

TypeScript

- ▶ TypeScript es un lenguaje de programación fuertemente tipado que se basa en JavaScript de tipado estático y basado en clases, lo que le brinda mejores herramientas a cualquier escala. Lanzado en el 2012 por Microsoft.
- ▶ <https://www.typescriptlang.org/>



- ▶ **TypeScript (TS) es un lenguaje de programación construido a un nivel superior de JavaScript (JS).** Esto quiere decir que TypeScript dota al lenguaje de varias características adicionales que hacen que podamos escribir código con menos errores, más sencillo, coherente y fácil de probar, en definitiva, más limpio y sólido.
- ▶ Fue creado por **Microsoft** en 2012 y, desde entonces, su adopción no ha hecho más que crecer. Especialmente, desde que **Google** decidió adoptarlo como lenguaje por defecto para desarrollar con **Angular**. Aunque, hoy en día, podemos desarrollar con TypeScript en cualquiera de los **frameworks o librerías** más punteras, como son **React** para el frontend o **Node** para el backend.

Como lo obtenemos?

- ▶ Instalamos el nodeJS para obtener el gestor de paquetes de Javascript
- ▶ Y luego instalamos typescript con npm





Welcome to the Node.js Setup Wizard



The Setup Wizard will install Node.js on your computer.

Back

Next

Cancel

Comprobamos por consola que tenemos instalado el nodeJS correctamente

 Símbolo del sistema

```
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.
```

```
C:\Users\dell>npm -v
6.14.9
```

```
C:\Users\dell>
```

Ahora instalamos typeScript a nivel global

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\dell>npm -v
6.14.9

C:\Users\dell>npm i -g typescript
C:\Users\dell\AppData\Roaming\npm\tsc -> C:\Users\dell\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\dell\AppData\Roaming\npm\tsserver -> C:\Users\dell\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@4.5.2
added 1 package from 1 contributor in 6.369s

C:\Users\dell>tsc -v
Version 4.5.2

C:\Users\dell>
```

Si quisiéramos compilarlos nos generaría un archivo js

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

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** index.ts - typescript - Visual Studio Code [Administrador]
- Sidebar:** EXPLORADOR, EDITORES ABIERTOS (with index.ts selected), and TYPESCRIPT (with index.js and index.ts).
- Editor:** The main editor window contains the code:

```
1 console.log("hola como vamos");
```
- Terminal:** An open terminal window shows the command-line process:

```
C:\Users\dell>cd Desktop
C:\Users\dell\Desktop>cd typescript
C:\Users\dell\Desktop\typescript>tsc index.ts
C:\Users\dell\Desktop\typescript>_
```

Finalmente compilamos el js para obtener el resultado

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

- Menú superior:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Título:** index.ts - typescript - Visual Studio Code [Administrador]
- Área de trabajo:**
 - EXPLORADOR:** Muestra carpetas y archivos.
 - EDITORES ABIERTOS:** Muestra "index.ts" (TS).
 - TYPESCRIPT:** Muestra "index.js" (JS) y "index.ts" (TS).
- Terminal:** Muestra la ejecución del comando "node index" que imprime "hola como vamos".

```
1  console.log("hola como vamos");
```

```
C:\Users\dell\Desktop\typescript>node index
hola como vamos

C:\Users\dell\Desktop\typescript>
```

Instalamos este componente para poder ejecutar los ts con un solo comando

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\dell>npm install -g ts-node
C:\Users\dell\AppData\Roaming\npm\ts-node -> C:\Users\dell\AppData\Roaming\npm\node_modules\ts-node\dist\bin.js
C:\Users\dell\AppData\Roaming\npm\ts-script -> C:\Users\dell\AppData\Roaming\npm\node_modules\ts-node\dist\bin-script-deprecated.js
C:\Users\dell\AppData\Roaming\npm\ts-node-script -> C:\Users\dell\AppData\Roaming\npm\node_modules\ts-node\dist\bin-script.js
C:\Users\dell\AppData\Roaming\npm\ts-node-transpile-only -> C:\Users\dell\AppData\Roaming\npm\node_modules\ts-node\dist\bin-transpile.js
C:\Users\dell\AppData\Roaming\npm\ts-node-cwd -> C:\Users\dell\AppData\Roaming\npm\node_modules\ts-node\dist\bin-cwd.js
npm WARN ts-node@10.4.0 requires a peer of @types/node@* but none is installed. You must install peer dependencies yourself.
npm WARN ts-node@10.4.0 requires a peer of typescript@>=2.7 but none is installed. You must install peer dependencies yourself.

+ ts-node@10.4.0
added 14 packages from 45 contributors in 6s

C:\Users\dell>
```

A tener en cuenta

- ▶ Dentro de visual studio utilizar npx en frente de los comandos

Preparamos nuestro entorno de prueba

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

- File Explorer (Left):** Shows the project structure with files: `index.js`, `index.ts`, `intro` (expanded), `app.ts`, and `index.html`. The `index.html` file is selected and highlighted with a red box.
- Code Editor (Right):** Displays the `index.html` file content. A red box highlights the `<script src="app.js"></script>` line.
- Top Bar:** Shows the menu bar with options like Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda, and the title index.html - typescript - Visual Studio Code [Administrador].
- Bottom Bar:** Shows tabs for PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, and TERMINAL.
- Terminal:** Shows the command prompt at the bottom: `PS C:\Users\dell\Desktop\typescript\intro>`.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="app.js"></script>
    <title>Document</title>
</head>
<body>
    <h1>TypeScript</h1>
</body>
</html>
```

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The top navigation bar includes Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, and Ayuda. The title bar displays app.ts - typescript - Vis. The left sidebar (Explorador) shows a file tree under EDITORES ABIERTOS / TYPESCRIPT / intro. The files listed are index.html, app.ts (which is selected and highlighted with a red box), index.js, and index.ts. The main editor pane shows the contents of app.ts:

```
1 console.log("Hola iniciamos con esto")
```

. The bottom navigation bar includes PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, and TERMINAL. The terminal pane at the bottom shows the command PS C:\Users\dell\Desktop\typescript\intro> followed by a command prompt icon.

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda app.ts - typescript - Visual Studio Code [Ad]

EXPLORADOR ...

EDITORES ABIERTOS

TYPESCRIPT

- intro
- app.js
- app.ts**
- index.html
- index.js
- index.ts

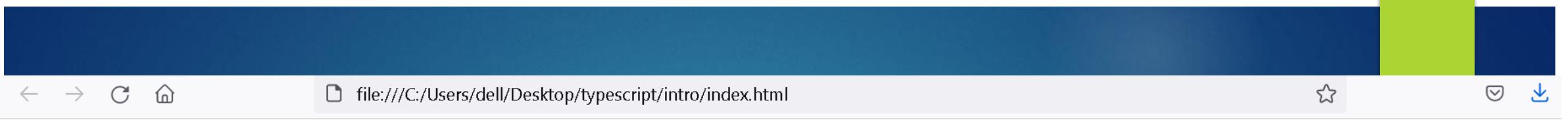
index.html TS app.ts X

intro > TS app.ts

1 `console.log("Hola iniciamos con esto")`

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
PS C:\Users\dell\Desktop\typescript\intro> npx tsc app.ts
PS C:\Users\dell\Desktop\typescript\intro> node app.js
Hola iniciamos con esto
PS C:\Users\dell\Desktop\typescript\intro>
```



A screenshot of the Chrome DevTools console tab. The tab bar at the top includes "Inspector", "Consola" (which is selected), "Depurador", "Red", "Editor de estilos", "Rendimiento", "Memoria", and three more tabs. Below the tab bar is a toolbar with icons for "Filtrar salida" (Filter output) and several status indicators: "Errores" (Errors), "Advertencias" (Warnings), "Registros" (Logs), "Información" (Information), and "Depurar" (Breakpoint). At the bottom of the console area, the output "Hola iniciamos con esto" is displayed, along with a file path "app.js:" and a double arrow icon.

Para no escribir doble comando al compilar con el watch actualizamos al instante los archivos ts y js

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda app.ts - typescript - Visual Studio Code [Administrador]

EXPLORADOR ...

> EDITORES ABIERTOS

✓ TYPESCRIPT

✓ intro

JS app.js

TS app.ts

▷ index.html

JS index.js

TS index.ts

TS app.ts X JS app.js

intro > TS app.ts

1 console.log("Vamos a ello")

JS app.js X

intro > JS app.js

1 console.log("Vamos a ello");

2

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

PS C:\Users\dell\Desktop\typescript\intro> npx tsc app -w

+ ▼ ^

powershell

powershell

powershell

Ver Ir Ejecutar Terminal Ayuda app.ts - typescript - Visual Studio Code [Administrador] — □ X

TS app.ts X JS app.js ...

intro > TS app.ts

```
1 console.log("TypeScript Mola bastante")
```

JS app.js X ...

intro > JS app.js

```
1 console.log("TypeScript Mola bastante");
```

2

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL + ▼ ^ X

[5:51:17] File change detected. Starting incremental compilation...

[5:51:17] Found 0 errors. Watching for file changes.

[]

[>] powershell
[>] powershell
[>] node

← → ⌛ ⌂ file:///C:/Users/dell/Desktop/typescript/intro/index.html ☆ ↴ ↴

TypeScript

Inspector Consola Depurador Red Editor de estilos Rendimiento Memoria ...

⌫ Filtrar salida Errores Advertencias Registros Información Depurar CSS XHR Peticiones :

TypeScript Mola bastante app.js:1

» (

Tipos de valores - Boolean

TS app.ts 1 X JS app.js

intro > TS app.ts > ...

```
4 let estadoCivil:boolean = false;
5 estadoCivil = true;
6
7 if(estadoCivil){
8     console.log("casado");
9 }else {
10     console.log("Soltero");
11 }
12
13
```

TypeScript

Inspector Consola Depurador ...

Filtrar salida

Errores Advertencias Registros Información Depurar CSS XHR Peticiones

casado app.js:6:1

»

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

[6:03:28] Found 0 errors. Watching for file changes.

node

Tipos de valores - number

The screenshot shows a Visual Studio Code interface with two files open:

- app.ts**:

```
1 let numero:number = 12.9;
2
3 let edad:number = 18;
4 if (edad >= 18) {
5     console.log("Es mayor de edad");
6 } else {
7     console.log("Es menor de edad");
8 }
9
10 edad = ObtenerEdad();
11 function ObtenerEdad(){
12     return 30;
13 }
```
- app.js**:

```
1 "use strict";
2 var numero = 12.9;
3 var edad = 18;
4 if (edad >= 18) {
5     console.log("Es mayor de edad");
6 } else {
7     console.log("Es menor de edad");
8 }
9
10 function ObtenerEdad() {
11     return 30;
12 }
```

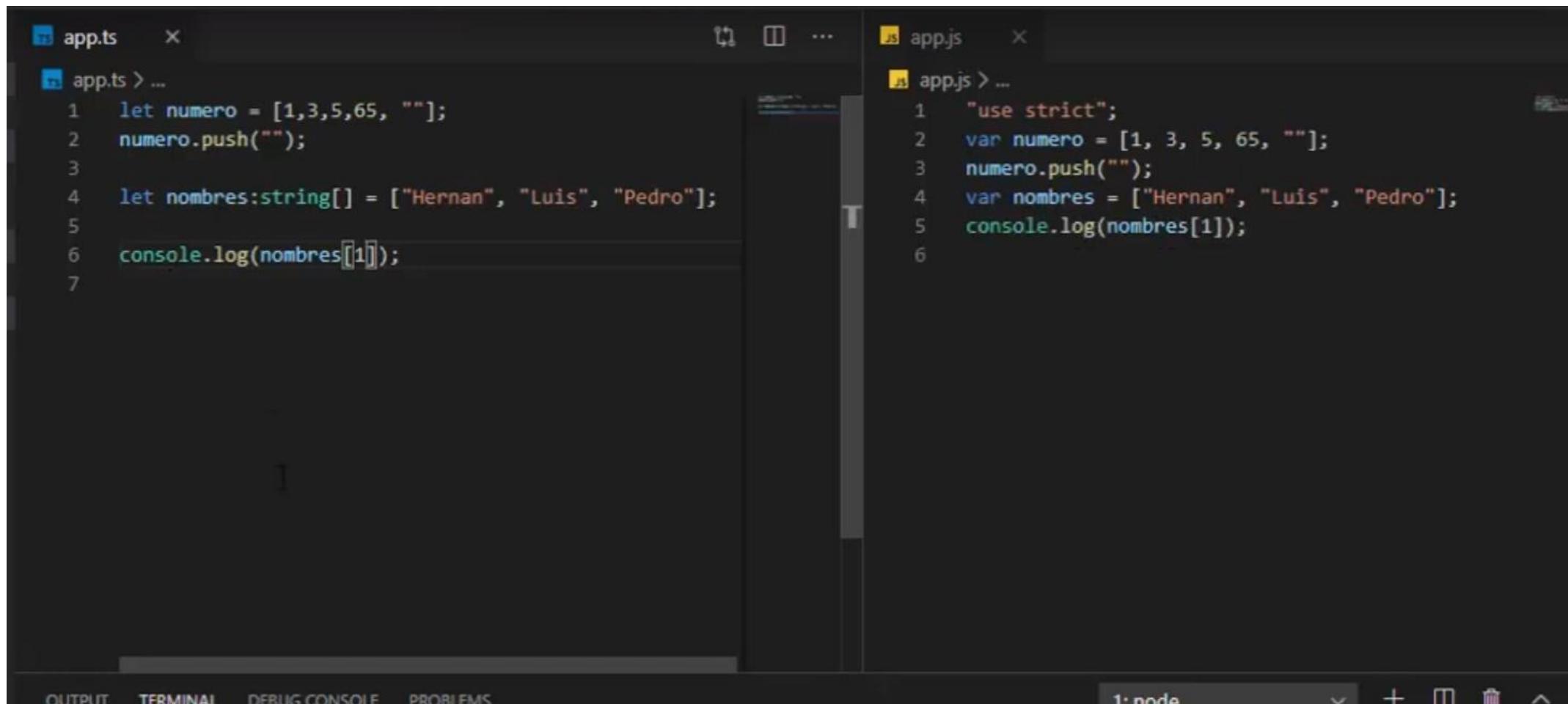
The status bar at the bottom indicates the terminal is set to run `node`.

Tipos de valores - String

```
app.ts    X
app.ts > ...
1 let nombre:string = "Douglas";
2 let apellido:string = "Narváez";
3
4 let resultado = nombre + " " + apellido;
5 console.log(resultado);
6
7 let concatenar:string = `${nombre} ${apellido}`;
8 console.log(concatenar.toUpperCase());
```

```
app.js    X
app.js > ...
1 "use strict";
2 var nombre = "Douglas";
3 var apellido = "Narváez";
4 var resultado = nombre + " " + apellido;
5 console.log(resultado);
6 var concatenar = nombre + " " + apellido;
7 console.log(concatenar.toUpperCase());
```

Tipos de valores - Array



The image shows a screenshot of the Visual Studio Code (VS Code) interface. It features two side-by-side code editors.

Left Editor (app.ts):

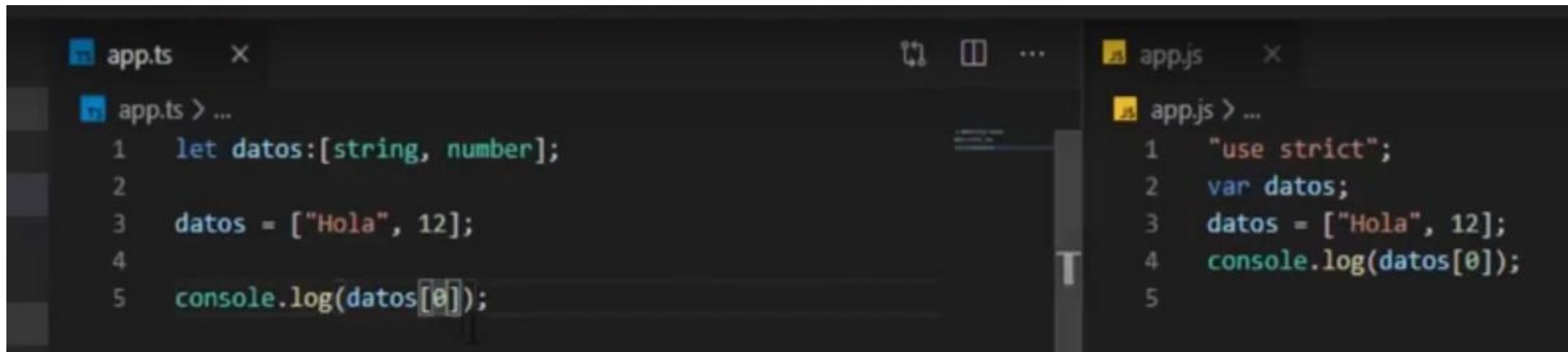
```
app.ts    X
app.ts > ...
1 let numero = [1,3,5,65, ""];
2 numero.push("");
3
4 let nombres:string[] = ["Hernan", "Luis", "Pedro"];
5
6 console.log(nombres[1]);
7
```

Right Editor (app.js):

```
app.js    X
app.js > ...
1 "use strict";
2 var numero = [1, 3, 5, 65, ""];
3 numero.push("");
4 var nombres = ["Hernan", "Luis", "Pedro"];
5 console.log(nombres[1]);
6
```

Both files contain code that declares arrays and performs operations like pushing values onto them. The left file uses TypeScript syntax, while the right file uses JavaScript syntax. The VS Code interface includes a top bar with file tabs, a toolbar with icons, and a bottom navigation bar with tabs for OUTPUT, TERMINAL, DEBUG CONSOLE, and PROBLEMS.

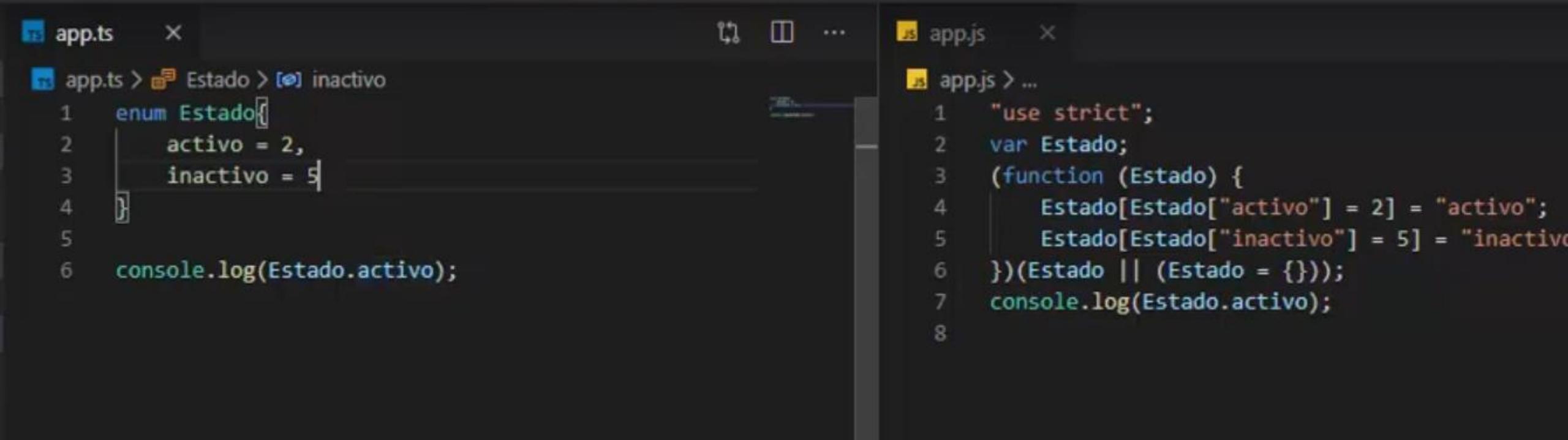
Tipos de valores - Array



```
app.ts      x
app.ts > ...
1  let datos:[string, number];
2
3  datos = ["Hola", 12];
4
5  console.log(datos[0]);
```

```
app.js      x
app.js > ...
1  "use strict";
2  var datos;
3  datos = ["Hola", 12];
4  console.log(datos[0]);
5
```

Enums



The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left, there is a TypeScript file named 'app.ts' with the following code:

```
ts app.ts    X
ts app.ts > Estado > inactivo
1 enum Estado{
2     activo = 2,
3     inactivo = 5
4 }
5
6 console.log(Estado.activo);
```

On the right, there is a JavaScript file named 'app.js' with the following code:

```
js app.js    X
js app.js > ...
1 "use strict";
2 var Estado;
3 (function (Estado) {
4     Estado[Estado["activo"] = 2] = "activo";
5     Estado[Estado["inactivo"] = 5] = "inactivo";
6 })(Estado || (Estado = {}));
7 console.log(Estado.activo);
8
```

The code illustrates the transpilation of a TypeScript enum into a JavaScript object. The enum 'Estado' is converted into a variable 'Estado' containing a function that creates an object with properties 'activo' and 'inactivo'.

Any

```
app.ts
app.ts > ...
1 let cualquierValor;
2
3 cualquierValor = "Hola";
4 console.log(cualquierValor.length);
5
6 cualquierValor = false;
7 console.log(cualquierValor);
8
9 cualquierValor = 12;

app.js
app.js > ...
1 "use strict";
2 var cualquierValor;
3 cualquierValor = "Hola";
4 console.log(cualquierValor.length);
5 cualquierValor = false;
6 console.log(cualquierValor);
7
```

Void

Terminal Help app.ts - TiposBásico - Visual Studio Code

app.ts x app.js x

app.ts > ... app.js > ...

```
function Saludar():void{
    console.log("Hola");
}

let valor = Saludar();
console.log(valor);
```

```
"use strict";
function Saludar() {
    console.log("Hola");
}

let valor = Saludar();
console.log(valor);
```

Null y Undefined

The screenshot shows a Visual Studio Code interface with two open files: `app.ts` and `app.js`. The title bar indicates the project is named `TiposBásico`.

app.ts:

```
1 let monto:number = null;
2 let miVariable;
3 console.log(monto);
4 console.log(miVariable);
5
6 console.log(typeof monto);
7 console.log(typeof miVariable);
```

app.js:

```
1 var monto = null;
2 var miVariable;
3 console.log(monto);
4 console.log(miVariable);
5 console.log(typeof monto);
6 console.log(typeof miVariable);
7
```

Type assertions

The image shows a code editor interface with two tabs: `app.ts` and `app.js`.

app.ts:

```
app.ts > [e] resultado2
1 let valor:any = "Hola soy una cadena";
2
3 let resultado:number = (<string> valor).length;
4 let resultado2:number = (valor as string).length;
```

app.js:

```
app.js > ...
1 "use strict";
2 var valor = "Hola soy una cadena";
3 var resultado = valor.length;
4 var resultado2 = valor.length;
5
```

Const

The image shows a code editor interface with two tabs: "app.ts" and "app.js".

app.ts

```
1  const ESTADO:Boolean = false;
2
3  if (true) {
4      const ESTADO = true;
5  } else {
6
7  }
8
9  for (const iterator of [1,2,3,4,5,6,7]) {
10     console.log(iterator);
11 }
12
```

app.js

```
1  "use strict";
2  const ESTADO = false;
3  if (true) {
4      const ESTADO = true;
5  } else {
6  }
7
8  for (const iterator of [1, 2, 3, 4, 5, 6, 7]) {
9      console.log(iterator);
10 }
11
```

Array destructuring

```
app.ts      X  
app > app.ts > ...  
1  let frutas:string[] = ["Manzana", "Uva", "Piña"];  
2  
3  let [item1, item2, item3] = frutas;  
4  console.log(item1);  
5  console.log(item2);  
6  console.log(item3);  
7
```

Tuple destructuring

```
app.ts  ✘  
app > app.ts > ...  
1  
2 let persona: [string[], number, string, string[]];  
3 persona = [["Luis", "Narvaéz"], 24, "masculino", ["musica", "lectura"]];  
4  
5 let [nombre, edad, genero, interese] = persona;  
6 console.log(nombre);  
7 console.log(edad);  
8
```

Object destructuring

The image shows a code editor interface with two files: `app.ts` and `app.js`. The `app.ts` file contains TypeScript code, and the `app.js` file contains the corresponding JavaScript output after compilation.

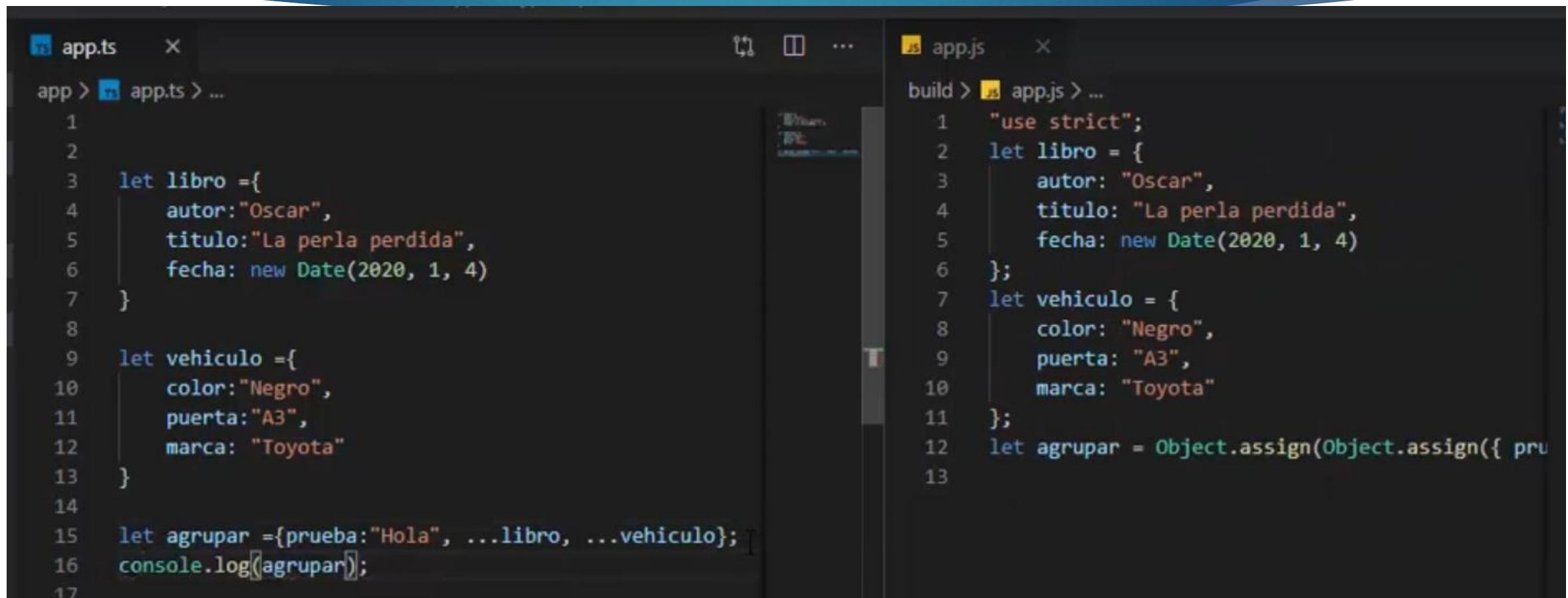
app.ts

```
1 let persona = {  
2     nombre:["Luis", "Narvaéz"],  
3     edad:34,  
4     genero:"masculino",  
5     intereses:["musica", "lectura"]  
6 }  
7     let nombre: string[]  
8 let {nombre, edad, genero, intereses} = persona;  
9 console.log(nombre);  
10 console.log(genero);  
11  
12
```

app.js

```
1 "use strict";  
2 let persona = {  
3     nombre: ["Luis", "Narvaéz"],  
4     edad: 34,  
5     genero: "masculino",  
6     intereses: ["musica", "lectura"]  
7 };  
8 let { nombre, edad, genero, intereses } = persona;  
9 console.log(nombre);  
10 console.log(genero);  
11
```

Spread Operador de propagación



```
app.ts      x
app > app.ts > ...
1
2
3 let libro ={
4   autor:"Oscar",
5   titulo:"La perla perdida",
6   fecha: new Date(2020, 1, 4)
7 }
8
9 let vehiculo ={
10   color:"Negro",
11   puerta:"A3",
12   marca: "Toyota"
13 }
14
15 let agrupar ={prueba:"Hola", ...libro, ...vehiculo};
16 console.log(agrupar);
17

app.js      x
build > app.js > ...
1 "use strict";
2 let libro ={
3   autor: "Oscar",
4   titulo: "La perla perdida",
5   fecha: new Date(2020, 1, 4)
6 };
7 let vehiculo = {
8   color: "Negro",
9   puerta: "A3",
10  marca: "Toyota"
11 };
12 let agrupar = Object.assign(Object.assign({ pru
```

Default Values

The image shows a code editor with two files side-by-side:

app.ts

```
1  function ObtenerValores(objeto: { valor1: string,
2                                 valor2?: number }) {
3     let { valor1, valor2 } = objeto;
4     console.log(valor1);
5     console.log(valor2);
6 }
7
8 ObtenerValores({valor1:"Camisa volcom"});
9
10 function Saludar(texto:string = "Hola2"){
11     console.log(texto);
12 }
13
14
15 Saludar("Hola");
16
```

app.js

```
1  function ObtenerValores(objeto) {
2     let { valor1, valor2 } = objeto;
3     console.log(valor1);
4     console.log(valor2);
5 }
6
7 ObtenerValores({ valor1: "Camisa volcom" });
8
9 function Saludar(texto = "Hola2") {
10     console.log(texto);
11 }
12
13
14
15 Saludar("Hola");
16
```

Clases

Terminal Help

• app.ts - Typescript - Visual Studio Code

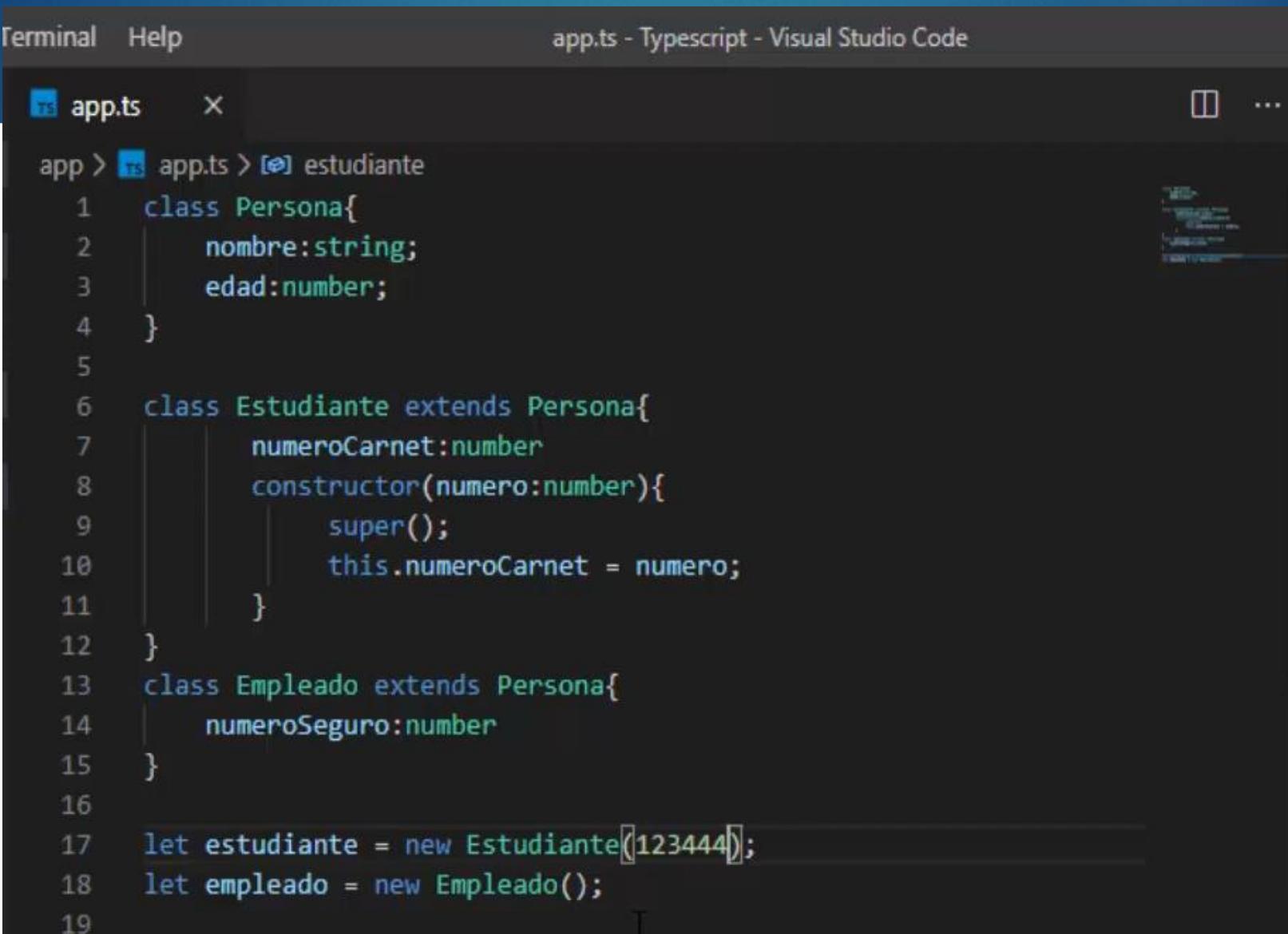
app.ts

```
app > app.ts > Persona > constructor
1  class Persona{
2      nombre:string = "Janeth";
3      apellido:string;
4      edad:number;
5      fechaNacimiento:Date;
6
7      constructor(nombre:string){
8          this.nombre = nombre;
9      }
10 }
11
12 let persona = new Persona("Luis");
13 console.log(persona);
```

app.js

```
build > app.js > ...
1  "use strict";
2  class Persona {
3      constructor(nombre) {
4          this.nombre = "Janeth";
5          this.nombre = nombre;
6      }
7  }
8  let persona = new Persona("Luis");
9  console.log(persona);
10
```

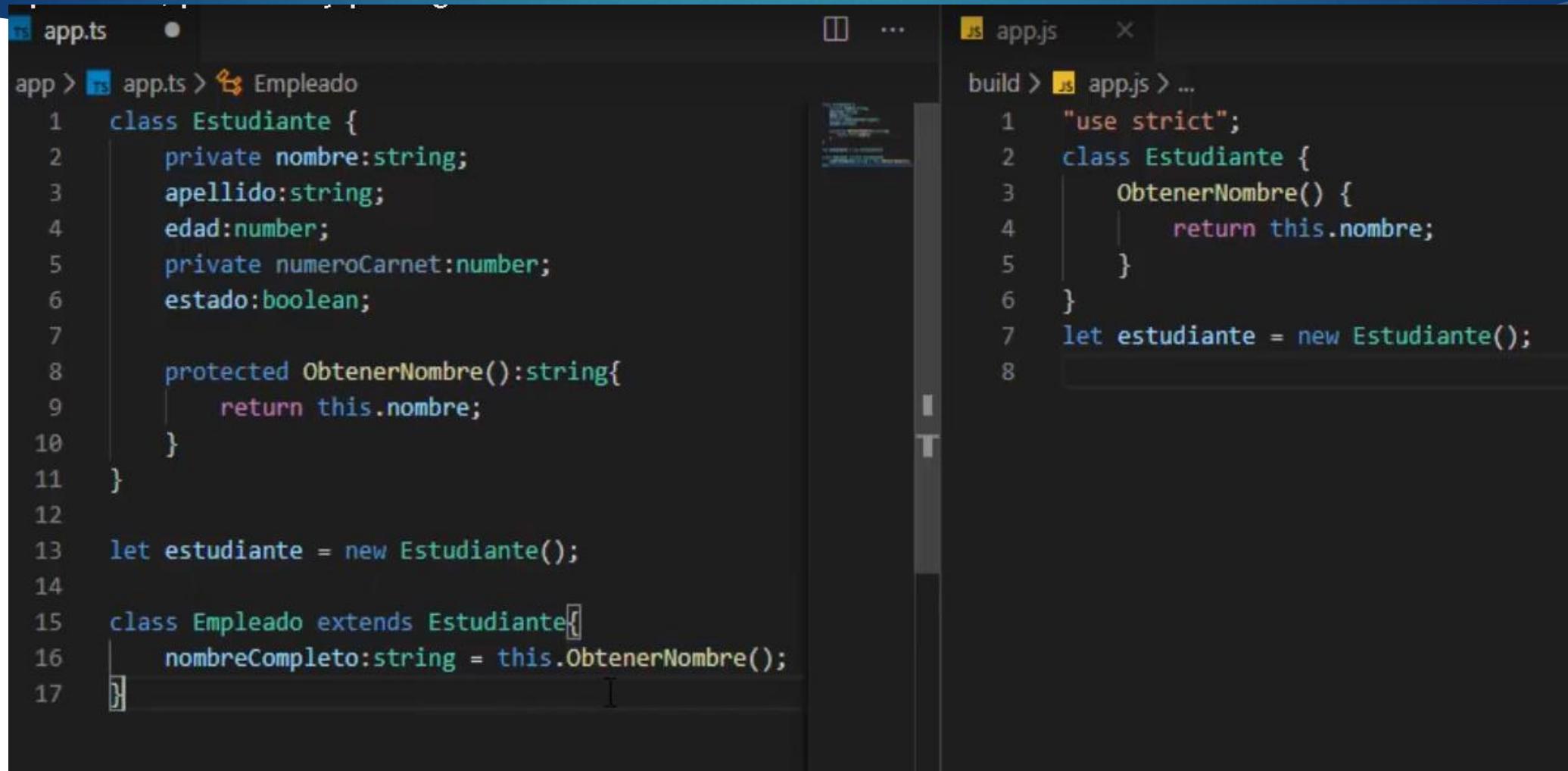
herencia



The screenshot shows a Visual Studio Code interface with a dark theme. The title bar reads "app.ts - Typescript - Visual Studio Code". The left sidebar shows a file tree with "app > app.ts > estudiante". The main editor area contains the following TypeScript code:

```
1  class Persona{
2      nombre:string;
3      edad:number;
4  }
5
6  class Estudiante extends Persona{
7      numeroCarnet:number
8      constructor(numero:number){
9          super();
10         this.numeroCarnet = numero;
11     }
12 }
13 class Empleado extends Persona{
14     numeroSeguro:number
15 }
16
17 let estudiante = new Estudiante(123444);
18 let empleado = new Empleado();
```

Modificadores de Acceso



The image shows a code editor interface with two files: `app.ts` and `app.js`.

app.ts:

```
app > app.ts > Empleado
1  class Estudiante {
2    private nombre:string;
3    apellido:string;
4    edad:number;
5    private numeroCarnet:number;
6    estado:boolean;
7
8    protected ObtenerNombre():string{
9      return this.nombre;
10 }
11 }
12
13 let estudiante = new Estudiante();
14
15 class Empleado extends Estudiante{
16   nombreCompleto:string = this.ObtenerNombre();
17 }
```

app.js:

```
build > app.js > ...
1 "use strict";
2 class Estudiante {
3   ObtenerNombre() {
4     return this.nombre;
5   }
6 }
7 let estudiante = new Estudiante();
8
```

Modificador Readonly

The image shows a code editor interface with two files: `app.ts` and `app.js`.

app.ts:

```
1 class Libro{  
2     private readonly autor:string;  
3     titulo:string;  
4     fechaPublicacion:Date;  
5 }  
6  
7 let libro = new Libro();  
8 libro.autor = "Pablo";
```

A tooltip is displayed over the line `libro.autor = "Pablo";`, showing the properties of the `Libro` object: `fechaPublicacion` and `titulo`. The `titulo` property is highlighted.

app.js:

```
1 "use strict";  
2 class Libro {  
3 }  
4 let libro = new Libro();  
5 libro.autor = "Pablo";  
6
```

GET y SET

app.ts

```
1  class Cliente {  
2      private _nombre : string;  
3      public get nombre(): string {  
4          if (this._nombre) {  
5              return this._nombre;  
6          } else {  
7              return this._nombre;  
8          }  
9      }  
10     public set nombre(v: string) {  
11         this._nombre = v;  
12     }  
13  
14     constructor(nombre?:string) {  
15         this._nombre = nombre;  
16     }  
17  
18 }  
19  
20 let cliente = new Cliente();  
21 console.log(cliente.nombre);
```

app.js

```
1  class Cliente {  
2      constructor(nombre) {  
3          this._nombre = nombre;  
4      }  
5      get nombre() {  
6          return this._nombre;  
7      }  
8      set nombre(v) {  
9          this._nombre = v;  
10     }  
11 }  
12 let cliente = new Cliente();  
13 console.log(cliente);  
14
```

Static

```
app > app.ts > valor
1  class Utilidad{
2      static pi:number = 3.1416;
3
4      static CalcularAreaDeCirculo(radio:number):number{
5          return radio * radio * this.pi;
6      }
7  }
8
9  let valor:number = Utilidad.CalcularAreaDeCirculo(23);
10
11
12
```

Clase Abstracta

```
app > app.ts > Perro > Ruido
1 abstract class Animal{
2     abstract Ruido():void;
3 }
4
5 class Gato extends Animal {
6     Ruido(){
7         console.log("miau");
8     }
9 }
10
11 class Perro extends Animal {
12     Ruido(){
13         console.log("wau");
14     }
15 }
16
17
```

```
build > app.js > ...
1 "use strict";
2 class Animal {
3 }
4 class Gato extends Animal {
5     Ruido() {
6         console.log("miau");
7     }
8 }
9 class Perro extends Animal {
10    Ruido() {
11        console.log("wau");
12    }
13 }
14
```

Interfaces

```
ts app.ts •  
app > ts app.ts > ...  
1  interface IMatematica{  
2    total?:number,  
3    Sumar(a:number, b:number):number,  
4    Restar(a:number, b:number):number  
5  }  
6  
7  class Utilidad implements IMatematica {  
8    Sumar(a:number, b:number):number{  
9      return a + b;  
10   }  
11   Restar(a:number, b:number):number{  
12     return a - b;  
13   }  
14 }  
15  
16  
17  app > ts app.ts > ...  
18    a - b;  
19  }  
20  }  
21  }  
22  }  
23  let figura2 = {} as IFigura2;  
24  figura2.color = "BLANCO";|  
25  
26  
27
```

Decoradores de clases

app > app.ts > DecoradorClase

```
1  function DecoradorClase(target:Function){  
2      target.prototype.Saludar = function(){  
3          console.log("Hola");  
4      }  
5  }  
6  @DecoradorClase  
7  class Persona {  
8      constructor() {  
9      }  
10 }  
11 }  
12  
13 let persona = new Persona();  
14 persona.Saludar();  
15 }
```

build > app.js > ...

```
1  var __decorate = (this &  
2  var c = arguments.le  
3  if (typeof Reflect =  
4  else for (var i = de  
5  return c > 3 && r &&  
6  };  
7  function DecoradorClase(  
8      target.prototype.Sal  
9          console.log("Hol  
10     );  
11 }  
12 let Persona = class Pers  
13     constructor() {  
14     }  
15 };  
16 Persona = __decorate([  
17     DecoradorClase  
18 ], Persona);  
19 }
```

Angular

- ▶ Angular es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página.
- ▶ <https://angular.io/>

Características

- ▶ Aplicaciones web responsivas
- ▶ SPA (Single Page Application)
- ▶ Reactivo RxJs
- ▶ Asíncrono

Angular se compone de

- ▶ Componentes
- ▶ Rutas
- ▶ Directivas
- ▶ Servicios
- ▶ Módulos

Componentes

- ▶ Básicamente se compone de un archivo HTML y una clase Typescript

Servicios

- Son lugares centralizados de información

Directivas

- ▶ Directivas de componentes, tiene un pedazo de código html reutilizable
- ▶ Directivas Estructurales, modifica el DOM o html ya sea añadiendo/removiendo elementos
- ▶ Directivas de atributos, cambia la apariencia o el comportamiento de un elemento o otra directiva

Rutas

- ▶ Nos ayuda a mostrar diferentes componentes basados en el URL del navegador web

Módulos

- ▶ Nos permite agrupar componentes, rutas , directivas y servicios

Angular contiene

- ▶ **Componentes**
- ▶ **Plantillas**
- ▶ **Directivas**
- ▶ **Eventos**
- ▶ **Pipe**
- ▶ **Rutas de navegación**
- ▶ **Decoradores**
- ▶ **Servicios y HttpClient**
- ▶ **Clases TypeScript**
- ▶ **Formulario**
- ▶ **Data Binding ngModel**
- ▶ **Validaciones**
- ▶ **Módulos**
- ▶ **Angular Material**

Versions

← → ⌂ 🔒 angular.io/guide/releases

HELP ANGULAR BY TAKING A 1 MINUTE SURVEY! [GO TO SURVEY](#) X

≡  ANGULAR FEATURES DOCS RESOURCES EVENTS BLOG Search | [!\[\]\(cc54820bc5d5e0b6da15383f035162b9_img.jpg\)](#) [!\[\]\(098e14923e9f68dd779599c3137afb08_img.jpg\)](#) [!\[\]\(8faf47828cab43f13b95bc17202f7471_img.jpg\)](#)

Introduction

Getting Started

Understanding Angular

Developer guides

Best Practices

Angular Tools

Tutorials

Release Information

Release Practices

Roadmap

patches are released.

The following table provides the status for Angular versions under support.

VERSION	STATUS	RELEASED	ACTIVE ENDS	LTS ENDS
^13.0.0	Active	Nov 04, 2021	May 04, 2022	May 04, 2023
^12.0.0	LTS	May 12, 2021	Nov 12, 2021	Nov 12, 2022
^11.0.0	LTS	Nov 11, 2020	May 11, 2021	May 11, 2022
^10.0.0	LTS	Jun 24, 2020	Dec 24, 2020	Dec 24, 2021

Angular versions v4, v5, v6, v7, v8, and v9 are no longer under support.

Angular versioning and releases

Angular versioning

Supported update paths

Preview releases

Release frequency

• [Support policy and schedule](#)

LTS fixes

Deprecation practices

Public API surface

Angular Labs

Lenguaje utilizado TypeScript

- ▶ Tipos basados en clases e interfaces
- ▶ Tipos de datos mas fuerte
- ▶ Atributos métodos y constructores
- ▶ Decoradores y mas

Herramientas necesarias

- ▶ Instalar Nodejs
- ▶ Instalar TypeScript
 - npm i -g typescript**
 - tsc -v**
- ▶ Instalar angular CLI

Si queremos instalar una versión específica

The screenshot shows a web browser displaying the Angular v11 guide at v11.angular.io/guide/setup-local. The page title is "Install the Angular CLI". The left sidebar lists various sections under "Getting Started": "What is Angular?", "Try it", "Setup", "Understanding Angular", "Developer Guides", "Best Practices", "Angular Tools", "Tutorials", "Release Information", and "Reference". The main content area starts with a heading "Install the Angular CLI". It explains that the CLI is used for creating projects, generating application and library code, and performing development tasks like testing, bundling, and deployment. It provides the command `npm install -g @angular/cli@11` for installing the CLI. Below this, there's a section titled "Create a workspace and initial application" with a sub-section about developing in a workspace.

v11.angular.io/guide/setup-local

Angular

FEATURES DOCS RESOURCES EVENTS BLOG

Search

Introduction

Getting Started

What is Angular?

Try it

Setup

Understanding Angular

Developer Guides

Best Practices

Angular Tools

Tutorials

Release Information

Reference

Install the Angular CLI

You use the Angular CLI to create projects, generate application and library code, and perform a variety of ongoing development tasks such as testing, bundling, and deployment.

To install the Angular CLI, open a terminal window and run the following command:

```
npm install -g @angular/cli@11
```

Create a workspace and initial application

You develop apps in the context of an Angular [workspace](#).

To create a new workspace and initial starter app:

Setting up the local environment and workspace

Prerequisites

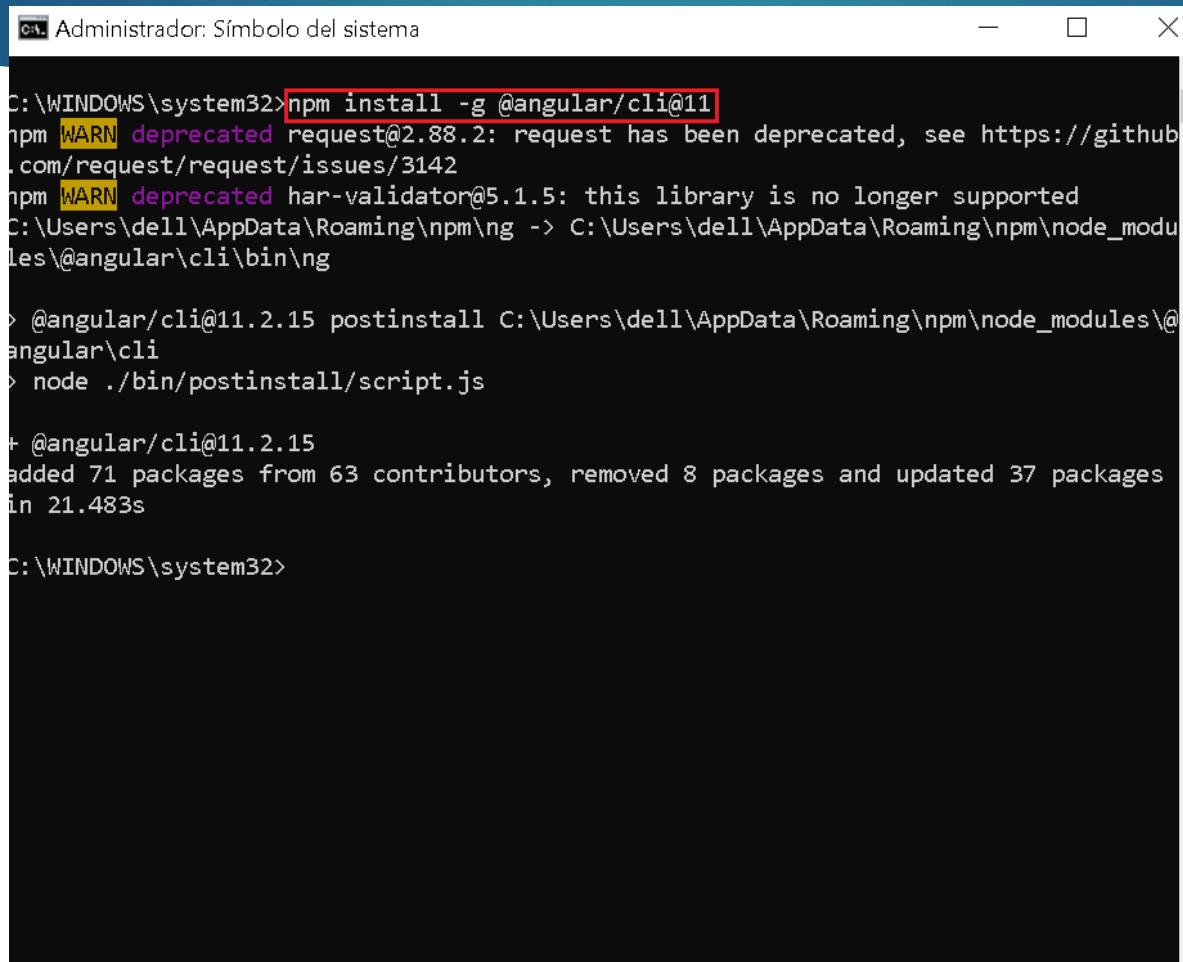
- Install the Angular CLI

Create a workspace and initial application

Run the application

Next steps

En la consola cmd como administrador



```
C:\WINDOWS\system32>npm install -g @angular/cli@11
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
C:\Users\dell\AppData\Roaming\npm\ng -> C:\Users\dell\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng

> @angular/cli@11.2.15 postinstall C:\Users\dell\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js

+ @angular/cli@11.2.15
added 71 packages from 63 contributors, removed 8 packages and updated 37 packages in 21.483s

C:\WINDOWS\system32>
```

Comprobamos que fue exitosa la instalación

```
C:\WINDOWS\system32>ng --version

Angular CLI: 11.2.15
Node: 16.13.0
OS: win32 x64

Angular:
...
Ivy Workspace:

Package          Version
-----
@angular-devkit/architect    0.1102.15 (cli-only)
@angular-devkit/core         11.2.15 (cli-only)
@angular-devkit/schematics   11.2.15 (cli-only)
@schematics/angular          11.2.15 (cli-only)
@schematics/update           0.1102.15 (cli-only)
```

Instalamos extensiones necesarias para Visual Studio Code

EXTENSIONES: MARKETPLACE

angular esse

- Angular Essentials (Version 12)** ⚡ 749K ★ 4.5
Essential extensions for Angular developers
John Papa [Instalar](#)
- Angular Snippets (Version 12)** ⚡ 2.8M ★ 5
Angular version 12 snippets by John Papa
John Papa [Instalar](#)
- Angular Language Service** ⚡ 2.7M ★ 3.5
Editor services for Angular templates
Angular [Instalar](#)
- Angular 10 Snippets - TypeScript, H...** ⚡ 1.9M ★ 4.5
242 Angular Snippets (TypeScript, Html, Angular Mat...
Mikael Morlund [Instalar](#)
- Angular Schematics** ⚡ 556K ★ 5
Angular schematics (CLI commands) from files Explore...
Cyrille Tuzi [Instalar](#)
- Angular Files** ⚡ 431K ★ 4
Quickly scaffold angular file templates
Alexander Ivanichev [Instalar](#)
- VSCode simpler Icons with Angular** ⚡ 293K ★ 4.5
A fork from vscode-great-icons. With fewer different i...
davidbabel [Instalar](#)
- Angular 8 and TypeScript/HTML VS ...** ⚡ 419K ★ 5
VS Code snippets for Angular and TypeScript/HTML
Dan Wahlin [Instalar](#)
- Angular 2 TypeScript Emmet** ⚡ 306K ★ 2.5
Support zen-coding syntax for Angular 2 typescript fil...

Extensión: Angular Essentials (Version 12) - Visual Studio Code [Administrador]

Angular Essentials (Version 12) v12.0.0

John Papa | ⚡ 749.460 | ★★★★★(20)

Essential extensions for Angular developers

[Instalar](#)

Detalles **Registro de cambios**

Paquete de extensión (12)

- ESLint** Integrates ESLint JavaScript into VS Code.
Microsoft [Instalar](#)
- Prettier - Code formatter** Code formatter using prettier
Prettier [Instalar](#)
- Material Icon Theme** Material Design Icons for Visual Studio Code
Philipp Kief [Instalar](#)
- [Deprecated] Debugger for Chrome**

Categorías

- Extension Packs

Recursos

- Marketplace
- Repositorio
- Licencia

Más información

- Publicado el 23/5/2017 5:06:
- Última actualización 27/5/2021 15:22:07
- Identificador johnpapa.angular_essentials

Angular Essentials - Extension Pack for VS Code

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

EXTENSIONES: MARKETPLACE

angular esse

- Angular Essentials (Version 12)** ⚡ 749K ★ 4.5
Essential extensions for Angular developers
John Papa [Instalar](#)
- Angular Snippets (Version 12)** ⚡ 2.8M ★ 5
Angular version 12 snippets by John Papa
John Papa [Instalar](#)
- Angular Language Service** ⚡ 2.7M ★ 3.5
Editor services for Angular templates
Angular [Instalar](#)
- Angular 10 Snippets - TypeScript, H...** ⚡ 1.9M ★ 4.5
242 Angular Snippets (TypeScript, Html, Angular Mat...
Mikael Morlund [Instalar](#)
- Angular Schematics** ⚡ 556K ★ 5
Angular schematics (CLI commands) from files Explore...
Cyrille Tuzi [Instalar](#)
- Angular Files** ⚡ 431K ★ 4
Quickly scaffold angular file templates
Alexander Ivanichev [Instalar](#)
- VSCode simpler Icons with Angular** ⚡ 293K ★ 4.5
A fork from vscode-great-icons. With fewer different i...
davidbabel [Instalar](#)
- Angular 8 and TypeScript/HTML VS ...** ⚡ 419K ★ 5
VS Code snippets for Angular and TypeScript/HTML
Dan Wahlin [Instalar](#)
- Angular 2 TypeScript Emmet** ⚡ 306K ★ 2.5
Support zen-coding syntax for Angular 2 typescript fil...
jokethashbi [Instalar](#)

Extensión: Angular Snippets (Version 12) - Visual Studio Code [Administrador]

Extensión: Angular Snippets (Version 12) X

Angular Snippets (Version 12) v12.0.0

John Papa | ⚡ 2.873.304 | ★★★★★(95)

Angular version 12 snippets by John Papa

[Instalar](#)

Detalles **Contribuciones de características** **Registro de cambios**

```
export class ThingsService {  
    constructor(private httpClient: HttpClient) { }  
  
    getThings(): Observable<Thing[]> {  
        return this.httpClient.get<Thing[]>(this.url);  
    }  
}
```

Categorías

Snippets

Recursos

Marketplace

Repositorio

Licencia

Más información

Publicado el 13/11/2015

Última actualización 27/5/2021

Identificador johnpapa.angular-snippets

See the [CHANGELOG](#) for the latest changes

Angular Essentials

Check out the [Angular Essentials extension](#) for more great extensions for developing with JavaScript and Angular.

EXTENSIONES: MARKETPLACE

angular esse

- Angular Essentials (Version 12)** ⚡ 749K ★ 4.5
Essential extensions for Angular developers
John Papa [Instalar](#)
- Angular Snippets (Version 12)** ⚡ 2.8M ★ 5
Angular version 12 snippets by John Papa
John Papa [Instalar](#)
- Angular Language Service** ⚡ 2.7M ★ 3.5
Editor services for Angular templates
Angular [Instalar](#)
- Angular 10 Snippets - TypeScript, H...** ⚡ 1.9M ★ 4.5
242 Angular Snippets (TypeScript, Html, Angular Mat...
Mikael Morlund [Instalar](#)
- Angular Schematics** ⚡ 556K ★ 5
Angular schematics (CLI commands) from files Explore...
Cyrille Tuzi [Instalar](#)
- Angular Files** ⚡ 431K ★ 4
Quickly scaffold angular file templates
Alexander Ivanichev [Instalar](#)
- VSCode simpler Icons with Angular** ⚡ 293K ★ 4.5
A fork from vscode-great-icons. With fewer different i...
davidbabel [Instalar](#)
- Angular 8 and TypeScript/HTML VS ...** ⚡ 419K ★ 5
VS Code snippets for Angular and TypeScript/HTML
Dan Wahlin [Instalar](#)
- Angular 2 TypeScript Emmet** ⚡ 306K ★ 2.5
Support zen-coding syntax for Angular 2 typescript fil...
Emmet [Instalar](#)

Extensión: Angular Language Service X

Angular Language Service v13.0.0

Angular ⚡ 2.786.784 ★★★★☆(197)
Editor services for Angular templates

[Instalar](#)

[Detalles](#) [Contribuciones de características](#) [Registro de cambios](#)

Angular Language Service

app.component.html

```
1 <h1>{{title}}
```

title number

Categorías

- Programming Languages

Recursos

Marketplace Repository

Más información

Publicado el 4/4/2017 0:31:2
Última actualización 3/11/2021 21:57:12
Identificador angular.ng-template



EXTENSIONES: MARKETPLACE



material icon the

Material Icon Theme

Material Design Icons for Visual Studio Code

Philipp Kief

10M ★ 5



Material Icon Theme

v4.11.0

Philipp Kief

10.090.799

★★★★★(226)

Material Design Icons for Visual Studio Code

Instalar



Material Theme Icons

Material Theme Icons, the most epic icons theme for V...

Equinusocio

646K ★ 5



Community Material Theme

The official community maintained Material Theme wi...

Equinusocio

675K ★ 4.5



Material Theme

The most epic theme now for Visual Studio Code

Equinusocio

582K ★ 5



Icon Fonts

Snippets for popular icon fonts such as Font Awesom...

idleberg

296K ★ 5



TabNine - Code Faster with the All-L...

JavaScript, Python, Java, Typescript & all other l...

TabNine

2.9M ★ 4.5



Sublime Material Theme

Port of the Material Theme for Sublime Text.

Jarvis Prestige

475K ★ 4.5



Nomo Dark Icon Theme

Nomo Dark Icon Theme

be5invis

283K ★ 5



Dark+ Material

The default Dark+ Theme with material palette and el...

324K ★ 4.5



Extensión: Material Icon Theme

Material Icon Theme

v4.11.0

Philipp Kief

10.090.799

★★★★★(226)

Material Design Icons for Visual Studio Code

Instalar



Detalles

Contribuciones de características

Registro de cambios

Categorías

Themes

Recursos

Marketplace

Repositorio

Licencia

Más información

Publicado el 10/9/2016

22:28:56

Última actualización 25/11/2021

actualización 21:32:59

Identificador pkief.material-icon-theme

icon-theme

Material Icon Theme



EXTENSIONES: MARKETPLACE

Y ⌂ ⌄ ...

html su

- HTML CSS Support**  v1.10.2
CSS Intellisense for HTML
ecmel | ⚡ 8.1M ★ 3.5
[Instalar](#)
- HTML Snippets** 
Full HTML tags including HTML5 Snippets
Mohamed Abusaid | ⚡ 8.100.126 ★ 4.5
[Instalar](#)
- IntelliSense for CSS class names in H...** 
CSS class name completion for the HTML class attribu...
Zignd | ⚡ 4M ★ 3.5
[Instalar](#)
- JS-CSS-HTML Formatter** 
Format ,prettyfy and beautify JS, CSS, HTML code by u...
lonely | ⚡ 2.7M ★ 2
[Instalar](#)
- Kite AutoComplete AI Code: Python...** 
AI code completions for all languages, intellisense, co...
Kite | ⚡ 3.1M ★ 3.5
[Instalar](#)
- HTML Preview** 
Provides ability to preview HTML documents.
Thomas Haakon Townsend | ⚡ 1.4M ★ 3
[Instalar](#)
- HTML Boilerplate** 
A basic HTML5 boilerplate snippet generator.
sidthesloth | ⚡ 1.2M ★ 5
[Instalar](#)
- TabNine - Code Faster with the All-L...** 
JavaScript, Python, Java, Typescript & all other l...
TabNine | ⚡ 2.9M ★ 4.5
[Instalar](#)
- HTML SCSS Support** 
HTML and SCSS support for VS Code
ecmel | ⚡ 160K ★ 4
[Instalar](#)

Extensión: HTML CSS Support X



HTML CSS Support v1.10.2

ecmel | ⚡ 8.100.126 ★ 4.5 (84)

CSS Intellisense for HTML

[Instalar](#)



[Detalles](#) [Contribuciones de características](#) [Registro de cambios](#)

Visual Studio Code CSS Intellisense for HTML

HTML **id** and **class** attribute completion for Visual Studio Code.

Features

- HTML **id** and **class** attribute completion.
- Supports linked and embedded style sheets.
- Supports template inheritance.
- Supports additional style sheets.
- Supports other HTML like languages.
- Validates CSS selectors on demand.

Usage

Categorías

Programming Languages

Recursos

Marketplace

Repositorio

Licencia

Más información

Publicado el 14/7/2016

23:02:33

Última actualización 10/2/2021

actualización 16:07:29

Identificado ecmel.vscode-html-css

Extensión: HTML Snippets - Visual Studio Code [Administrador]

EXTENSIONES: MARKETPLACE

html su

HTML CSS Support by ecmel (95ms)
CSS Intellisense for HTML

HTML Snippets by Mohamed Abusaid (highlighted with a red box)
Full HTML tags including HTML5 Snippets

IntelliSense for CSS class names in H... by Zignd (4M)
CSS class name completion for the HTML class attribu...

JS-CSS-HTML Formatter by lonefy (2.7M)
Format ,prettyify and beautify JS, CSS, HTML code by u...

Kite AutoComplete AI Code: Python... by Kite (3.1M)
AI code completions for all languages, intellisense, co...

HTML Preview by Thomas Haakon Townsend (1.4M)
Provides ability to preview HTML documents.

HTML Boilerplate by sidthesloth (1.2M)
A basic HTML5 boilerplate snippet generator.

Tabnine - Code Faster with the All-L... by TabNine (2.9M)
JavaScript, Python, Java, Typescript & all other l...

HTML SCSS Support by SCSS + HTML (160K)
SCSS support for HTML documents

HTML Snippets v0.2.1 by Mohamed Abusaid (6.490.175)
Full HTML tags including HTML5 Snippets

Deshabilitar Desinstalar

Esta extensión está habilitada globalmente.

Detalles Contribuciones de características Estado en tiempo de ejecución

README

Visual Studio Code HTML Snippets

DISABLING THIS EXTENSION AS IT CURRENTLY CONFLICTS WITH EXISTING VS CODE HTML EXTENSION

This extension adds rich language support for the HTML Markup to VS Code, including:

- Full HTML5 Tags
- Colorization
- Snippets
- [partially implemented] Quick Info
- description mentions if tag deprecated

Categorías
Snippets

Recursos

Marketplace
Licencia

Más información

Publicado el 24/12/2015
Última actualización 28/12/2017
Identificador abusaidm.html-snippets
20:27:06
16:51:21

Creamos nuestro primer proyecto

- ▶ Nos dirigimos al directorio deseado desde consola y ejecutamos
- ▶ **ng new nombredeproyecto**

Administrador: Símbolo del sistema

```
C:\Users\dell\Desktop\formacion_angular>ng new primero
```

```
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
```

```
This setting helps improve maintainability and catch bugs ahead of time.
```

```
For more information, see https://angular.io/strict Yes
```

```
? Would you like to add Angular routing? No
```

```
? Which stylesheet format would you like to use? CSS
```

```
CREATE primero/angular.json (3631 bytes)
```

```
CREATE primero/package.json (1208 bytes)
```

```
CREATE primero/README.md (1017 bytes)
```

```
CREATE primero/tsconfig.json (783 bytes)
```

```
CREATE primero/tslint.json (3185 bytes)
```

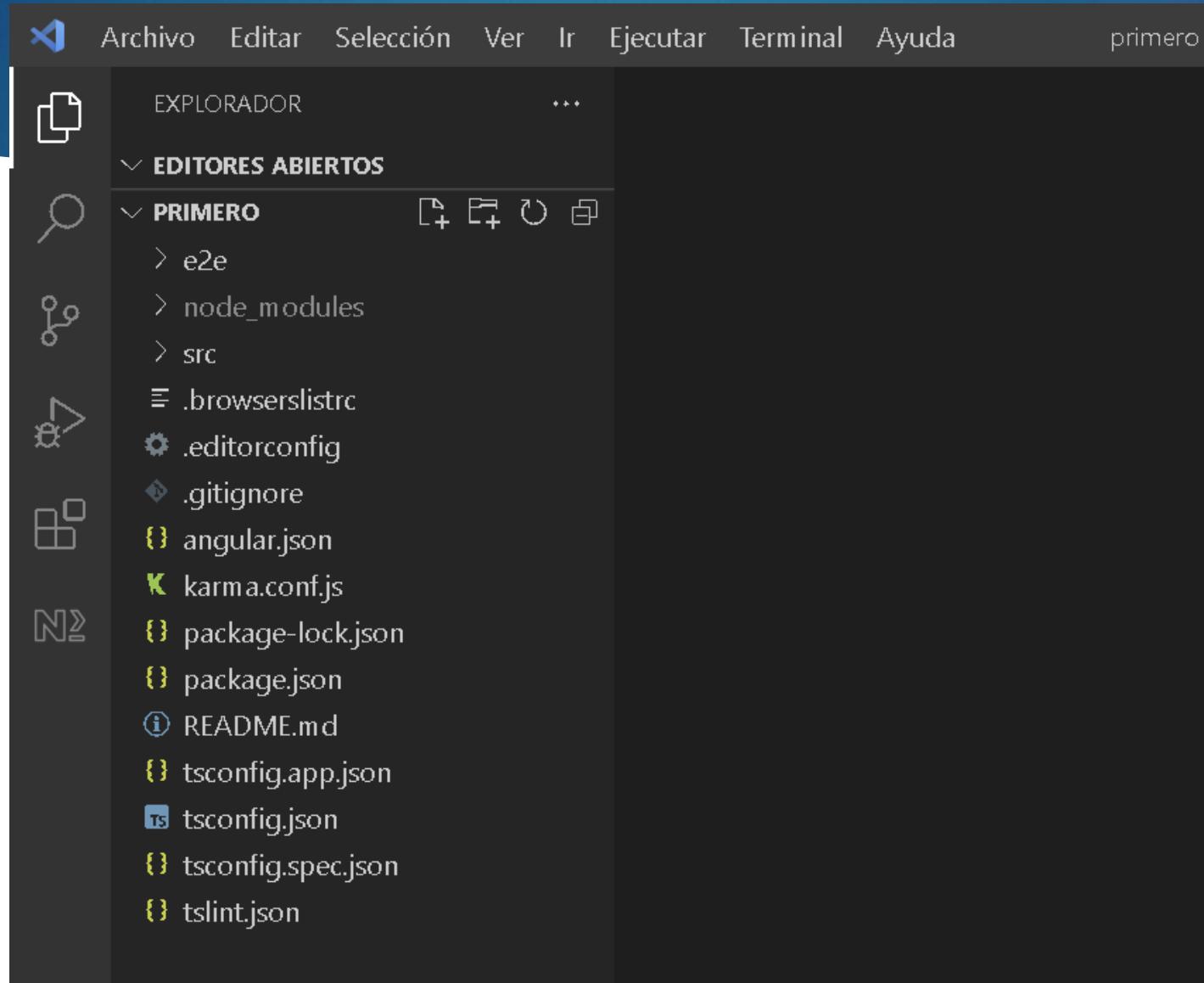
```
CREATE primero/.editorconfig (274 bytes)
```

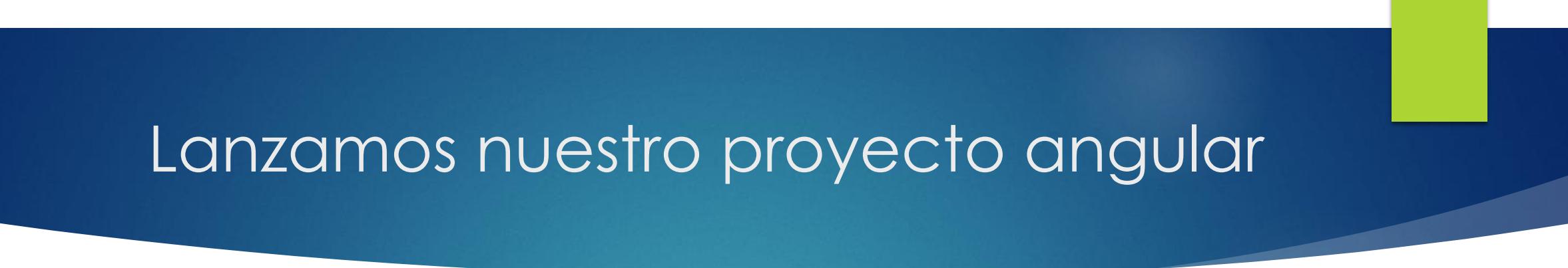
```
CREATE primero/.gitignore (631 bytes)
```

```
CREATE primero/.browserslistrc (703 bytes)
```

```
CREATE primero/karma.conf.js (1424 bytes)
```

Una vez creado el proyecto podemos abrirlo el proyecto con nuestro editor





Lanzamos nuestro proyecto angular

► **ng serve --open**

Para el que tiene inconvenientes con su consola desde
Visual Studio Code

PROBLEMAS

SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL



```
PS C:\Users\dell\Desktop\pruebas\primero> Set-ExecutionPolicy Unrestricted
PS C:\Users\dell\Desktop\pruebas\primero> ng serve --open
✖ Generating browser application bundles (phase: setup)...
```



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

primero - Visual Studio Code [Administrador]

— □ X



EXPLORADOR

...

EDITORES ABIERTOS

PRIMERO

- > .angular
- > node_modules
- > src
- ≡ .browserslistrc
- ⚙ .editorconfig
- ❖ .gitignore
- { } angular.json
- K karma.conf.js
- { } package-lock.json
- { } package.json
- ⓘ README.md
- { } tsconfig.app.json
- ts tsconfig.json
- { } tsconfig.spec.json



> ESQUEMA

> LÍNEA DE TIEMPO

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

main.js	main	51.43 kB
runtime.js	runtime	6.85 kB
	Initial Total	2.26 MB

[node] + × □ ^ ×

Build at: 2021-12-03T04:09:38.616Z - Hash: ebcd2cd3e172d7df - Time: 18485ms

** Angular Live Development Server is listening on localhost:4200, open your browser on <http://localhost:4200/> **

✓ Compiled successfully.

The screenshot shows the Angular welcome page running locally at `localhost:4200`. The page features a blue header with the Angular logo and the word "Welcome". Below the header is a central message: "primero app is running!" accompanied by a red rocket icon. A large white cloud graphic overlays the content. The main section is titled "Resources" and includes links to "Learn Angular", "CLI Documentation", "Angular Material", "Angular Blog", and "Angular DevTools". The "Angular Blog" link is currently active, indicated by a blue background. Below this is the "Next Steps" section, which asks "What do you want to do next with your app?" and lists several options: "New Component", "Angular Material", "Add PWA Support", "Add Dependency", "Run and Watch Tests", and "Build for Production".

localhost:4200

Welcome

primero app is running!

Resources

Here are some links to help you get started:

Learn Angular >

CLI Documentation >

Angular Material >

Angular Blog >

Angular DevTools >

Next Steps

What do you want to do next with your app?

+ New Component

+ Angular Material

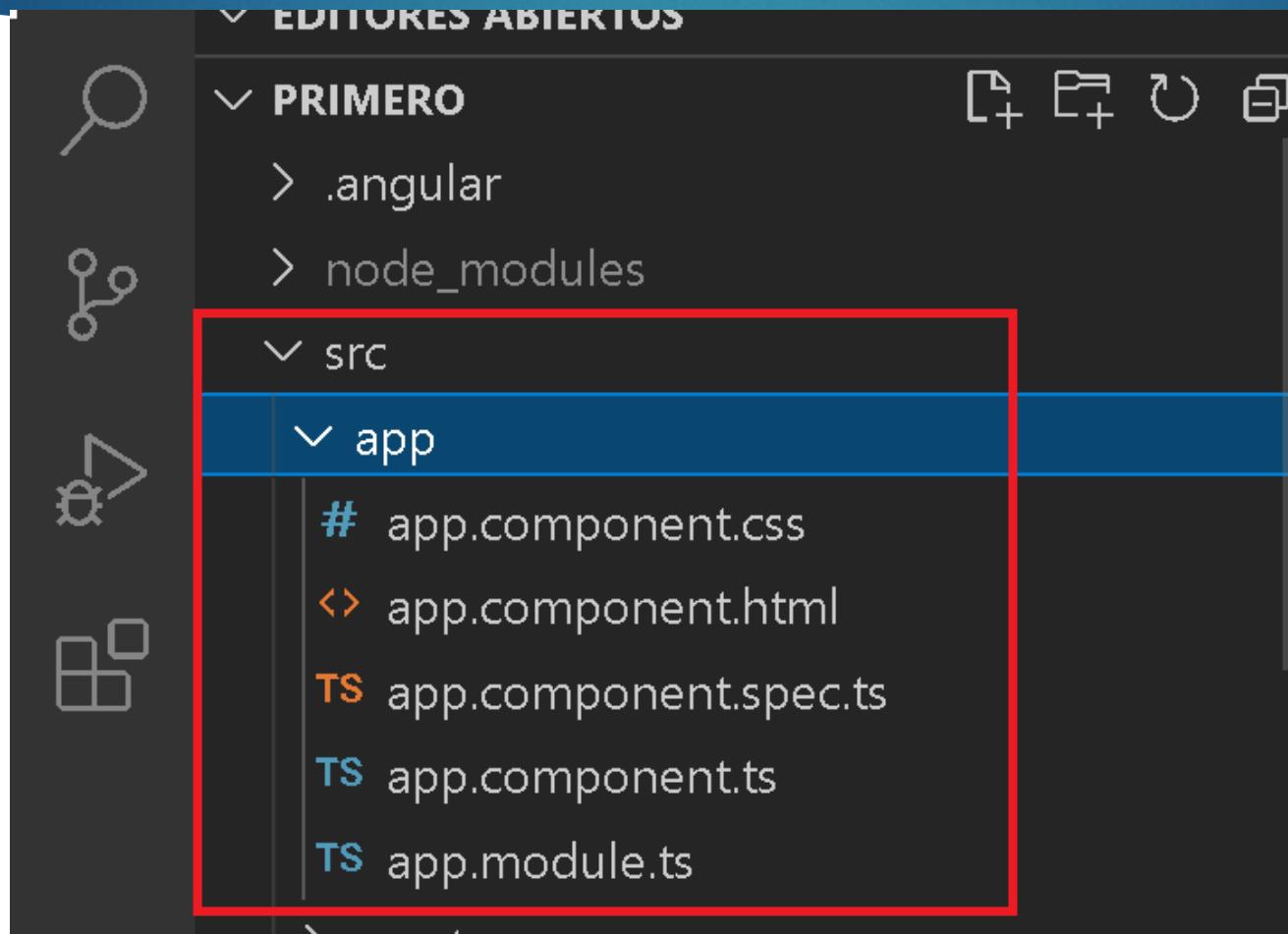
+ Add PWA Support

+ Add Dependency

+ Run and Watch Tests

+ Build for Production

Nos dirigimos a este directorio



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

EXPLORADOR EDITORES ABIERTOS PRIMERO

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   title = 'primero';
10 }
11
```

Agregamos esto en app.component.html

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

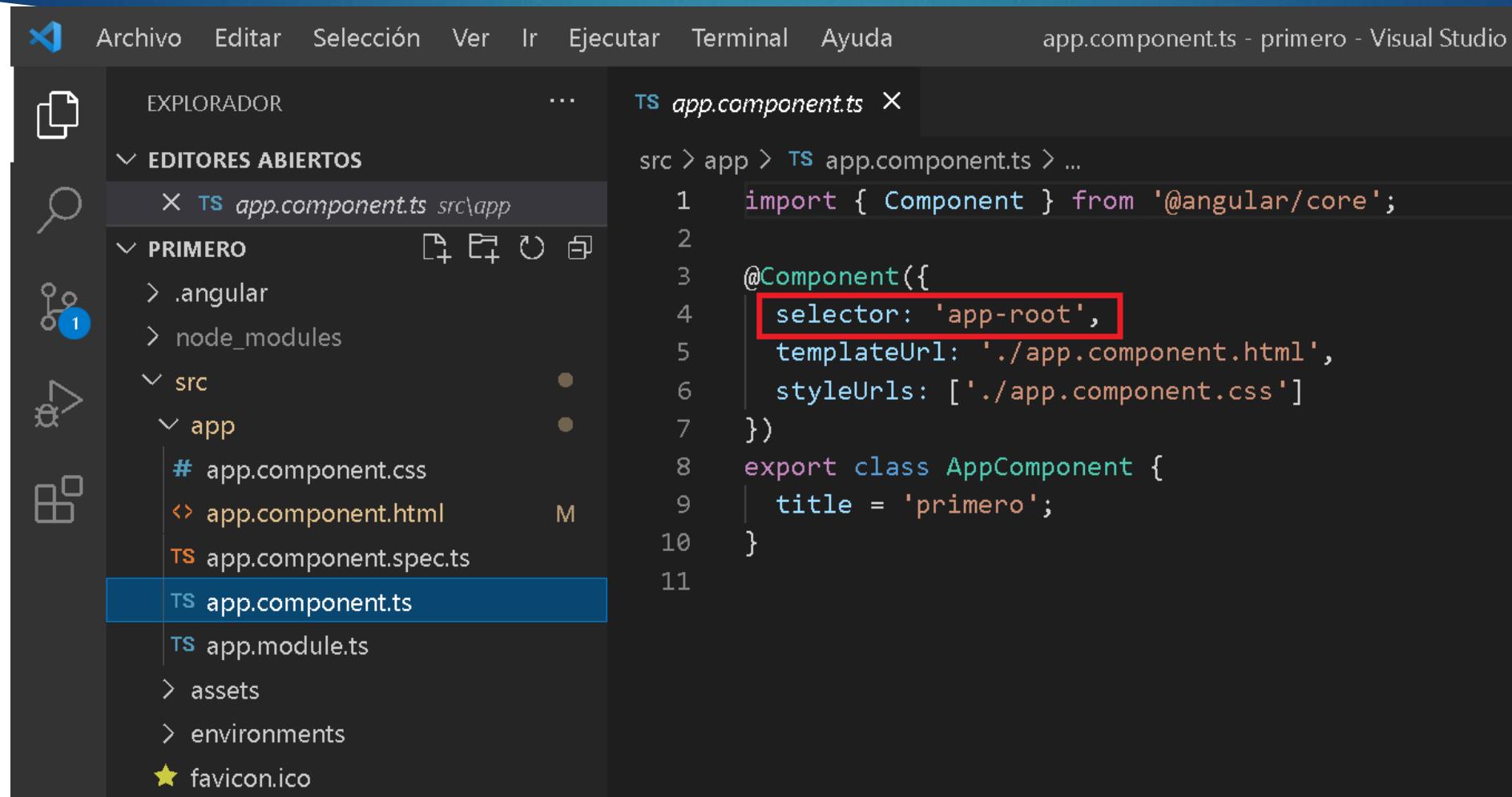
- File Menu:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** app.component.html - primero - Visual Studio Code.
- Explorer:** Shows the project structure:
 - EDITORES ABIERTOS: app.component.html (highlighted with a red box).
 - PRIMERO: .angular, node_modules, src (highlighted with a red box), app (highlighted with a red box), app.component.css (highlighted with a red box), app.component.html (highlighted with a red box).
 - Other files listed: app.component.spec.ts, app.component.ts, app.module.ts, assets, environments, favicon.ico, index.html, main.ts.
- Editor:** The code editor displays the content of app.component.html:

```
src > app > app.component.html > h1
1 | <h1>
2 |   Hola como vamos desde {{title}}
3 | </h1>
```
- Bottom Navigation:** PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL.



Hola como vamos desde primero

El selector app-root es incluida en el index principal para agregar todo el componente a la vista html



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

- File Explorer (Left):** Shows the project structure:
 - EDITORES ABIERTOS: app.component.ts (selected)
 - PRIMERO: .angular, node_modules, src (expanded), app (expanded), app.component.css, app.component.html, app.component.spec.ts, app.component.ts (highlighted in blue)
 - assets, environments, favicon.ico
- Code Editor (Right):** The file app.component.ts contains the following TypeScript 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   title = 'primero';
10 }
11
```
- Status Bar:** app.component.ts - primero - Visual Studio Code

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

- File Menu:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** index.html - primero - Visual Studio Code [Administrador].
- Left Sidebar (Explorador):**
 - EDITORES ABIERTOS: index.html src
 - PRIMERO:
 - .angular
 - node_modules
 - src
 - app
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
- Code Editor:** The index.html file is open, showing the following code:

```
src > <!doctype html>
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Primero</title>
6   <base href="/">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="icon" type="image/x-icon" href="favicon.ico">
9 </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
```

The line `<app-root></app-root>` is highlighted with a red border.
- Bottom Status Bar:** PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL.
 - TERMINAL: node +
 - SALIDA: 3 unchanged chunks
 - CONSOLA DE DEPURACIÓN: Build at: 2021-12-03T04:17:33.210Z - Hash: 683009ac500d4dbe - Time: 420ms

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

- File Explorer (Left):** Shows the project structure. The `src/app/app.component.ts` file is selected and highlighted with a red box.
- Code Editor (Right):** Displays the `app.component.ts` file content. A red box highlights the variable declarations:

```
nombre:string = "Rolando";
formacion:string = "Angular";
```
- Top Bar:** Shows the menu bar (Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda) and the title "app.component.ts - primero - Visual Studio Code".
- Bottom Bar:** Shows the status bar with tabs for PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, and TERMINAL.

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

- File Menu:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** app.component.html - primero - Visual Studio Code [Ad].
- Left Sidebar:** EXPLORADOR, EDITORES ABIERTOS (app.component.ts, app.component.html), PRIMERO (src, app, .angular, node_modules, assets, environments, favicon.ico, index.html, main.ts, polyfills.ts), and other icons.
- Editor Area:** The file `app.component.html` is open and highlighted with a red box. The code content is:

```
src > app > app.component.html > ul
1 <h1>
2   Bienvenidos al la formacion de {{formacion}}
3
4 </h1>
5
6 <ul>
7   <li>El formador {{nombre}}</li>
8   <li>El proyecto es {{title}}</li>
9 </ul>
```
- Bottom Navigation:** PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL.
- Status Bar:** 3 unchanged chunks.

Estructura de los componentes

Son pequeñas partes o bloques que forman parte de nuestra aplicación, y cumplen un rol específico dentro, en términos de programación son clases de typescript

- ▶ Menú de navegación
- ▶ Barra lateral
- ▶ Un pie de pagina
- ▶ Contenido dinámico o principal

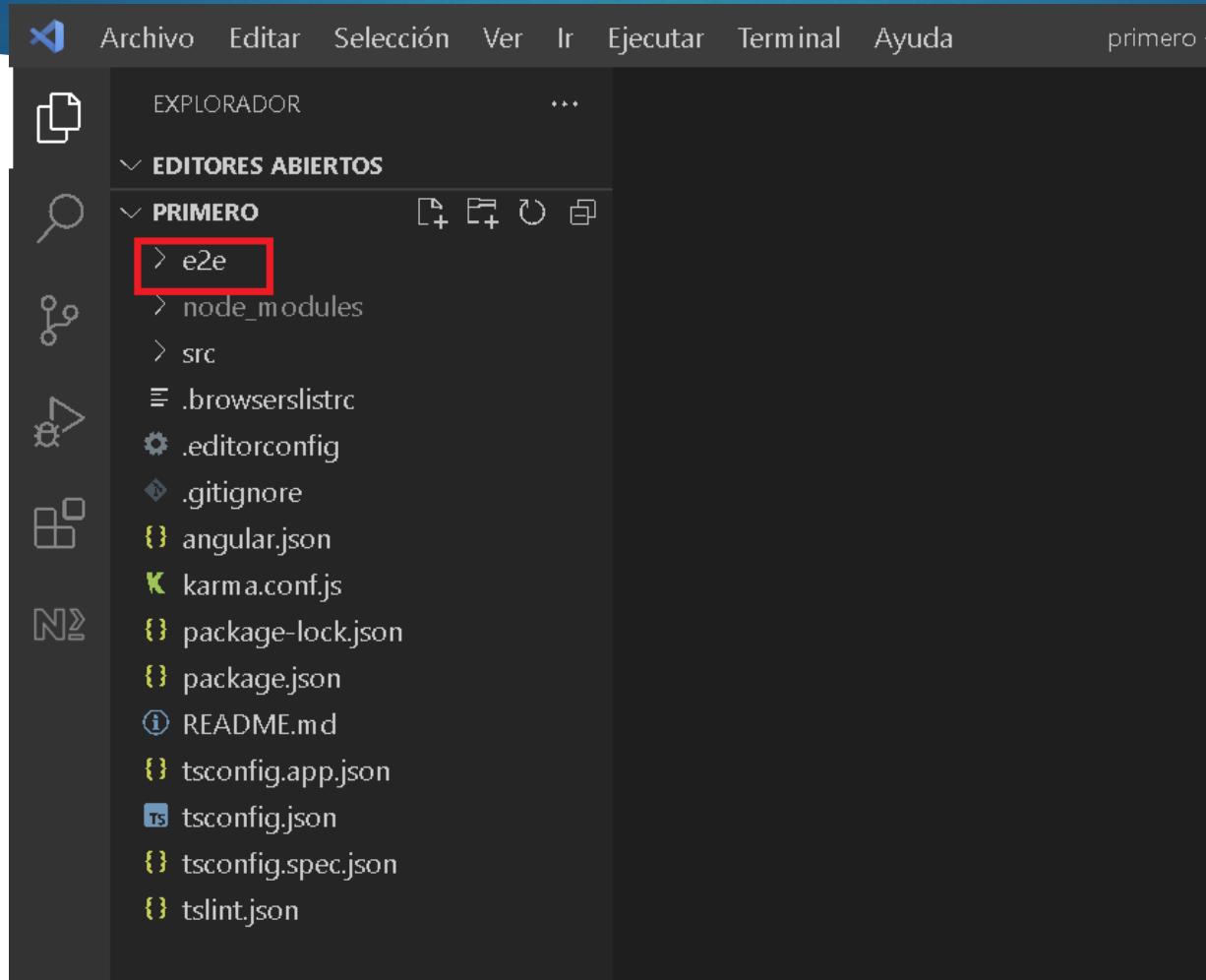
Características de los componentes

- ▶ Clase TS con una función
- ▶ Patrón composite
- ▶ Se puede anidar
- ▶ Se puede enrutar
- ▶ Ciclo de vida
- ▶ MVC
- ▶ Asíncronos
- ▶ Inyección de dependencia
- ▶ Tiene sus propios estilos

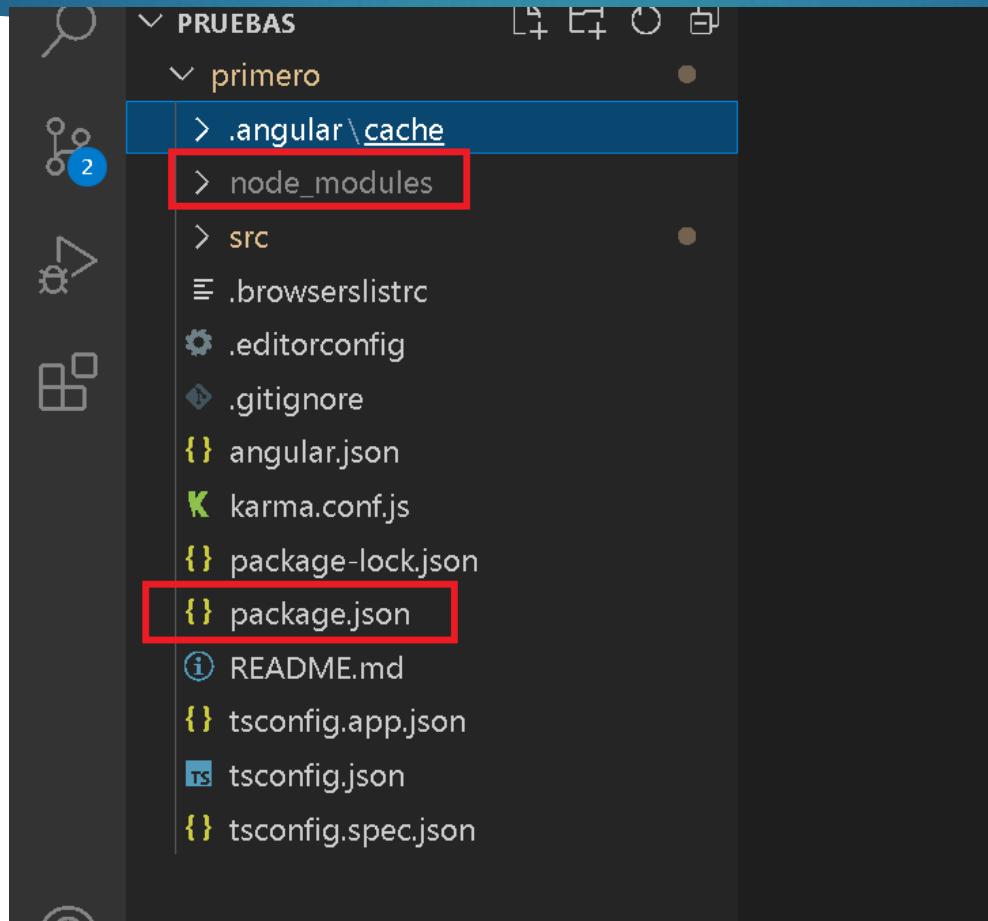
Archivos del proyecto Angular

- ▶ Estaremos viendo los archivo dentro del proyecto angular

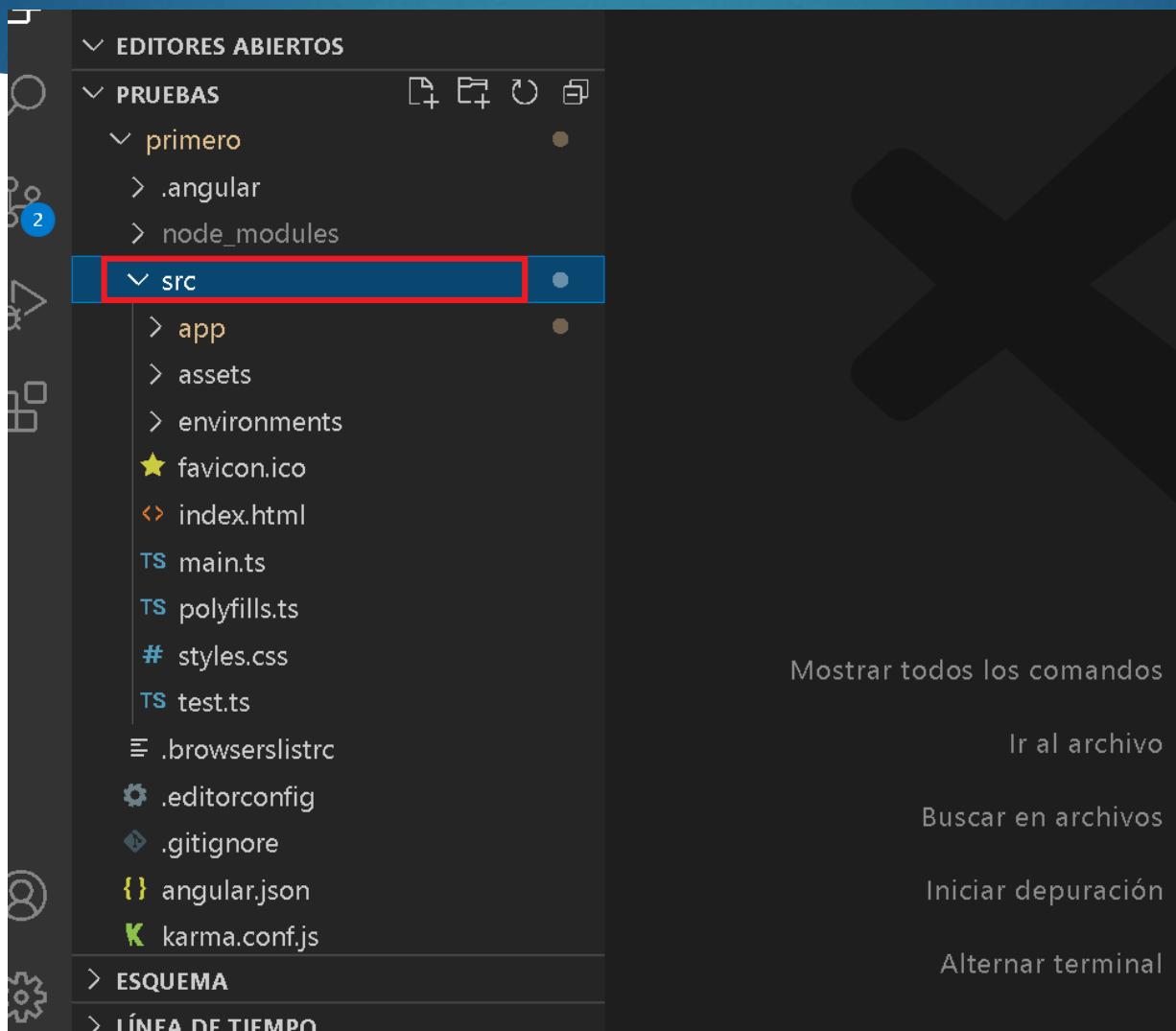
El directorio e2e para pruebas unitarias



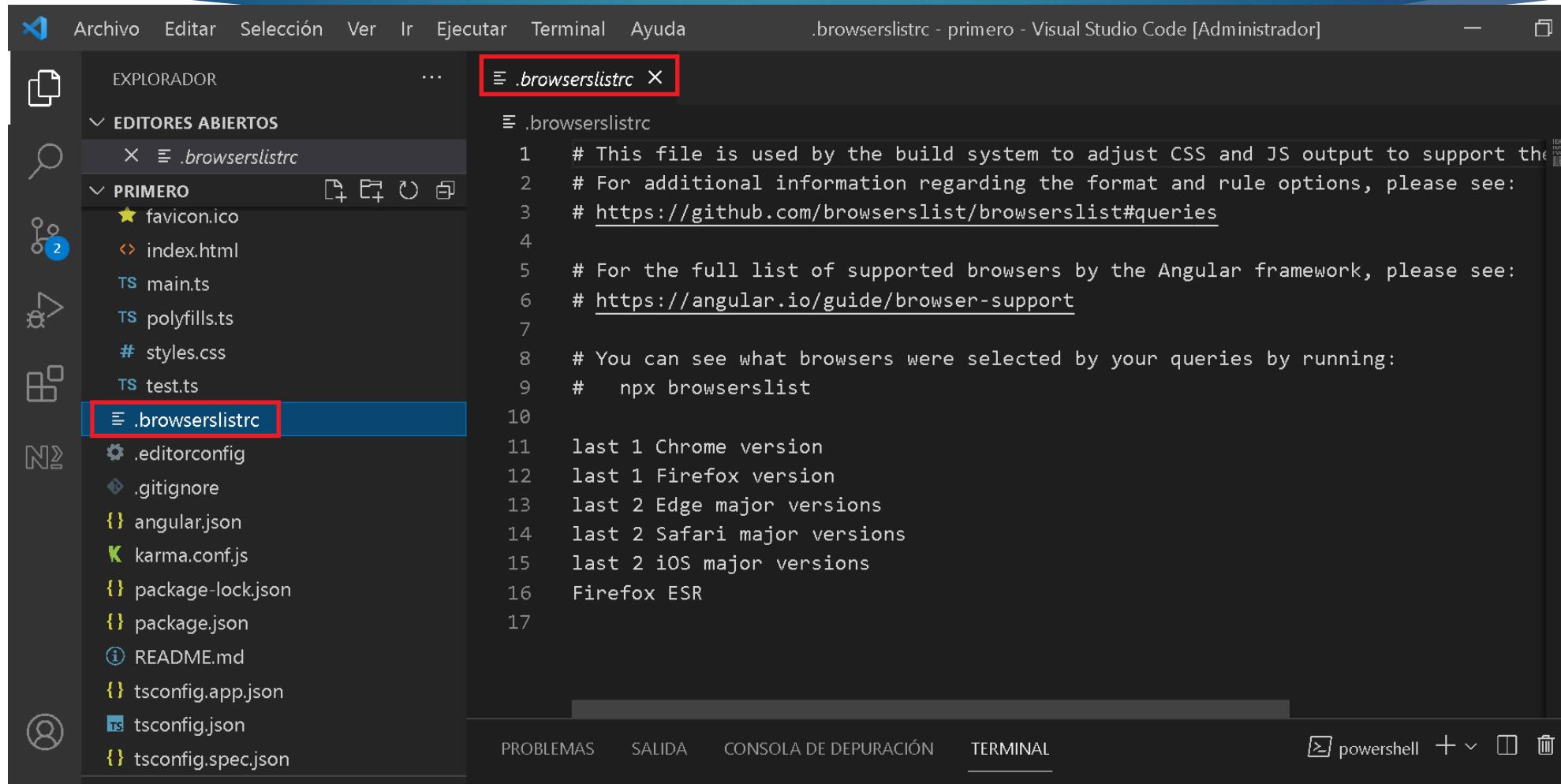
En `node_modules` encuentra todas las librerías y dependencias del proyecto, también maneja dependencias todo lo relacionado con dependencias de angular en general, y es manejado por el `package.json` de forma automática no deberíamos de tocarlo



La carpeta src es super importante contiene todo el código de nuestra aplicación



El archivo browserslistrc para incrementar/modificar compatibilidad con navegadores de internet existentes



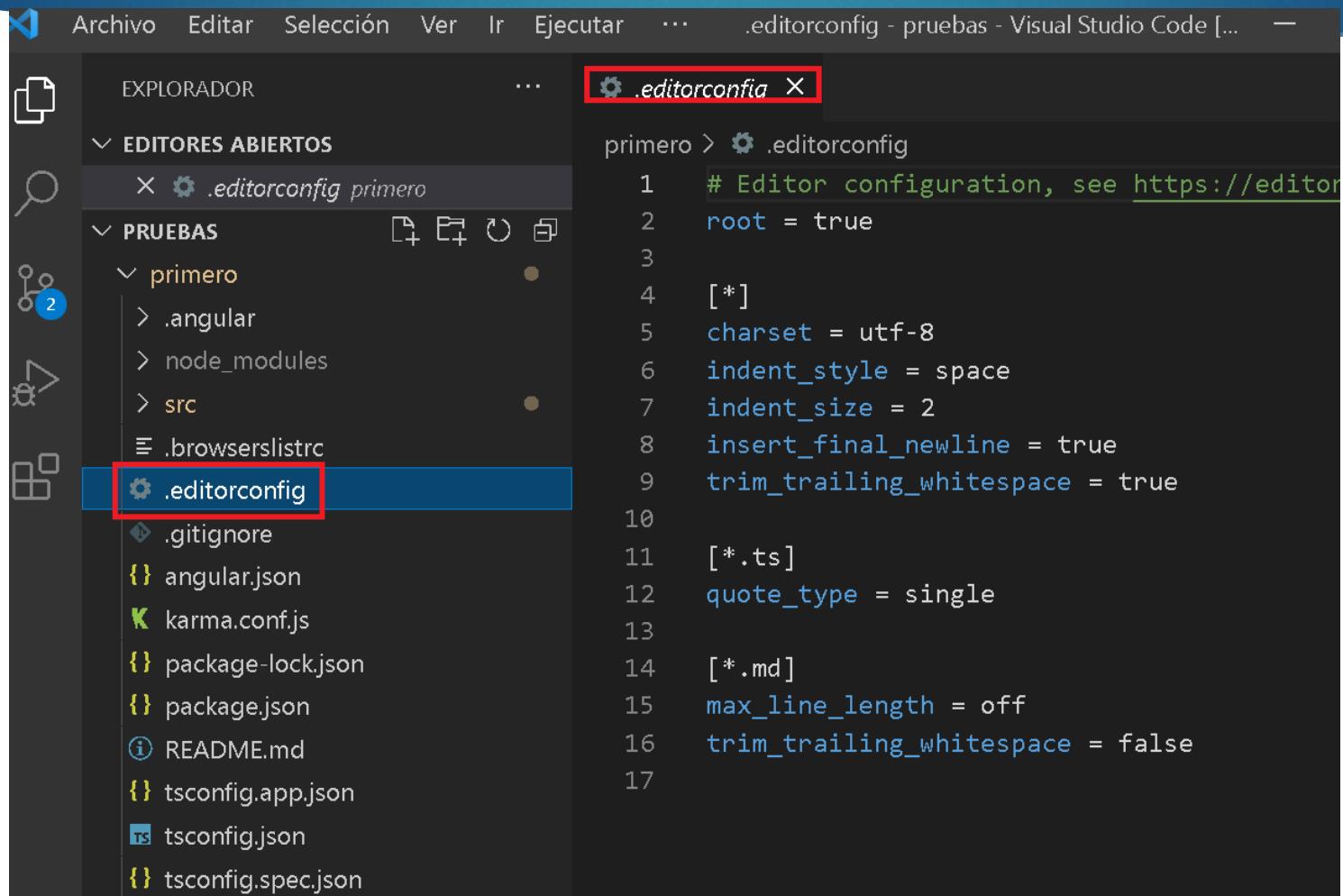
The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a project structure with files like `index.html`, `main.ts`, `polyfills.ts`, `styles.css`, `test.ts`, and configuration files like `.editorconfig`, `.gitignore`, `angular.json`, `karma.conf.js`, `package-lock.json`, `package.json`, `README.md`, `tsconfig.app.json`, `tsconfig.json`, and `tsconfig.spec.json`.
- Editor (Center):** Displays the contents of the `.browserslistrc` file. The file is a comment block starting with `/*` and ending with `*/`, containing instructions for the build system to adjust CSS and JS output for supported browsers.
- Bottom Navigation Bar:** Includes tabs for PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, and TERMINAL, along with icons for powershell, terminal, and other tools.

```
/*
 * This file is used by the build system to adjust CSS and JS output to support the
 * full list of supported browsers by the Angular framework, please see:
 * https://angular.io/guide/browser-support
 *
 * You can see what browsers were selected by your queries by running:
 * npx browserslist
 */

last 1 Chrome version
last 1 Firefox version
last 2 Edge major versions
last 2 Safari major versions
last 2 iOS major versions
Firefox ESR
```

El archivo editorconfig no se ocupa generalmente contiene toda la configuración del editor que podríamos modificar si lo necesitamos



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files and folders, with ".editorconfig" highlighted by a red box. The Editor tab on the right displays the content of the ".editorconfig" file, also with a red box around its title bar. The code in the editor is as follows:

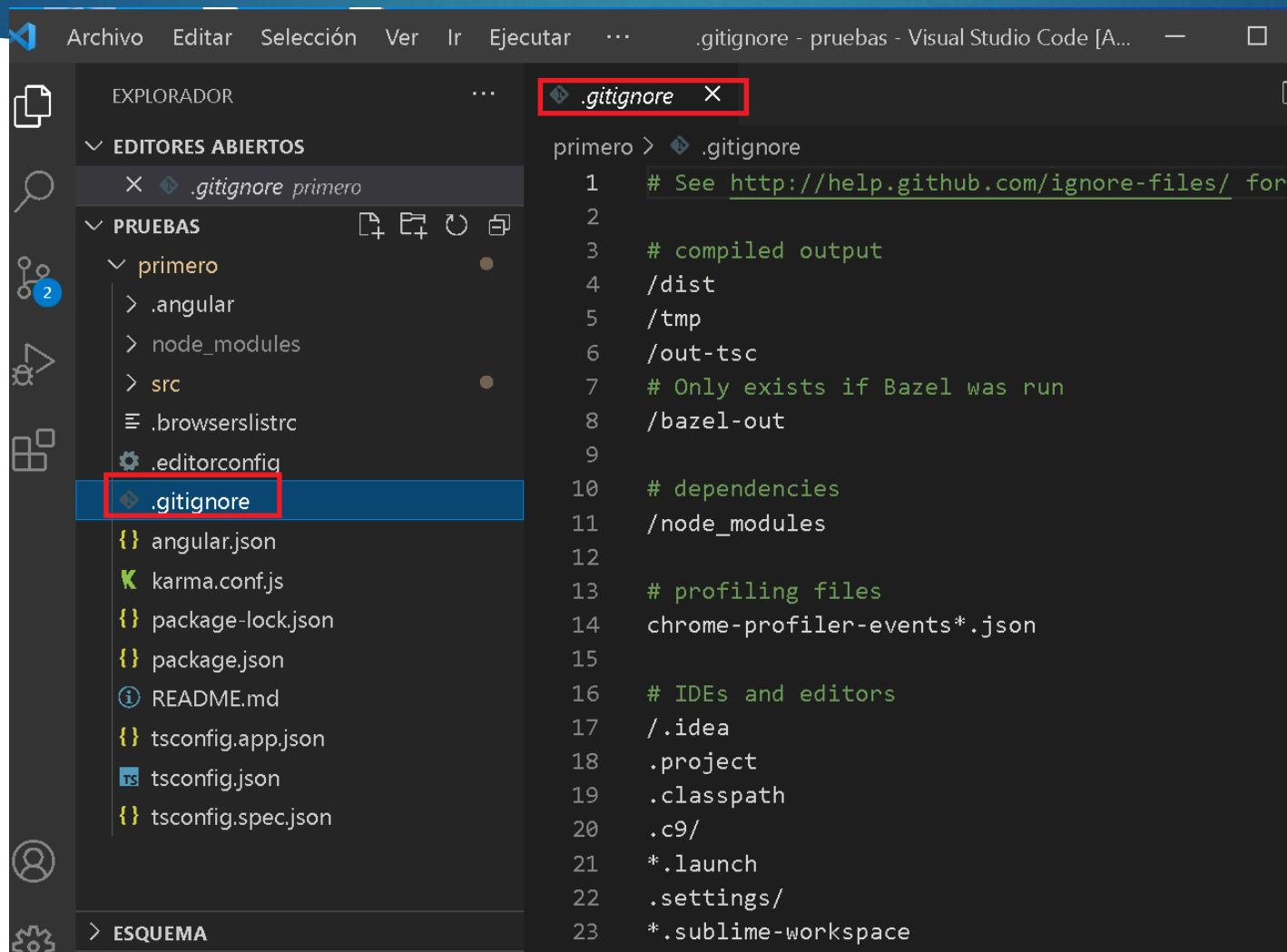
```
# Editor configuration, see https://editorconfig.org
root = true

[*]
charset = utf-8
indent_style = space
indent_size = 2
insert_final_newline = true
trim_trailing whitespace = true

[*.ts]
quote_type = single

[*.md]
max_line_length = off
trim_trailing whitespace = false
```

El archivo `.gitignore` es un archivo de github que nos permite ocultar ignorar algún archivo o directorio que no quisiéramos compartir en un repositorio git

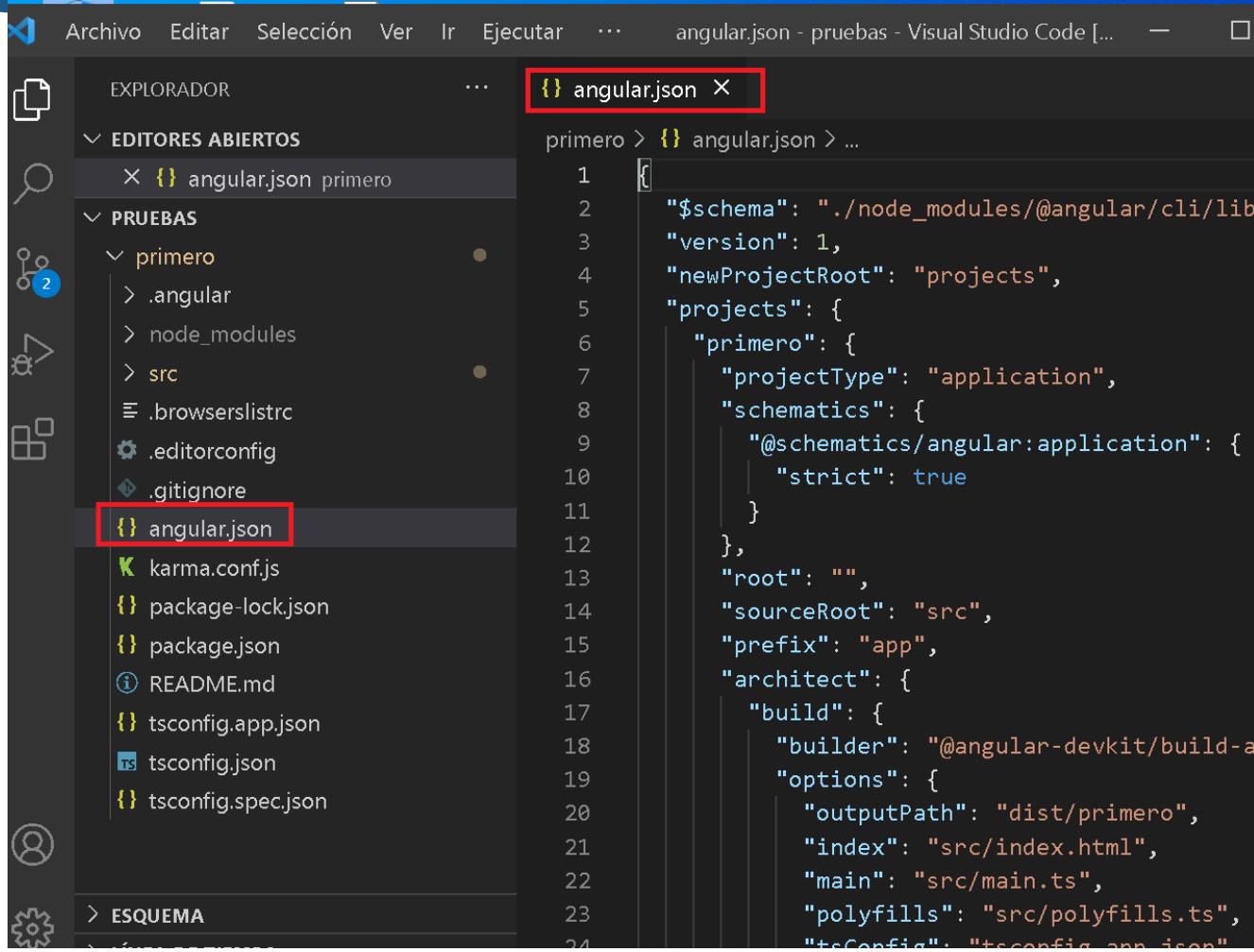


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a tree view of files and folders. A blue selection bar highlights the `.gitignore` file in the `PRUEBAS/primero` folder.
- Editor (Right):** Displays the content of the `.gitignore` file. The first few lines of the code are:

```
1 # See http://help.github.com/ignore-files/ for
2
3 # compiled output
4 /dist
5 /tmp
6 /out-tsc
7 # Only exists if Bazel was run
8 /bazel-out
9
10 # dependencies
11 /node_modules
12
13 # profiling files
14 chrome-profiler-events*.json
15
16 # IDEs and editors
17 /.idea
18 .project
19 .classpath
20 .c9/
21 *.launch
22 .settings/
23 *.sublime-workspace
```

Angular.json es el archivo principal de configuración del proyecto



The screenshot shows the Visual Studio Code interface. The title bar indicates the file is "angular.json - pruebas - Visual Studio Code [...]" with a tab icon. The left sidebar has icons for Explorer, Search, Problems, and others. The main area shows the "EXPLORADOR" (Explorer) view with a tree structure. The "EDITORES ABIERTOS" (Open Editors) section shows a file named "angular.json" with a red box around it. The "PRUEBAS" (Tests) section shows a folder named "primero" containing ".angular", ".node_modules", "src", ".browserslistrc", ".editorconfig", ".gitignore", and another "angular.json" file, also with a red box around it. The "ESQUEMA" (Schema) section shows ".tsconfig.app.json", ".tsconfig.json", and ".tsconfig.spec.json". The right-hand editor pane displays the JSON configuration for the "angular.json" file, which defines the project schema, version, new project root, and various build configurations for the "primero" application.

```
{ "$schema": "./node_modules/@angular/cli/lib", "version": 1, "newProjectRoot": "projects", "projects": { "primero": { "projectType": "application", "schematics": { "@schematics/angular:application": { "strict": true } }, "root": "", "sourceRoot": "src", "prefix": "app", "architect": { "build": { "builder": "@angular-devkit/build-angular:browser", "options": { "outputPath": "dist/primero", "index": "src/index.html", "main": "src/main.ts", "polyfills": "src/polyfills.ts", "tsConfig": "tsconfig.app.json" } } } } } }
```

El archivo karma.conf.js son las configuraciones para pruebas unitarias basadas en karma

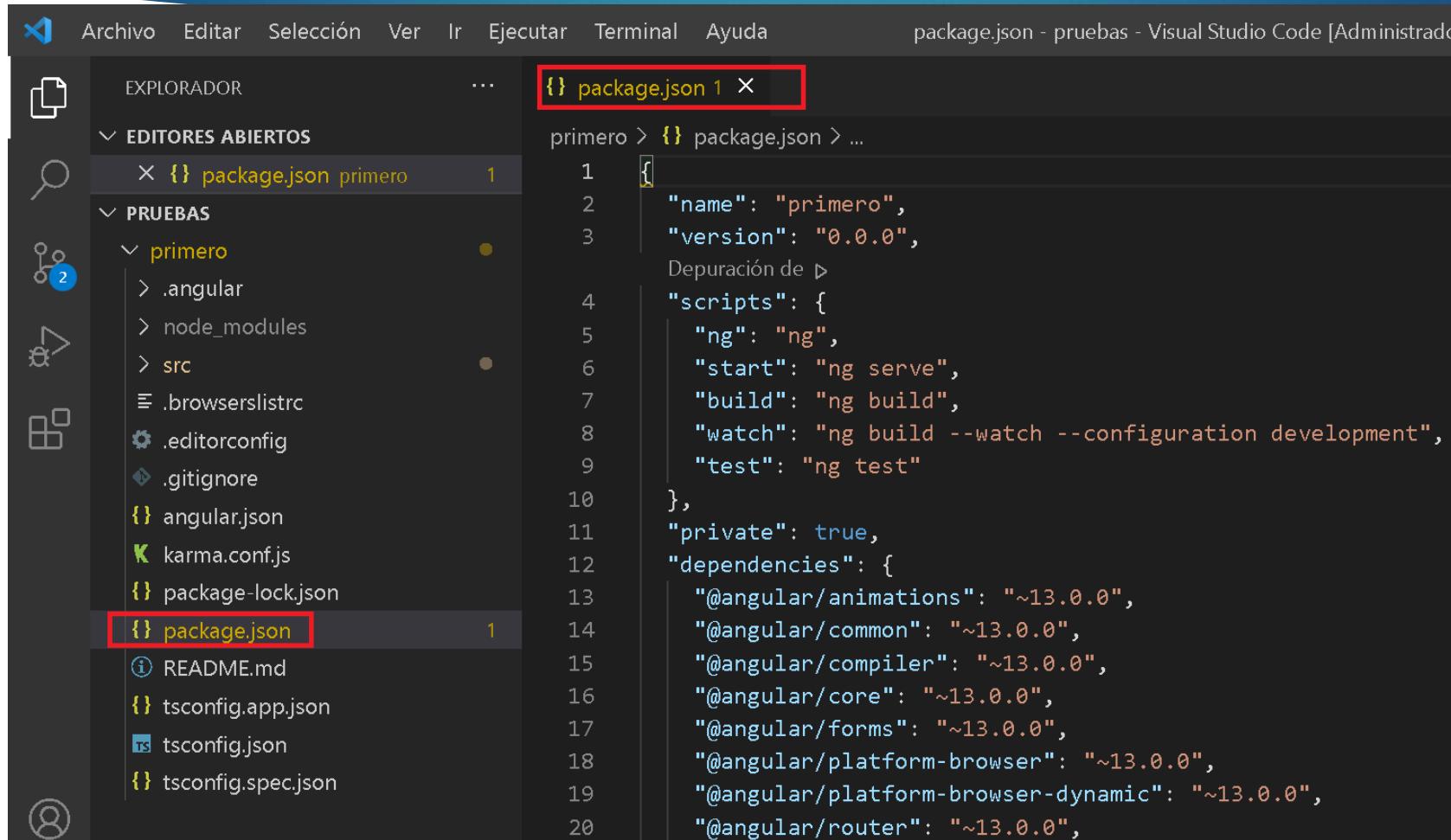
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure with files like index.html, main.ts, polyfills.ts, styles.css, test.ts, .browserslistrc, .editorconfig, .gitignore, angular.json, package-lock.json, package.json, README.md, tsconfig.app.json, tsconfig.json, and tsconfig.spec.json.
- Editor (Center):** Displays the contents of the karma.conf.js file. The file is a Karma configuration object with various settings for frameworks, plugins, and client configurations.
- Bottom Status Bar:** Shows the path PS C:\Users\dell\Desktop\pruebas\primero> and the terminal tab.

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html

module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage'),
      require('@angular-devkit/build-angular/plugins/karma')
    ],
    client: {
      jasmine: {
        // you can add configuration options for Jasmine here
        // the possible options are listed at https://jasmine.github.io/api/edge/
        // for example, you can disable the random execution with `random: false`
        // or set a specific seed with `seed: 4321`
      }
    }
  });
}
```

El package.json contiene todas las dependencias del proyecto, versiones

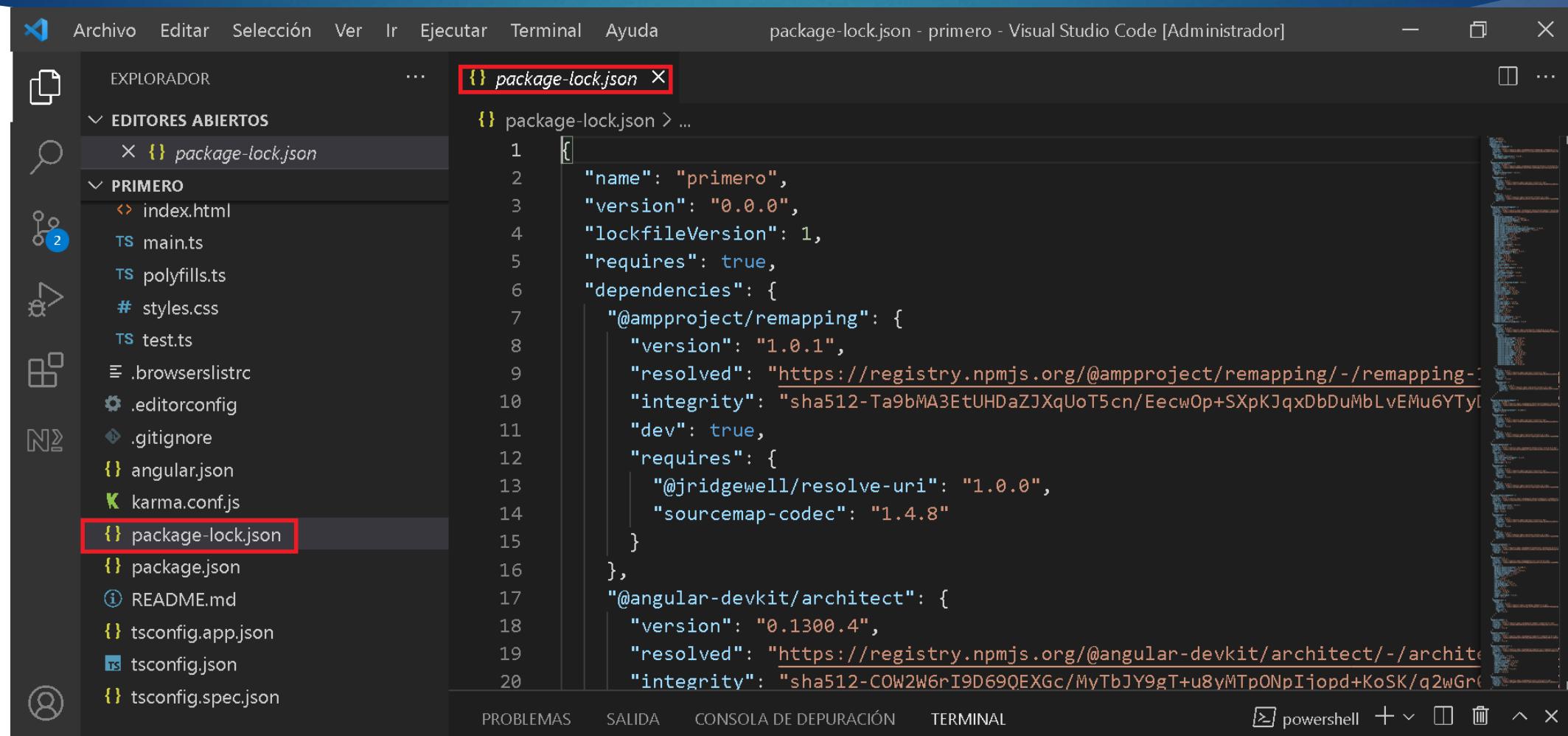


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure with files like `package.json`, `README.md`, `tsconfig.app.json`, `tsconfig.json`, and `tsconfig.spec.json`.
- Terminal (Top Right):** Shows the path `primero > package.json > ...`.
- Code Editor (Right):** Displays the `package.json` file content. The first few lines are:

```
1 {  
2   "name": "primero",  
3   "version": "0.0.0",  
4   "scripts": {  
5     "ng": "ng",  
6     "start": "ng serve",  
7     "build": "ng build",  
8     "watch": "ng build --watch --configuration development",  
9     "test": "ng test"  
10    },  
11    "private": true,  
12    "dependencies": {  
13      "@angular/animations": "~13.0.0",  
14      "@angular/common": "~13.0.0",  
15      "@angular/compiler": "~13.0.0",  
16      "@angular/core": "~13.0.0",  
17      "@angular/forms": "~13.0.0",  
18      "@angular/platform-browser": "~13.0.0",  
19      "@angular/platform-browser-dynamic": "~13.0.0",  
20      "@angular/router": "~13.0.0",
```

Es un archivo de registro automático de como se construye los módulos de node

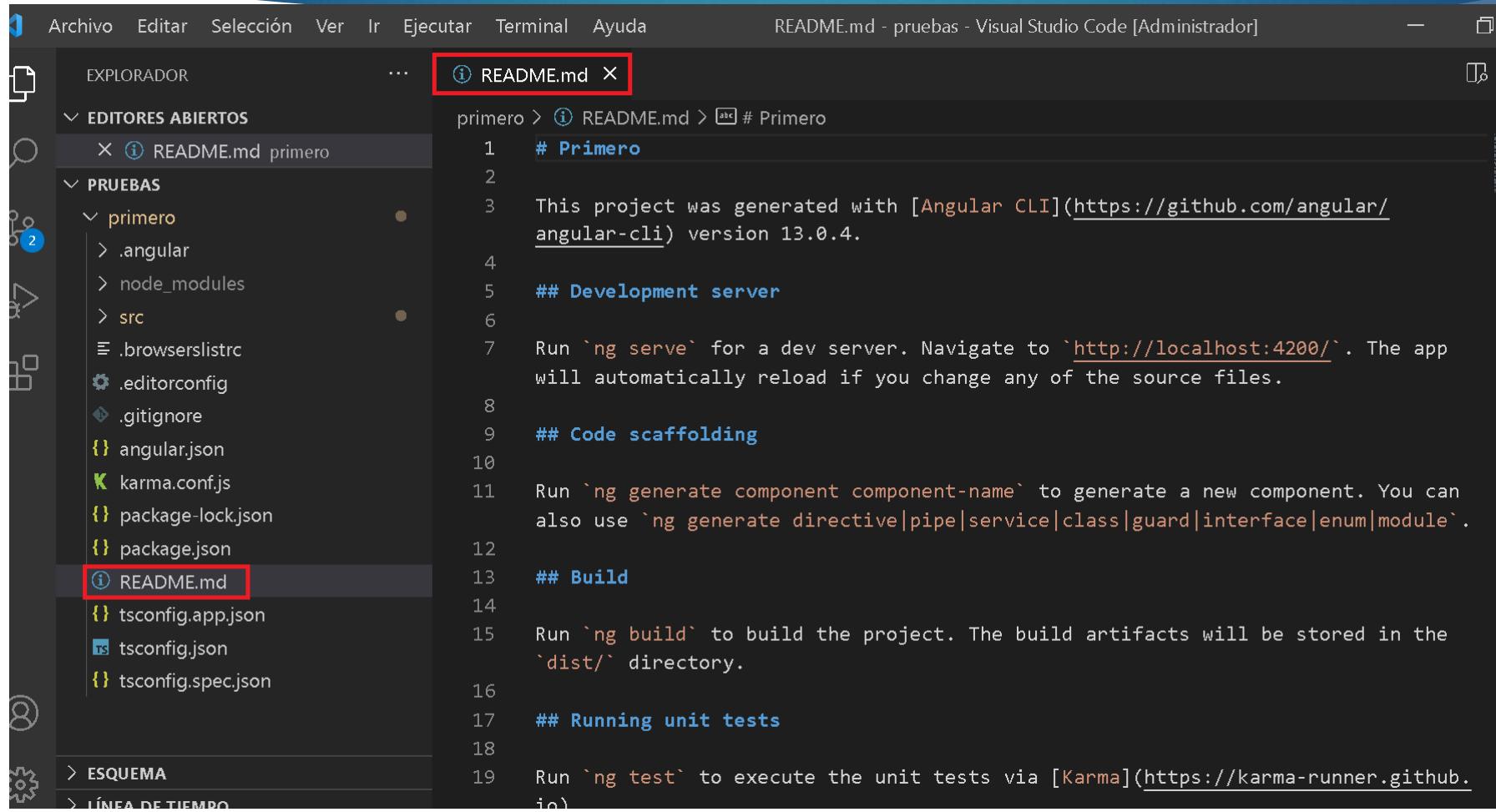


The screenshot shows a Visual Studio Code interface with the following details:

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** package-lock.json - primero - Visual Studio Code [Administrador].
- Left Sidebar (Explorador):** Shows the project structure with files like index.html, main.ts, polyfills.ts, styles.css, test.ts, .browserslistrc, .editorconfig, .gitignore, angular.json, karma.conf.js, package.json, README.md, tsconfig.app.json, tsconfig.json, and tsconfig.spec.json. The package-lock.json file is highlighted with a red box.
- Editor Area:** Displays the content of the package-lock.json file. The first few lines are:

```
1 {  
2   "name": "primero",  
3   "version": "0.0.0",  
4   "lockfileVersion": 1,  
5   "requires": true,  
6   "dependencies": {  
7     "@ampproject/remapping": {  
8       "version": "1.0.1",  
9       "resolved": "https://registry.npmjs.org/@ampproject/remapping/-/remapping-1.0.1.tgz",  
10      "integrity": "sha512-Ta9bMA3EtUHDaZJXqUoT5cn/EecwOp+SXpKJqxDbDuMbLvEMu6YTyD...",  
11      "dev": true,  
12      "requires": {  
13        "@jridgewell/resolve-uri": "1.0.0",  
14        "sourcemap-codec": "1.4.8"  
15      }  
16    },  
17    "@angular-devkit/architect": {  
18      "version": "0.1300.4",  
19      "resolved": "https://registry.npmjs.org/@angular-devkit/architect/-/architect-0.1300.4.tgz",  
20      "integrity": "sha512-COW2W6rI9D69QEXGc/MyTbJY9gT+u8yMTpONpIjopd+KoSK/q2wGr...",  
21    },  
22    "@angular/animations": {  
23      "version": "13.0.3",  
24      "resolved": "https://registry.npmjs.org/@angular/animations/-/animations-13.0.3.tgz",  
25      "integrity": "sha512-...",  
26    },  
27    "@angular/common": {  
28      "version": "13.0.3",  
29      "resolved": "https://registry.npmjs.org/@angular/common/-/common-13.0.3.tgz",  
30      "integrity": "sha512-...",  
31    },  
32    "@angular/compiler": {  
33      "version": "13.0.3",  
34      "resolved": "https://registry.npmjs.org/@angular/compiler/-/compiler-13.0.3.tgz",  
35      "integrity": "sha512-...",  
36    },  
37    "@angular/compiler-cli": {  
38      "version": "13.0.3",  
39      "resolved": "https://registry.npmjs.org/@angular/compiler-cli/-/compiler-cli-13.0.3.tgz",  
40      "integrity": "sha512-...",  
41    },  
42    "@angular/core": {  
43      "version": "13.0.3",  
44      "resolved": "https://registry.npmjs.org/@angular/core/-/core-13.0.3.tgz",  
45      "integrity": "sha512-...",  
46    },  
47    "@angular/forms": {  
48      "version": "13.0.3",  
49      "resolved": "https://registry.npmjs.org/@angular/forms/-/forms-13.0.3.tgz",  
50      "integrity": "sha512-...",  
51    },  
52    "@angular/platform-browser": {  
53      "version": "13.0.3",  
54      "resolved": "https://registry.npmjs.org/@angular/platform-browser/-/platform-browser-13.0.3.tgz",  
55      "integrity": "sha512-...",  
56    },  
57    "@angular/platform-browser-dynamic": {  
58      "version": "13.0.3",  
59      "resolved": "https://registry.npmjs.org/@angular/platform-browser-dynamic/-/platform-browser-dynamic-13.0.3.tgz",  
60      "integrity": "sha512-...",  
61    },  
62    "@angular/router": {  
63      "version": "13.0.3",  
64      "resolved": "https://registry.npmjs.org/@angular/router/-/router-13.0.3.tgz",  
65      "integrity": "sha512-...",  
66    },  
67    "esbuild": {  
68      "version": "0.14.2",  
69      "resolved": "https://registry.npmjs.org/esbuild/-/esbuild-0.14.2.tgz",  
70      "integrity": "sha512-...",  
71    },  
72    "node-sass": {  
73      "version": "7.0.1",  
74      "resolved": "https://registry.npmjs.org/node-sass/-/node-sass-7.0.1.tgz",  
75      "integrity": "sha512-...",  
76    },  
77    "sass": {  
78      "version": "1.57.0",  
79      "resolved": "https://registry.npmjs.org/sass/-/sass-1.57.0.tgz",  
80      "integrity": "sha512-...",  
81    },  
82    "typescript": {  
83      "version": "4.8.4",  
84      "resolved": "https://registry.npmjs.org/typescript/-/typescript-4.8.4.tgz",  
85      "integrity": "sha512-...",  
86    },  
87  },  
88}
```
- Bottom Status Bar:** PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL, powershell, +, □, X.

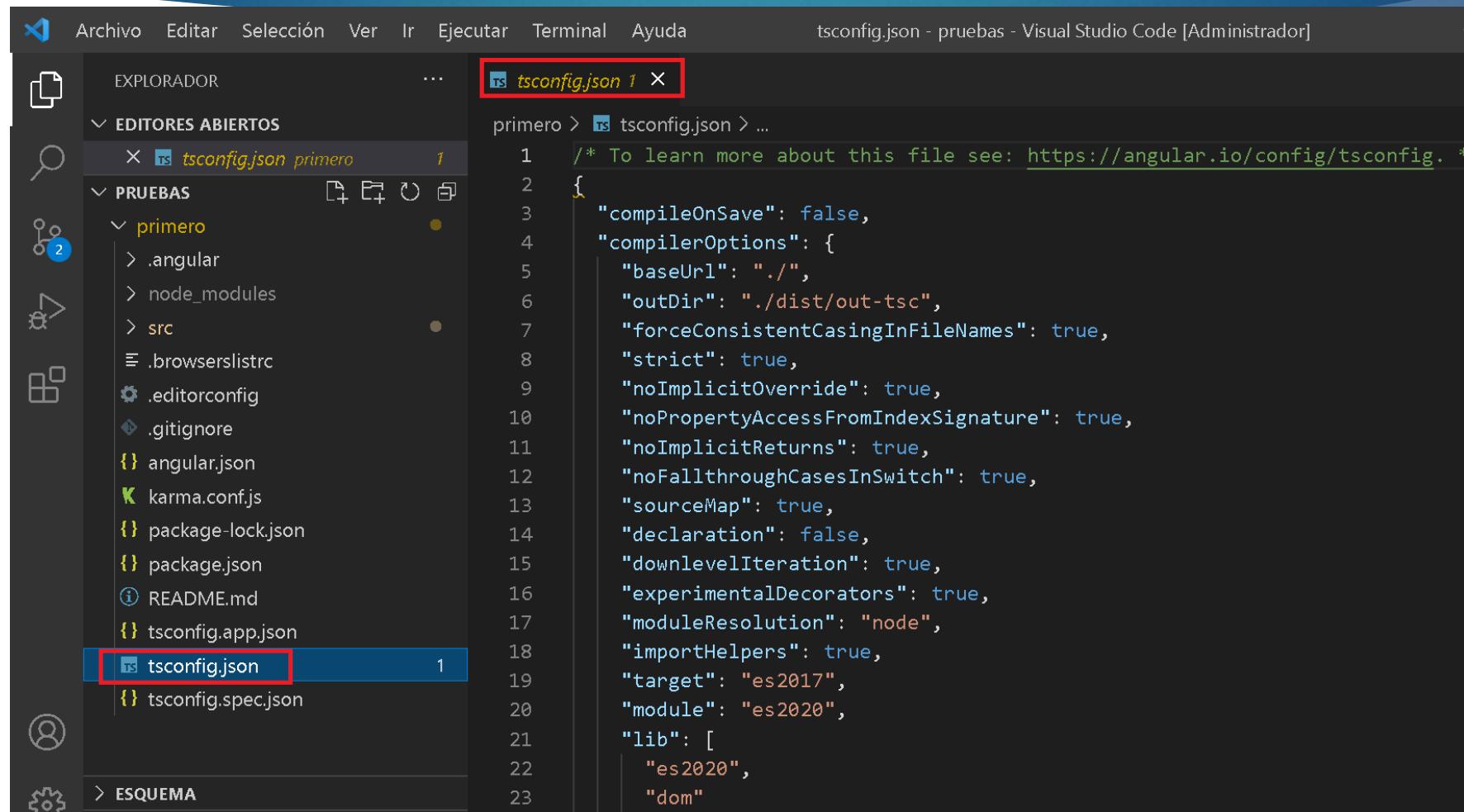
El archivo README.md es básicamente una documentación o guía para el proyecto



The screenshot shows a Visual Studio Code interface. The title bar reads "README.md - pruebas - Visual Studio Code [Administrador]". The left sidebar (Explorador) shows a project structure with files like .angular, node_modules, src, .browserslistrc, .editorconfig, .gitignore, angular.json, karma.conf.js, package-lock.json, package.json, and tsconfig files. The README.md file is listed twice: once in the top-level "EDITORES ABIERTOS" section and once in the "PRUEBAS" section under the "primero" folder. Both instances of the file are highlighted with a red border. The main editor area displays the content of the README.md file, which is a standard MD file providing project documentation.

```
primero > README.md > # Primero
1 # Primero
2
3 This project was generated with [Angular CLI](https://github.com/angular/angular-cli) version 13.0.4.
4
5 ## Development server
6
7 Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The app
will automatically reload if you change any of the source files.
8
9 ## Code scaffolding
10
11 Run `ng generate component component-name` to generate a new component. You can
also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.
12
13 ## Build
14
15 Run `ng build` to build the project. The build artifacts will be stored in the
`dist/` directory.
16
17 ## Running unit tests
18
19 Run `ng test` to execute the unit tests via [Karma](https://karma-runner.github.io).
```

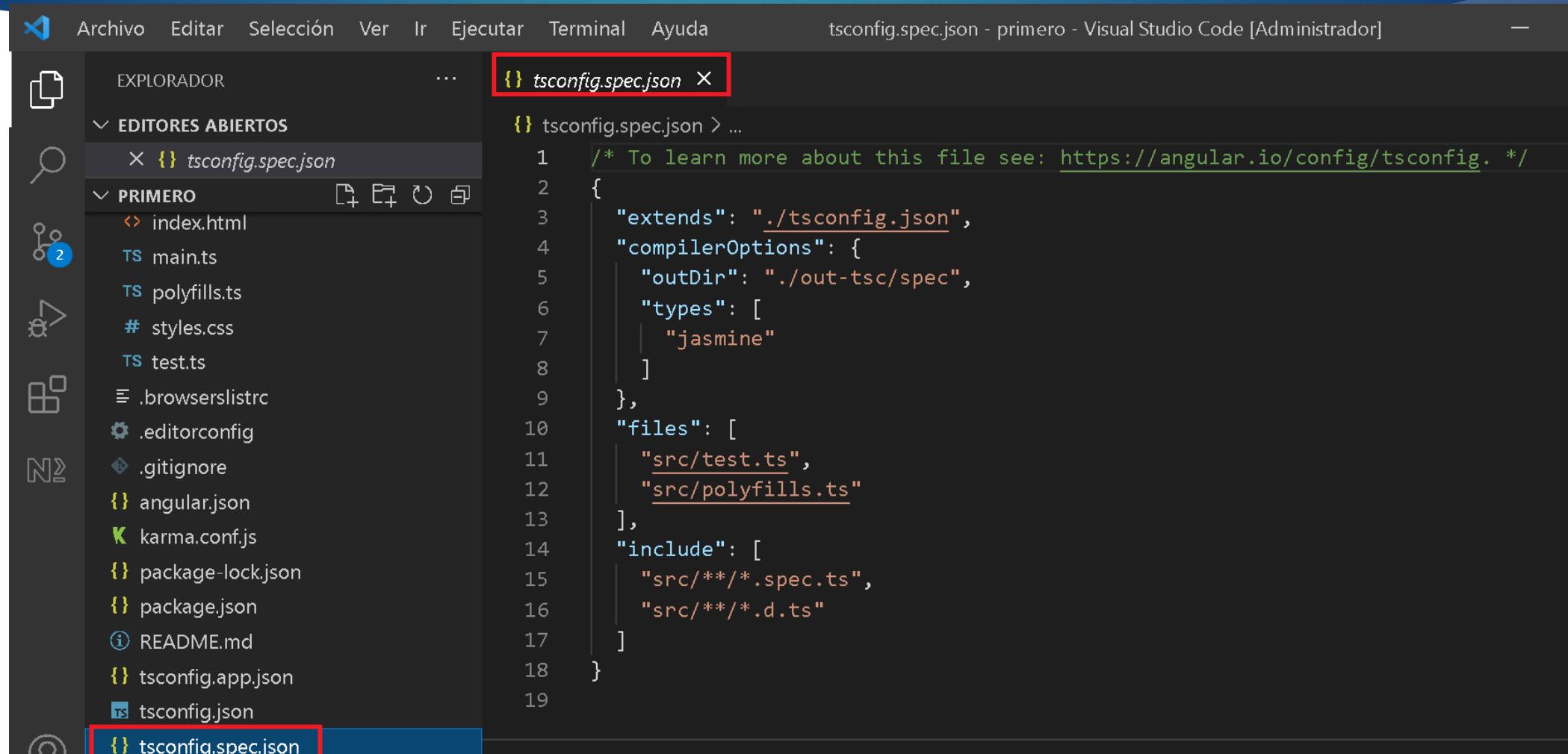
El archivo tsconfig nos ayuda con la alertas, variables, clases ,decoradores todo lo relacionado a como queremos que funcione typescript en nuestro proyecto



The screenshot shows the Visual Studio Code interface with the title bar "tsconfig.json - pruebas - Visual Studio Code [Administrador]". The left sidebar displays the file tree with files like .angular, node_modules, src, .browserslistrc, .editorconfig, .gitignore, angular.json, karma.conf.js, package-lock.json, package.json, README.md, tsconfig.app.json, and tsconfig.json. The tsconfig.json file is currently selected and highlighted with a red box. The main editor area shows the contents of the tsconfig.json file:

```
/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "noImplicitOverride": true,
    "noPropertyAccessFromIndexSignature": true,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "sourceMap": true,
    "declaration": false,
    "downlevelIteration": true,
    "experimentalDecorators": true,
    "moduleResolution": "node",
    "importHelpers": true,
    "target": "es2017",
    "module": "es2020",
    "lib": [
      "es2020",
      "dom"
    ]
}
```

El archivo `tsconfig.spec.json`, llama al archivo `tsconfig.json` y añade mas configuraciones relacionadas a pruebas unitarias



```
/* To learn more about this file see: https://angular.io/config/tsconfig. */\n{\n  \"extends\": \"./tsconfig.json\", \n  \"compilerOptions\": { \n    \"outDir\": \"./out-tsc/spec\", \n    \"types\": [ \n      \"jasmine\" \n    ] \n  }, \n  \"files\": [ \n    \"src/test.ts\", \n    \"src/polyfills.ts\" \n  ], \n  \"include\": [ \n    \"src/**/*.spec.ts\", \n    \"src/**/*.d.ts\" \n  ] \n}\n\n
```

El archivo tsconfig.app.json llama al archivo tsconfig.json y agrega mas configuraciones respecto a la aplicación

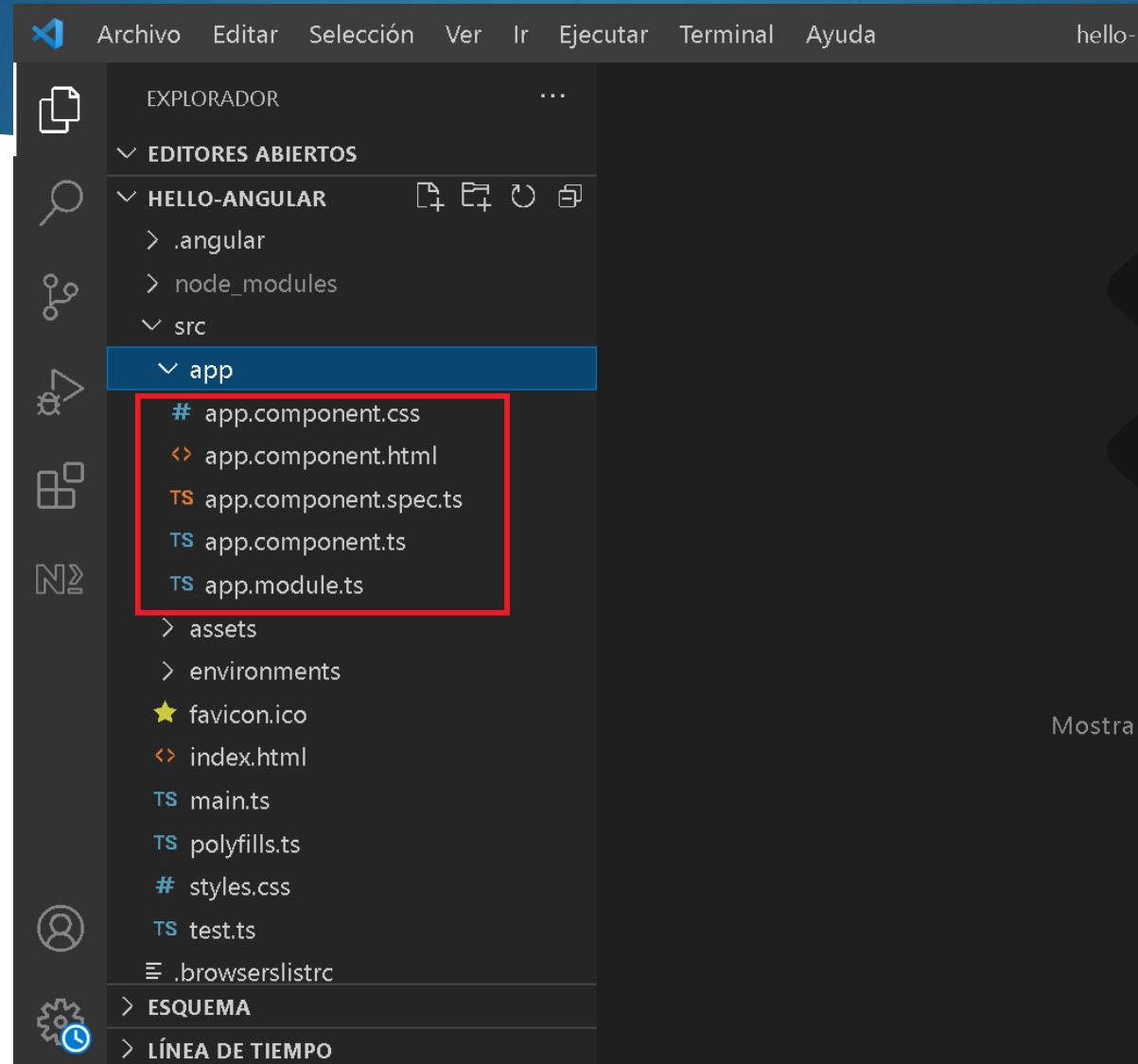
The screenshot shows the Visual Studio Code interface with the following details:

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** tsconfig.app.json - primero - Visual Studio Code [Administrador].
- Left Sidebar (Explorador):** Shows the project structure with files like index.html, main.ts, polyfills.ts, styles.css, test.ts, .browserslistrc, .editorconfig, .gitignore, angular.json, karma.conf.js, package-lock.json, package.json, and README.md. The file "tsconfig.app.json" is highlighted with a red box at the bottom of the sidebar.
- Editor Area:** Displays the contents of the tsconfig.app.json file. The code is as follows:

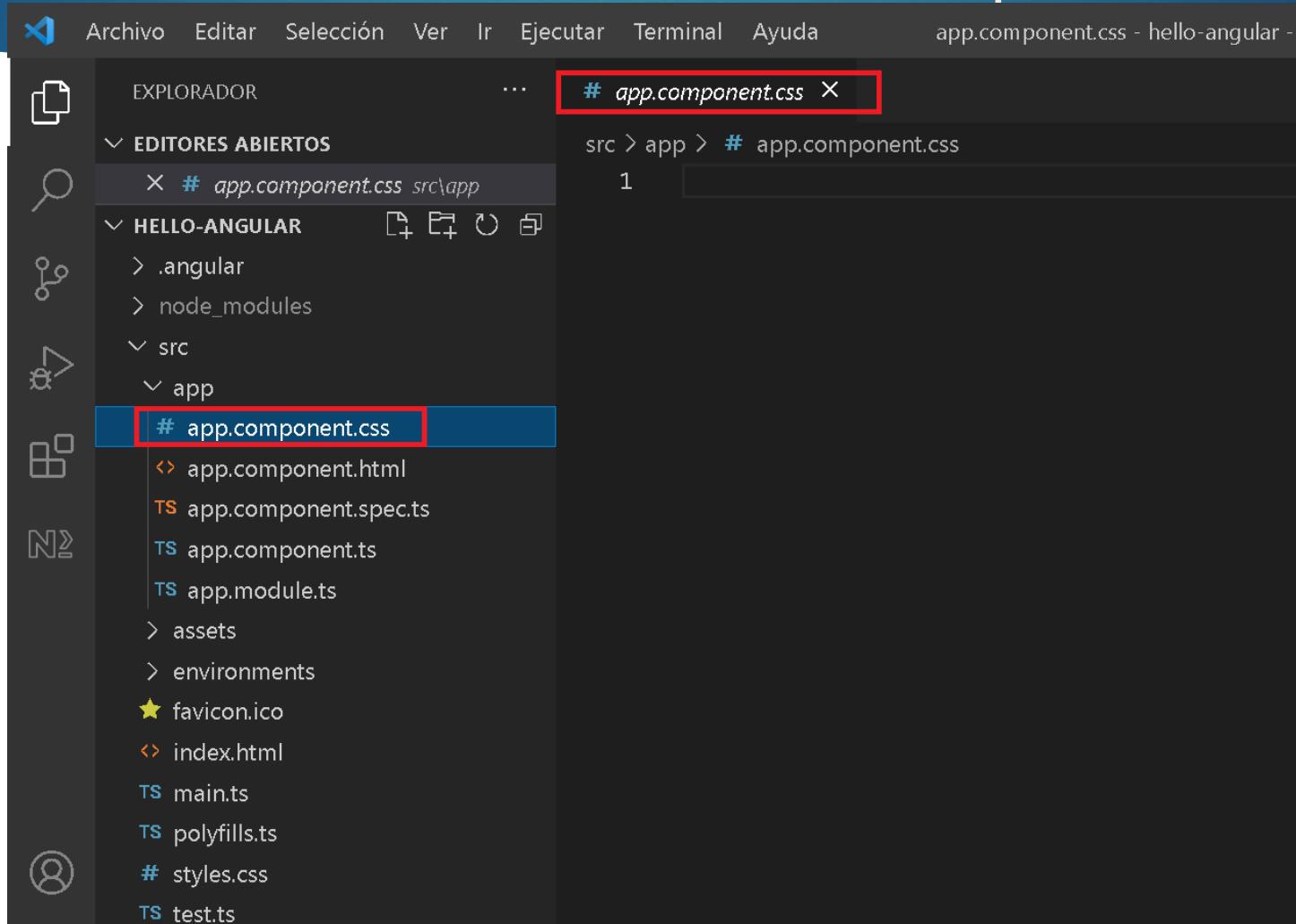
```
/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/app",
    "types": []
  },
  "files": [
    "src/main.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}
```

Explorando los archivos internos de src

Directorio app, van los componentes de nuestra aplicacion

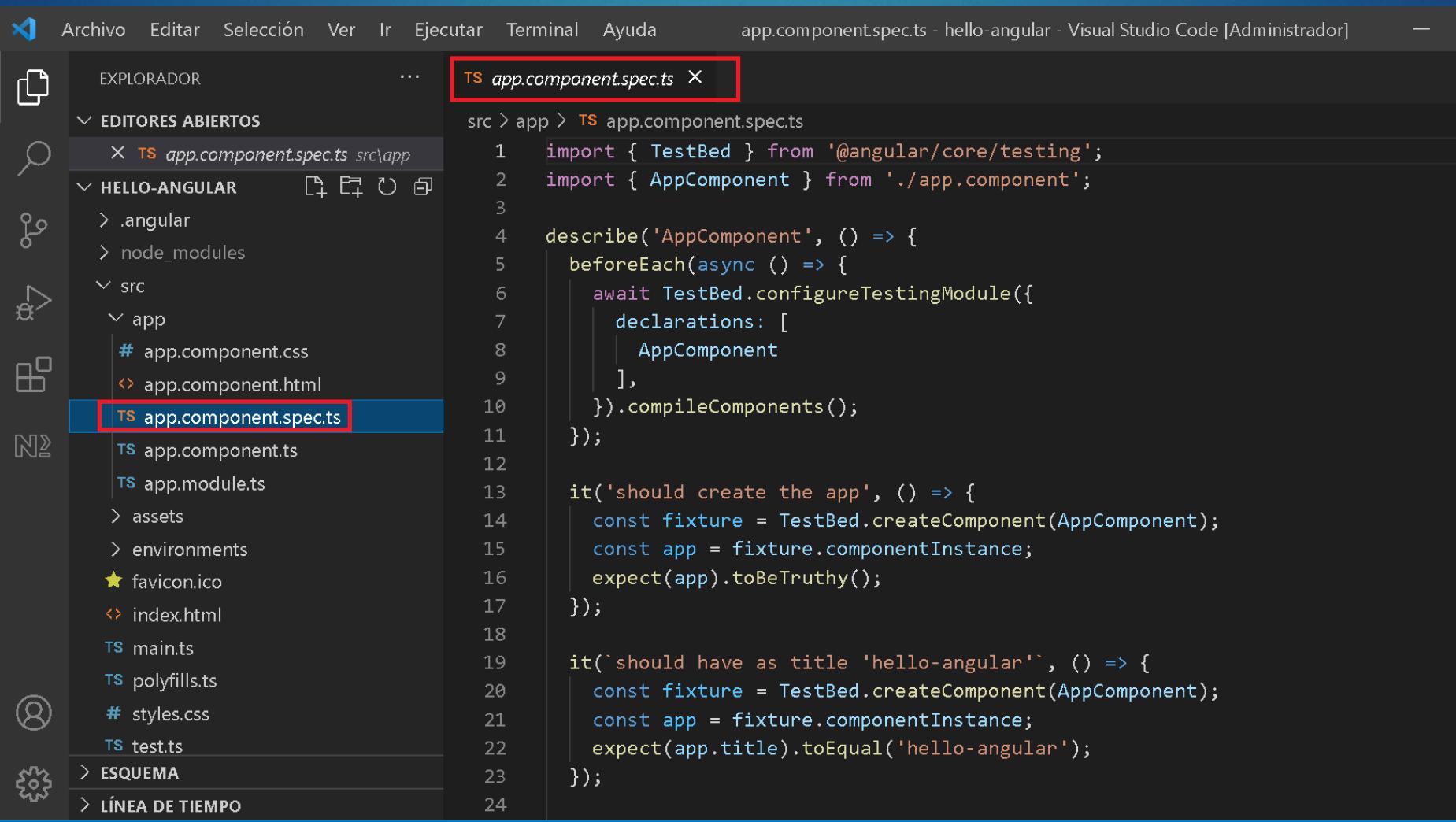


App.component.css se encarga de todo el contenido css del componente



App.component.html contiene todo el html o vista del componente

App.component.spec.ts para realizar pruebas unitarias

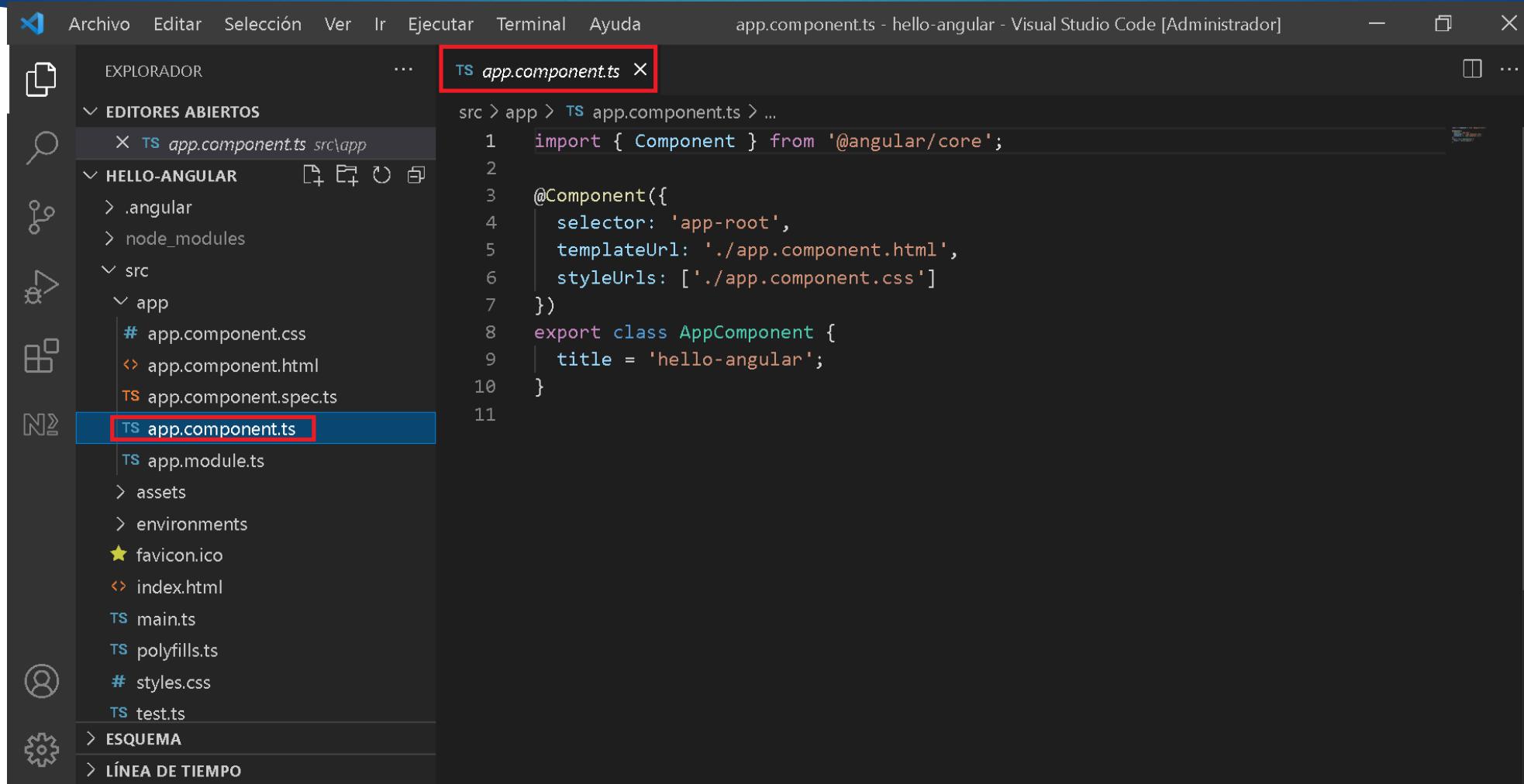


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "EDITORES ABIERTOS". The file "app.component.spec.ts" is highlighted with a blue selection bar and has its path "src > app > app.component.spec.ts" displayed above it.
- Editor (Right):** Displays the content of the "app.component.spec.ts" file, which is a unit test for the AppComponent. The code uses Jasmine and TestBed to test the component's creation and title.
- Top Bar:** Shows the menu items Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda, and the title "app.component.spec.ts - hello-angular - Visual Studio Code [Administrador]".

```
src > app > app.component.spec.ts
1 import { TestBed } from '@angular/core/testing';
2 import { AppComponent } from './app.component';
3
4 describe('AppComponent', () => {
5   beforeEach(async () => {
6     await TestBed.configureTestingModule({
7       declarations: [
8         AppComponent
9       ],
10    }).compileComponents();
11  });
12
13  it('should create the app', () => {
14    const fixture = TestBed.createComponent(AppComponent);
15    const app = fixture.componentInstance;
16    expect(app).toBeTruthy();
17  });
18
19  it(`should have as title 'hello-angular'`, () => {
20    const fixture = TestBed.createComponent(AppComponent);
21    const app = fixture.componentInstance;
22    expect(app.title).toEqual('hello-angular');
23  });
24}
```

Es una clase que representa a una parte de nuestra aplicación web



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

- File Menu:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** app.component.ts - hello-angular - Visual Studio Code [Administrador].
- Explorer Bar:** Shows the project structure:
 - src > app > app.component.ts
 - HELLO-ANGULAR
 - .angular
 - node_modules
 - src
 - app
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - assets
 - environments
 - favicon.ico
 - index.html
 - ESQUEMA
 - LÍNEA DE TIEMPO
- Editor Area:** The file app.component.ts is open, showing the following code:

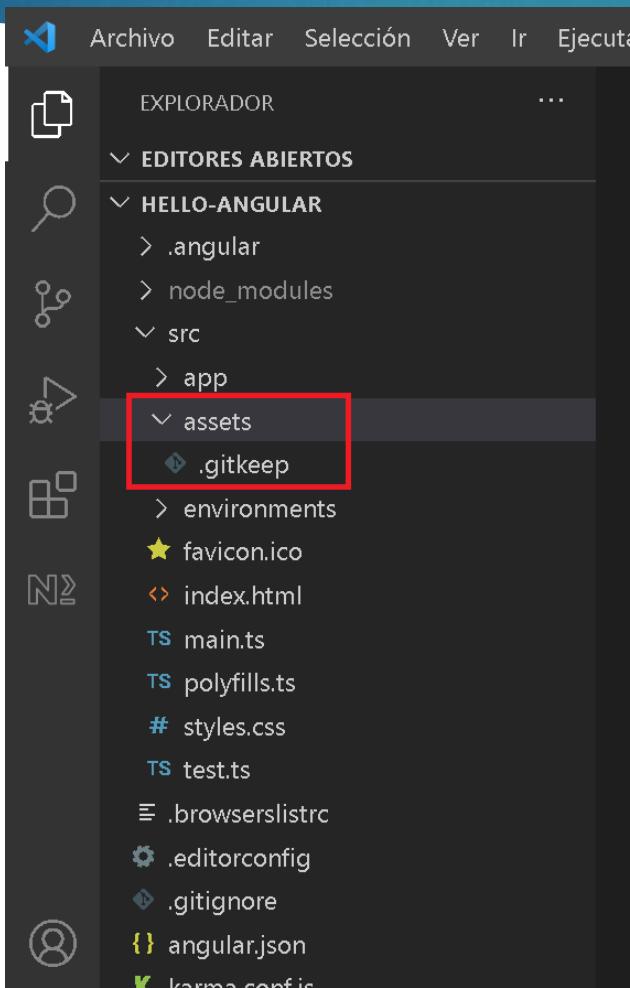
```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'hello-angular';
10 }
```

app.modules.ts es como un repositorio donde se registra nuestro componente y todo lo que podriamos utilizar en el

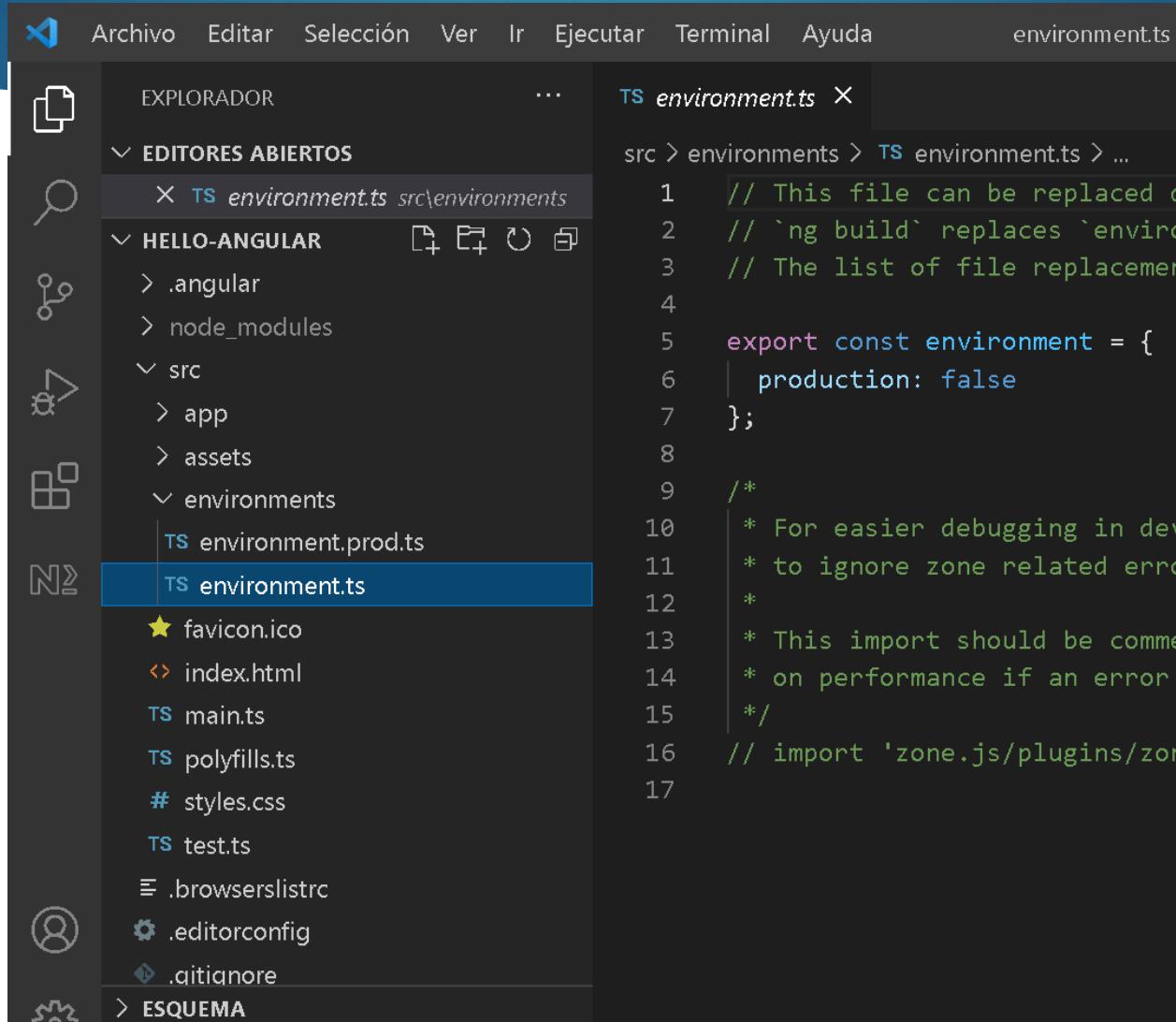
The screenshot shows the Visual Studio Code interface with the following details:

- Toolbar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** app.module.ts - hello-angular - Visual Studio Code [Administrador].
- Left Sidebar (Explorador):** Shows the project structure:
 - HELLO-ANGULAR
 - .angular
 - node_modules
 - src
 - app
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - ESQUEMA
 - LÍNEA DE TIEMPO
- Editor Area:** The file app.module.ts is open, showing its code. The file path is src > app > app.module.ts. The code defines the AppModule with imports from @angular/core and @angular/platform-browser, declarations of AppComponent, imports of BrowserModule, providers of [], and bootstrap of AppComponent. The file is saved at line 17.

En el directorio assets se guarda todo el contenido estático de nuestra aplicación



El directorio environments contiene los archivos de configuración de ambientes de desarrollo y producción de la aplicación

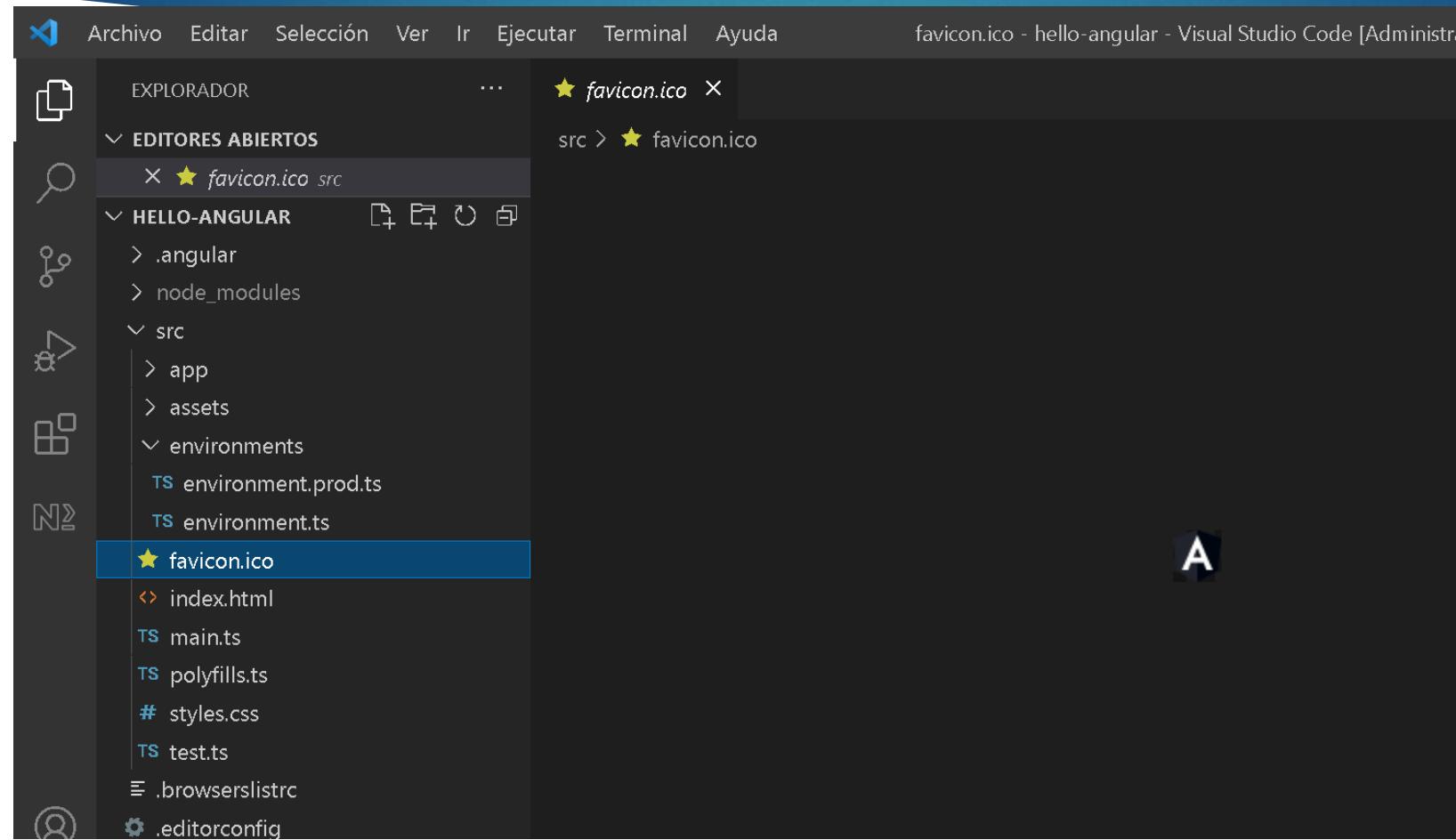


The screenshot shows the Visual Studio Code interface with the following details:

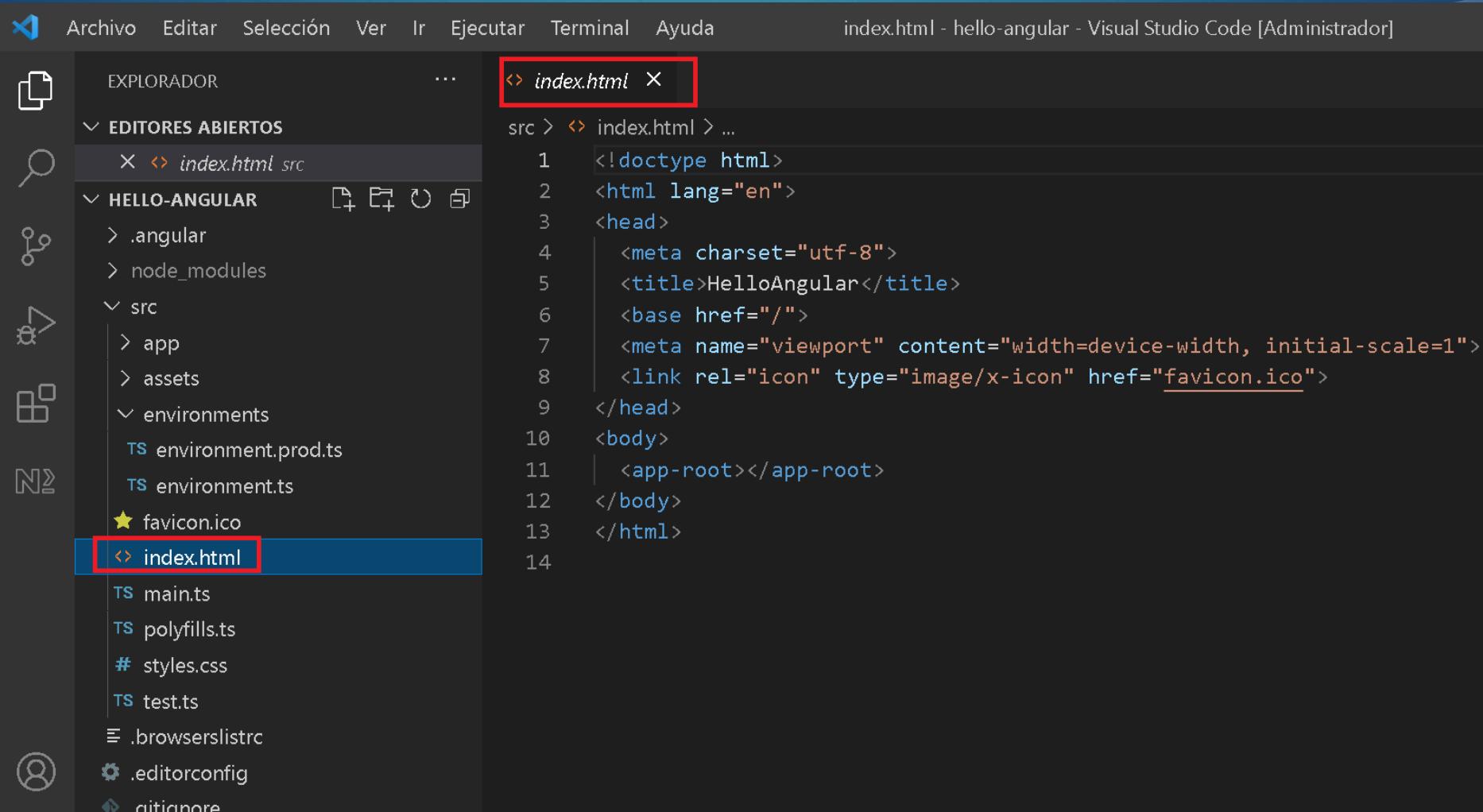
- Toolbar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Left Sidebar:** EXPLORADOR, EDITORES ABIERTOS (showing environment.ts), HELLO-ANGULAR (with .angular, node_modules, src, app, assets, environments), and other files like favicon.ico, index.html, main.ts, polyfills.ts, styles.css, test.ts, .browserslistrc, .editorconfig, and .gitignore.
- Right Editor Area:** The file environment.ts is open, showing the following code:

```
// This file can be replaced during the build` replaces `environment.ts`  
// The list of file replacement  
  
export const environment = {  
  production: false  
};  
  
/*  
 * For easier debugging in development, you can ignore zone related errors.  
 */  
  
/* This import should be commented out for better performance if an error is  
 */  
// import 'zone.js/plugins/zone-error';
```

El archivo favicon.ico es la imagen que nos muestra por default en el tab de los navegadores



El archivo index.html es la pagina principal de la aplicacion

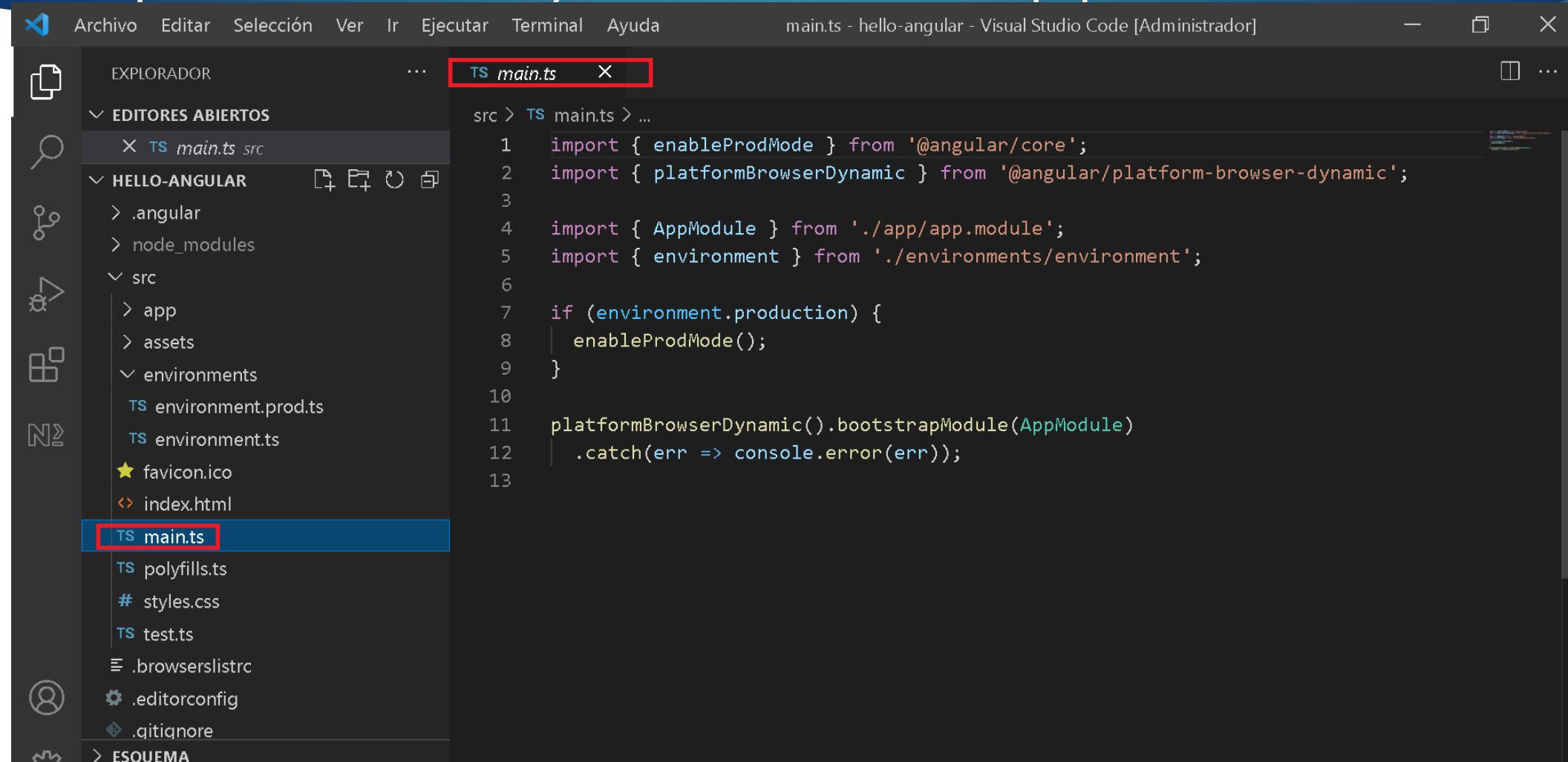


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure:
 - EDITORES ABIERTOS:** index.html (highlighted with a red box)
 - HELLO-ANGULAR:** .angular, node_modules, src
 - src:** app, assets, environments
 - environment.prod.ts (TS), environment.ts (TS)
 - favicon.ico**
- Editor (Right):** The file index.html is open, showing its content:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>HelloAngular</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```
- Status Bar:** index.html - hello-angular - Visual Studio Code [Administrador]

El archivo main.ts es la clase principal que levanta y arranca el app.module



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

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** main.ts - hello-angular - Visual Studio Code [Administrador].
- Left Sidebar (Explorador):** Shows the project structure:
 - src > main.ts (highlighted with a red box)
 - src > polyfills.ts
 - # styles.css
 - TS test.ts
 - .browserslistrc
 - .editorconfig
 - .gitignore
 - > ESQUEMA
- Central Area:** The main.ts file content is displayed:

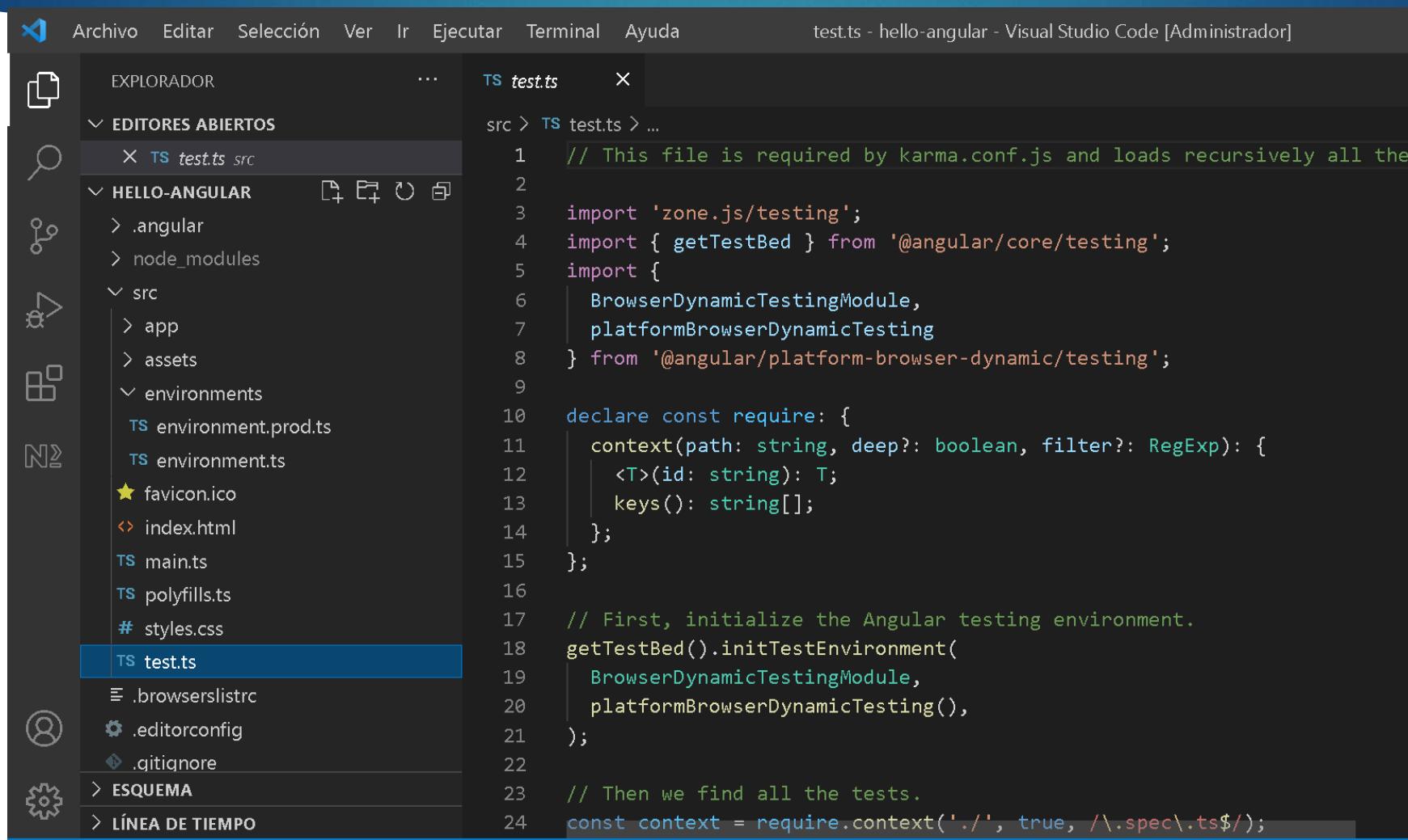
```
src > TS main.ts > ...
1 import { enableProdMode } from '@angular/core';
2 import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4 import { AppModule } from './app/app.module';
5 import { environment } from './environments/environment';
6
7 if (environment.production) {
8   enableProdMode();
9 }
10
11 platformBrowserDynamic().bootstrapModule(AppModule)
12   .catch(err => console.error(err));
13
```

El archivo style.css para aplicar estilos globales en la aplicación

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

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** styles.css - hello-angular - Visual Studio Code [Administrador]
- Explorer View:** Shows the project structure:
 - EDITORES ABIERTOS: # styles.css (highlighted with a red box)
 - HELLO-ANGULAR
 - .angular
 - node_modules
 - src
 - app
 - assets
 - environments
 - environment.prod.ts
 - environment.ts
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - # styles.css (highlighted with a blue bar)
 - test.ts
 - browserslistrc
 - editorconfig
 - gitignore
 - ESQUEMA
 - LÍNEA DE TIEMPO

El archivo test.ts para configuración de pruebas unitarias



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

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, Ayuda.
- Title Bar:** test.ts - hello-angular - Visual Studio Code [Administrador]
- Explorer View (Left):**
 - EXPLORADOR
 - EDITORES ABIERTOS
 - X TS testts src
 - HELLO-ANGULAR
 - > .angular
 - > node_modules
 - src
 - > app
 - > assets
 - environments
 - TS environment.prod.ts
 - TS environment.ts
 - ★ favicon.ico
 - ↳ index.html
 - TS main.ts
 - TS polyfills.ts
 - # styles.css
 - TS test.ts
 - .browserslistrc
 - .editorconfig
 - .gitignore
 - ESQUEMA
 - LÍNEA DE TIEMPO
- Editor View (Right):** The code editor displays the content of the 'test.ts' file, which is a configuration file for Karma testing. It includes imports for zone.js/testing, TestBed, and BrowserDynamicTestingModule, and defines a require function to initialize the Angular testing environment.

```
// This file is required by karma.conf.js and loads recursively all the
// angular2 related files.
import 'zone.js/testing';
import { TestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    <T>(id: string): T;
    keys(): string[];
  };
};

// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
);

// Then we find all the tests.
const context = require.context('./', true, /\.spec\.ts$/);
```