



Angular

2023

Indice

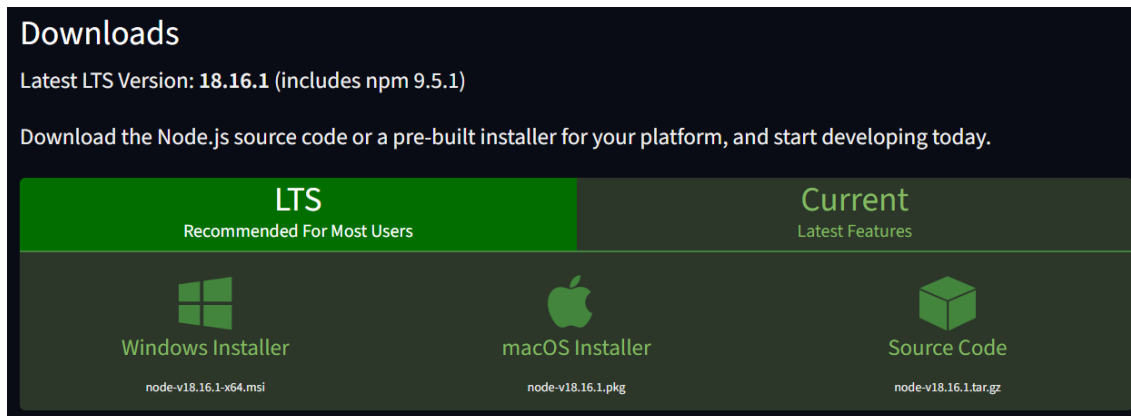
Node

Cliente de Comandos (CLI)

Creación de un primer proyecto

Node

Se puede instalar NodeJS desde la página oficial <https://nodejs.org/en/download>



La versión LTS es la más estable, mientras que Current es la última sin soporte actualizado.

Una vez descargado e instalado, comprobar que la instalación fue exitosa abriendo la consola del sistema y verificar la versión instalada.

```
CA: Command Prompt
Microsoft Windows [Version 10.0.17758.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Admin>Node --version
v14.17.3

C:\Users\Admin>npm --version
6.14.13

C:\Users\Admin>
```

— Verificación de la instalación de Node.js en Windows.



Cliente de comandos de Angular (CLI)

Desde el ambiente de Node

```
> node -v
v18.12.1
> npm i -g @angular/cli
npm WARN deprecated @npmcli/move-file@2.0.1: This functionality has been moved to
@npmcli/fs
npm WARN deprecated source-map-codec@1.4.8: Please use @jridgewell/source-map-codec
instead

changed 232 packages, and audited 233 packages in 8s

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

found 0 vulnerabilities
```

Para comprobar que este instalado

```
> ng version

Angular CLI
Angular CLI: 15.0.4
Node: 18.12.1
Package Manager: npm 8.19.2
```

Es recomendable descargar **Angular Dev-Tools** como extensión del navegador.

Creación de un primer proyecto

Para iniciar un proyecto nuevo con Angular, nos manejamos desde la consola de Node con el siguiente comando

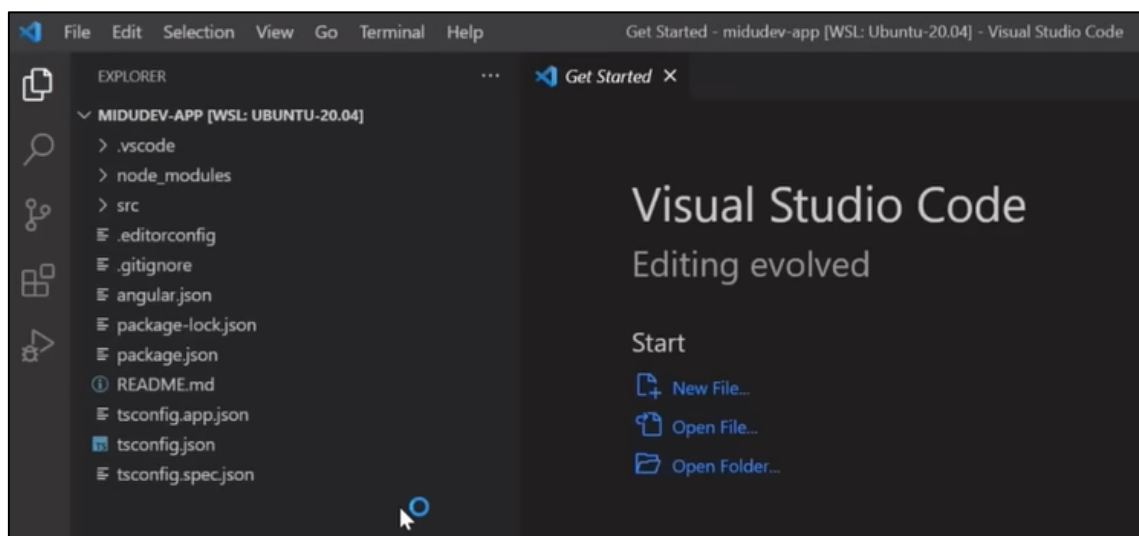
```
ng new nombre-app
```

Tras esto, nos preguntará algunas configuraciones

```
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use?
> CSS
```

Luego, comenzará a instalarse en una carpeta con el mismo nombre de la app.

Finalmente, para comenzar a programar, se ingresa dentro de la carpeta creada y se abre el Visual Studio Code. `> code .`

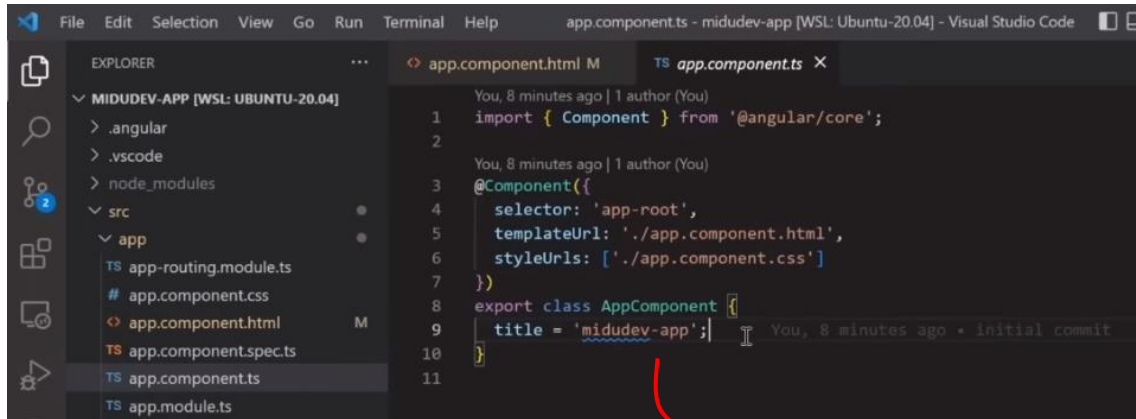


Comandos disponibles

- npm run start
 - ng serve
- genera una primera página de inicio y la muestra

Primeros pasos

Una vez creado un nuevo proyecto, se creará una carpeta llamada **src** donde se alojará nuestro código.



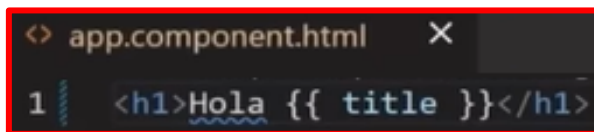
The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays the project structure: `MIDUDEV-APP [WSL: UBUNTU-20.04]` with subfolders `.angular`, `.vscode`, `node_modules`, and `src`. The `src` folder contains an `app` subfolder with files `app-routing.module.ts`, `app.component.css`, `app.component.html`, `app.component.spec.ts`, `app.component.ts`, and `app.module.ts`. The main editor shows the `app.component.ts` file 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   title = 'midudev-app';
10 }
11
```

Dentro del archivo **app.component.html** irá nuestro **index** de HTML.

En el archivo **app.component.css** irán los estilos.

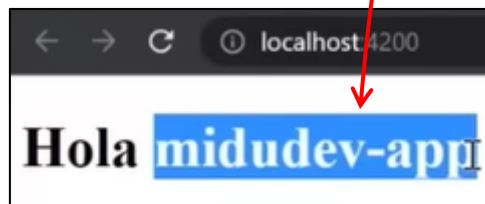
Una primera página de ejemplo puede ser la siguiente



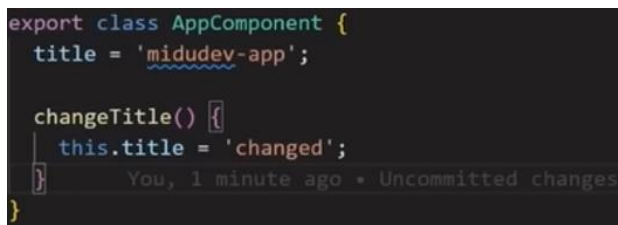
The screenshot shows the `app.component.html` file with the following code:

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

Se pasa como
parámetro a
nuestro HTML



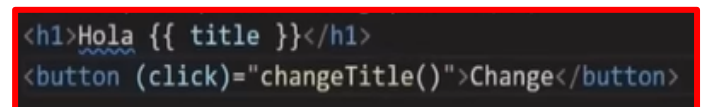
No sólo se pueden pasar variables como parámetros, sino funciones



The screenshot shows the `app.component.ts` file with the following code:

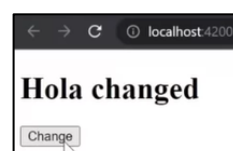
```
export class AppComponent {
  title = 'midudev-app';

  changeTitle() {
    this.title = 'changed';
  }
}
```



The screenshot shows the `app.component.html` file with the following code:

```
<h1>Hola {{ title }}</h1>
<button (click)="changeTitle()">Change</button>
```





Creación de nuevos componentes

Para crear un nuevo componente se utiliza la consola de Node para generarlo.

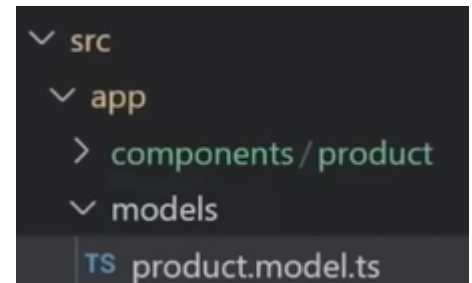
nombre del nuevo componente

```
> ng g c components/product
CREATE src/app/components/product/product.component.css (0 bytes)
CREATE src/app/components/product/product.component.html (22 bytes)
CREATE src/app/components/product/product.component.spec.ts (606 bytes)
CREATE src/app/components/product/product.component.ts (206 bytes)
UPDATE src/app/app.module.ts (490 bytes)
```

Prompt

El envío de datos de un componente a otro se debe realizar utilizando un model, el cual se establece como una **interface** que contendrá los atributos esperados.

```
export interface Product {
  id: number;
  title: string;
  price: number;
  images: string[];
}
```



Luego se debe indicar a nuestro nuevo componente que podrá recibir parámetros pormedio de un **input**.

Finalmente, para poder pasar de una página a otra, se utiliza el **selector** dentro de nuestro HTML.

```
TS product.model.ts TS product.component.ts U ●
1 import { Component, Input } from '@angular/
2
3 @Component({
4   selector: 'app-product',
5   templateUrl: './product.component.html',
6   styleUrls: ['./product.component.css']
7 })
8 export class ProductComponent {
9   @Input() title: string = '';
10 }
```



<https://youtu.be/sS90VVmBPcg?t=1499>