

32 - Herramienta Angular CLI - comando: ng new

Hemos visto que con el comando 'new' de la herramienta Angular CLI se crea un esqueleto de una aplicación Angular:

```
ng new proyecto025
```

Durante la generación del proyecto se nos consulta como queremos administrar las hojas de estilo (por defecto archivos CSS):

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\angulardevya> ng new proyecto025
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS [ https://developer.mozilla.org/docs/Web/CSS ]
  Sass (SCSS) [ https://sass-lang.com/documentation/syntax#scss ]
  Sass (Indented) [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]
```

Seguidamente se nos cuestiona si queremos activar el renderizado en el servidor (por defecto si presionamos la tecla enter se selecciona que NO queremos):

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\angulardevya> ng new proyecto025
✓ Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS ]
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? (y/N) |
```

El comando new podemos pasar otros parámetros para adaptar la aplicación generada.

- Podemos especificar el prefijo a los selectores generados mediante el parámetro --prefix

```
ng new proyecto025 --prefix ele
```

Luego se crea con prefijo 'ele' en lugar del valor por defecto 'app':

Si abrimos la componente creada por defecto podemos ver que el prefijo es 'ele':

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';

@Component({
  selector: 'ele-root',
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'proyecto025';
}
```

En lugar de 'app':

```
selector: 'app-root'
```

Luego si creamos una componente dentro del proyecto también se respetará el prefijo definido en la creación del proyecto:

```
ng generate component dado
```

El selector de la componente dado queda con el siguiente nombre:

```
import { Component } from '@angular/core';

@Component({
  selector: 'ele-dado',
  imports: [],
  templateUrl: './dado.component.html',
  styleUrls: ['./dado.component.css']
})
export class DadoComponent {

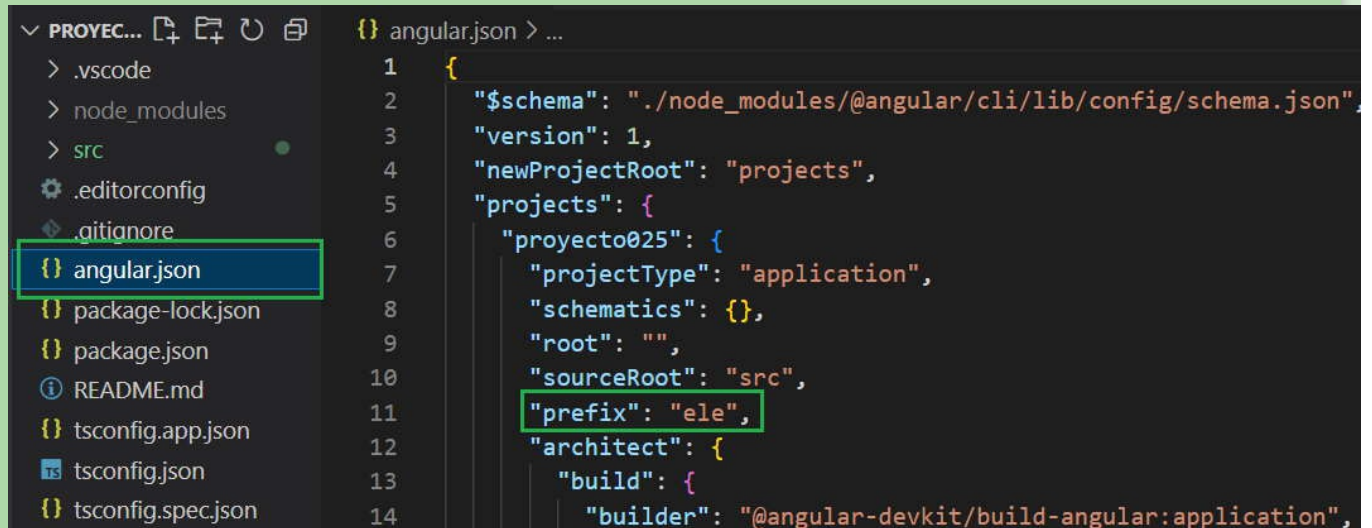
}
```

La definición de prefijos en las componentes de Angular permiten diferenciar las etiquetas nativas del navegador (ej. 'button') con las etiquetas propias de Angular (ej. ele-button)

Al crearse un proyecto uno de los archivos fundamentales que almacenan datos de configuración del mismo es 'angular.json' (se encuentra en

la carpeta raíz del proyecto)

En el archivo 'angular.json' se encuentra almacenado el prefijo definido al crearse el proyecto:



```
angular.json > ...
1  {
2    "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
3    "version": 1,
4    "newProjectRoot": "projects",
5    "projects": {
6      "proyecto025": {
7        "projectType": "application",
8        "schematics": {},
9        "root": "",
10       "sourceRoot": "src",
11       "prefix": "ele",
12       "architect": {
13         "build": {
14           "builder": "@angular-devkit/build-angular:application",
```

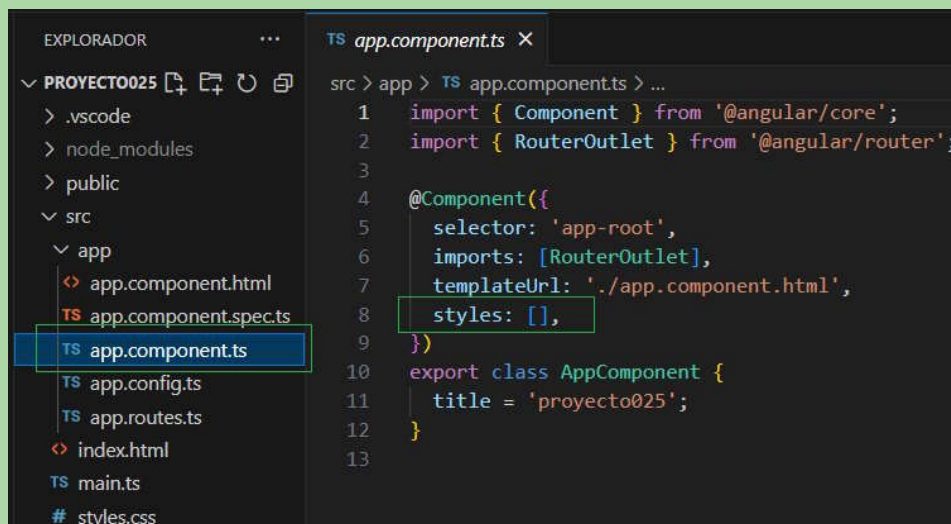
Si queremos escribir menos con Angular CLI al crear el proyecto podemos utilizar el alias -p en lugar de --prefix:

```
ng new proyecto025 -p ele
```

- Mediante el parámetro --inline-style (alias: -s) podemos evitar que se cree el archivo *.css y la definición de los estilos se deba hacer directamente en el archivo *.ts:

```
ng new proyecto025 --inline-style
```

Con esto no se crearán los archivos *.css para las componentes. El archivo *.ts de la componente tienen un lugar donde definir los estilos:



```
src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3
4  @Component({
5    selector: 'app-root',
6    imports: [RouterOutlet],
7    templateUrl: './app.component.html',
8    styles: [],
9  })
10 export class AppComponent {
11   title = 'proyecto025';
12 }
13
```

Puede tener sentido si la componente es bastante sencilla y no requiere la definición de una hoja de estilo compleja.

Probar modificar los archivos *.ts y *.html de la componente del proyecto.

app.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styles: ['.titulo { color:red; font-size:2rem }',
    '.parrafo {color:black; font-size:1.1rem }'],
})
export class AppComponent {
  tit = 'Prueba de inline-style';
}
```

app.component.html

```

<div style="text-align:center">
  <h1 class="titulo">{{tit}}</h1>
  <p class="parrafo">Esto es un párrafo.</p>
</div>

<router-outlet />

```

Si ejecutamos el proyecto podemos ver que los estilos se están recuperando del archivo *.ts:

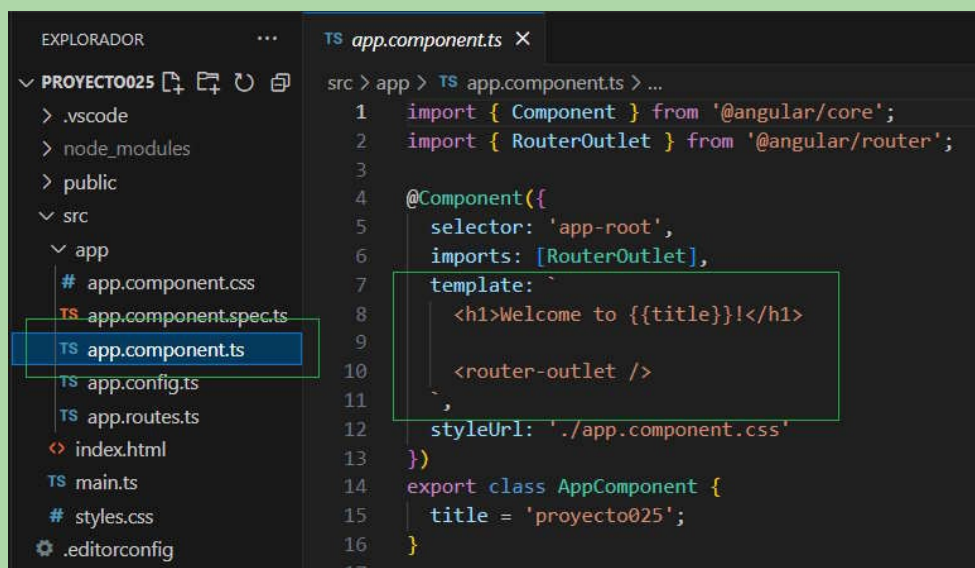


- Mediante el parámetro `--inline-template` (alias: `-s`) podemos evitar que se cree el archivo *.html y la definición del HTML se hace directamente en el archivo *.ts (borre primero el proyecto025 antes de crearlo nuevamente con esta nueva configuración):

```
ng new proyecto025 --inline-template
```

Con este parámetro estamos indicando que no se cree el archivo *.html para la componente y la definición del HTML se haga en el mismo archivo *.ts

Luego de creado el proyecto solo se han creado los archivos *.css y *.ts:



Nuevamente decimos que esto tiene sentido si la complejidad de la componente Angular no es grande.

Podemos inclusive evitar que se creen tanto el archivo *.css y *.html indicando ambos parámetros al crear la componente:

```
ng new proyecto025 --inline-style --inline-template
```

- Si tratamos de crear un proyecto y ya existe uno en la carpeta actual se produce un error al ejecutar:

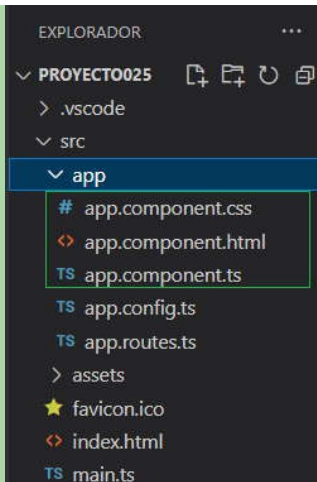
```
ng new proyecto025
```

Si queremos forzar la creación del proyecto y sobrescribir los archivos actuales debemos añadir el parámetro `--force` (alias `-f`):

```
ng new proyecto025 --force
```

- Si por algún motivo no queremos que Angular.CLI nos genere el archivo de test debemos pasar el parámetro `--skip-tests` (alias `-s`):

```
ng new proyecto025 --skip-tests
```



Como podemos comprobar no se ha generado el archivo 'app.component.spec.ts'.

- Otro parámetro más que podemos utilizar cuando vamos a crear un proyecto es `--dry-run` (alias: `-d`)

Mediante esta opción Angular CLI nos informa que archivos se crearán (no los crea) y a partir de estos datos poder tomar la decisión de crear o no el proyecto:

```
ng new proyecto025 --dry-run
```

```
PS C:\angulardevya> ng new proyecto025 --dry-run
? Which stylesheet format would you like to use? CSS
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? No
CREATE proyecto025/angular.json (2617 bytes)
CREATE proyecto025/package.json (1042 bytes)
CREATE proyecto025/README.md (1065 bytes)
CREATE proyecto025/tsconfig.json (909 bytes)
CREATE proyecto025/.editorconfig (274 bytes)
CREATE proyecto025/.gitignore (548 bytes)
CREATE proyecto025/tsconfig.app.json (263 bytes)
CREATE proyecto025/tsconfig.spec.json (273 bytes)
CREATE proyecto025/.vscode/extensions.json (130 bytes)
CREATE proyecto025/.vscode/launch.json (470 bytes)
CREATE proyecto025/.vscode/tasks.json (938 bytes)
CREATE proyecto025/src/main.ts (250 bytes)
CREATE proyecto025/src/favicon.ico (15086 bytes)
CREATE proyecto025/src/index.html (297 bytes)
CREATE proyecto025/src/styles.css (80 bytes)
CREATE proyecto025/src/app/app.component.html (20887 bytes)
CREATE proyecto025/src/app/app.component.spec.ts (931 bytes)
CREATE proyecto025/src/app/app.component.ts (372 bytes)
CREATE proyecto025/src/app/app.component.css (0 bytes)
CREATE proyecto025/src/app/app.config.ts (227 bytes)
CREATE proyecto025/src/app/app.routes.ts (77 bytes)
CREATE proyecto025/src/assets/.gitkeep (0 bytes)

NOTE: The "--dry-run" option means no changes were made.
PS C:\angulardevya> |
```

Retornar