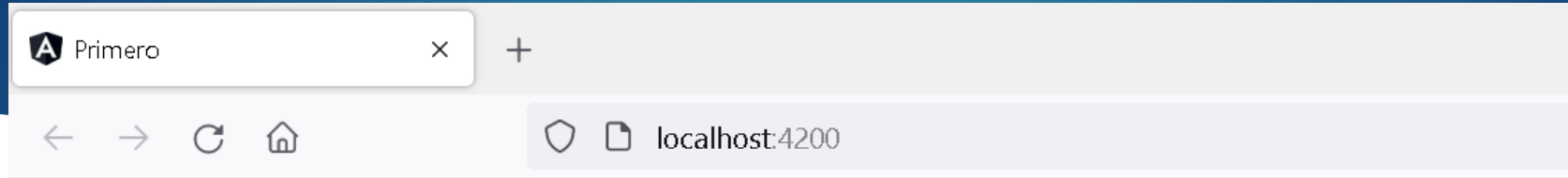
PRIMERO

- > .angular
- > node_modules
- > src
 - > app
 - # app.component.css
 - <> app.component.html M
 - TS app.component.spec.ts
 - TS app.component.ts M
 - TS app.module.ts
 - > assets
 - > environments
 - ★ favicon.ico
 - <> index.html
 - TS main.ts
 - TS polyfills.ts
 - # styles.css M

```
src > app > TS app.component.ts > AppComponent
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    // templateUrl: './app.component.html',
6    template: `
7    <h1>Hola como vamos </h1>
8    <ul>
9    <li>tarea 1</li>
10   <li>tarea 2</li>
11   <li>tarea 3</li>
12   </ul>
13   `,
14   styleUrls: ['./app.component.css']
15 })
16
17
18 export class AppComponent {
19   title = 'primero';
20 }
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Comprobamos que podemos tener definida el html dentro del ts



Hola como vamos

- tarea 1
- tarea 2
- tarea 3

Creo un ejemplo de contador

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

app.component.html - primero - Visual Studio Code [Administrador]

```
TS app.component.ts M x <> app.component.html M ...
src > app > TS app.component.ts > AppComponent > titulo
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
9
10 export class AppComponent {
11   titulo = 'Contador';
12
13
14 }
15
```

```
<> app.component.html M x
src > app > <> app.component.html > span
Go to component
1  <h1>
2    {{titulo}}
3
4  </h1>
5
6  <span> 0 </span>
7
```

PROBLEMAS

SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL



TS app.component.ts M X <> app.component.html M

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

```

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
9
10 export class AppComponent {
11   titulo:string = 'Contador';
12   numero:number = 10;
13
14 }
15

```

<> app.component.html M X

src > app > <> app.component.html > button

Go to component

```

1  <h1>
2    {{titulo}}
3
4  </h1>
5
6  <button> +1 </button>
7  <span>{{numero}} </span>
8  <button> -1 </button>
9

```

Primerο



localhost:4200

Contador

+1

10

-1

Agregamos un evento al botón

Visual Studio Code interface showing the implementation of an Angular component and its template.

Left Panel (app.component.ts):

```
src > app > TS app.component.ts > AppComponent > numero
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
9
10 export class AppComponent {
11   titulo:string = 'Contador';
12   numero:number = 10;
13
14 }
15
```

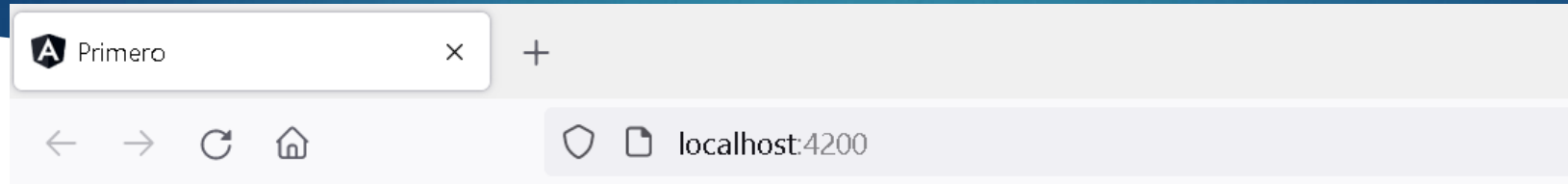
Right Panel (app.component.html):

```
src > app > app.component.html > ...
Go to component
1  <h1>
2    {{titulo}}
3  </h1>
4
5
6  <button (click)="numero = numero+1"> +1 </button>
7  <span>{{numero}} </span>
8  <button> -1 </button>
9
```

The button on line 6 of the template is highlighted with a red box, indicating the event being added.

Bottom Panel: PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL

Nos debería de aumentar en 1 el contador



Contador



Así quedaría agregando funcionalidad al botón -1



The image shows a Visual Studio Code editor window with the following elements:

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, and a dropdown menu (three dots) followed by app.componente.
- Tab Bar:** Two tabs are visible: `TS app.component.ts M` and `<> app.component.html M X`. The second tab is highlighted with a red rectangular box.
- Editor Area:** The active file is `app.component.html`. The breadcrumb navigation shows `src > app > <> app.component.html > ...`. The text `Go to component` is displayed above the code. The code content is as follows:

```
1 <h1>
2   {{titulo}}
3
4 </h1>
5
6 <button (click)="numero = numero+1"> +1 </button>
7 <span>{{numero}} </span>
8 <button (click)="numero = numero-1"> -1 </button>
9
```
- Left Sidebar:** Contains icons for Explorer, Search, Source Control (with a blue circle containing the number 3), Run and Debug, and Extensions.

Mejoramos nuestro código

Visual Studio Code interface showing two files: `app.component.ts` and `app.component.html`.

app.component.ts (Left Panel):

```
src > app > TS app.component.ts > AppComponent > sumar
3  @Component({
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
14   sumar(){
15     this.numero +=1;
16   }
17
18 }
19
```

app.component.html (Right Panel):

```
src > app > app.component.html > ...
Go to component
1  <h1>
2    {{titulo}}
3
4  </h1>
5
6  <button (click)="sumar()"> +1 </button>
7  <span>{{numero}} </span>
8  <button (click)="numero = numero-1"> -1 </button>
9
```

Visual Studio Code interface showing the bottom status bar with tabs: PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, and TERMINAL. The terminal tab is active, showing a prompt `>` and the text `node`.

TS app.component.ts M <> app.component.html M

src > app > TS app.component.ts > AppComponent > restar

```

3  @Component({
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
14   sumar(){
15     this.numero +=1;
16   }
17
18   restar(){
19     this.numero -=1;
20   }
21
22 }

```

<> app.component.html M

src > app > <> app.component.html > button

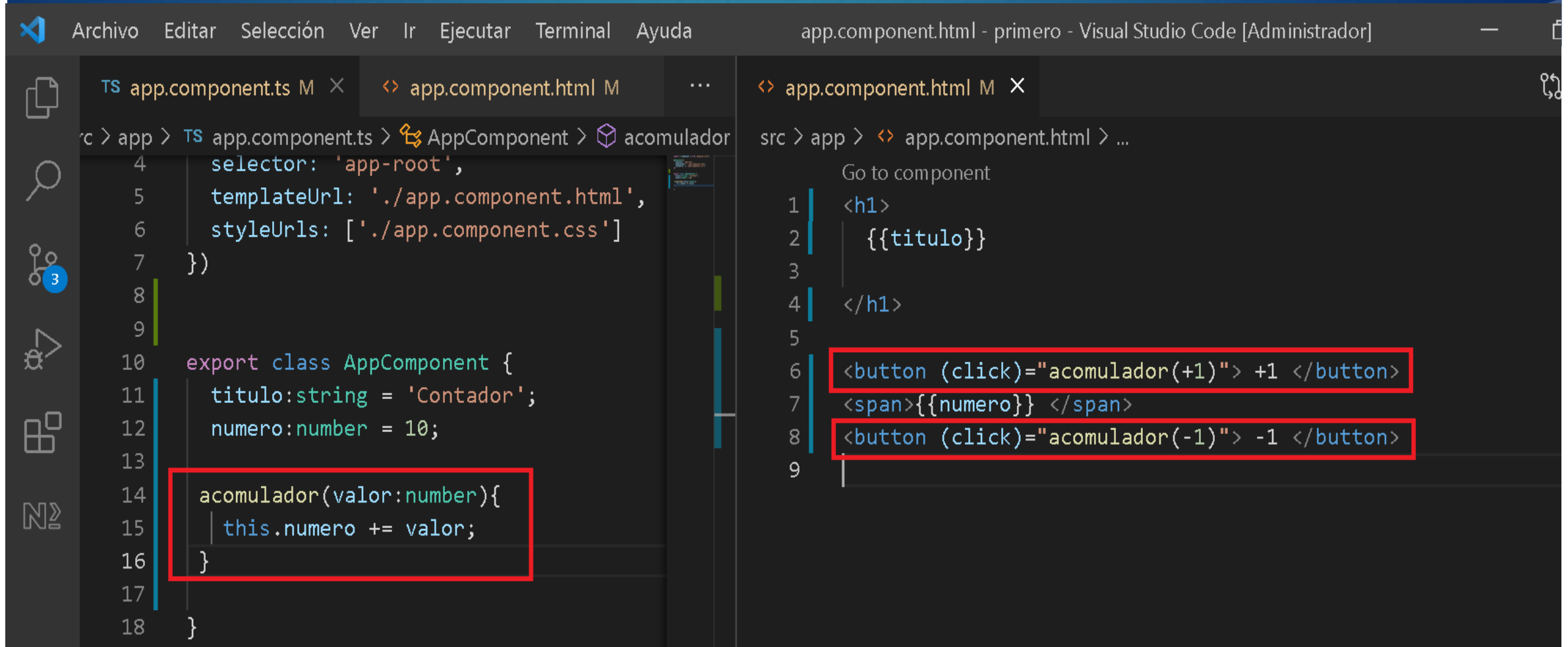
Go to component

```

1  <h1>
2    {{titulo}}
3
4  </h1>
5
6  <button (click)="sumar()"> +1 </button>
7  <span>{{numero}} </span>
8  <button (click)="restar()"> -1 </button>
9

```

Optimizamos aun mas



The image shows a screenshot of the Visual Studio Code editor with two files open: `app.component.ts` and `app.component.html`. The `app.component.ts` file on the left contains TypeScript code for an `AppComponent` class. It has a `selector` of `'app-root'`, a `templateUrl` of `'./app.component.html'`, and `styleUrls` of `['./app.component.css']`. The class has two properties: `titulo: string = 'Contador';` and `numero: number = 10;`. A method `acomulador(valor: number)` is defined, which increments `this.numero` by `valor`. The `app.component.html` file on the right shows the template. It starts with `<h1>` and `{{titulo}}`. There are two buttons: one with `(click)="acomulador(+1)"` and another with `(click)="acomulador(-1)"`. Both buttons are highlighted with red boxes. The `acomulador` method name in the HTML is misspelled as `acomulador` instead of `acumular`.

```
src > app > TS app.component.ts > AppComponent > acumulador
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
14   acumulador(valor:number){
15     this.numero += valor;
16   }
17
18 }
```

```
src > app > app.component.html > ...
Go to component
1 <h1>
2   {{titulo}}
3
4 </h1>
5
6 <button (click)="acomulador(+1)"> +1 </button>
7 <span>{{numero}} </span>
8 <button (click)="acomulador(-1)"> -1 </button>
9
```

Ejercicio

- Convertir el contador en un acumulador, teniendo en cuenta que desde ts se podría cambiar cuando se quiera el valor a acumular/decrementar por ejemplo tener poder acumular de 2 en 2 , de 5 en 5

TS app.component.ts M X <> app.component.html M ...

src > app > TS 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    acumulador(valor:number){
16      this.numero += valor;
17    }
18
19  }
20

```

<> app.component.html M X

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

```

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

```


Pasamos el Acumulador a un componente

The screenshot shows the Visual Studio Code interface with the following elements:

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, ...
- Tab Bar:** acumulador.component.ts - primero - Visual Studio Code
- EXPLORADOR (Left Panel):**
 - EDITORES ABIERTOS
 - × TS acumulador.component.t... U
 - PRIMERO
 - > .angular
 - > node_modules
 - > src
 - > app
 - TS acumulador.component.ts U** (highlighted with a red box)
 - # app.component.css
 - <> app.component.html M
 - TS app.component.spec.ts
 - TS app.component.ts M
 - TS app.module.ts
 - > assets
 - > environments
 - ★ favicon.ico
- Editor (Main Area):**
 - src > app > TS acumulador.component.ts > ...
 - Line 1: `import { Component } from "@angular/core";`
 - Line 2: (empty)
 - Line 3: (empty)
 - Line 4: `@Component({`
 - Line 5: `selector: 'app-acumulador',`
 - Line 6: `template: ``
 - Line 7: ```
 - Line 8: `})`
 - Line 9: `})`
 - Line 10: (empty)
 - Line 11: `export class ContadorComponent{`
 - Line 12: (empty)
 - Line 13: (empty)
 - Line 14: `}`
 - Line 15: (empty)
- Bottom Panel:** PROBLEMAS, SALIDA, TERMINAL, ...

TS acumulador.component.ts U X

<> app.component.html M



<> app.component.html M X

src > app > TS acumulador.component.ts > ...

```
1 import { Component } from "@angular/core";
2
3 |
4 @Component({
5   selector: 'app-acumulador',
6   template: `
7
8   `
9 })
10
11 export class ContadorComponent{
12
13
14 }
15
```

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
```



TS acumulador.component.ts 9, U X TS app.component.ts M



src > app > TS acumulador.component.ts > ContadorComponent



```
1  import { Component } from "@angular/core";
2
3
4  @Component({
5    selector: 'app-acumulador',
6    template: `
7
8    <h1>
9      {{titulo}}
10     Base de acumulador {{base}}
11   </h1>
12
13   <button (click)="acomulador(base)"> +{{base}} </button>
14   <span>{{numero}} </span>
15   <button (click)="acomulador(-base)"> -{{base}} </button>
16
17   `
18 })
19
20 export class ContadorComponent{
21
22
23
24 }
```

TS acumulador.component.ts 9, U ✕

TS app.component.ts M

src > app > TS acumulador.component.ts > ContadorComponent

```

1  import { Component } from "@angular/core";
2
3
4  @Component({
5    selector: 'app-acumulador',
6    template: `
7
8    <h1>
9      {{titulo}}
10     Base de acumulador {{base}}
11   </h1>
12
13   <button (click)="acomulador(base)"> +{{base}} </button>
14   <span>{{numero}} </span>
15   <button (click)="acomulador(-base)"> -{{base}} </button>
16
17   `
18 })
19
20 export class ContadorComponent{
21
22
23 }
24

```

TS app.component.ts M ✕

src > app > TS app.component.ts > AppComponent

```

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

```



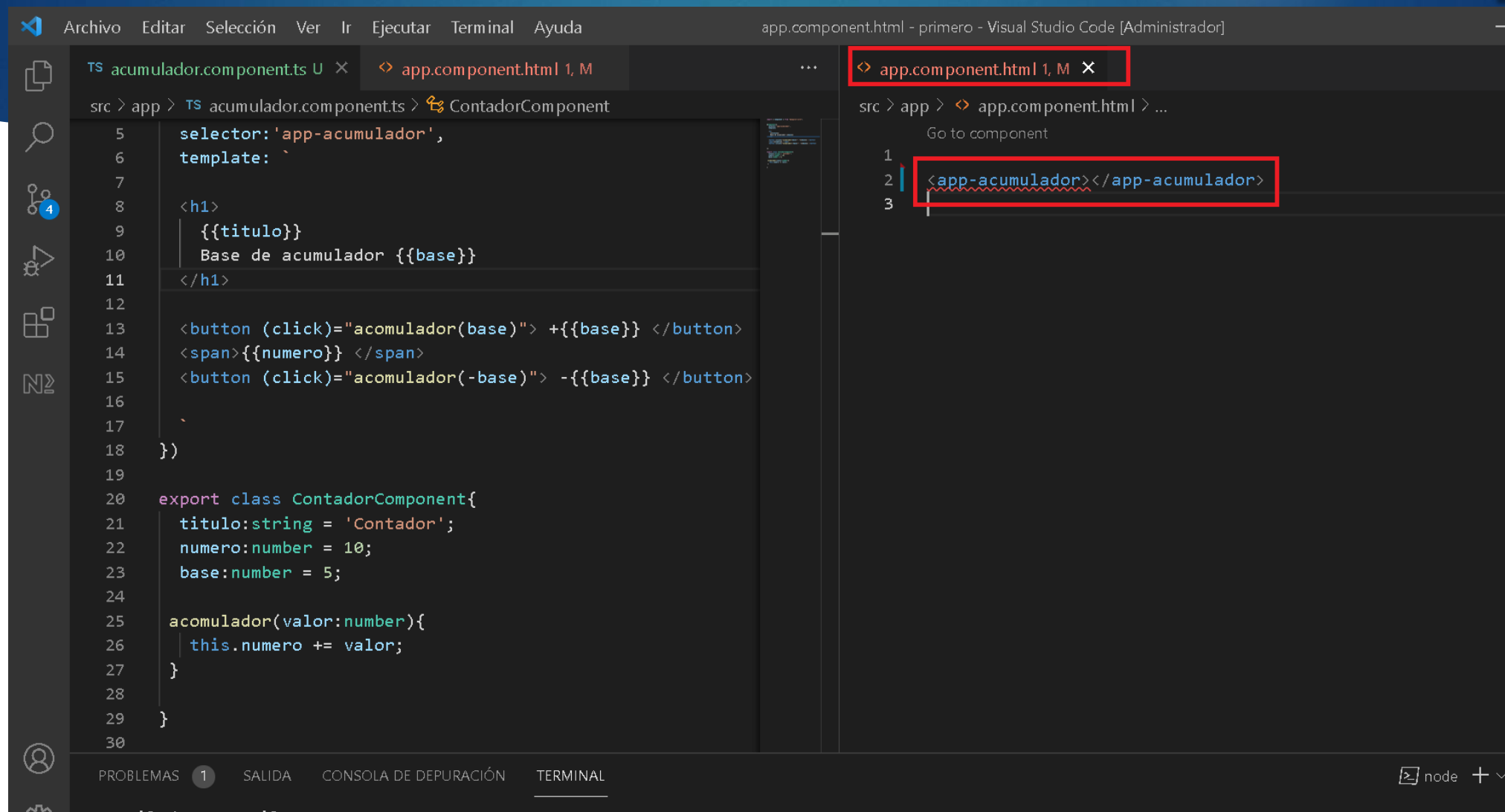
TS acumulador.component.ts U X TS app.component.ts M



src > app > TS acumulador.component.ts > ContadorComponent > acumulador

```
5     selector: 'app-acumulador',
6     template: `
7
8     <h1>
9         {{titulo}}
10        Base de acumulador {{base}}
11    </h1>
12
13    <button (click)="acumulador(base)"> +{{base}} </button>
14    <span>{{numero}} </span>
15    <button (click)="acumulador(-base)"> -{{base}} </button>
16
17    `
18  })
19
20  export class ContadorComponent{
21      titulo:string = 'Contador';
22      numero:number = 10;
23      base:number = 5;
24
25      acumulador(valor:number){
26          this.numero += valor;
27      }
28
29  }
30
```

Ahora agregamos este nuevo componente al principal



The image shows a Visual Studio Code editor with two files open. The left file is `TS acumulador.component.ts` and the right file is `app.component.html`. Both files are highlighted with red boxes.

Left File: `TS acumulador.component.ts`

```
src > app > TS acumulador.component.ts > ContadorComponent
5   selector: 'app-acumulador',
6   template: `
7
8   <h1>
9     {{titulo}}
10    Base de acumulador {{base}}
11  </h1>
12
13  <button (click)="acomulador(base)" +{{base}} </button>
14  <span>{{numero}} </span>
15  <button (click)="acomulador(-base)" -{{base}} </button>
16
17  `
18  })
19
20  export class ContadorComponent{
21    titulo:string = 'Contador';
22    numero:number = 10;
23    base:number = 5;
24
25    acomulador(valor:number){
26      this.numero += valor;
27    }
28
29  }
30
```

Right File: `app.component.html`

```
src > app > app.component.html > ...
Go to component
1
2  <app-acumulador></app-acumulador>
3
```

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 component 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.

× Failed to compile.

TS app.module.ts M X

TS acumulador.component.ts U



src > app > TS app.module.ts > AppModule

```

1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { ContadorComponent } from './acumulador.component';
4
5  import { AppComponent } from './app.component';
6
7  @NgModule({
8    declarations: [
9      AppComponent,
10     ContadorComponent
11   ],
12   imports: [
13     BrowserModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
19

```

TS acumulador.component.ts U X

src > app > TS acumulador.component.ts > ContadorComponent

```

6  template: `
7
8    <h1>
9      {{titulo}}
10     Base de acumulador {{base}}
11   </h1>
12
13   <button (click)="acomulador(base)"> +{{base}} </button>
14   <span>{{numero}} </span>
15   <button (click)="acomulador(-base)"> -{{base}} </button>
16
17 `
18 })
19
20 export class ContadorComponent{
21   titulo:string = 'Contador';
22   numero:number = 10;
23   base:number = 5;
24
25   acomulador(valor:number){
26     this.numero += valor;
27   }
28
29 }
30

```

PROBLEMAS 1

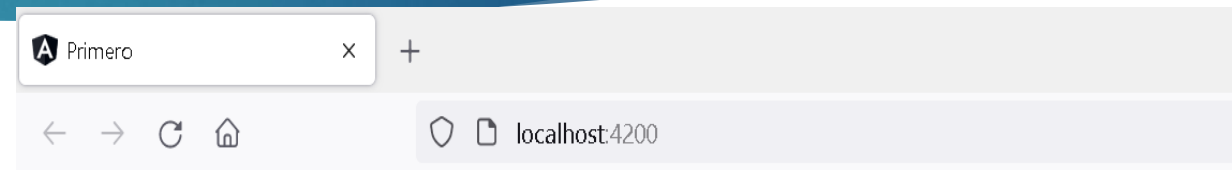
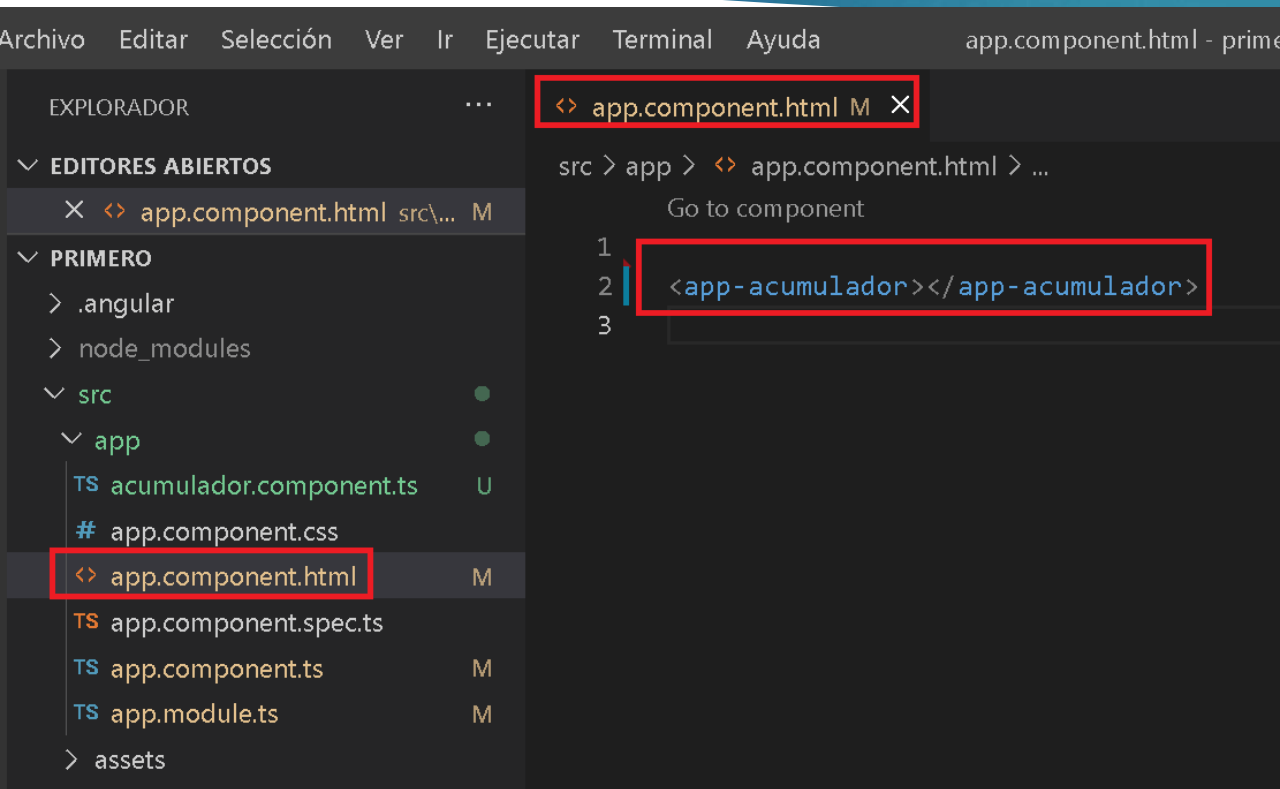
SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL

node + - [] [X] ^ X

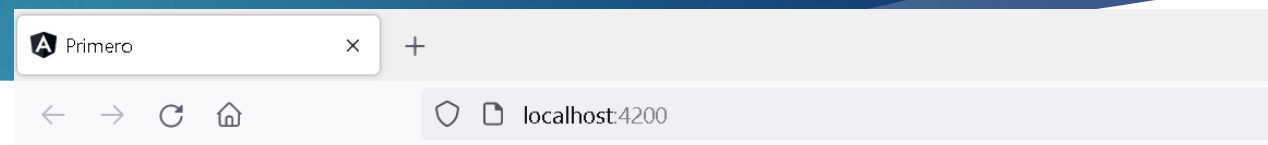
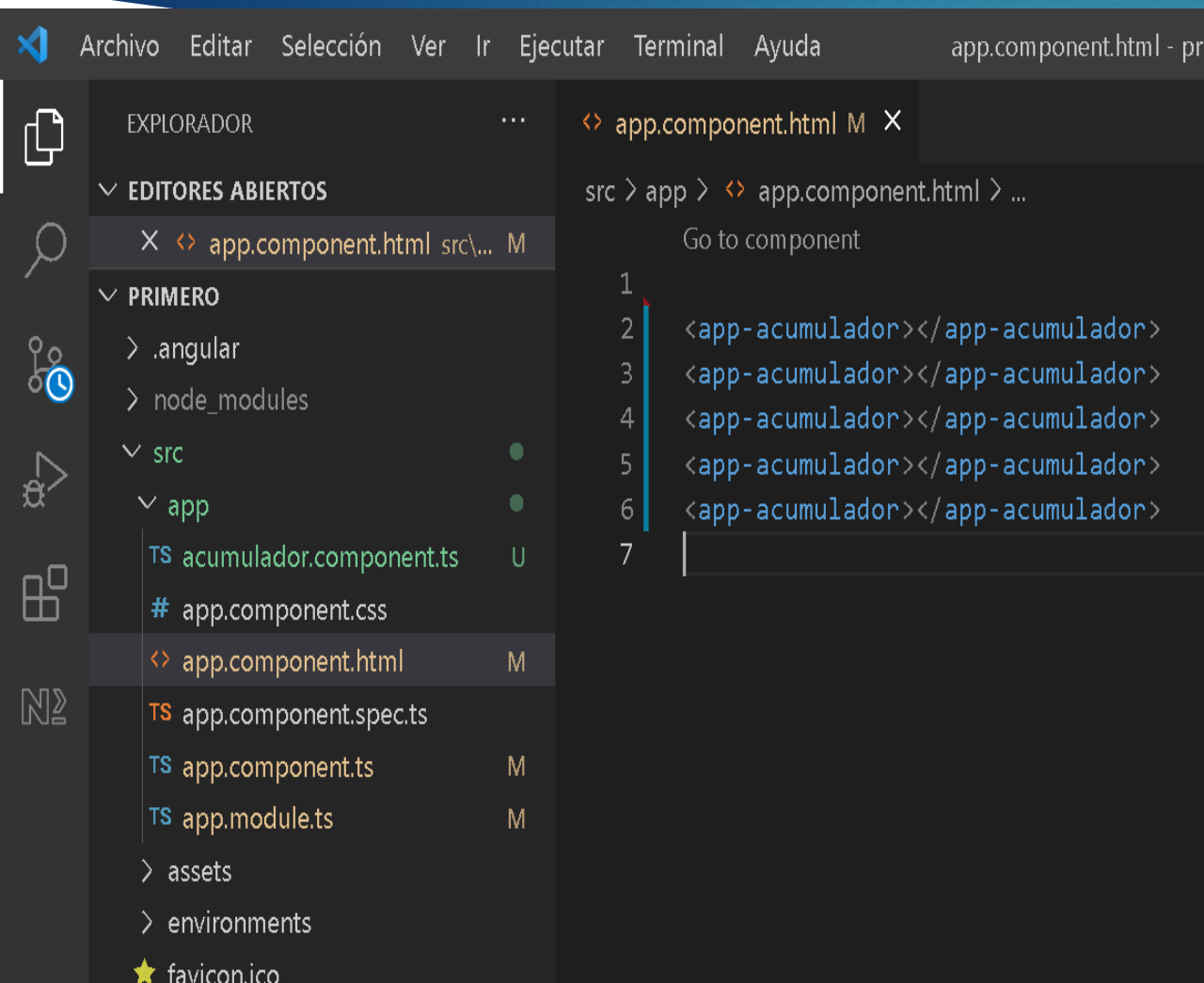
Con esto ya nos funciona el llamado al componente



Contador Base de acumulador 5

+5 5 -5

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



Contador Base de acumulador 5

+5 -10 -5

Contador Base de acumulador 5

+5 30 -5

Contador Base de acumulador 5

+5 55 -5

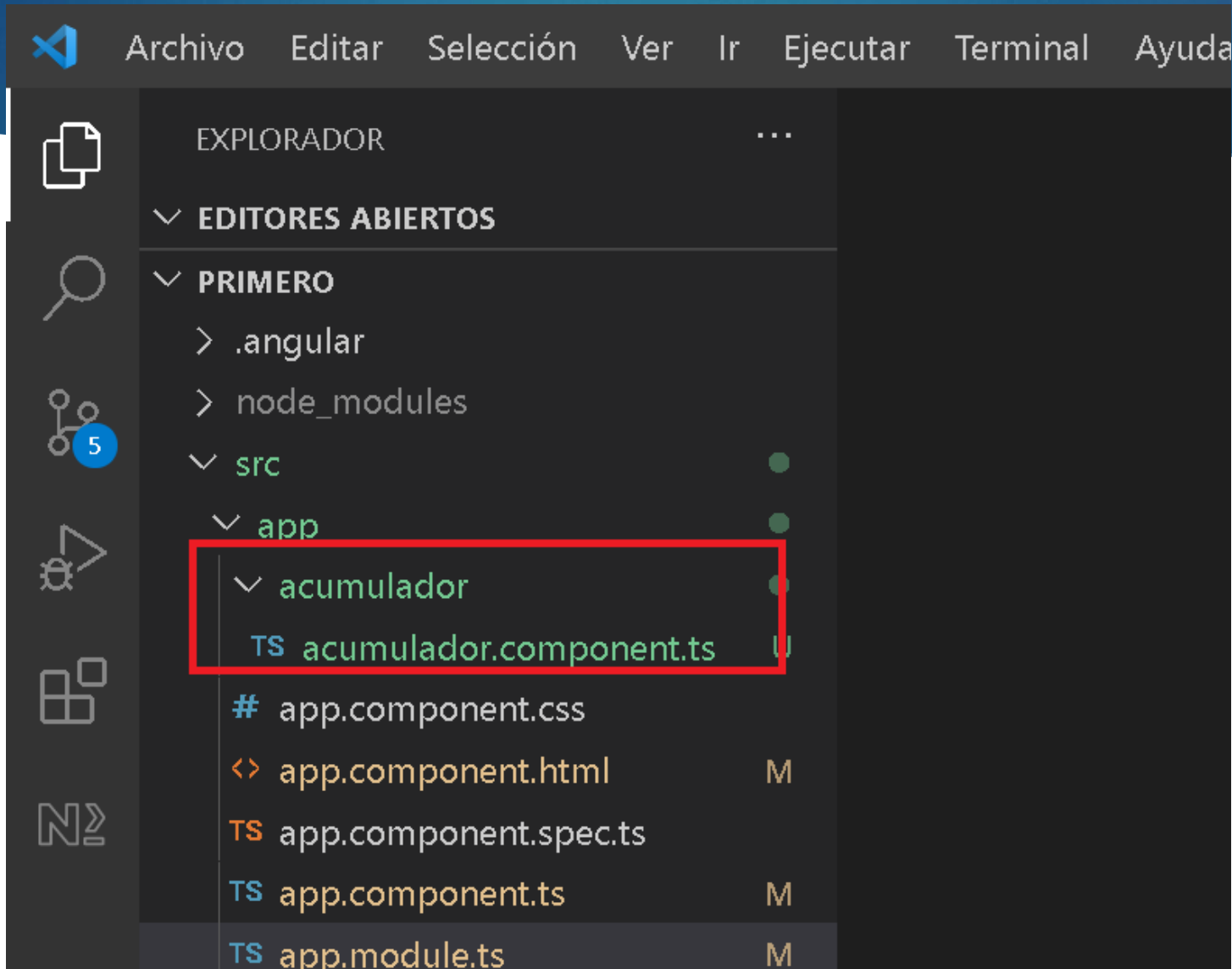
Contador Base de acumulador 5

+5 -30 -5

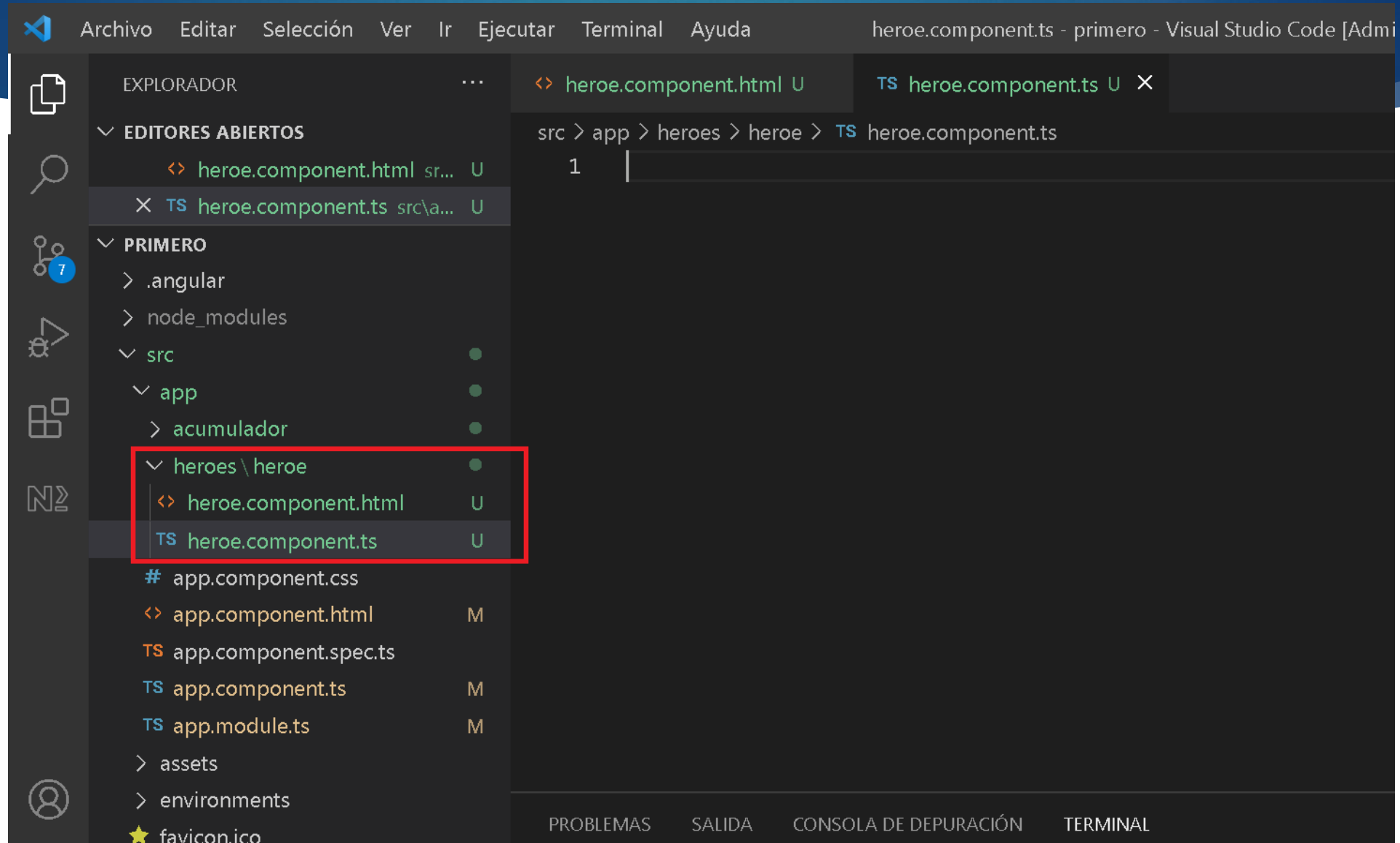
Contador Base de acumulador 5

+5 65 -5

Podemos crear directorios para acomodar mejor los componentes



Ahora creamos un nuevo componente



EXPLORADOR

EDITORES ABIERTOS

<> hero.component.html sr... U

× TS hero.component.ts src\a... U

PRIMERO

> .angular

> node_modules

src

app

> acumulador

heroes \ heroe

<> hero.component.html U

TS hero.component.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts M

<> hero.component.html U

TS hero.component.ts U ×

src > app > heroes > heroe > TS hero.component.ts > HeroComponent

1 import { Component } from "@angular/core";

2

3

4 @Component({

5 selector: "app-heroe",

6 templateUrl: "hero.component.html"

7 })

8

9 export class HeroComponent{

10

11 }

12

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

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

app.module.ts - primero - Visual Studio Code [Administrador]

EXPLORADOR

✓ EDITORES ABIERTOS

<> hero.component.html src\... U

TS hero.component.ts src\... U

✕ TS app.module.ts src\app M

✓ PRIMERO

> .angular

> node_modules

✓ src

✓ app

> acumulador

✓ heroes \ heroe

<> hero.component.html U

TS hero.component.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts M

TS app.module.ts M

src > app > TS app.module.ts > AppModule

```
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { ContadorComponent } from '../acumulador/acumulador.component';
4
5  import { AppComponent } from '../app.component';
6  import { HeroComponent } from '../heroes/heroe/heroe.component';
7
8  @NgModule({
9    declarations: [
10     AppComponent,
11     ContadorComponent,
12     HeroComponent
13   ],
14   imports: [
15     BrowserModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
```



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

heroe.component.html - primer



EXPLORADOR



EDITORES ABIERTOS

× <> heroe.component.html sr... U

TS heroe.component.ts src\a... U

TS app.module.ts src\app M

PRIMERO

> .angular

> node_modules

src

app

acumulador

heroes \ heroe

<> heroe.component.html U

TS heroe.component.ts U

app.component.css

<> app.component.html M

<> heroe.component.html U X

TS heroe.component.ts U

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

Go to component

1 <h1>Componente de Heroe</h1>

2

EXPLORADOR

EDITORES ABIERTOS

X <> app.component.html src\... M

PRIMERO

src

app

> acumulador

heroes \ heroe

<> heroe.component.html U

TS heroe.component.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts M

TS app.module.ts M

> assets

> environments

★ favicon.ico

<> index.html

<> app.component.html M X

src > app > <> app.component.html > app-heroe

Go to component

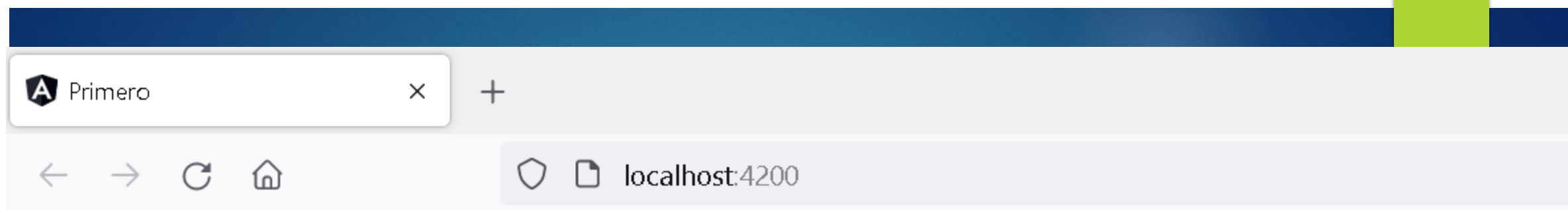
1 <!--<app-acumulador></app-acumulador>-->

2

3 <app-heroe></app-heroe>

4

5



Componente de Heroe

Agregamos lo siguiente

Visual Studio Code interface showing the development of a component.

File Explorer (Left):

- src > app > heroes > hero > TS **hero.component.ts** U X

Code Editor (Left):

```
3
4 @Component({
5   selector: "app-heroe",
6   templateUrl: "hero.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 }
```

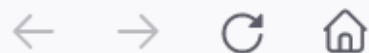
File Explorer (Right):

- src > app > heroes > hero > <> **hero.component.html** U X

Code Editor (Right):

```
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
```

Primerο



localhost:4200

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

hero.component.html - primero - Visual Studio Code [Administrador]

TS hero.component.ts U X

src > app > heroes > hero > TS hero.component.ts > HeroComponent > cambiarNombre

```
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 }
```

<> hero.component.html U X

src > app > heroes > hero > <> hero.component.html > dl > button

```
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
```

Componente de Heroe

Nombre:

Spiderman

Edad:

30

Funcion:

Spiderman - 30

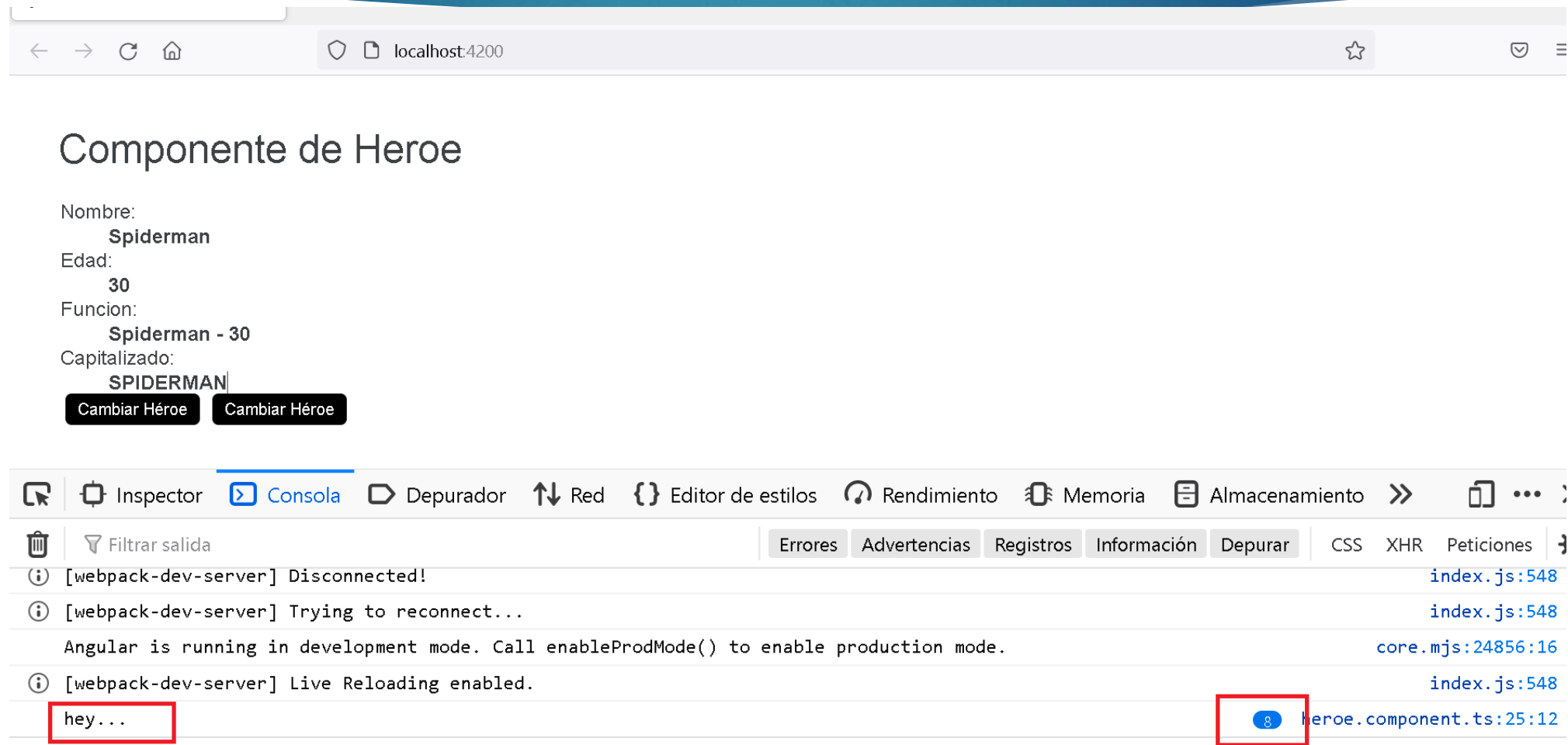
Capitalizado:

SPIDERMAN

Cambiar Héroe

Cambiar Héroe

One Wey Data Binding – Enlazado en una sola vía



The screenshot shows a web browser at `localhost:4200` displaying a form titled "Componente de Heroe". The form contains the following fields and values:

- Nombre: **Spiderman**
- Edad: **30**
- Funcion: **Spiderman - 30**
- Capitalizado: **SPIDERMAN**

Below the form are two buttons, both labeled "Cambiar Héroe".

The browser's developer console is open, showing the following messages:

- [webpack-dev-server] Disconnected! `index.js:548`
- [webpack-dev-server] Trying to reconnect... `index.js:548`
- Angular is running in development mode. Call `enableProdMode()` to enable production mode. `core.mjs:24856:16`
- [webpack-dev-server] Live Reloading enabled. `index.js:548`
- hey... `heroe.component.ts:25:12`

The last log entry, "hey...", is highlighted with a red box, and the corresponding source code location, `heroe.component.ts:25:12`, is also highlighted with a red box.

También podemos generar componentes por comando

PROBLEMAS

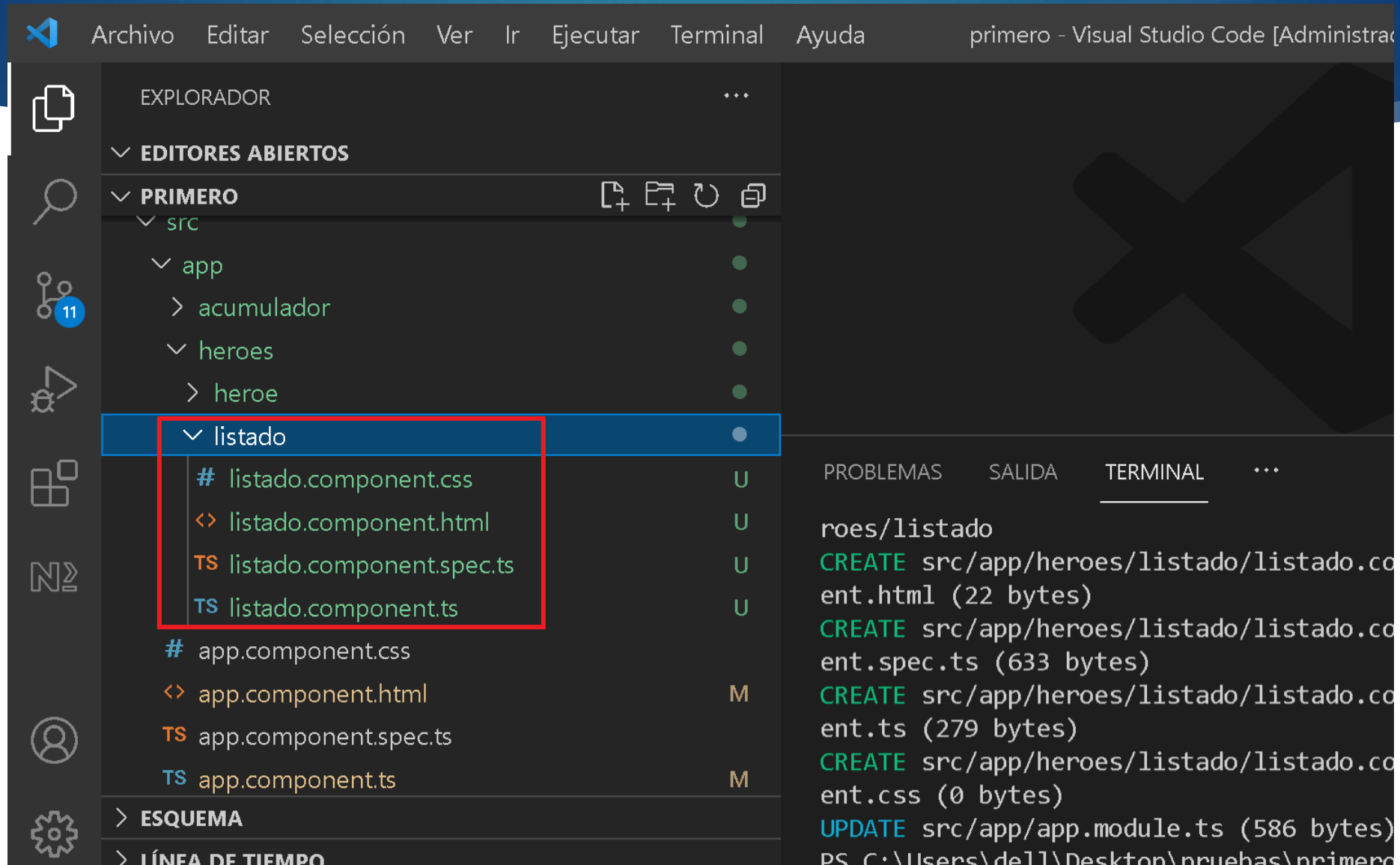
SALIDA

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



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is titled 'EXPLORADOR' and shows a folder structure. The 'PRIMERO' folder is expanded, showing a 'src' folder, which is also expanded to show 'app', 'acumulador', 'heroes', and 'listado'. The 'listado' folder is selected, and a list of files is shown below it:

- # listado.component.css
- <> listado.component.html
- TS listado.component.spec.ts
- TS listado.component.ts

The 'listado' folder is highlighted with a blue bar, and the list of files is enclosed in a red box. The main editor area on the right shows a large 'X' icon, indicating that no file is currently open. The bottom status bar shows the file path: 'PS C:\Users\de11\Desktop\pruebas\primero'.

También agrega la función al app.module.ts

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

app.module.ts - primero - Visual Studio

app.module.ts M X

src > app > TS app.module.ts > ...

```
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { ContadorComponent } from '../acumulador/acumulador.component';
4
5  import { AppComponent } from '../app.component';
6  import { HeroeComponent } from '../heroes/heroe/heroe.component';
7  import { ListadoComponent } from '../heroes/listado/listado.component';
8
9  @NgModule({
10   declarations: [
11     AppComponent,
12     ContadorComponent,
13     HeroeComponent,
14     ListadoComponent
15   ],
16   imports: [
17     BrowserModule
18   ],
19   providers: [],
20   bootstrap: [AppComponent]
```



EXPLORADOR



EDITORES ABIERTOS

<> listado.component.html src\app\heroes\lista... U

X <> app.component.html src\app M

PRIMERO

heroes

hero

listado

listado.component.css U

<> listado.component.html U

TS listado.component.spec.ts U

TS listado.component.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts M

TS app.module.ts M

<> listado.component.html U

<> app.component.html M X

src > app > <> app.component.html > app-listado

Go to component

1 <!--<app-acumulador></app-acumulador>-->

2

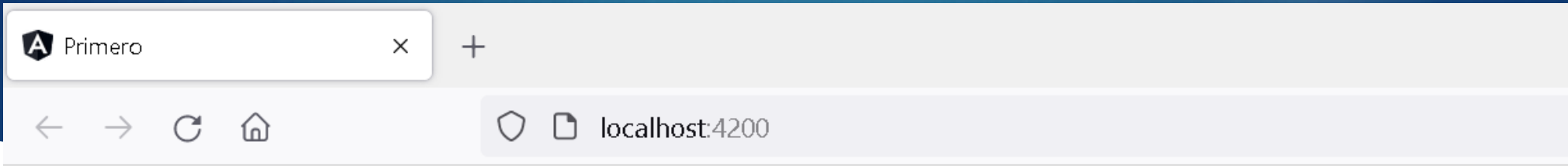
3 <!--<app-heroe></app-heroe>-->

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
 TS listado.component.spec.ts U
 TS listado.component.ts U
 # app.component.css
 <> app.component.html M
 TS app.component.spec.ts
 TS app.component.ts M
 TS app.module.ts M

<> listado.component.html U ✕

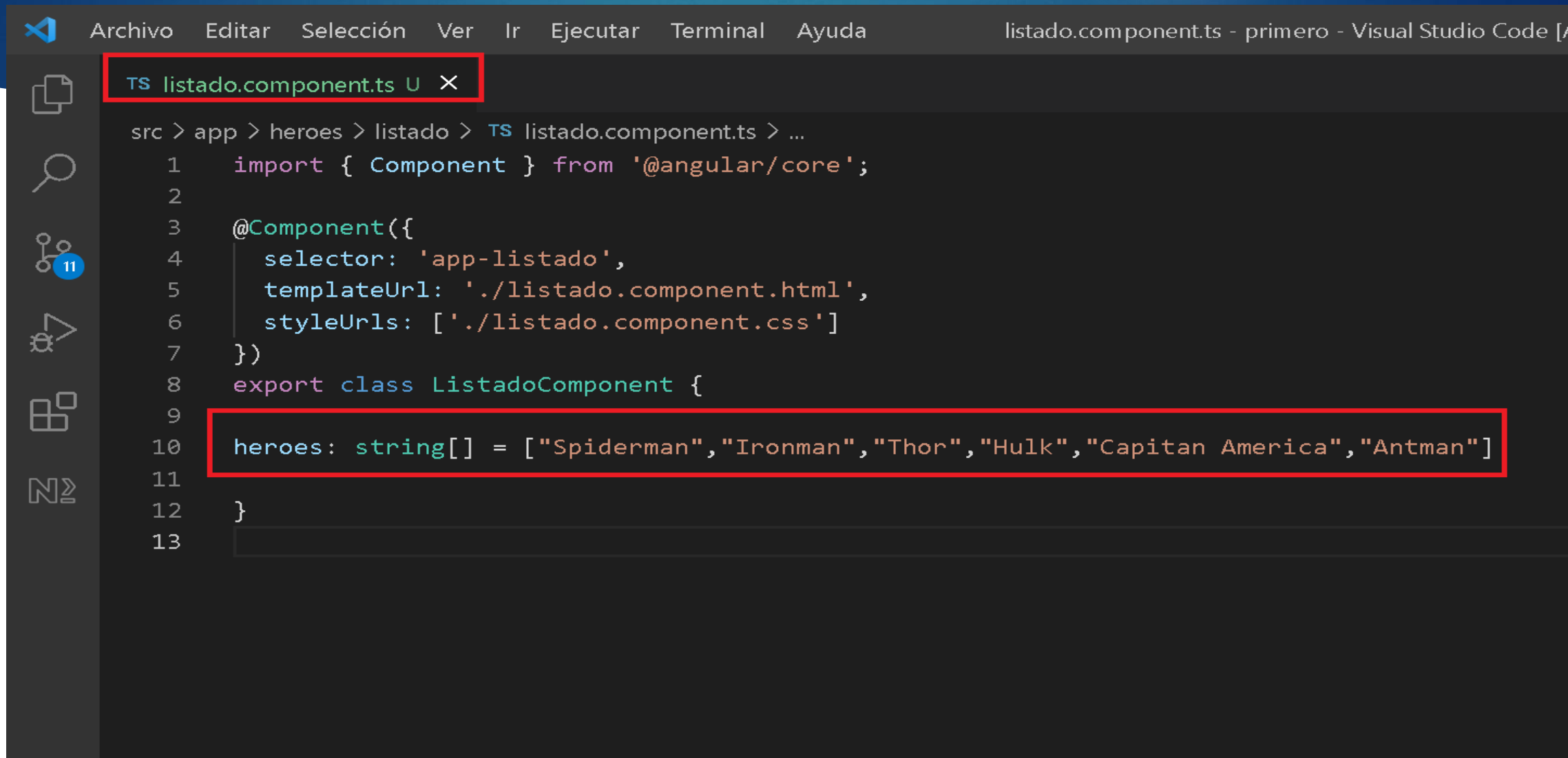
<> app.component.html M

src > app > heroes > listado > <> listado.component.html > ...

Go to component

```
1 <p>listado works!</p>  
2 |
```

Directiva *ngFor



```
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  }
13
```



Archivo

Editar

Selección

Ver

Ir

Ejecutar



listado.componente



TS listado.component.ts U

<> listado.component.html U X



src > app > heroes > listado > <> listado.component.html > ...

Go to component

```
1  <p>listado de Héroes</p>
2
3  <ul>
4    <li *ngFor="let hero of heroes; let i = index">
5      | | {{i+1}} - {{hero}}
6    </li>
7  </ul>
8
```

listado de Héroes

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

Ejercicio

- Agregamos un botón en `listado.component.html` y una función en `listado.component.ts` que borre un elemento de la lista, entonces al pulsar el botón debería borrar un elemento del array `heroes`

TS listado.component.ts U X

<> 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.c



TS listado.component.ts U

<> listado.component.html U X



src > app > heroes > listado > <> listado.component.html > ...

Go to component

1 <p>listado de Héroes</p>

2

3

4 | <li *ngFor="let hero of heroes; let i = index">

5 | | {{i+1}} - {{hero}}

6 |

7

8

9 <button (click)="borrarHeroe()">Borrar Héroe</button>

10



Ejercicio

- ▶ Agregar una variable que guarde y muestre el héroe borrado.

TS listado.component.ts U X

<> listado.component.html U

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

```
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   heroeBorrado: string | undefined = "";
12
13   borrarHeroe(){
14     this.heroeBorrado = this.heroes.pop();
15   }
16 }
17
```



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

listado.component.html - prime



TS listado.component.ts U

<> listado.component.html U X



src > app > heroes > listado > <> listado.component.html > ...

Go to component

```
1  <p>listado de Héroes</p>
2
3  <ul>
4    <li *ngFor="let hero of heroes; let i = index">
5      | | {{i+1}} - {{hero}}
6    </li>
7  </ul>
8
9  <p>Heróe Borrado {{heroBorrado}}</p>
10
11
12  <button (click)="borrarHeroe()">Borrar Héroe</button>
13
```