

SUB-MÓDULOS

Sub-módulos y sus componentes

Como ya comentamos en el curso anterior cuando hablamos de los módulos, Angular es un *framework* que trabaja de forma modular por lo que podemos dividir nuestra aplicación en módulos más pequeños según funcionalidades.

Para crear un nuevo módulo usaremos el siguiente comando:

“

ng generate module nameModule

”

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

@NgModule({
  declarations: [],
  imports: [
    CommonModule
  ]
})
export class DashboardModule { }
```

SUB-MÓDULO

El código por defecto es similar al *AppMódulo* que tenemos en nuestro proyecto.

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

¿QUÉ MÓDULO USO?

Para determinar nuestro módulo raíz usaremos el archivo *main.ts*, que ya mencionamos en el curso anterior, para que Angular identifique a su módulo principal.

CREAR COMPONENTES DENTRO DE LOS SUB-MÓDULOS

“

ng generate component sub-modulo/componente

”