



red.es

PROFESIONALES
DIGITALES

Formación
Continua

El FSE invierte en tu futuro



UNIÓN EUROPEA

Conceptos avanzados de Angular



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

NgModule API

```
@NgModule({  
  // Static, that is compiler configuration  
  declarations: [], // Configure the selectors  
  entryComponents: [], // Generate the host factory  
  
  // Runtime, or injector configuration  
  providers: [], // Runtime injector configuration  
  
  // Composability / Grouping  
  imports: [], // composing NgModules together  
  exports: [] // making NgModules available to other parts of  
  the app  
})
```

Propiedad	Descripción
<i>declarations</i>	Clases declarables (componentes, directivas y <i>pipes</i>). Todas estas clases se declaran en un solo módulo, según necesitemos, si no dará fallo el compilador.
<i>providers</i>	Clases inyectables (<i>Services</i> y algunas clases). Angular tiene asociado a cada módulo un <u>instructor</u> , por lo que el módulo raíz tiene su instructor raíz y cada submódulo tiene el suyo propio que parte del principal.
<i>imports</i>	Todos aquellos módulos o librerías de Angular que se vayan a utilizar en la aplicación.
<i>exports</i>	Clases declarables (componentes, directivas y <i>pipes</i>) que quieran ser importadas por otro módulo
<i>bootstrap</i>	Componente de arranque de la aplicación
<i>entryComponents</i>	Componentes que pueden ser cargados dinámicamente en la vista



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

***El FSE** invierte en tu futuro*

Multimodulado

Angular es modular, tiene la posibilidad de dividir sus aplicaciones en diferentes áreas. Esto aligera la primera petición al servidor para cargar la aplicación web además de organizar nuestro proyecto por diferentes funcionalidades.

```
ng generate module nameModule
```

Una vez creado el submódulo, para crear cualquier componente y añadirlo a ese submódulo:

```
ng generate component nameComponent
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Para visualizar los componentes de esos submódulos podemos hacerlo de dos maneras:

- A través de los *exports* del submódulo.
- Añadiendo rutas al submódulo y a sus componentes.

Para esta segunda opción debemos de ir al *app-routing* e introducir la ruta del submódulo:

```
{ path: 'customer', loadChildren: '.. rutadelarchivo/customer-dashboard.module#CustomerDashboarModule' }
```

Una vez añadida la ruta al submódulo, dentro de este debemos crear las rutas pertinentes a sus componentes:

```
...
imports: [
  ...
  RouterModule.forChild([
    { path: "", component: CustomerDashboardComponent }
  ]),
  ...
]
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Providers

Providers es una instrucción del Inyector de Dependencias para obtener el valor de una dependencia. Esto quiere decir que todos los archivos que Angular detecte que se almacenan en el ID deben de ser importados en el *providers* para poder ser utilizados. La mayoría de estos archivos suelen ser *services*. Dependiendo del nivel donde importemos ese archivo podremos usarlo:

- Toda la aplicación
 - Providers del módulo principal
 - `@Injectable({ providedIn: 'root' })`
- Para un solo submódulo
 - Providers de ese submódulo
 - `@Injectable({ providedIn: nombreModulo })`
- Para un solo componente
 - Providers de ese componente



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Routing - Children

Hasta ahora se había visto como crear rutas sencillas entre componentes. Pero se pueden hacer varios niveles de enrutado para una aplicación web.

```
const crisisCenterRoutes: Routes = [  
  {  
    path: 'crisis-center', component: CrisisCenterComponent, children: [  
      {  
        path: '', component: CrisisListComponent, children: [  
          {  
            path: ':id', component: CrisisDetailComponent  
          },  
          {  
            path: '', component: CrisisCenterHomeComponent  
          }  
        ]  
      }  
    ]  
  }  
];
```




red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Routing - CanActivate

Se utiliza para limitar a los usuarios el acceso a las rutas que nosotros no queramos que accedan.

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot,
RouterStateSnapshot } from '@angular/router';

@Injectable({
  providedIn: 'root',
})
export class AuthGuard implements CanActivate {
  canActivate( next: ActivatedRouteSnapshot, state:
RouterStateSnapshot): boolean {
    // Código para comprobar el token o clave para
devolver true o false. Lo habitual justo antes de devolver
false se reenvie con la clase router de Angular a otra
vista, si no la app se quedará parada en este punto.
  }
}
```

{ path: 'categorias', canActivate: [AuthGuard], component:
'CategoriasComponent' }

El FSE invierte en tu futuro

Eventos

Durante la navegación, el objeto *Router* emite una serie de eventos a través de la propiedad *Router.events*.

```
this.router.events.subscribe( (event) =>
{
  console.log(event);
});
```

NavigationStart	NavigationEnd	NavigationError
GuardsCheckEnd	ChildActivationStart	ResolveStart