

Tabla de contenido

[**BOOTSTRAP 4** 2](#_Toc64484473)

[Opción 1: Bootstrap + Ng-bootstrap 2](#_Toc64484474)

[Opción 2: Sólo Bootstrap CSS 2](#_Toc64484475)

[**FONT-AWESOME** 3](#_Toc64484476)

[**USO FORMULARIOS** 3](#_Toc64484477)

[**SINTAXIS PLANTILLAS** 4](#_Toc64484478)

[**NG-MODULES** 6](#_Toc64484479)

[**COMPONENTES** 7](#_Toc64484480)

[**CICLO DE VIDA** 8](#_Toc64484481)

[**DECORADORES DE CLASE** 9](#_Toc64484482)

[**DECORADORES** 9](#_Toc64484483)

[**DIRECTIVAS** 10](#_Toc64484484)

[**ROUTING & NAVEGACIÓN** 13](#_Toc64484485)

[Configuración 13](#_Toc64484486)

[Sistema de navegación 14](#_Toc64484487)

[Routing con código 17](#_Toc64484488)

[Routing con parámetros 18](#_Toc64484489)

[Ir a la pantalla anterior 19](#_Toc64484490)

[Redirigir con programación 20](#_Toc64484491)

[**SERVICIOS** 22](#_Toc64484492)

[Configuración 22](#_Toc64484493)

[**ANGULAR CLI** 23](#_Toc64484494)

# **BOOTSTRAP 4**

## Opción 1: Bootstrap + Ng-bootstrap

1. **Dependencia Bootstrap:**

npm install bootstrap --save

1. **Importar el css de BS en styles.css:**

Añadir: @import "~bootstrap/dist/css/bootstrap.css";

1. **Instalación ng-bootstrap:**

npm install --save @ng-bootstrap/ng-bootstrap

1. **Añadir @angular/localize:**

ng add @angular/localize

1. **Añadir import de NgbModule en src/app/app.module.ts:**

import {NgbModule} from '@ng-bootstrap/ng-bootstrap';

@NgModule({

// [...]

imports: [NgbModule, /\* [...] \*/],

// [...]

})

export class AppModule {

}

## Opción 2: Sólo Bootstrap CSS

1. **Dependencia Bootstrap:**

npm install bootstrap --save

1. **Importar el css de BS en styles.css:**

Añadir: @import "~bootstrap/dist/css/bootstrap.css";

# **FONT-AWESOME**

1. **Instalación**

npm install font-awesome --save

1. **Añadir a angular.json**

"styles": [

"src/styles.css",

"node\_modules/font-awesome/css/font-awesome.css"],

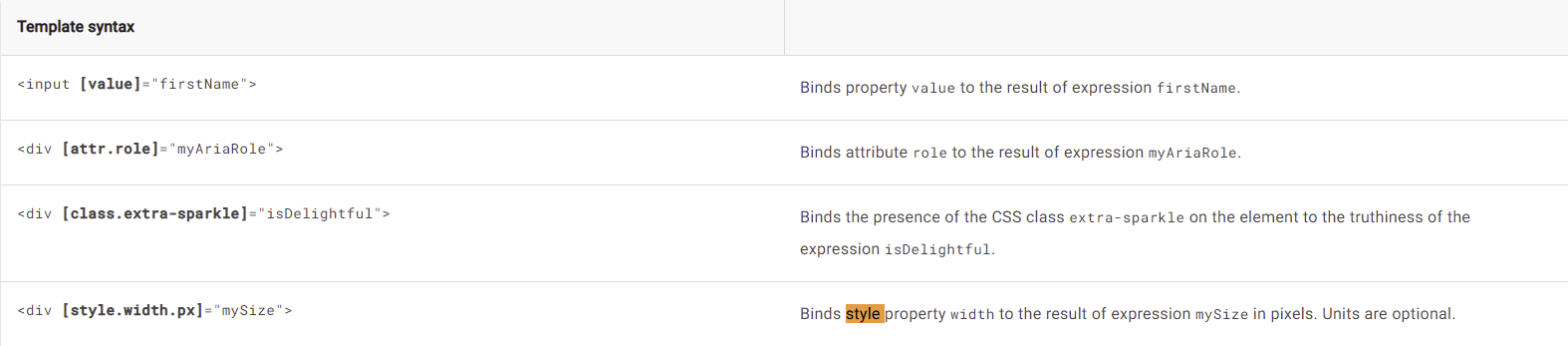
# **USO FORMULARIOS**

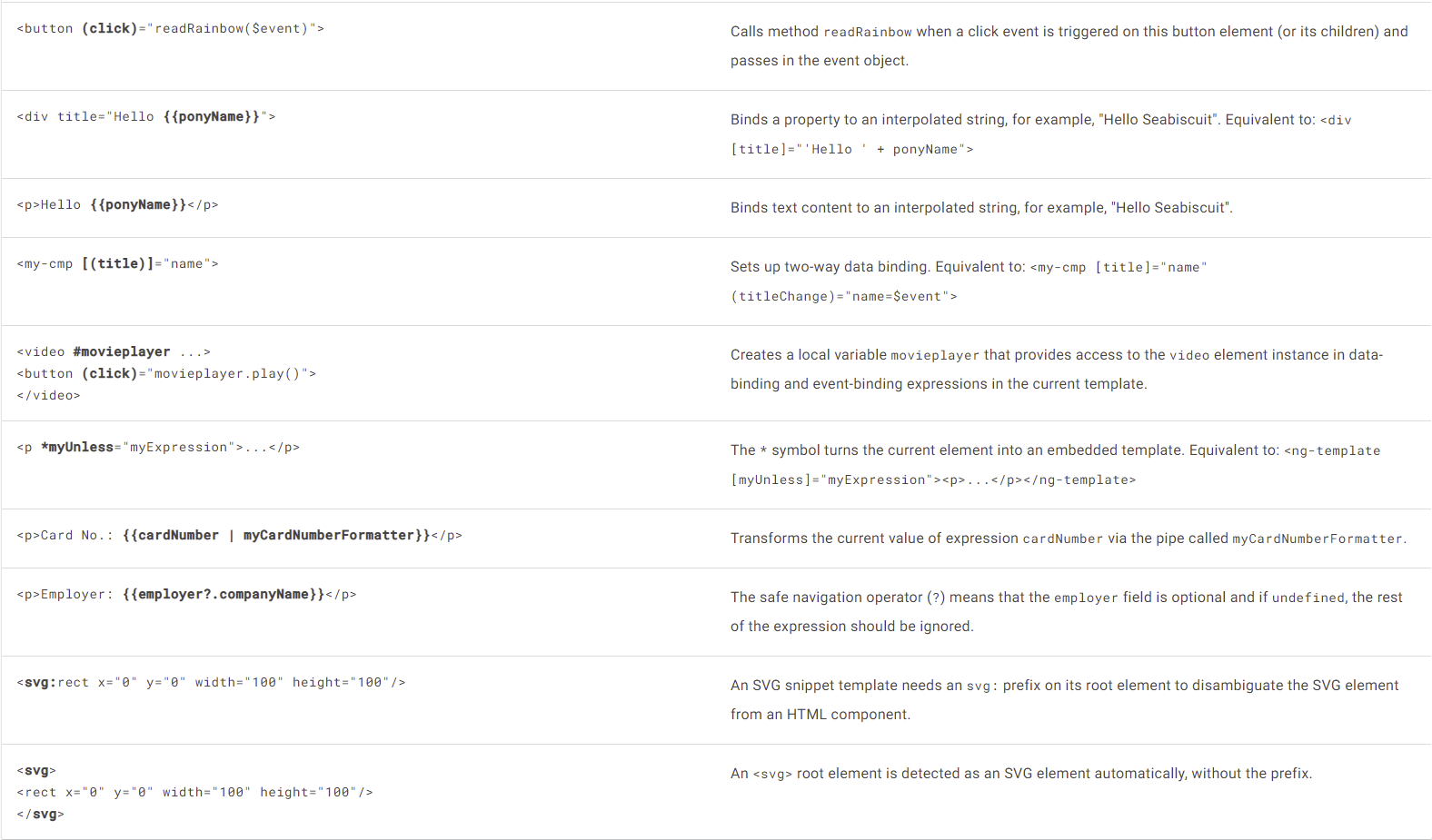
1. **Importar en apps.module.ts**

FormsModule

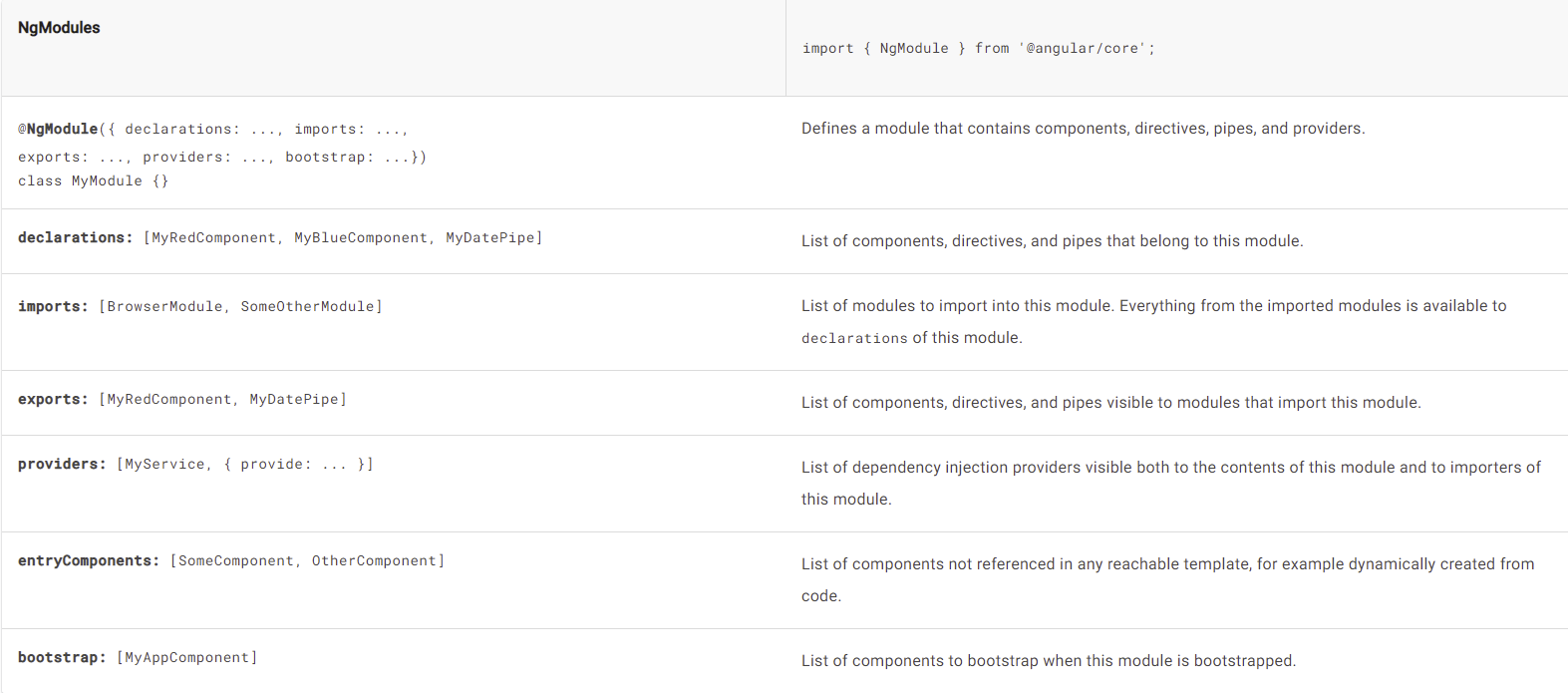
ReactiveFormsModule

# **SINTAXIS PLANTILLAS**

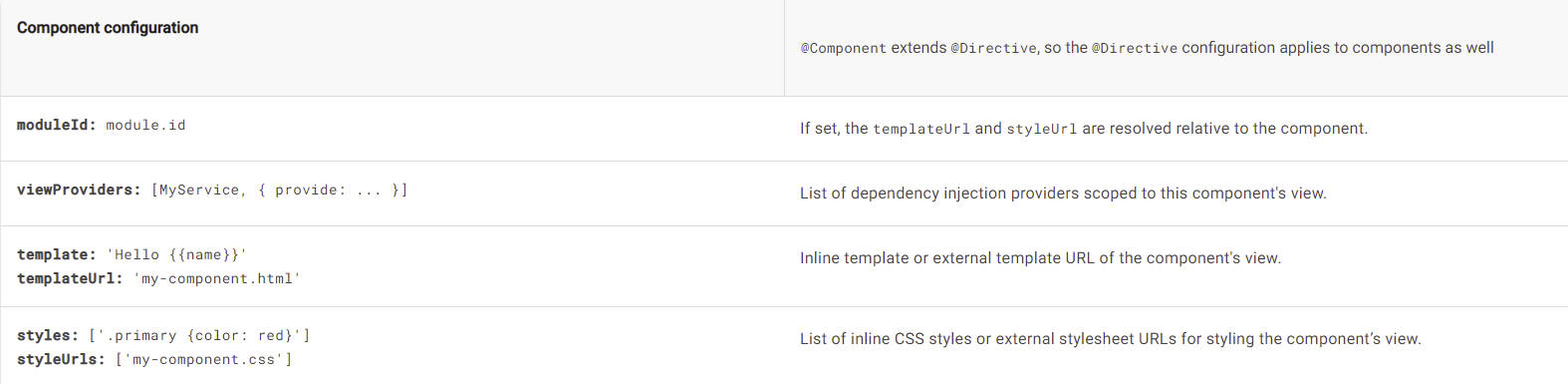




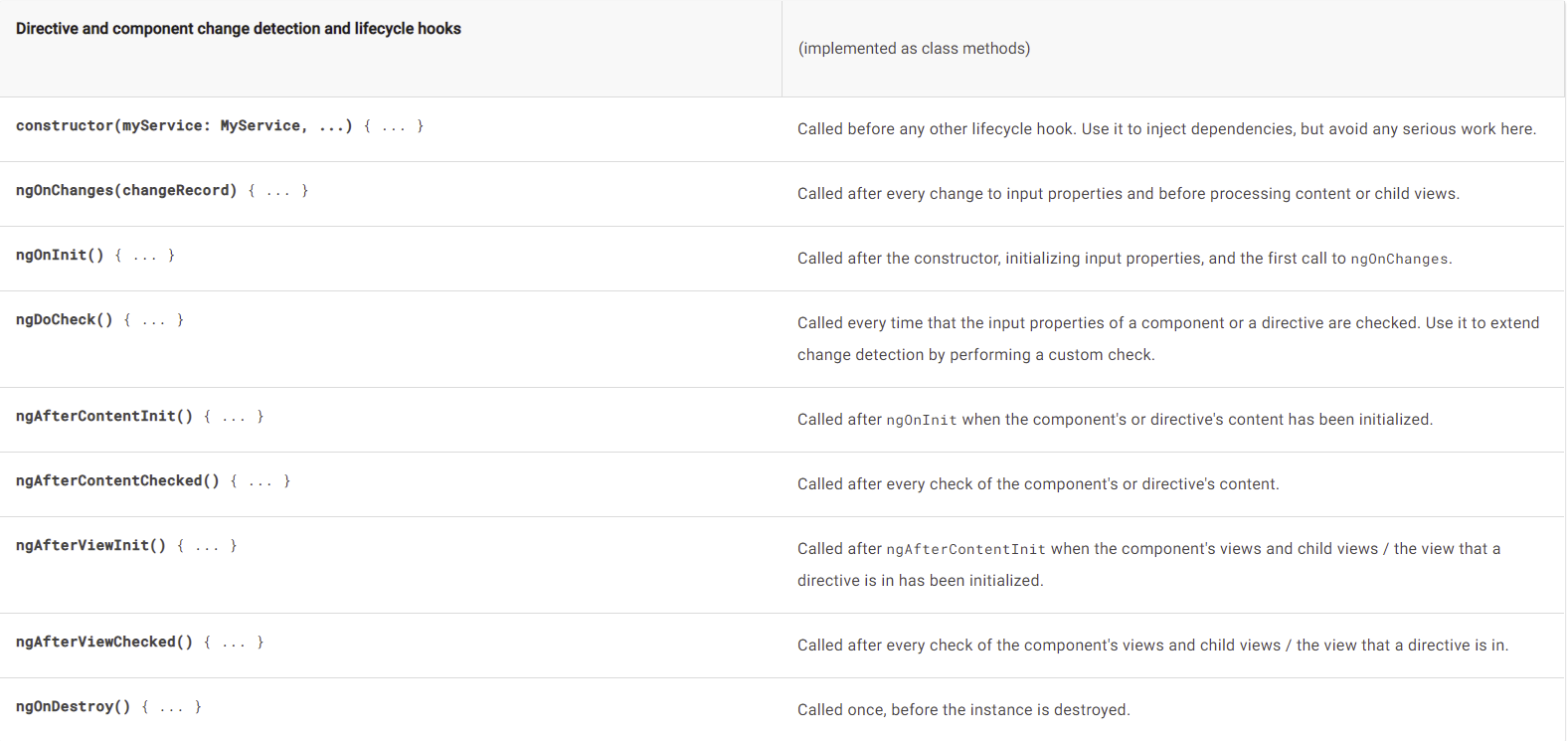
# **NG-MODULES**

****

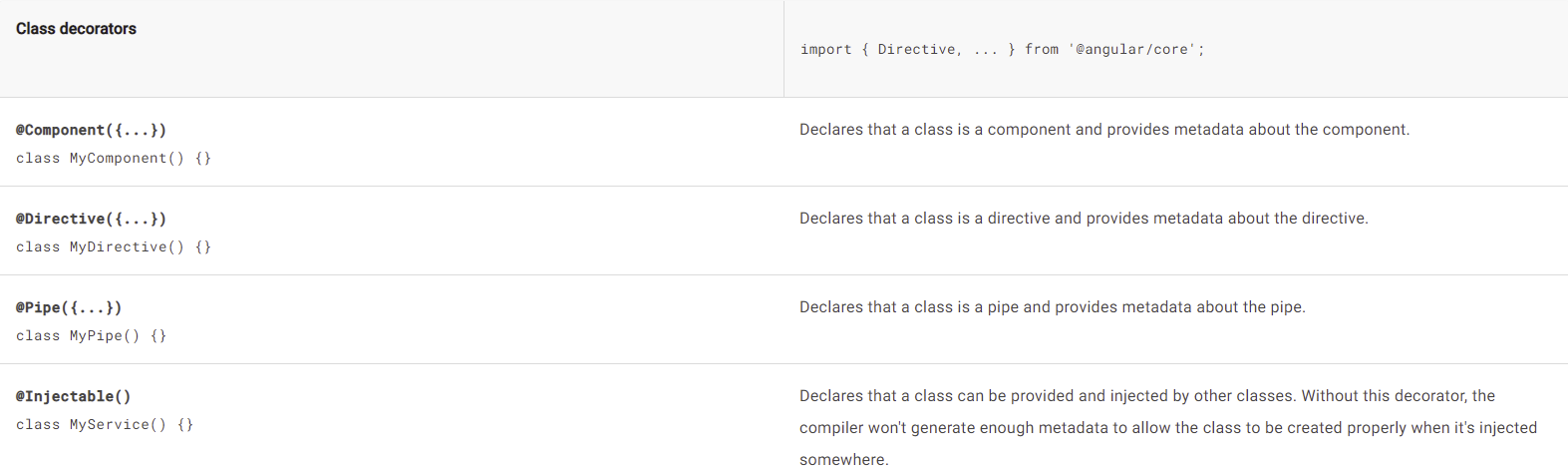
# **COMPONENTES**



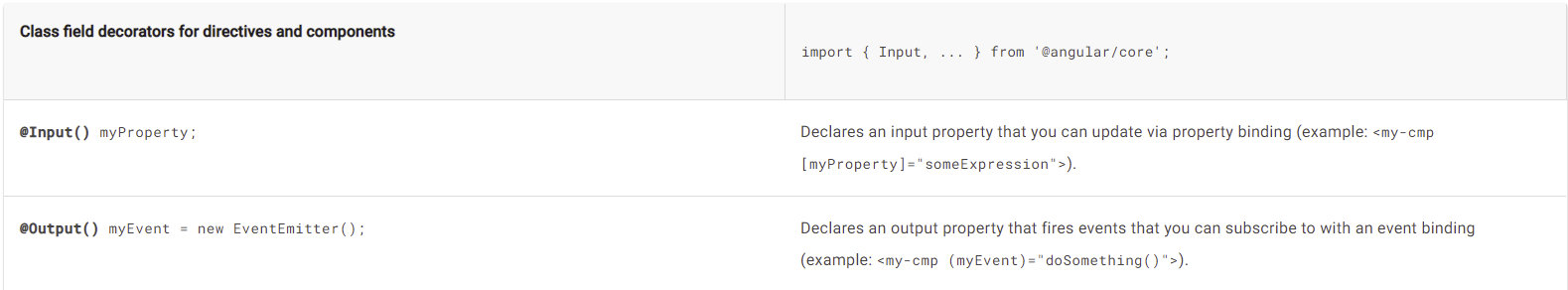
# **CICLO DE VIDA**



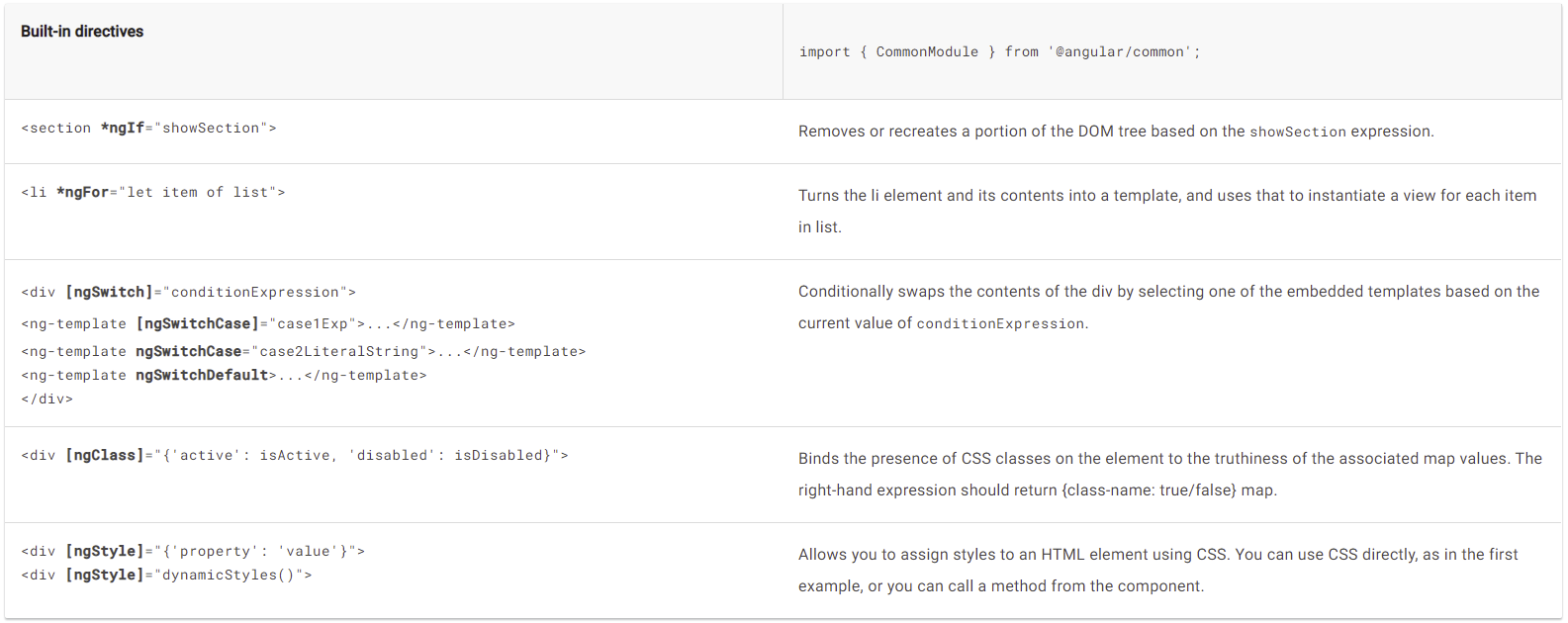
# **DECORADORES DE CLASE**

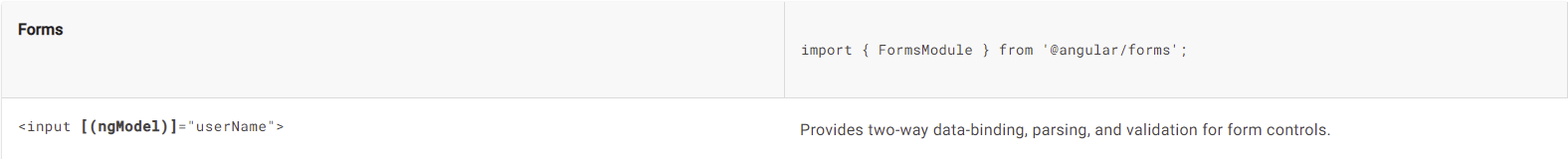
****

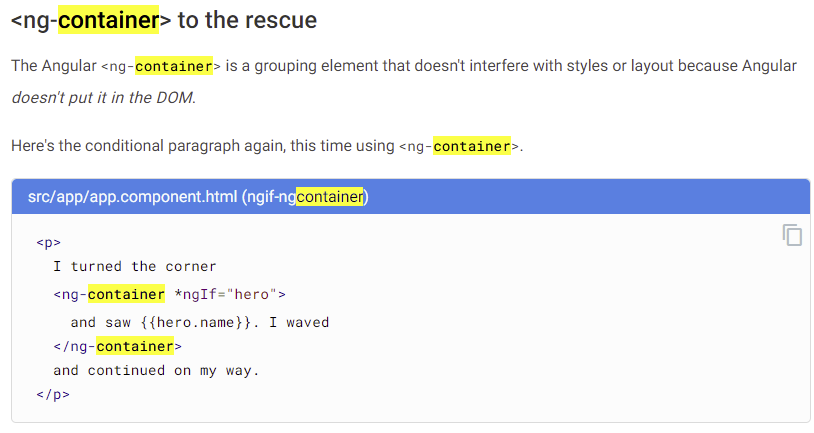
# **DECORADORES**

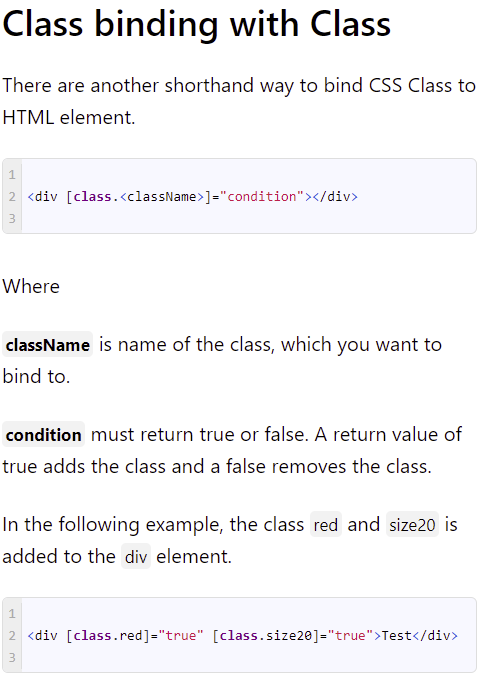


# **DIRECTIVAS**









# **ROUTING & NAVEGACIÓN**

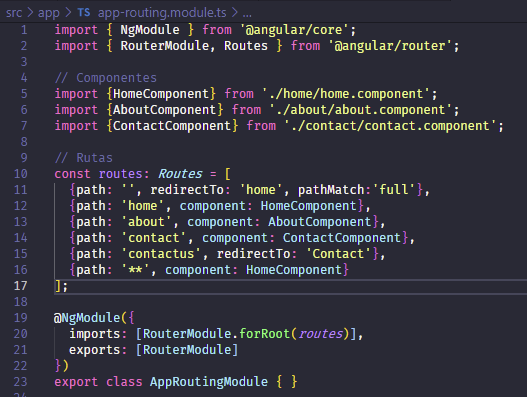
## Configuración

1. **Generar módulo AppRoutingmodule:**

ng generate module app-routing –flat

1. **Se puede borrar lo relativo al declarations de @NgModule y lo relativo al CommonModule.**
2. **Añadir el import de RouterModule y Routes:**

import { RouterModule, Routes } from '@angular/router';

1. **Importar los componentes que queramos usar.**
2. **Añadir las rutas.**
3. **Inicializar el router:**

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

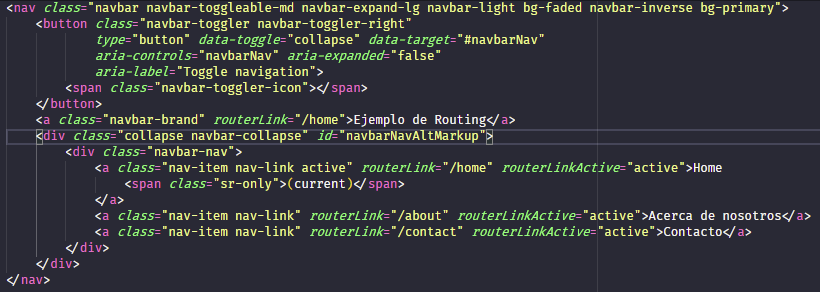
})

1. **Añadir <router-oulet> donde queramos que renderice.**

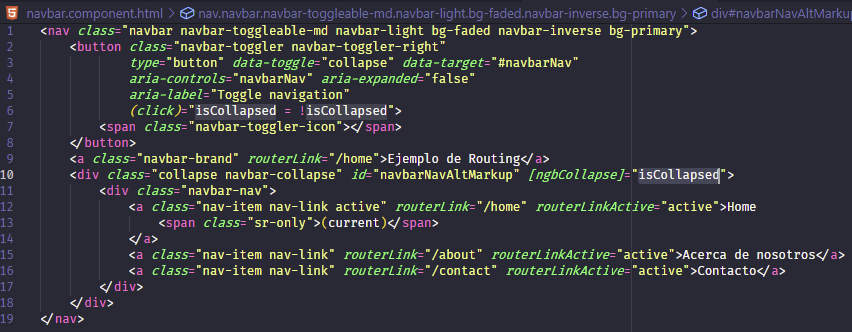
## Sistema de navegación

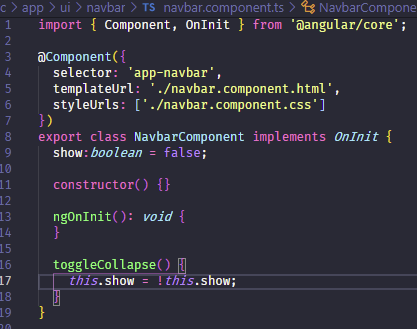
1. **Unificar en carpeta UI.**
2. **Generar componente (navbar = 1 componente)**

ng generate module app-routing –flat

1. **Cambiar href por routerLink**
2. **Añadir routerLinkActive=”active” a cada routerLink.**
3. **Añadir navbar a la plantilla. **
4. **Para que la hamburguesa funcione.**

* Opción A: Con ng-bootstrap:

****

* ****Opción B: Mediante TS:

****

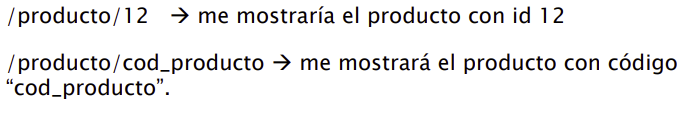
## Routing con código

**Con el servicio Router:**

****



## Routing con parámetros

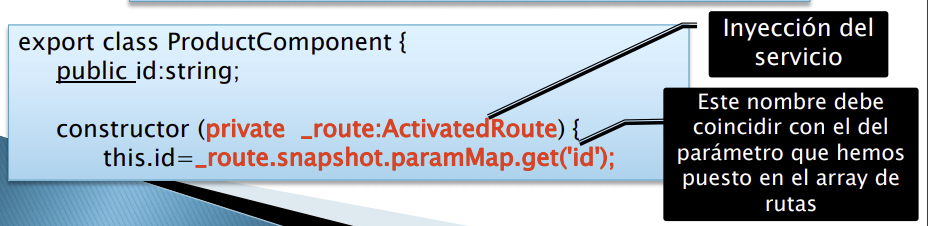
****



1. **Recibir parámetros:**

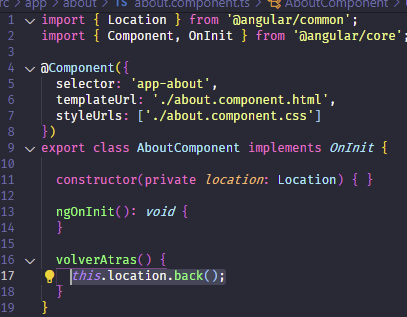
Inyectando el servicio ActivatedRoute.





## Ir a la pantalla anterior

**Con el servicio Location:**



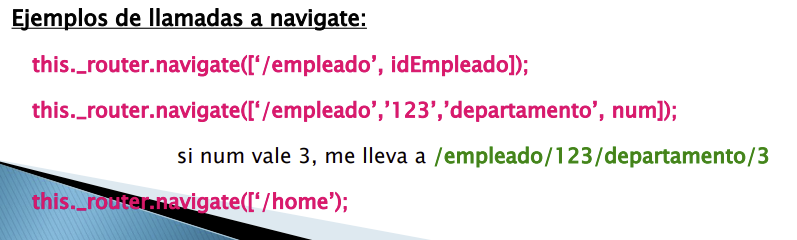


## Redirigir con programación

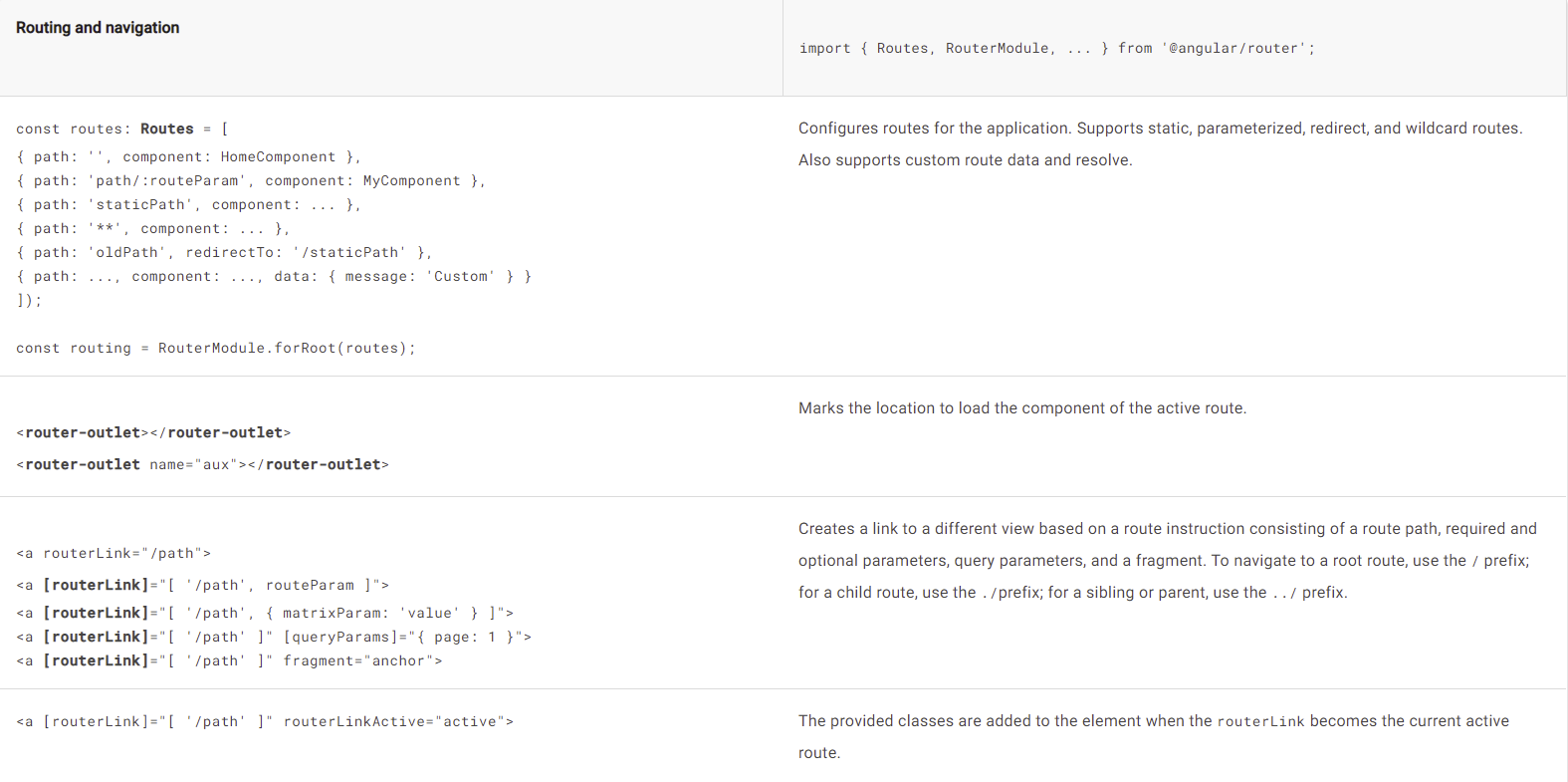
1. **import { Router } from ‘@angular/router’;**
2. **Inyectar el servicio Router:**

constructor ( private \_router: Router) {}

1. **Llamar al método navigate() pasando array con la dirección y subdirección**

****

\_\_



# **SERVICIOS**

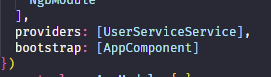
## Configuración

* Servicio = lógica de negocio && acceso a datos.
* Es una clase con el decorador @Injectable.
* Una única instancia compartida(Singleton).
* Cada clase que lo quiera usar lo inyecta en su constructor.
* Se pueden inyectar varios servicios.
* Un servicio puede ser inyectado dentro de otro.

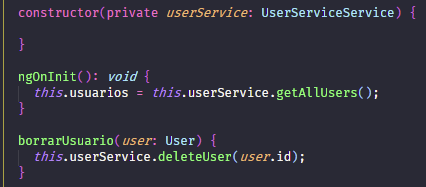
1. **Crear un servicio**

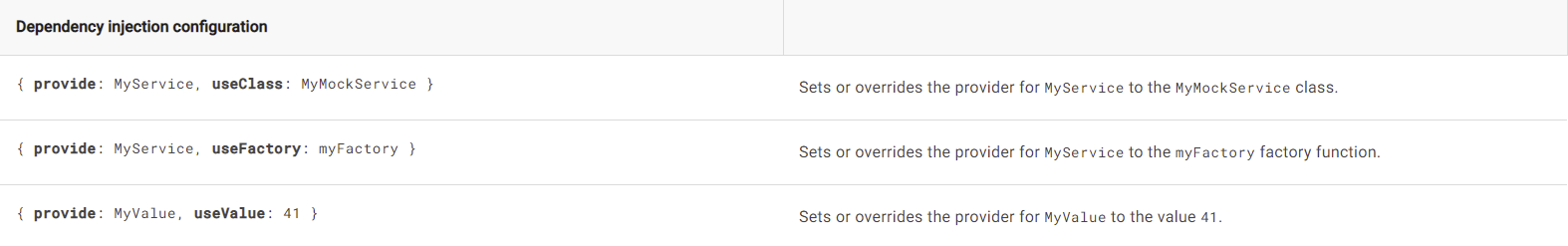
ng generate service nombreServicio

1. **Registrar el proveedor en app.module:**

****

1. **Inyectarlo donde lo queramos usar:**

****

****

# **ANGULAR CLI**

