



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE MILPA ALTA II

ASIGNATURA:

MEAN STACK FOR FRONT – END

TEMA:

- INSTALACION
- PRACTICA 1

ELABORA:

POZOS RIVERA ALEJANDRO

NÚMERO DE CONTROL:

171190025

PROFESOR:

ROLDAN AQUINO SEGURA

16 DE MAYO DE 2021

Instalación de angular

Como primer paso vamos a abrir cualquier tipo de terminal en nuestro sistema. En este caso abriré cmd. Primero verificaremos que ya estén instalados node y npm. Los comandos son `node --version` y `npm --version`.

```
Seleccionar Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19041.985]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>node -version
node: bad option: -version

C:\WINDOWS\system32>node --version
v14.16.1

C:\WINDOWS\system32>npm --version
7.9.0

C:\WINDOWS\system32>
```

Ahora si no tenemos angular solo escribiremos el siguiente comando `npm install -g @angular/cli`. Si ya lo tenemos instalado únicamente lo que hará es hacer un tipo de update.

```
Seleccionar Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19041.985]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>node -version
node: bad option: -version

C:\WINDOWS\system32>node --version
v14.16.1

C:\WINDOWS\system32>npm --version
7.9.0

C:\WINDOWS\system32>npm install -g @angular/cli
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

changed 234 packages, and audited 235 packages in 35s

23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\WINDOWS\system32>
```

Con el siguiente comando solo validaremos la versión de angular `ng --version`.

```
Administrador: C:\WINDOWS\system32\cmd.exe

found 0 vulnerabilities

C:\WINDOWS\system32>ng --version

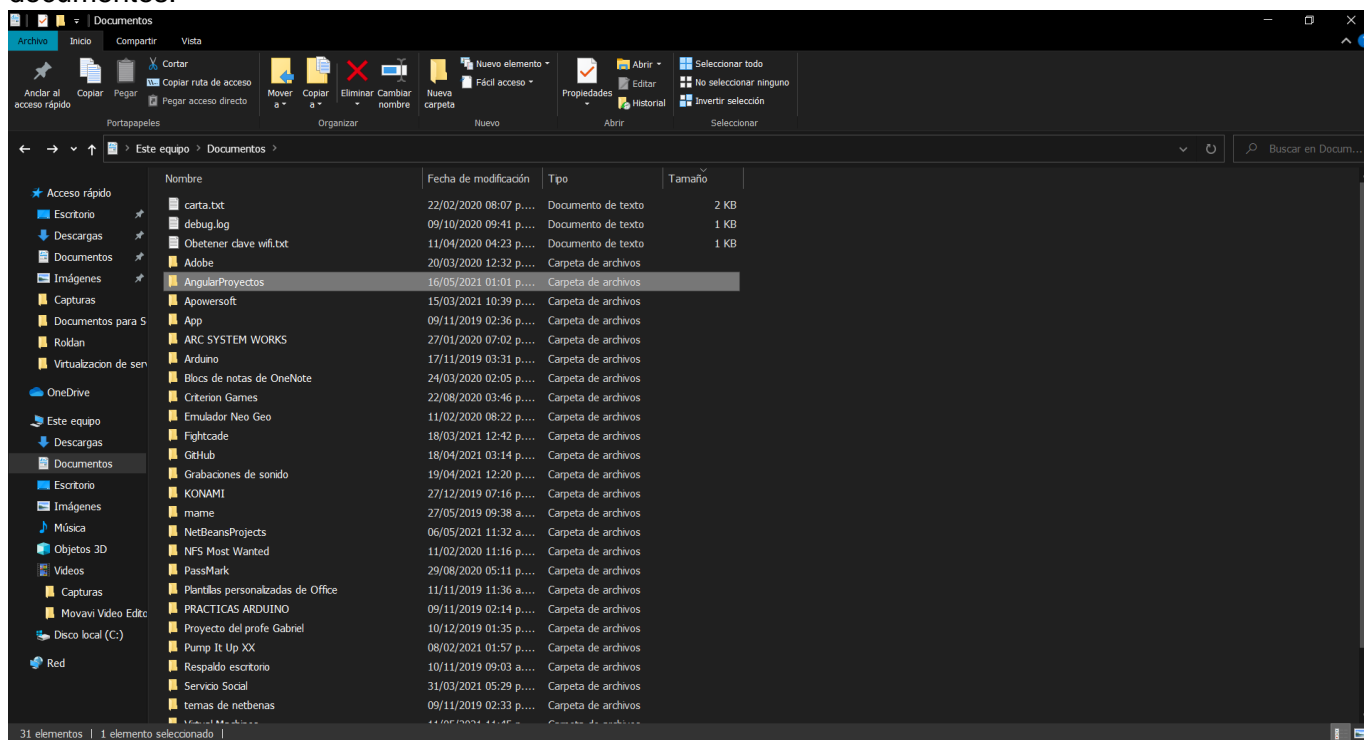
Angular CLI

Angular CLI: 12.0.0
Node: 14.16.1
Package Manager: npm 7.9.0
OS: win32 x64

Angular:
...

Package      Version
-----
@angular-devkit/architect    0.1200.0 (cli-only)
@angular-devkit/core         12.0.0 (cli-only)
@angular-devkit/schematics   12.0.0 (cli-only)
@schematics/angular          12.0.0 (cli-only)
```

Ahora vamos a crear la carpeta donde se empezarán a alojar los proyectos que se vayan generando. La carpeta que cree lleva como nombre AngularProyectos y está dentro de la carpeta principal de documentos.



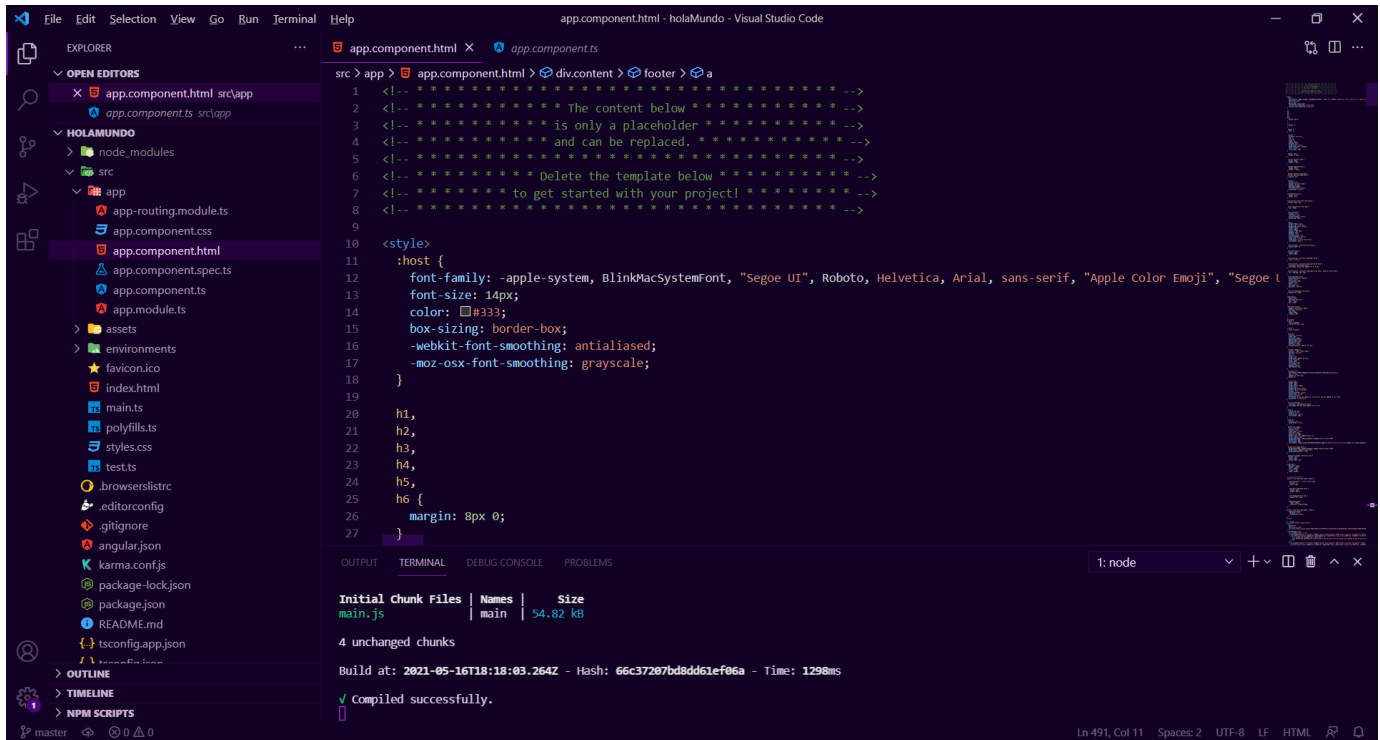
Creamos un nuevo proyecto, pero debemos ingresar la ruta donde se creó la carpeta, por ejemplo.
C:\Users\Admin\Documents\AngularProyectos.

```
Seleccíon nrm instal
C:\Users\Admin\Documents\AngularProyectos>ng new holaMundo
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE holaMundo/angular.json (3057 bytes)
CREATE holaMundo/package.json (1072 bytes)
CREATE holaMundo/README.md (999 bytes)
CREATE holaMundo/tsconfig.json (783 bytes)
CREATE holaMundo/.editorconfig (274 bytes)
CREATE holaMundo/.gitignore (604 bytes)
CREATE holaMundo/.browserslistrc (703 bytes)
CREATE holaMundo/karma.conf.js (1426 bytes)
CREATE holaMundo/tsconfig.app.json (287 bytes)
CREATE holaMundo/tsconfig.spec.json (333 bytes)
CREATE holaMundo/src/favicon.ico (948 bytes)
CREATE holaMundo/src/index.html (295 bytes)
CREATE holaMundo/src/main.ts (372 bytes)
CREATE holaMundo/src/polyfills.ts (2820 bytes)
CREATE holaMundo/src/styles.css (80 bytes)
CREATE holaMundo/src/test.ts (743 bytes)
CREATE holaMundo/src/assets/.gitkeep (0 bytes)
CREATE holaMundo/src/environments/environment.prod.ts (51 bytes)
CREATE holaMundo/src/environments/environment.ts (658 bytes)
CREATE holaMundo/src/app/app-routing.module.ts (245 bytes)
CREATE holaMundo/src/app/app.module.ts (393 bytes)
CREATE holaMundo/src/app/app.component.html (23809 bytes)
CREATE holaMundo/src/app/app.component.spec.ts (1066 bytes)
CREATE holaMundo/src/app/app.component.ts (213 bytes)
CREATE holaMundo/src/app/app.component.css (0 bytes)
```

Y finalizamos.

```
Administrador: C:\WINDOWS\system32\cmd.exe
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.spec.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.prod.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/polyfills.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
Successfully initialized git.
```

Ahora abriremos nuestro editor de textos y una vez estando ando ahí abriremos la ruta de nuestro proyecto.



```
src > app > app.component.html > div.content > footer > a
1 <!-- ***** The content below ***** -->
2 <!-- ***** The content below ***** -->
3 <!-- ***** is only a placeholder ***** -->
4 <!-- ***** and can be replaced. ***** -->
5 <!-- ***** -->
6 <!-- ***** Delete the template below ***** -->
7 <!-- ***** to get started with your project! ***** -->
8 <!-- ***** -->
9
10 <style>
11 :host {
12   font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif, "Apple Color Emoji", "Segoe UI Color Emoji";
13   font-size: 14px;
14   color: #333;
15   box-sizing: border-box;
16   -webkit-font-smoothing: antialiased;
17   -moz-osx-font-smoothing: grayscale;
18 }
19
20 h1,
21 h2,
22 h3,
23 h4,
24 h5,
25 h6 {
26   margin: 8px 0;
27 }
```

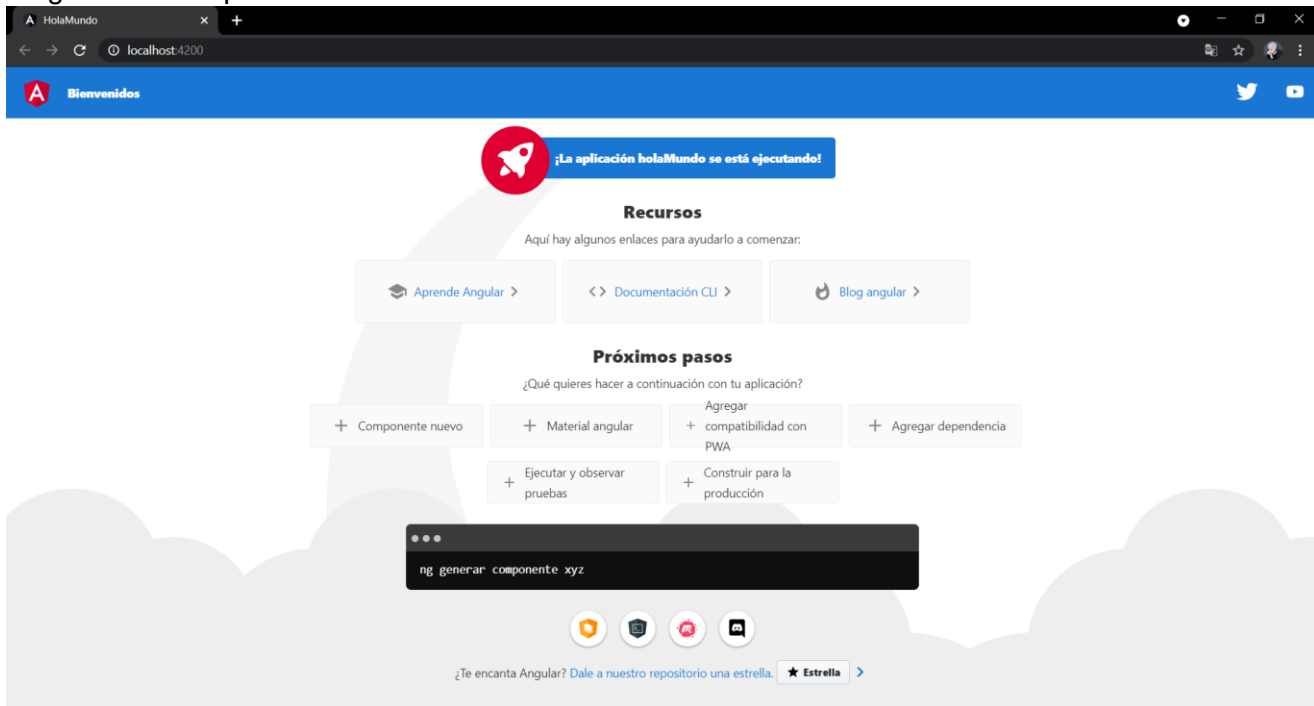
Initial Chunk Files | Names | Size
main.js | main | 54.82 kB

4 unchanged chunks

Build at: 2021-05-16T18:18:03.264Z - Hash: 66c37207bd8dd61ef06a - Time: 1298ms

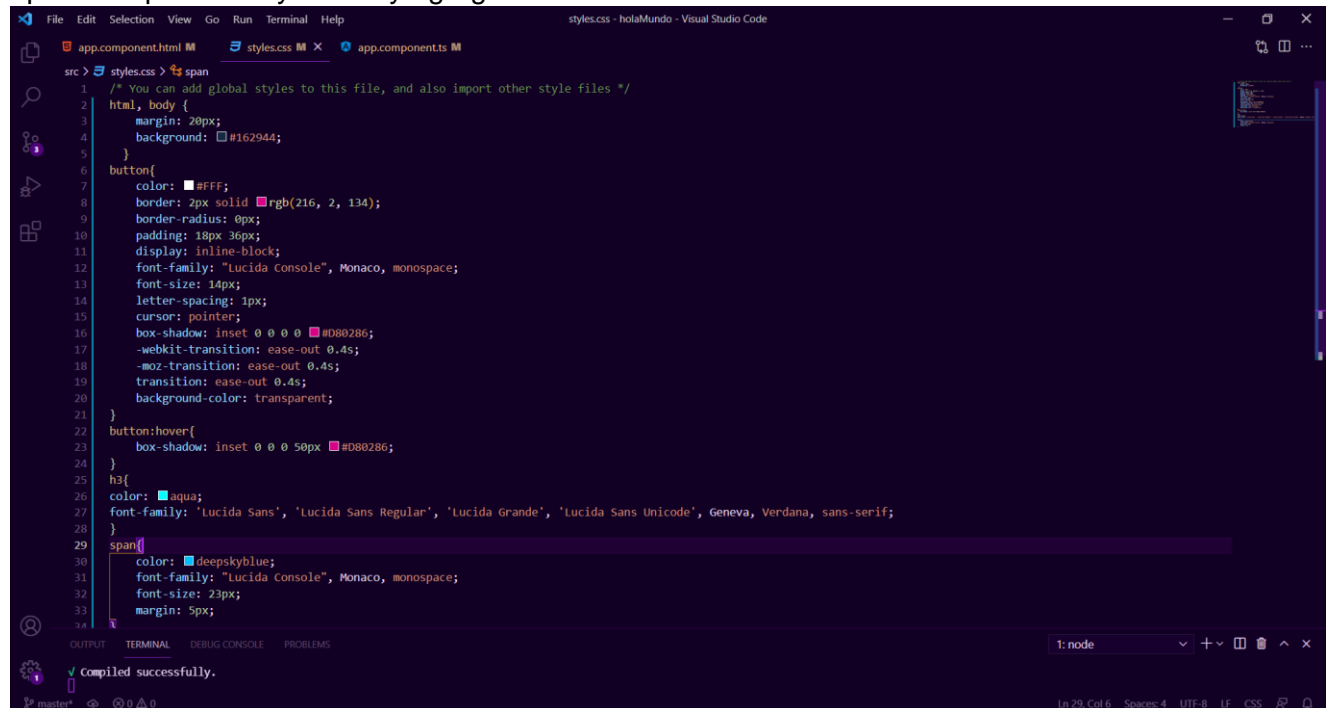
✓ Compiled successfully.

Abriremos la terminal del editor y escribiremos la siguiente línea de comando que es la que nos va a permitir ver los cambios dentro del servidor. `ng serve -o`. Y en automático se nos abrirá el navegador que tengamos como predeterminado.



Practica 1

Primero vamos a agregar un poco de estilos a nuestra pagina para ellos nos vamos a posicionar dentro del apartado que dice styles.css y agregaremos nuestros estilos.



```
1  /* You can add global styles to this file, and also import other style files */
2  html, body {
3    margin: 20px;
4    background: #162944;
5  }
6  button{
7    color: #FFFF;
8    border: 2px solid #rgb(216, 2, 134);
9    border-radius: 0px;
10   padding: 18px 36px;
11   display: inline-block;
12   font-family: "Lucida Console", Monaco, monospace;
13   font-size: 14px;
14   letter-spacing: 1px;
15   cursor: pointer;
16   box-shadow: inset 0 0 0 #D80286;
17   -webkit-transition: ease-out 0.4s;
18   -moz-transition: ease-out 0.4s;
19   transition: ease-out 0.4s;
20   background-color: transparent;
21 }
22 button: hover{
23   box-shadow: inset 0 0 0 50px #D80286;
24 }
25 h3{
26   color: #aqua;
27   font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
28 }
29 span{
30   color: #deepskyblue;
31   font-family: "Lucida Console", Monaco, monospace;
32   font-size: 23px;
33   margin: 5px;
34 }
```

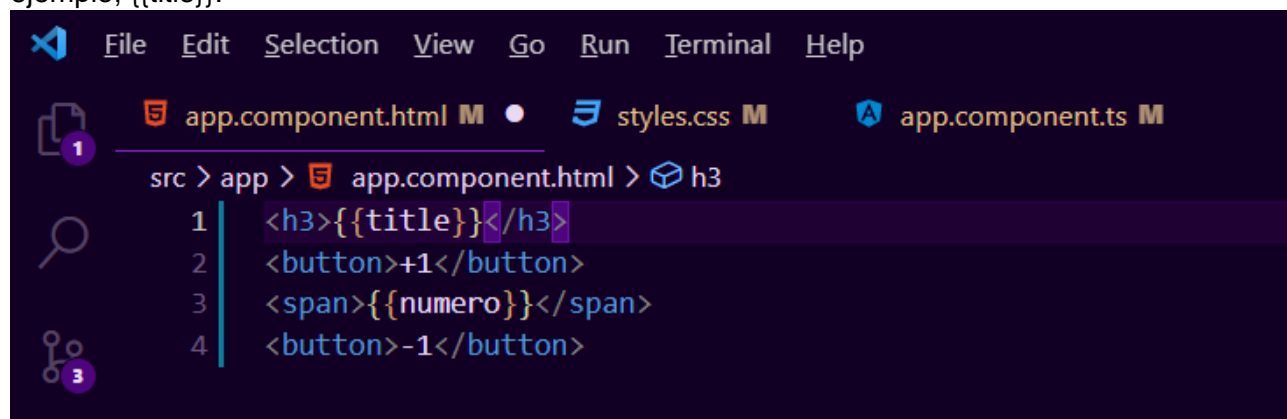
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: node

Compiled successfully.

Ln 29, Col 6 Spaces: 4 UTF-8 LF CSS

Ahora nos vamos a dirigir a la parte de app.component.html y vamos a agregar el titulo y los 2 botones, uno para incrementar y el otro para decrementar. Algo que hay que resaltar es que cuando ya sabes cómo se llaman los electos que vas a ocupar los puedes mandar a llamar por medio de dos juegos de llaves ejemplo, {{title}}.



```
1  <h3>{{title}}</h3>
2  <button>+1</button>
3  <span>{{numero}}</span>
4  <button>-1</button>
```

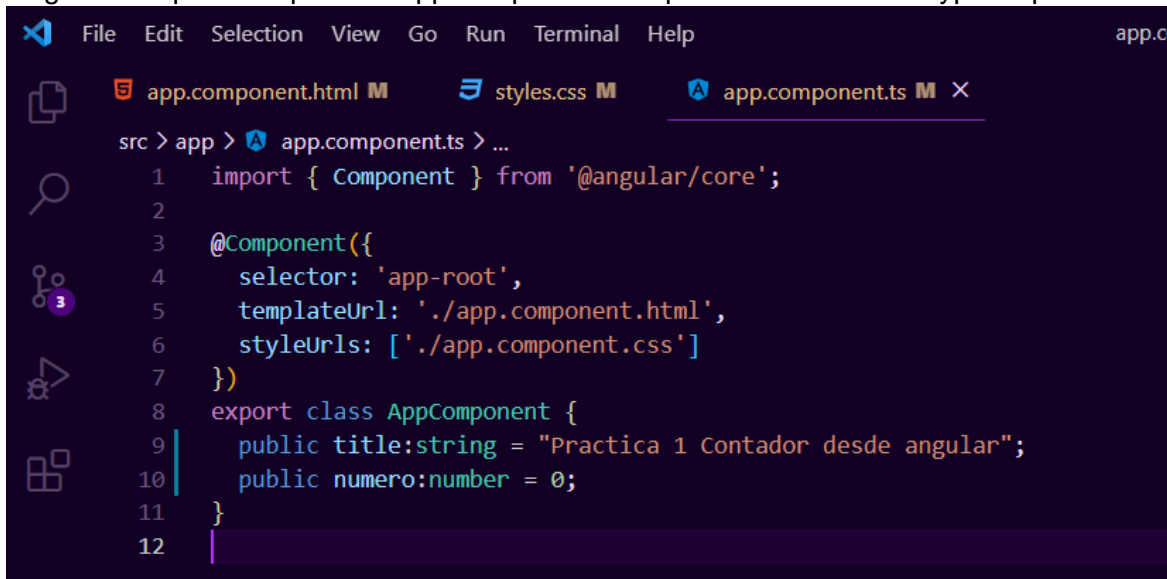
1

3

src > app > app.component.html > h3

Compiled successfully.


Para poder cambiar el nombre a algunos elementos como el título y otros que queramos agregar nos vamos a dirigir en el apartado que dice app.component.ts que son los archivos typescript de nuestro

A screenshot of the Visual Studio Code editor with a dark theme. The file explorer on the left shows a project structure with 'src' > 'app' > 'app.component.ts'. The editor window shows the content of 'app.component.ts' with the following code:

```
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   public title:string = "Practica 1 Contador desde angular";
10  public numero:number = 0;
11 }
12
```

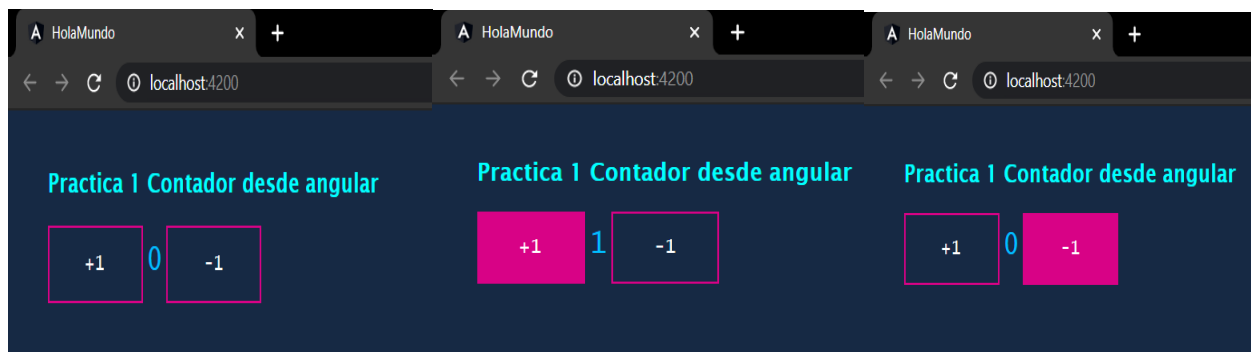
proyecto.

Como primer método lo que tenemos que hacer es crear (click) y asignarle la función que debe de realizar en caso de presionar el botón, para ello el primero método le asignaremos que "numero = numero + 1".

A screenshot of the Visual Studio Code editor with a dark theme. The file explorer on the left shows a project structure with 'src' > 'app' > 'app.component.html'. The editor window shows the content of 'app.component.html' with the following code:

```
1 <h3>{{title}}</h3>
2 <button (click)="numero = numero + 1">+1 </button>
3 <span>{{numero}}</span>
4 <button (click)="numero = numero - 1">-1</button>
5
```

Y visualizaremos los resultados en la página.



Para el segundo método lo que se tiene que hacer es por el mismo método (click) asignarle el método que creamos dentro de nuestro app.components.ts los cuales son sumar y restar.

```
File Edit Selection View Go Run Terminal Help app.component.ts - holaMundo - Visual

app.component.html styles.css app.component.ts X

src > app > app.component.ts > ...
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   public title:string = "Practica 1 Contador desde angular";
10  public numero:number = 0;
11
12  public sumar(){
13    this.numero++;
14  }
15
16  public restar(){
17    this.numero--;
18  }
19 }
20
```

Y lo mandamos a traer en nuestro archivo app.components.html, comentando el método anterior.

```
File Edit Selection View Go Run Terminal Help app.com

app.component.html X styles.css app.component.ts M

src > app > app.component.html > ...
1 <h3>{{title}}</h3>
2 <!-- <button (click)="numero = numero + 1">+1 </button>
3 <span>{{numero}}</span>
4 <button (click)="numero = numero - 1">-1</button> -->
5 <button (click)="sumar()">+1</button>
6 <span>{{numero}}</span>
7 <button (click)="restar()">-1</button>
8
```


Para nuestro método numero 3 de igual manera por medio del método (click) asignarle el método que creamos dentro de nuestro app.components.ts el cual es sumarRestar en donde le mandaremos por medio de parámetro un valor.



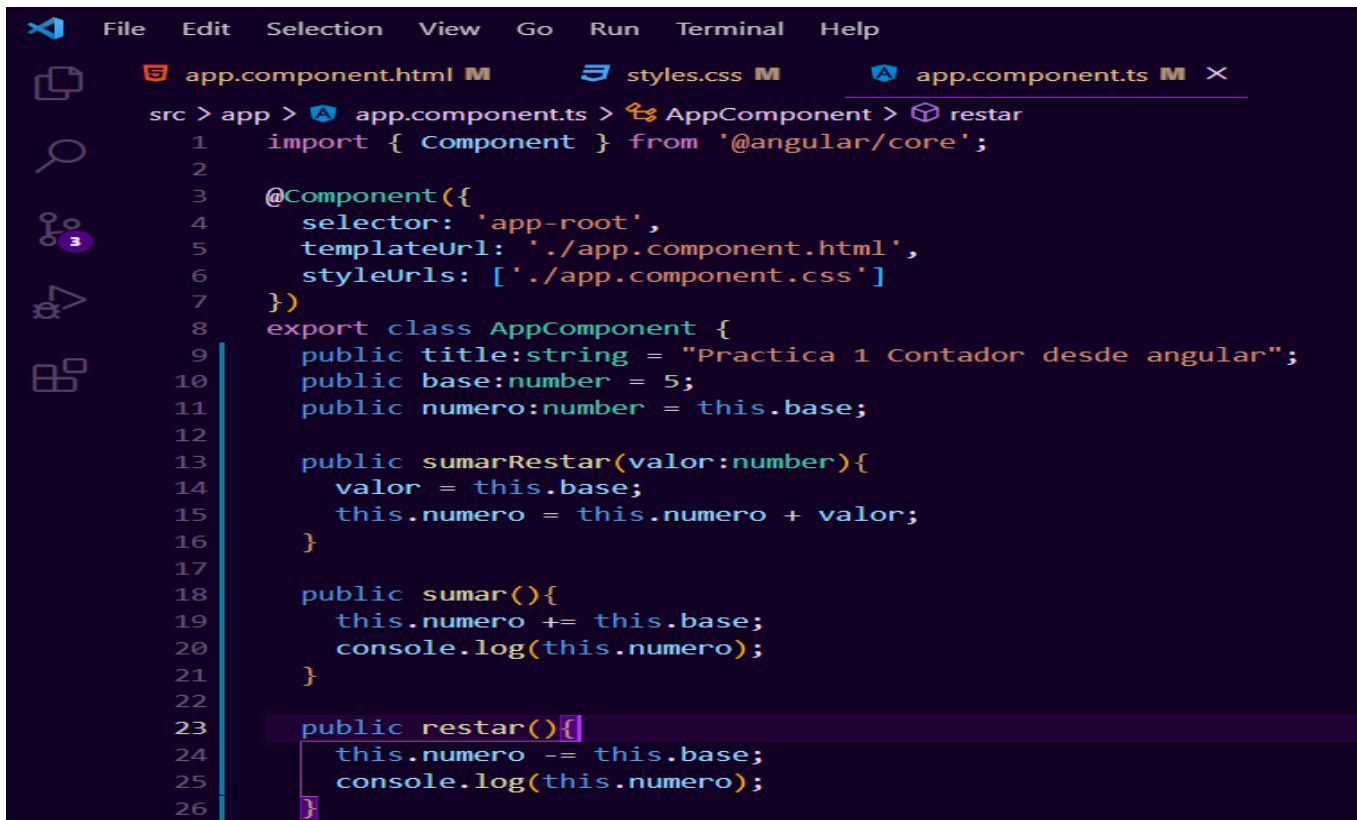
```
src > app > app.component.ts > AppComponent > sumarRestar
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    public title:string = "Practica 1 Contador desde angular";
10   public numero:number = 0;
11
12   public sumarRestar(valor:number){
13     this.numero = this.numero + valor;
14   }
15
16   public sumar(){
17     this.numero++;
18   }
19
20   public restar(){
21     this.numero--;
22   }
23 }
```

Y lo mandamos a traer en nuestro archivo app.components.html, comentando los métodos anteriores.



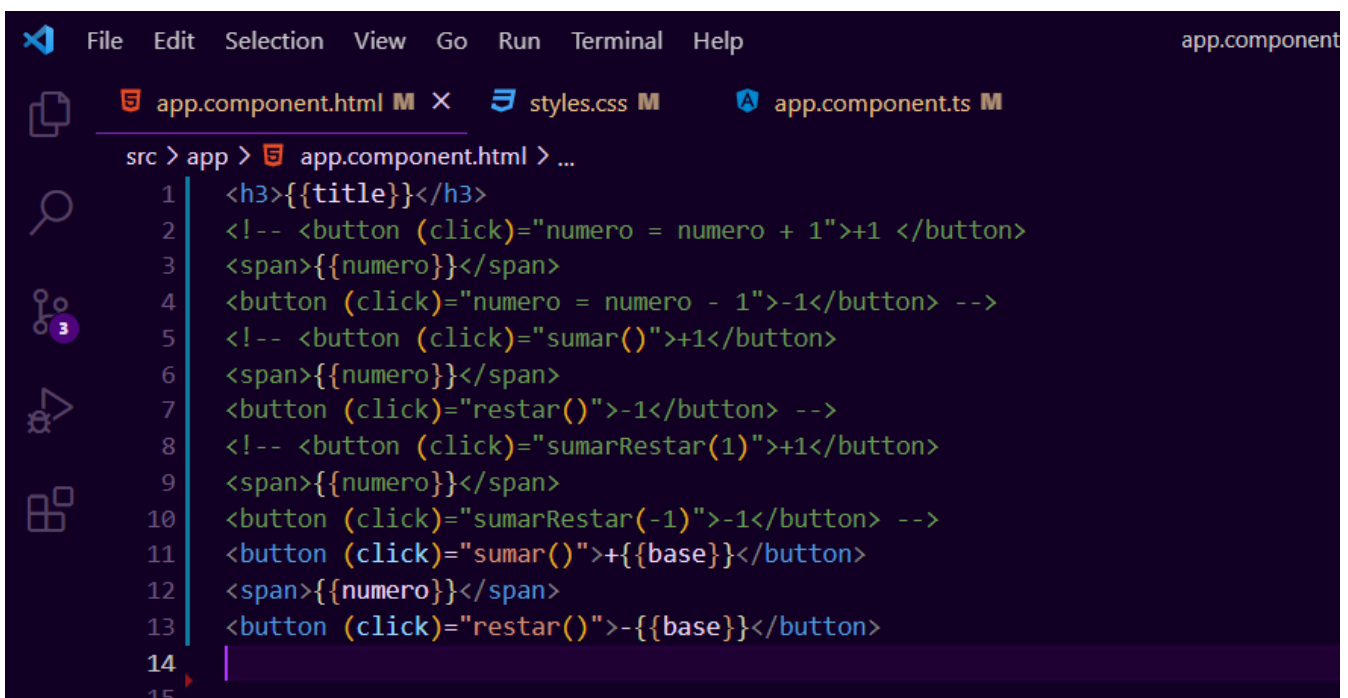
```
src > app > app.component.html > ...
1  <h3>{{title}}</h3>
2  <!-- <button (click)="numero = numero + 1">+1 </button>
3  <span>{{numero}}</span>
4  <button (click)="numero = numero - 1">-1</button> -->
5  <!-- <button (click)="sumar()">+1</button>
6  <span>{{numero}}</span>
7  <button (click)="restar()">-1</button> -->
8  <button (click)="sumarRestar(1)">+1</button>
9  <span>{{numero}}</span>
10 <button (click)="sumarRestar(-1)">-1</button>
11
```

Por último, ahora vamos a crear una nueva variable que se llame base y le asignaremos el valor de 5 para que ahora en lugar de empezar desde 0 empezara en 5 e ira decrementando o incrementando de 5 en 5.



```
src > app > app.component.ts > AppComponent > restar
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    public title:string = "Practica 1 Contador desde angular";
10   public base:number = 5;
11   public numero:number = this.base;
12
13   public sumarRestar(valor:number){
14     valor = this.base;
15     this.numero = this.numero + valor;
16   }
17
18   public sumar(){
19     this.numero += this.base;
20     console.log(this.numero);
21   }
22
23   public restar(){
24     this.numero -= this.base;
25     console.log(this.numero);
26   }
```

También en el html lo que se tiene que hacer es mandar a llamar los métodos de sumar y restar que son los que ocupamos.



```
src > app > app.component.html > ...
1  <h3>{{title}}</h3>
2  <!-- <button (click)="numero = numero + 1">+1 </button>
3  <span>{{numero}}</span>
4  <button (click)="numero = numero - 1">-1</button> -->
5  <!-- <button (click)="sumar()">+1</button>
6  <span>{{numero}}</span>
7  <button (click)="restar()">-1</button> -->
8  <!-- <button (click)="sumarRestar(1)">+1</button>
9  <span>{{numero}}</span>
10 <button (click)="sumarRestar(-1)">-1</button> -->
11 <button (click)="sumar()">+{{base}}</button>
12 <span>{{numero}}</span>
13 <button (click)="restar()">-{{base}}</button>
14
15
```

Verificamos que efectivamente empiecen los cambios desde 5 y que incremente o decremente según el caso.

