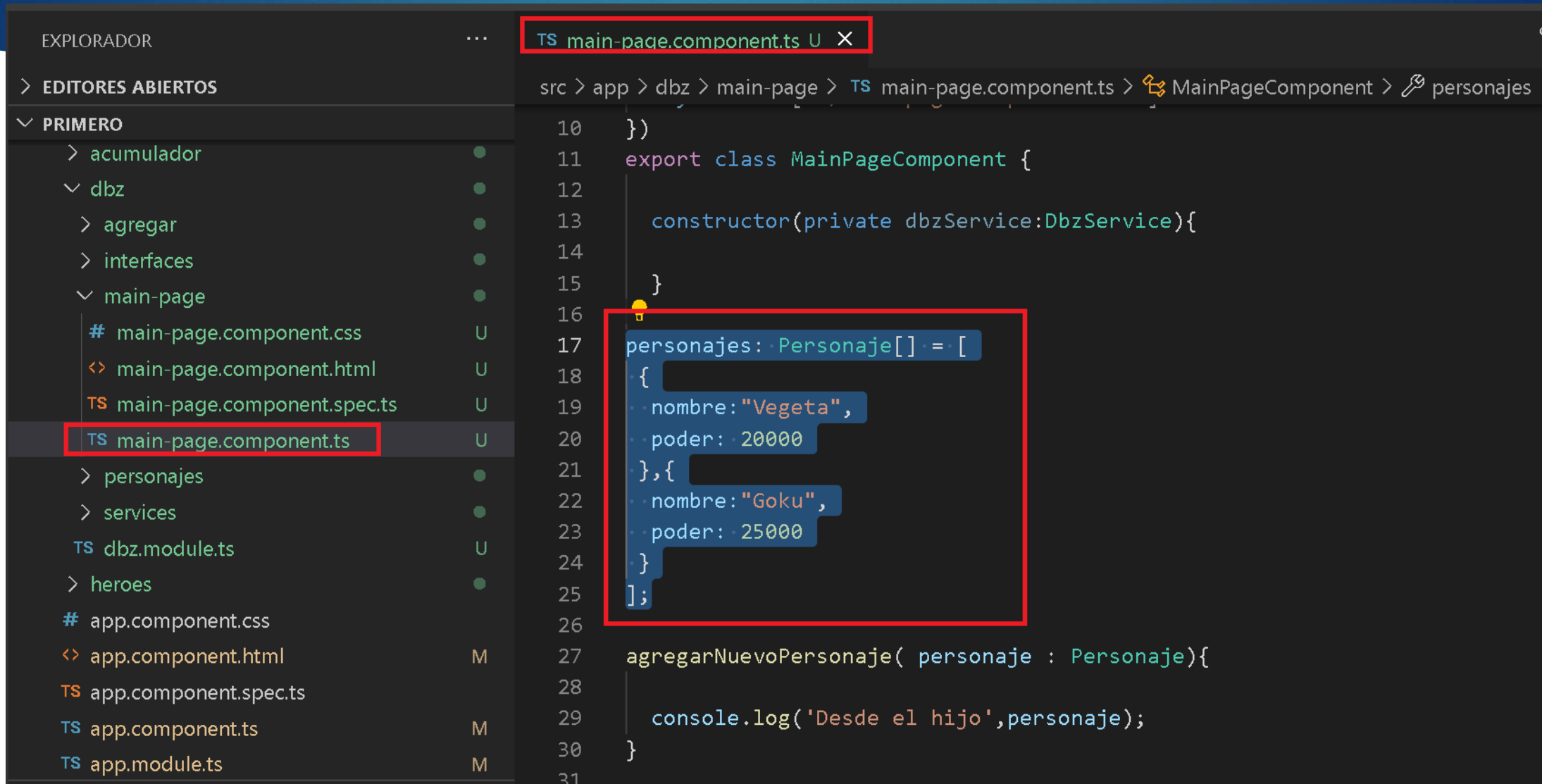


Centralizamos los datos en los servicios, pasamos nuestro array de personajes a nuestro servicio



```
EXPLORADOR
... TS main-page.component.ts U X

> EDITORES ABIERTOS
src > app > dbz > main-page > TS main-page.component.ts > MainPageComponent > personajes

v PRIMERO
  > acumulador
  v dbz
    > agregar
    > interfaces
    v main-page
      # main-page.component.css U
      <> main-page.component.html U
      TS main-page.component.spec.ts U
      TS main-page.component.ts U
      > personajes
      > services
    TS dbz.module.ts U
  > heroes
  # app.component.css
  <> app.component.html M
  TS app.component.spec.ts
  TS app.component.ts M
  TS app.module.ts M

10  })
11  export class MainPageComponent {
12
13    constructor(private dbzService: DbzService){
14
15    }
16
17    personajes: Personaje[] = [
18      {
19        nombre: "Vegeta",
20        poder: 20000
21      }, {
22        nombre: "Goku",
23        poder: 25000
24      }
25    ];
26
27    agregarNuevoPersonaje( personaje : Personaje){
28
29      console.log('Desde el hijo', personaje);
30    }
31  }
```

Lo pasamos al servicio creado

The screenshot shows an IDE with two panels. The left panel, titled 'EXPLORADOR', displays a file tree under 'EDITORES ABIERTOS'. The tree structure is as follows:

- PRIMERO
 - acumulador
 - dbz
 - agregar
 - interfaces
 - main-page
 - main-page.component.css (U)
 - main-page.component.html (U)
 - main-page.component.spec.ts (U)
 - main-page.component.ts (U)
 - personajes
 - services
 - dbz.service.ts (U)** (highlighted with a red box)
 - dbz.module.ts (U)
 - heroes
 - app.component.css
 - app.component.html (M)
 - app.component.spec.ts
 - app.component.ts (M)

The right panel shows the code for 'TS dbz.service.ts'. The breadcrumb path is 'src > app > dbz > services > TS dbz.service.ts > DbzService > personajes'. The code is as follows:

```
1 import { Injectable } from "@angular/core";
2 import { Personaje } from "../interfaces/dbz.interfaces";
3
4
5 @Injectable()
6 export class DbzService{
7
8
9     personajes: Personaje[] = [
10         {
11             nombre: "Vegeta",
12             poder: 20000
13         }, {
14             nombre: "Goku",
15             poder: 25000
16         }
17     ];
18     constructor(){
19         console.log('Servicio Iniciado');
20     }
21 }
22
```

A red box highlights the `personajes` array definition in the service class.

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



The screenshot shows the Visual Studio Code editor interface. On the left is the Explorer sidebar showing the project structure: `src > app > dbz > main-page`. The main editor area displays the `main-page.component.html` file. The breadcrumb navigation at the top indicates the path: `src > app > dbz > main-page > main-page.component.html > div.row > div.col > app-personajes`. The code in the editor is as follows:

```
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
```

Two instances of the attribute `[personajes]="personajes"` are highlighted with red rectangular boxes: one on line 9 within the `<app-personajes>` tag, and another on line 16 within the `<app-agregar>` tag.

Quedaría así nuestro main-page.component.html

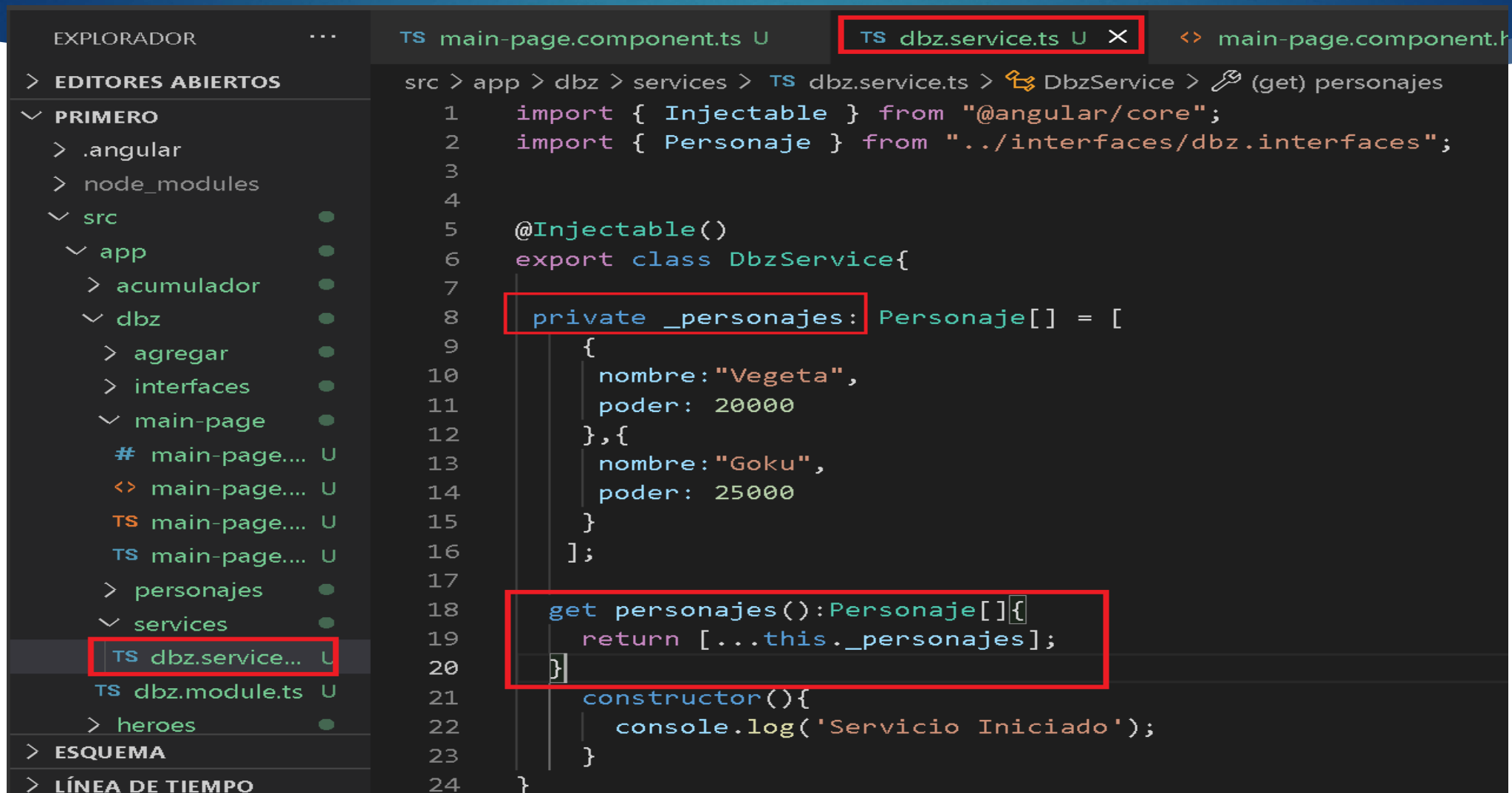
```
EXPLORADOR  ...  TS main-page.component.ts U  TS dbz.service.ts U  <> main-page.component.html U X

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

PRIMERO
  > .angular
  > node_modules
  > src
    > app
      > acumulador
      > dbz
        > agregar
        > interfaces
        > main-page
          # main-page... U
          <> main-page... U
          TS main-page... U
          TS main-page... U
        > personajes
      > services
        TS dbz.service... U
        TS dbz.module.ts U

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

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

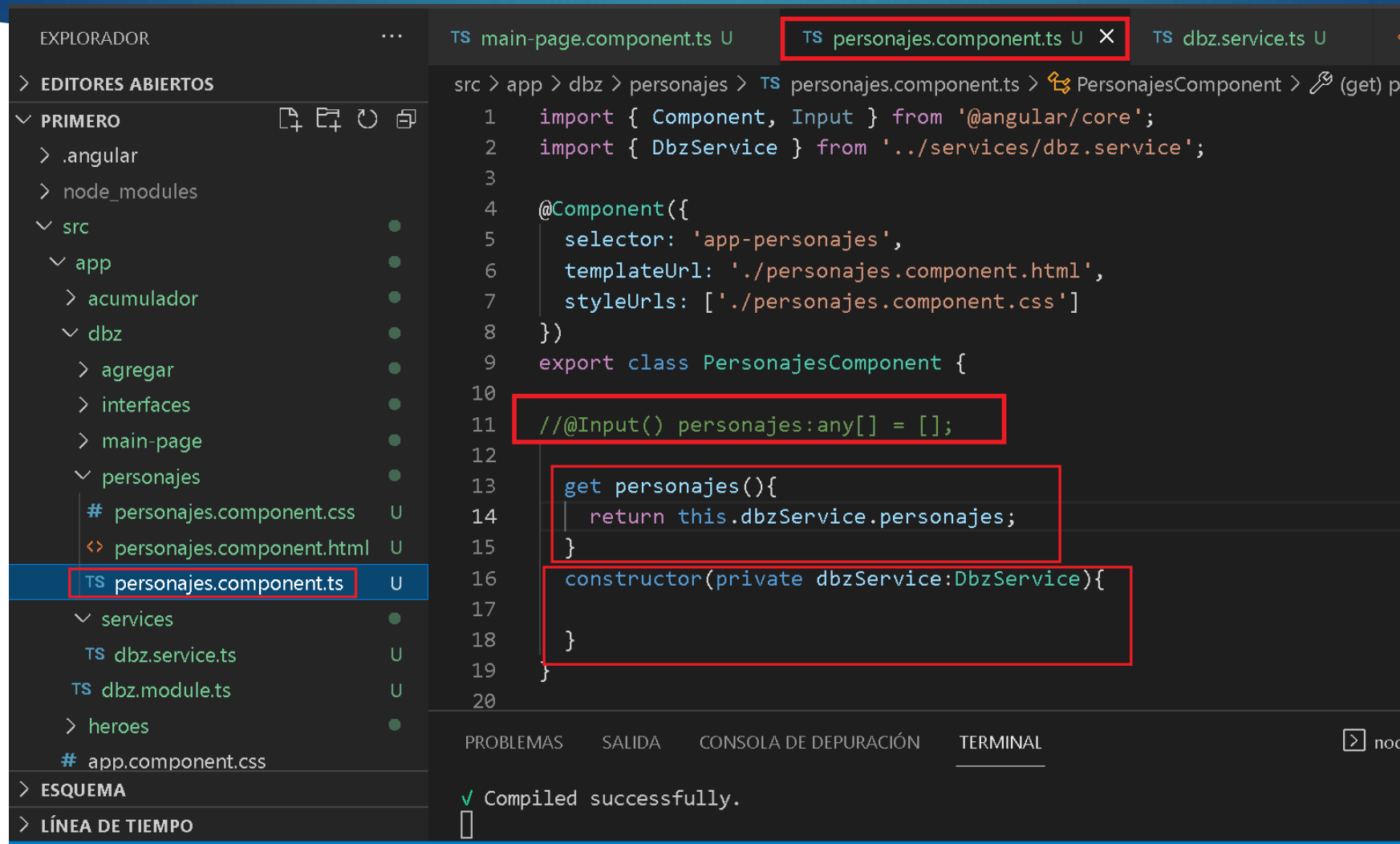


The screenshot shows the Visual Studio Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the project structure with 'src' expanded, and 'services' expanded, highlighting 'dbz.service.ts'. The Editor shows the code for 'dbz.service.ts' with the following content:

```
src > app > dbz > services > TS dbz.service.ts > DbzService > (get) personajes
1  import { Injectable } from "@angular/core";
2  import { Personaje } from "../interfaces/dbz.interfaces";
3
4
5  @Injectable()
6  export class DbzService{
7
8      private _personajes: Personaje[] = [
9          {
10             nombre: "Vegeta",
11             poder: 20000
12          }, {
13             nombre: "Goku",
14             poder: 25000
15          }
16      ];
17
18      get personajes(): Personaje[] {
19          return [...this._personajes];
20      }
21
22      constructor(){
23          console.log('Servicio Iniciado');
24      }
25  }
```

The code is written in TypeScript. It imports `Injectable` from `@angular/core` and `Personaje` from `../interfaces/dbz.interfaces`. The `DbzService` class is decorated with `@Injectable()`. It has a private property `_personajes` of type `Personaje[]` initialized with an array of two objects: `{ nombre: "Vegeta", poder: 20000 }` and `{ nombre: "Goku", poder: 25000 }`. A `get` method named `personajes` is defined, which returns a new array created by spreading the elements of `this._personajes` (`[...this._personajes]`). The `constructor` method logs `'Servicio Iniciado'` to the console.

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



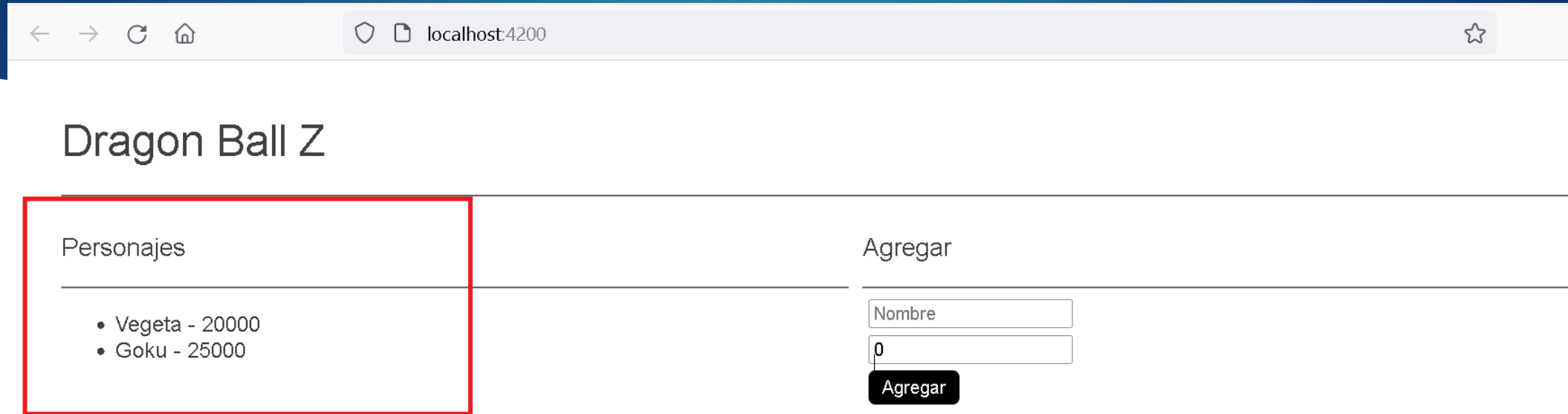
```
EXPLORADOR
...
EDITORES ABIERTOS
PRIMERO
  .angular
  node_modules
  src
    app
      acumulador
      dbz
        agregar
        interfaces
        main-page
        personajes
          # personajes.component.css
          <> personajes.component.html
          TS personajes.component.ts
        services
          TS dbz.service.ts
          TS dbz.module.ts
        heroes
        # app.component.css
  ESQUEMA
  LÍNEA DE TIEMPO

TS main-page.component.ts U
TS personajes.component.ts U X
TS dbz.service.ts U

src > app > dbz > personajes > TS personajes.component.ts > PersonajesComponent > (get) p
1  import { Component, Input } from '@angular/core';
2  import { DbzService } from '../services/dbz.service';
3
4  @Component({
5    selector: 'app-personajes',
6    templateUrl: './personajes.component.html',
7    styleUrls: ['./personajes.component.css']
8  })
9  export class PersonajesComponent {
10
11    // @Input() personajes: any[] = [];
12
13    get personajes() {
14      return this.dbzService.personajes;
15    }
16
17    constructor(private dbzService: DbzService) {
18
19    }
20
21  }

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL
✓ Compiled successfully.
```

Deberíamos de poder ver el listado desde el servicio



← → ↻ 🏠 localhost:4200 ☆

Dragon Ball Z

Personajes

- Vegeta - 20000
- Goku - 25000

Agregar

Nombre

0

Agregar

Ahora agregamos al servicio la funcionalidad de crear personaje

```
EXPLORADOR
...
TS dbz.service.ts U X

src > app > dbz > services > TS dbz.service.ts > ...

PRIMERO
> .angular
> node_modules
src
  app
    acumulador
    dbz
      agregar
      interfaces
      main-page
      personajes
        # personajes.component.css U
        <> personajes.component.html U
        TS personajes.component.ts U
      services
        TS dbz.service.ts U
        TS dbz.module.ts U

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

nombre: "Vegeta",
poder: 20000
},{
nombre: "Goku",
poder: 25000
}
];

get personajes(): Personaje[] {
  return [...this._personajes];
}

constructor() {
  console.log('Servicio Iniciado');
}

agregarPersonaje( personaje: Personaje ) {
  this._personajes.push(personaje);
}
```


EXPLORADOR

> EDITORES ABIERTOS

✓ PRIMERO

> .angular

> node_modules

✓ src

✓ app

> acumulador

✓ dbz

✓ agregar

agregar.component.css U

<> agregar.component.html U

TS agregar.component.ts U

> interfaces

> main-page

✓ personajes

personajes.component.css U

<> personajes.component.html U

TS personajes.component.ts U

✓ services

TS dbz.service.ts U

TS dbz.module.ts U

> heroes

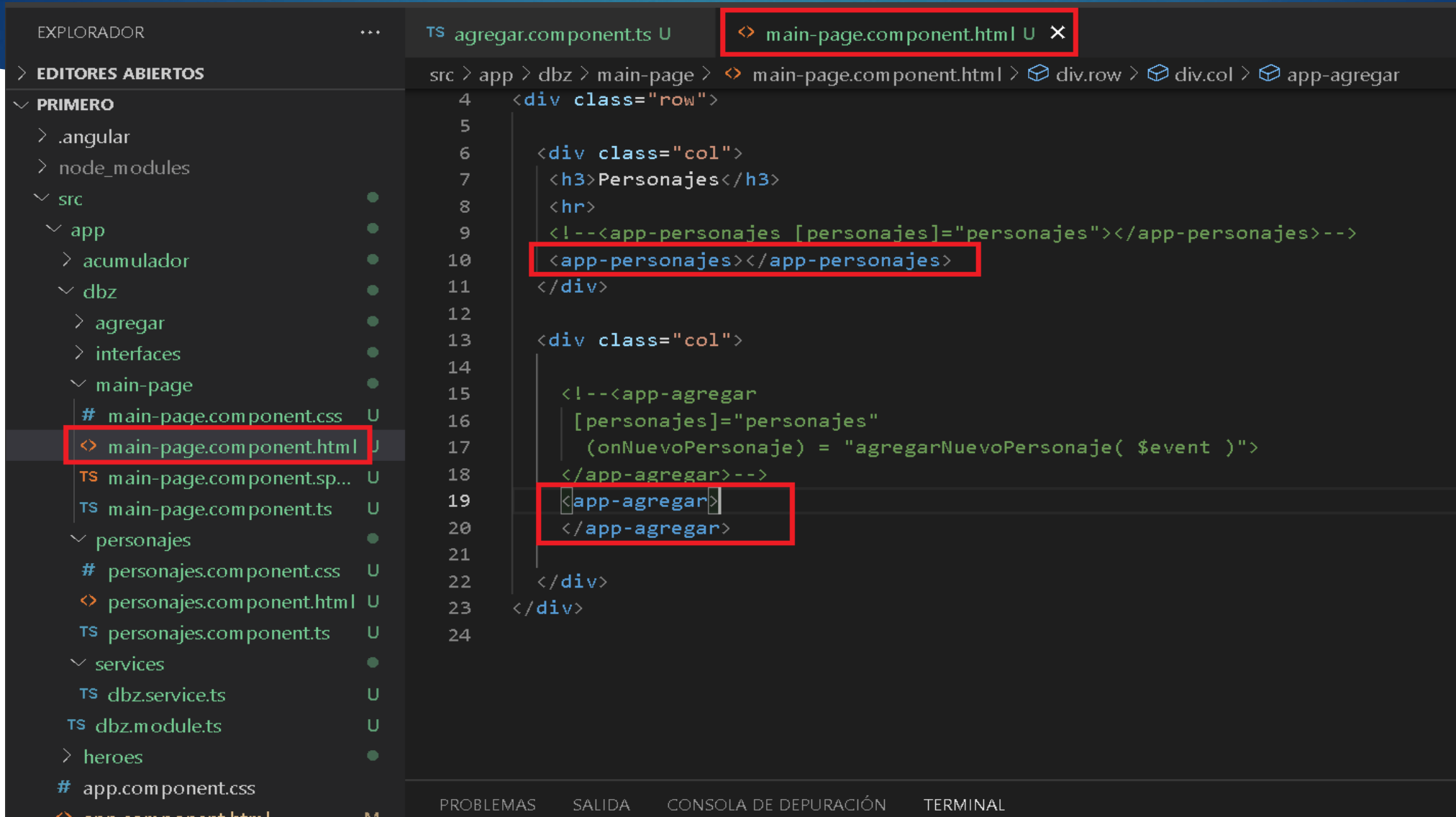
app.component.css

TS agregar.component.ts U X

src > app > dbz > agregar > TS 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

←

→

↻

🏠

🛡️

📄

localhost:4200

Dragon Ball Z

Personajes

- Vegeta - 20000
- Goku - 25000
- Majin Boo - 1500000000

Agregar

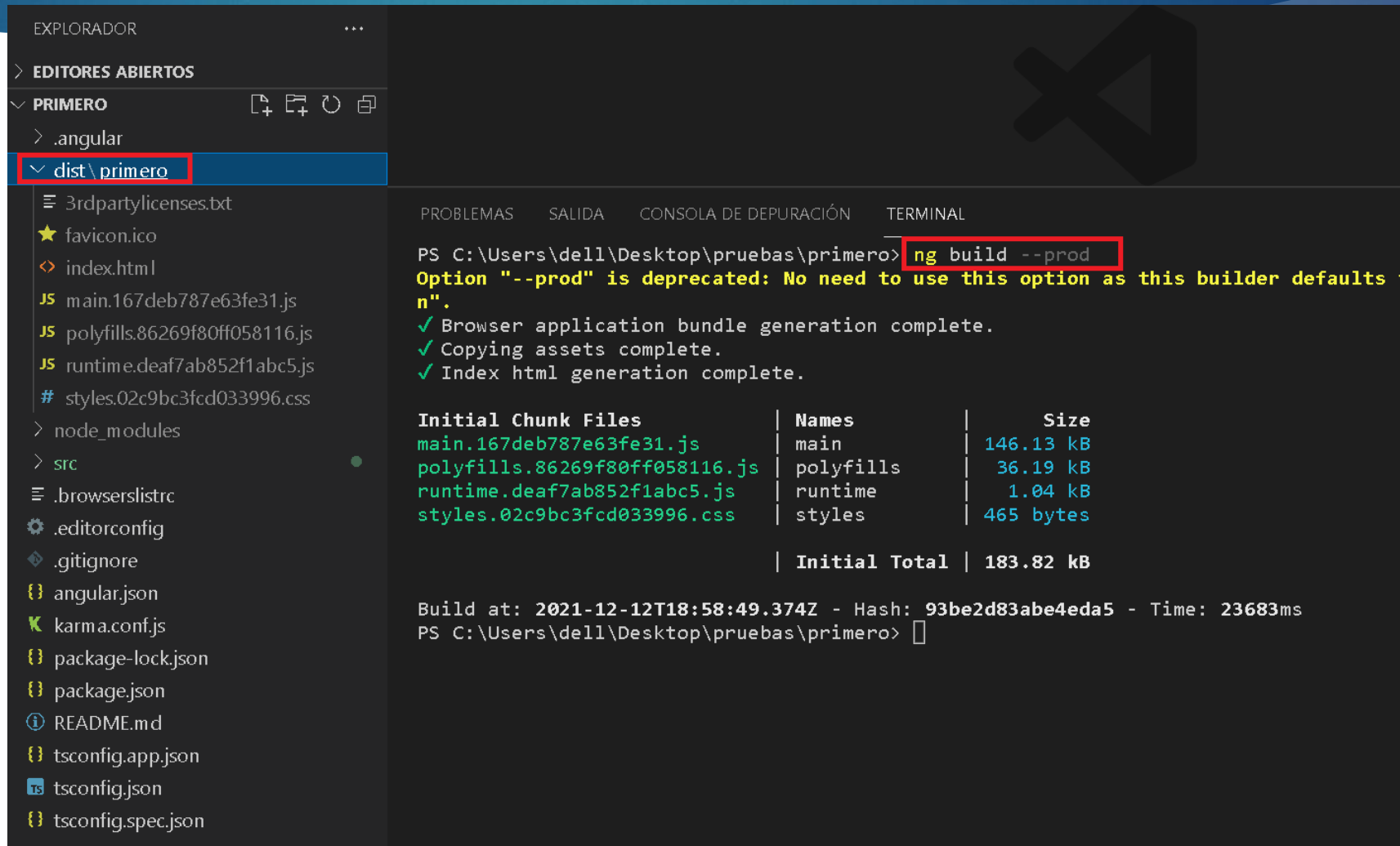
Nombre

0

Agregar

Lanzar nuestra app a producción

► `ng build --prod`



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Terminal on the right. The Explorer sidebar shows the file structure of the project, with the `dist\primero` folder highlighted. The Terminal shows the output of the `ng build --prod` command, which includes a deprecation warning and a summary of the build results.

EXPLORADOR

> EDITORES ABIERTOS

✓ PRIMERO

> .angular

✓ dist\primero

3rdpartylicenses.txt

★ favicon.ico

<> index.html

JS main.167deb787e63fe31.js

JS polyfills.86269f80ff058116.js

JS runtime.deaf7ab852f1abc5.js

styles.02c9bc3fcd033996.css

> node_modules

> src

.browserslistrc

.editorconfig

.gitignore

angular.json

karma.conf.js

package-lock.json

package.json

README.md

tsconfig.app.json

tsconfig.json

tsconfig.spec.json

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

PS C:\Users\dell\Desktop\pruebas\primero> `ng build --prod`

Option "--prod" is deprecated: No need to use this option as this builder defaults to "production".

✓ Browser application bundle generation complete.

✓ Copying assets complete.

✓ Index html generation complete.

Initial Chunk Files	Names	Size
main.167deb787e63fe31.js	main	146.13 kB
polyfills.86269f80ff058116.js	polyfills	36.19 kB
runtime.deaf7ab852f1abc5.js	runtime	1.04 kB
styles.02c9bc3fcd033996.css	styles	465 bytes
Initial Total		183.82 kB

Build at: 2021-12-12T18:58:49.374Z - Hash: 93be2d83abe4eda5 - Time: 23683ms

PS C:\Users\dell\Desktop\pruebas\primero>



 <https://www.netlify.com>



Platform ▾

Pricing

Enterprise ▾

Jamstack ▾

Community ▾

Docs



Contact sales

Log in

[Sign up](#) →

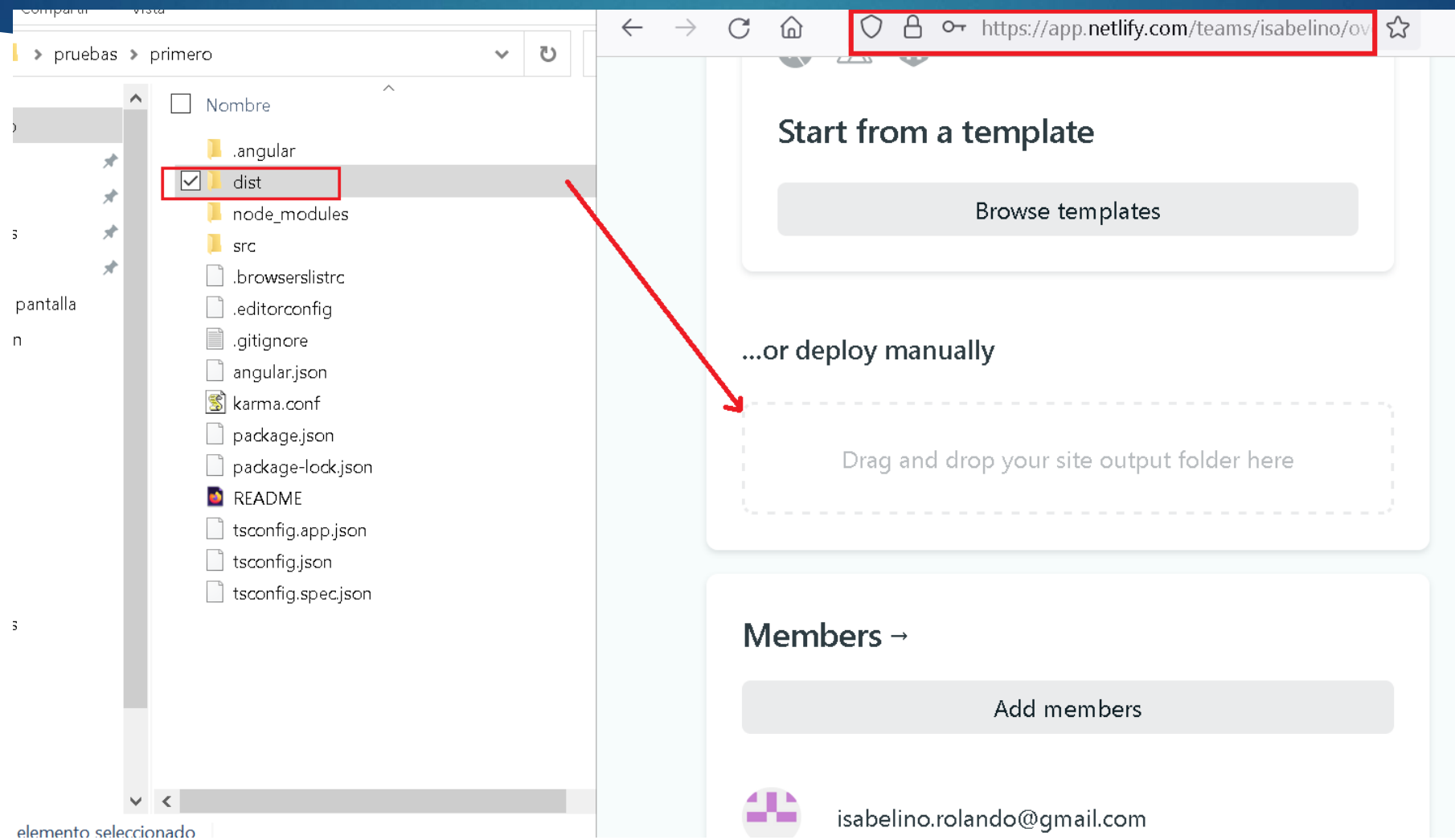
Build, deploy, & scale modern web projects

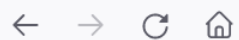
Millions of developers use Netlify to ship faster,
simplify their workflows, and scale effortlessly.

[Get started for free](#)

[Contact sales](#) →

Arrastramos el archivo dist de esta manera





https://app.netlify.com/sites/unruffled-poitras-80b380/deploys



myproject > 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





  https://unruffled-poitras-80b380.netlify.app



Dragon Ball Z

Personajes

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

Agregar

Lifecycle hook

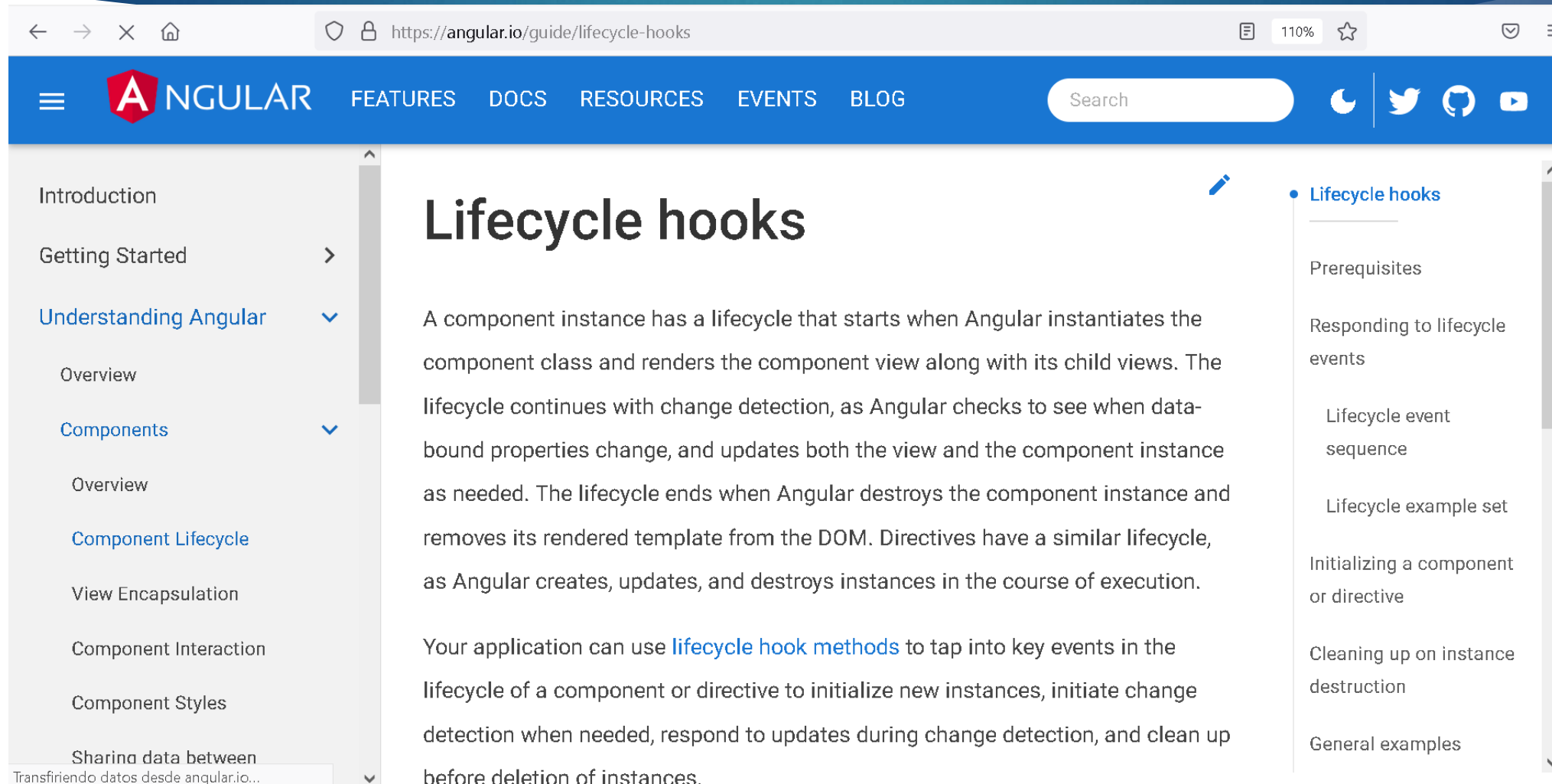
- ▶ Son métodos que podemos llamar cuando angular ejecute ciertos momentos como montaje de componentes, cuando se inicializa, cuando se actualiza el dom entre otros.

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)
CREATE ciclode-vida/browserconfig.xml (600 bytes)
```

La guía oficial



The screenshot shows the Angular.io website with the 'Lifecycle hooks' guide open. The browser address bar shows 'https://angular.io/guide/lifecycle-hooks'. The Angular logo and navigation links (FEATURES, DOCS, RESOURCES, EVENTS, BLOG) are in the top blue bar. A search bar and social media icons are on the right. The left sidebar contains a table of contents with 'Understanding Angular' and 'Components' expanded. The main content area has the title 'Lifecycle hooks' and two paragraphs of text. The right sidebar lists sub-topics under 'Lifecycle hooks'.

← → × 🏠 🔒 https://angular.io/guide/lifecycle-hooks 110% ☆ 📧

≡ **ANGULAR** FEATURES DOCS RESOURCES EVENTS BLOG Search 🌙 🐦 🐙 📺

Introduction

Getting Started >

Understanding Angular >

Overview

Components >

Overview

Component Lifecycle

View Encapsulation

Component Interaction

Component Styles

Sharing data between
Transfiriendo datos desde angular.io...

Lifecycle hooks

A component instance has a lifecycle that starts when Angular instantiates the component class and renders the component view along with its child views. The lifecycle continues with change detection, as Angular checks to see when data-bound properties change, and updates both the view and the component instance as needed. The lifecycle ends when Angular destroys the component instance and removes its rendered template from the DOM. Directives have a similar lifecycle, as Angular creates, updates, and destroys instances in the course of execution.

Your application can use [lifecycle hook methods](#) to tap into key events in the lifecycle of a component or directive to initialize new instances, initiate change detection when needed, respond to updates during change detection, and clean up before deletion of instances.

- Lifecycle hooks
 - Prerequisites
 - Responding to lifecycle events
 - Lifecycle event sequence
 - Lifecycle example set
 - Initializing a component or directive
 - Cleaning up on instance destruction
 - General examples

Creamos los siguientes componentes

EXPLORADOR

EDITORES ABIERTOS

CICLODE-VIDA

- node_modules
- src
 - app
 - pages \ **pagina1**
 - <> pagina1.component.html U
 - TS pagina1.component.ts U
 - # app.component.css
 - <> app.component.html
 - TS app.component.spec.ts
 - TS app.component.ts
 - TS app.module.ts M

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 m
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>
```

EXPLORADOR

EDITORES ABIERTOS

CICLODE-VIDA

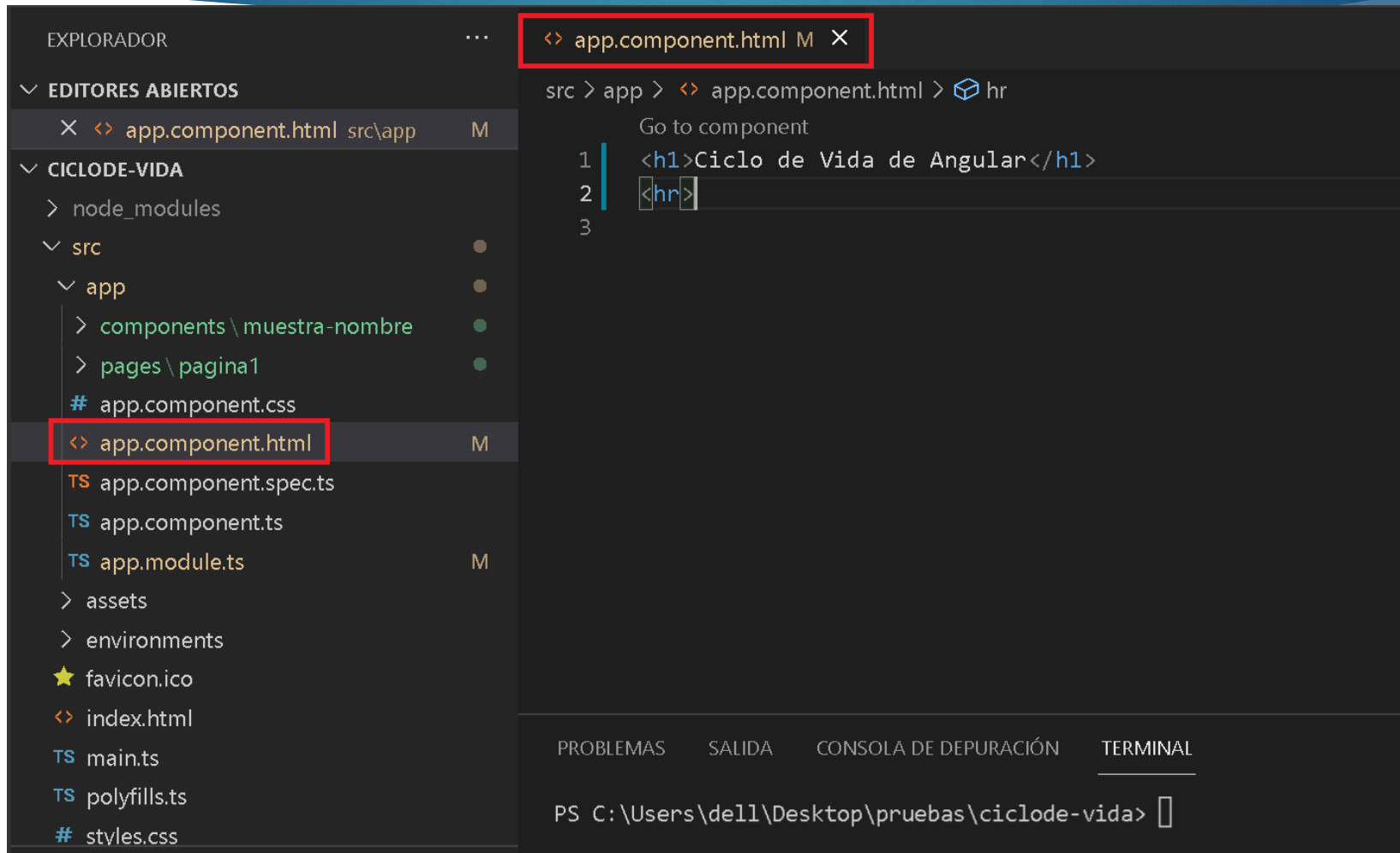
- node_modules
- src
 - app
 - components \ muestra-nombre**
 - muestra-nombre.component.html U
 - TS muestra-nombre.component.ts U
 - pages \ pagina1
 - pagina1.component.html U
 - TS pagina1.component.ts U
 - # app.component.css
 - <> app.component.html
 - TS app.component.spec.ts
 - TS app.component.ts
 - TS app.module.ts M
 - assets

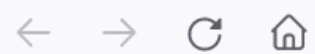
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

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 version.
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 version.
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
<code>ngOnChanges()</code>	<p>Responder cuando Angular establece o restablece propiedades de entrada vinculadas a datos. El método recibe un <code>SimpleChanges</code> objeto de valores de propiedad actuales y anteriores.</p> <p>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.</p>	<p>Se llama antes <code>ngOnInit()</code> (si el componente tiene entradas vinculadas) y cada vez que cambian una o más propiedades de entrada vinculadas a datos.</p> <p>Tenga en cuenta que si su componente no tiene entradas o lo usa sin proporcionar ninguna entrada, el marco no llamará <code>ngOnChanges()</code>.</p>

Método de gancho	Propósito	Momento
<code>ngOnInit()</code>	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 Iniciación de un componente o directiva en este documento.	Llamado una vez, después de la primera <code>ngOnChanges()</code> . <code>ngOnInit()</code> todavía se llama incluso cuando <code>ngOnChanges()</code> no lo es (que es el caso cuando no hay entradas vinculadas a la plantilla).

Método de gancho	Propósito	Momento
<code>ngDoCheck()</code>	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 <code>ngOnChanges()</code> de cada ejecución de detección de cambios e inmediatamente después <code>ngOnInit()</code> de la primera ejecución.

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

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

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

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

Método de gancho	Propósito	Momento
<code>ngOnDestroy()</code>	<p>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.</p> <p>Consulte los detalles en Limpieza de la destrucción de instancias en este documento.</p>	<p>Se llama inmediatamente antes de que Angular destruya la directiva o el componente.</p>

En pagina1.component.ts agregamos esto

TS pagina1.component.ts U X

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

```
1  import { AfterContentChecked, AfterContentInit,
2    AfterViewChecked, AfterViewInit,
3    Component, DoCheck, OnChanges, OnDestroy,
4    OnInit, SimpleChanges } from '@angular/core';
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, AfterContentInit,
14    AfterContentChecked, AfterViewInit, AfterViewChecked, OnDestroy {
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');
```


TS pagina1.component.ts U X

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 X

src > app > pages > pagina1 > TS pagina1.component.ts > Pagina1Compone

```
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
39   ngOnInit(): void {
40     |   console.log('ngOnInit')
41   }
42
43 }
```

EXPLORADOR

EDITORES ABIERTOS

TS pagina1.component.ts src\ap... U

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

CICLODE-VIDA

src

app

> components

pages \ pagina1

<> pagina1.component.html U

TS pagina1.component.ts U

app.component.css

<> app.component.html M

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts M

> assets

> ESQUEMA

> LÍNEA DE TIEMPO

TS pagina1.component.ts U

<> app.component.html M X

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.

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:40:14

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:33:14

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:33:14

[webpack-dev-server] Live Reloading enabled. index.js:548

>>

EXPLORADOR

> EDITORES ABIERTOS

▼ CICLODE-VIDA

▼ src

▼ app

> components

▼ pages \ pagina1

<> pagina1.component.html U

TS pagina1.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

> ESQUEMA

> LÍNEA DE TIEMPO

TS pagina1.component.ts U

TS app.component.ts M X

<> app.comp

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    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    mostrar:boolean = true;
10
11  }
```

PROBLEMAS

SALIDA

TERMINAL

...

> node

+ v

e: 353ms

✓ Compiled successfully.

EXPLORADOR

> EDITORES ABIERTOS

▼ CICLODE-VIDA

▼ src

▼ app

> components

▼ pages \ pagina1

<> pagina1.component.html U

TS pagina1.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

> ESQUEMA

> LÍNEA DE TIEMPO

component.ts U

TS app.component.ts M

<> app.component.html M X

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.

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

TS pagina1.component.ts

app.component.css

<> app.component.html

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts

assets

environments

★ favicon.ico

ESQUEMA

TS pagina1.component.ts U X

TS app.component.ts M

<> app.compor

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

38

39

40

41

42

43

44

45

46

47

ngOnInit(): void {

console.log('ngOnInit')

}

guardar(){

}

}

PROBLEMAS

SALIDA

TERMINAL

node

e: 245ms

✓ Compiled successfully.

EXPLORADOR

> EDITORES ABIERTOS

▼ CICLODE-VIDA

▼ src

▼ app

> components

▼ pages \ pagina1

<> pagina1.component.html U

TS pagina1.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

> ESQUEMA

TS pagina1.component.ts U

<> pagina1.component.html U X

src > app > pages > pagina1 > <> pagina1.component.html > ...

Go to component

1

<p>pagina1 works!</p>

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 Informadón Depurar CSS XHR Peticiones

ngDoCheck	pagina1.component.ts:21:4
ngAfterContentInit	pagina1.component.ts:27:4
ngAfterViewChecked	pagina1.component.ts:33:2

>>



EXPLORADOR

component.ts M

<> pagina1.component.html U

TS pagina1.component.ts U

TS app.module.ts M X

> EDITORES ABIERTOS

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

6 import { MuestraNombreComponent } from '../components/muestra-nombre/muestra-nombre.component';

7 import { FormsModule } from '@angular/forms';

8

9 @NgModule({

10 declarations: [

11 AppComponent,

12 Pagina1Component,

13 MuestraNombreComponent

14],

15 imports: [

16 BrowserModule,

17 FormsModule

18],

19 providers: [],

20 bootstrap: [AppComponent]

21 })

22 export class AppModule { }

23

src

app

components

pages \ pagina1

<> pagina1.component.html U

TS pagina1.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

EXPLORADOR

EDITORES ABIERTOS

CICLODE-VIDA

src

app

components

pages \ pagina1

pagina1.component.html U

TS pagina1.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

TS main.ts

TS polyfills.ts

styles.css

ESQUEMA

LÍNEA DE TIEMPO

component.ts M

pagina1.component.html U

TS pagina1.component.ts U

TS app.module.ts M

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

```
4      ngOnInit, SimpleChanges } from '@angular/core';
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

TS pagina1.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

component.ts M

<> pagina1.component.html U X

TS pagina1.component.ts U

TS app

src > app > pages > pagina1 > <> pagina1.component.html > ...

Go to component

1 <p>pagina1 works!</p>

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!

Click



Inspector



Consola



Depurador



Filtrar salida



Errores

Advertencias

Registros

Información

Depurar

CSS

XHR

Peticiones

ngDoCheck

pagina1.component.ts:23:1

ngAfterContentInit

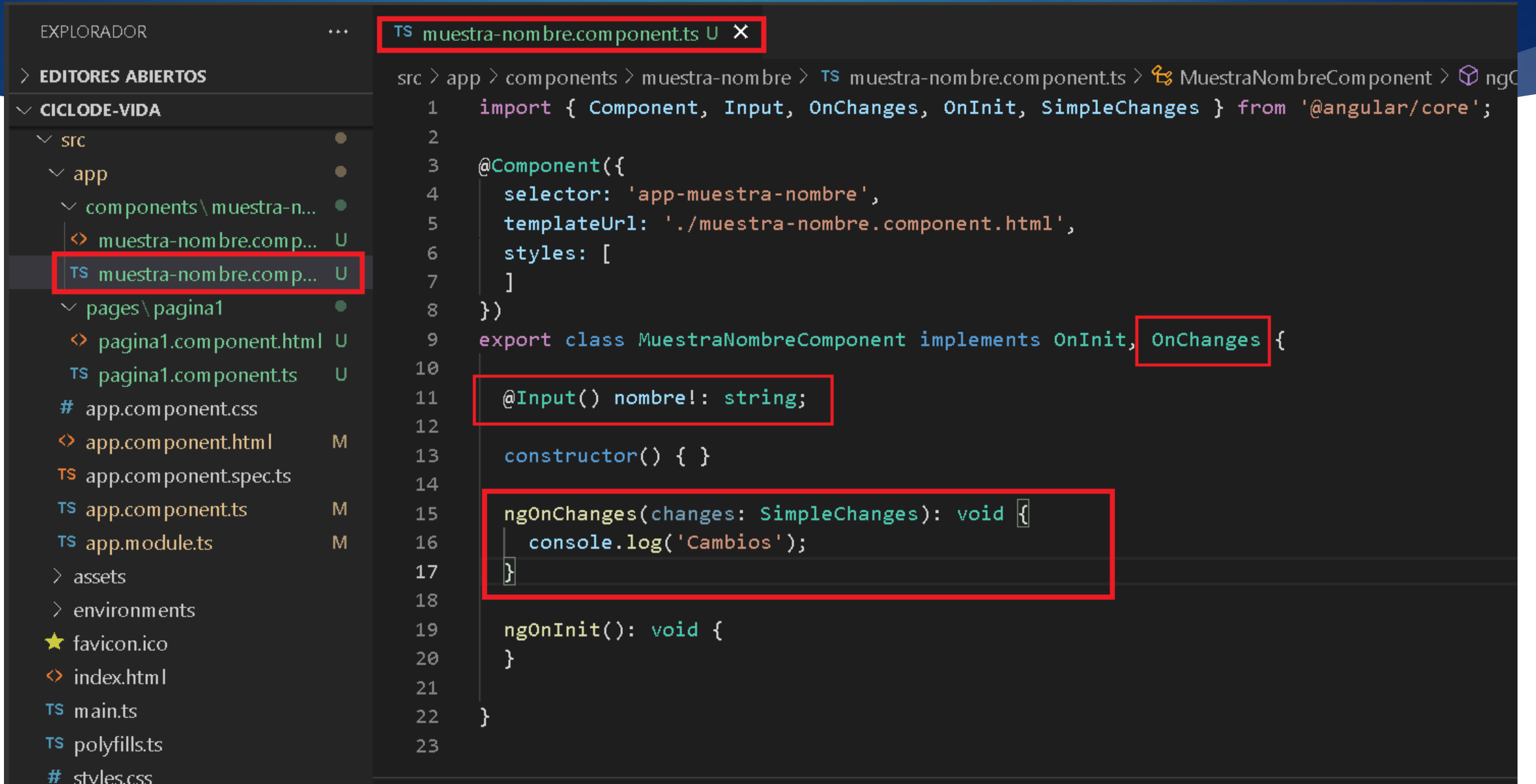
pagina1.component.ts:29:1

ngAfterViewChecked

pagina1.component.ts:35:1



Ejemplo de onChange



The image shows a screenshot of an IDE with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORADOR', shows a project structure with folders like 'src', 'app', 'components', 'pages', and 'assets'. The file 'muestra-nombre.component.ts' is highlighted in the 'components' folder. The code editor shows the implementation of the 'MuestraNombreComponent' class, which implements the 'OnChanges' interface. The code includes imports from '@angular/core', a component decorator, and the implementation of the 'ngOnChanges' method, which logs 'Cambios' to the console. Red boxes highlight the file name in the explorer, the 'OnChanges' interface in the class declaration, the '@Input()' decorator, and the 'ngOnChanges' method implementation.

```
EXPLORADOR
...
TS muestra-nombre.component.ts U X

src > app > components > muestra-nombre > TS muestra-nombre.component.ts > MuestraNombreComponent > ngOnChanges

1  import { Component, Input, OnChanges, OnInit, SimpleChanges } from '@angular/core';
2
3  @Component({
4    selector: 'app-muestra-nombre',
5    templateUrl: './muestra-nombre.component.html',
6    styles: [
7    ]
8  })
9  export class MuestraNombreComponent implements OnInit, OnChanges {
10
11    @Input() nombre!: string;
12
13    constructor() { }
14
15    ngOnChanges(changes: SimpleChanges): void {
16      console.log('Cambios');
17    }
18
19    ngOnInit(): void {
20    }
21
22  }
23
```

EXPLORADOR

> EDITORES ABIERTOS

▼ CICLODE-VIDA

▼ src

▼ app

▼ components\ muestra-n...

❏ muestra-nombre.comp... U

TS muestra-nombre.comp... U

▼ pages\ pagina1

❏ pagina1.component.html U

TS pagina1.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

TS main.ts

TS muestra-nombre.component.ts U

❏ pagina1.component.html U X

src > app > pages > pagina1 > ❏ pagina1.component.html > ...

Go to component

1 <p>pagina1 works!</p>

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!

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

