



*Apuntes de Angular. Juan Carlos Fernández García.*  
*03 – Angular , bootstrap y rutas*

### Primeros pasos con Angular.

En este documento vamos a hacer una aplicación base en Angular. Después vamos a comentar la estructura de la aplicación e incorporaremos bootstrap. Una vez visto lo anterior, vamos a crear 3 páginas dentro de la aplicación utilizando las rutas de Angular en las que iremos incorporando funcionalidades. De momento una página serán las directivas estructurales.

### Manos a la obra.

Lo primero que tenemos que tener claro es si tenemos angular instalado.

Para ello en la consola de comandos escribimos:

`ng -v`

```
$ ng -v

Angular CLI 6.1.5
Node: 10.14.2
OS: win32 x64
Angular:
...
Package                           Version
-----
@angular-devkit/architect         0.7.5
@angular-devkit/core              0.7.5
@angular-devkit/schematics        0.7.5
@schematics/angular              0.7.5
@schematics/update                0.7.5
rxjs                             6.2.2
typescript                       2.9.2
```

Si no lo tenemos instalado o es una versión anterior a la 6, pues hay que instalarlo con

```
npm install -g @angular/cli
```



*Apuntes de Angular. Juan Carlos Fernández García.*  
*03 – Angular , bootstrap y rutas*

Recuerdo que para poder instalarlo con la opción `-g` (global), hay que iniciar el cmd con permiso de administrador.

Una vez instalado, vamos a crear una carpeta donde tendremos los proyectos de angular y en ella vamos a crear nuestro primer proyecto.

Por ejemplo, en el raíz de c: he puesto una carpeta llamada `ProyectosAngular`

En ese directorio escribimos `ng new AngularApp` y de esta forma vamos a crear nuestra primera aplicación Angular.

Cuando termine, escribiremos `ng serve -o` para que ejecute el servidor y con la opción `-o` abra directamente el navegador.

```
D:\>cd ProyectosAngular

D:\ProyectosAngular>dir
El volumen de la unidad D es DATOS
El número de serie del volumen es: D428-F807

Directorio de D:\ProyectosAngular

10/01/2019  07:45    <DIR>          .
10/01/2019  07:45    <DIR>          ..
               0 archivos              0 bytes
               2 dirs  1.426.654.695.424 bytes libres

D:\ProyectosAngular>ng new AngularApp
CREATE AngularApp/angular.json (3584 bytes)
CREATE AngularApp/package.json (1316 bytes)
CREATE AngularApp/README.md (1027 bytes)
CREATE AngularApp/tsconfig.json (408 bytes)
CREATE AngularApp/tslint.json (2805 bytes)
CREATE AngularApp/.editorconfig (245 bytes)
CREATE AngularApp/.gitignore (503 bytes)
```

```
D:\ProyectosAngular>dir
El volumen de la unidad D es DATOS
El número de serie del volumen es: D428-F807

Directorio de D:\ProyectosAngular

10/01/2019  07:47    <DIR>          .
10/01/2019  07:47    <DIR>          ..
10/01/2019  07:49    <DIR>          AngularApp
               0 archivos              0 bytes
               3 dirs  1.426.329.427.968 bytes libres

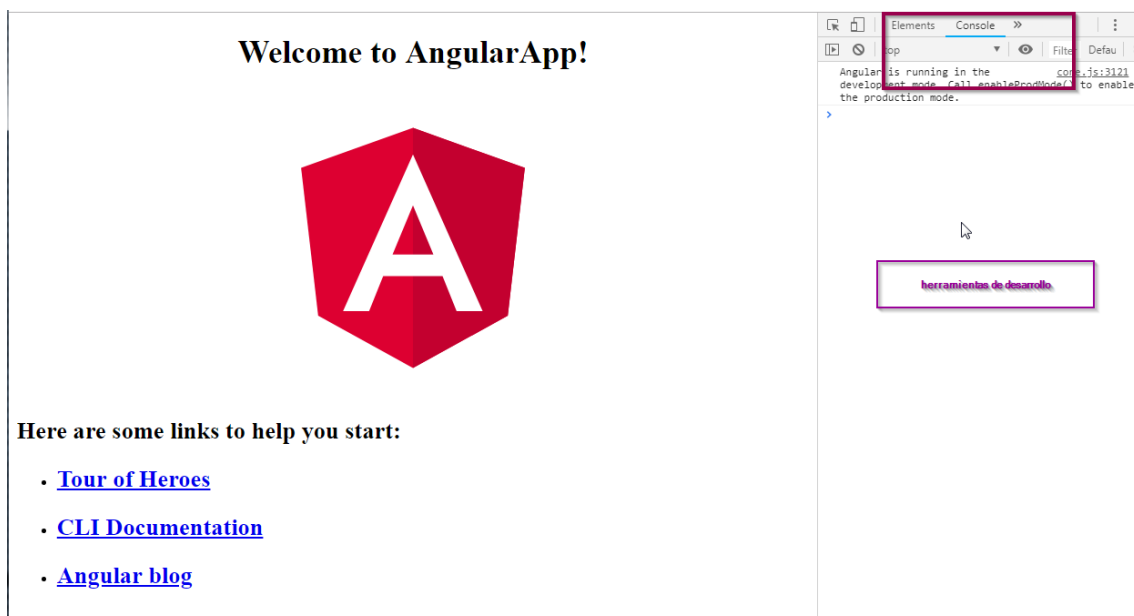
D:\ProyectosAngular>_
```

Cuando termine, veremos que ha creado una carpeta con el nombre `AngularApp`.

Dentro de la **carpeta** escribiremos `ng serve -o` y esperamos un momento....



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



Ya tenemos en el navegador en la url <http://localhost:4200/> nuestra aplicación corriendo.

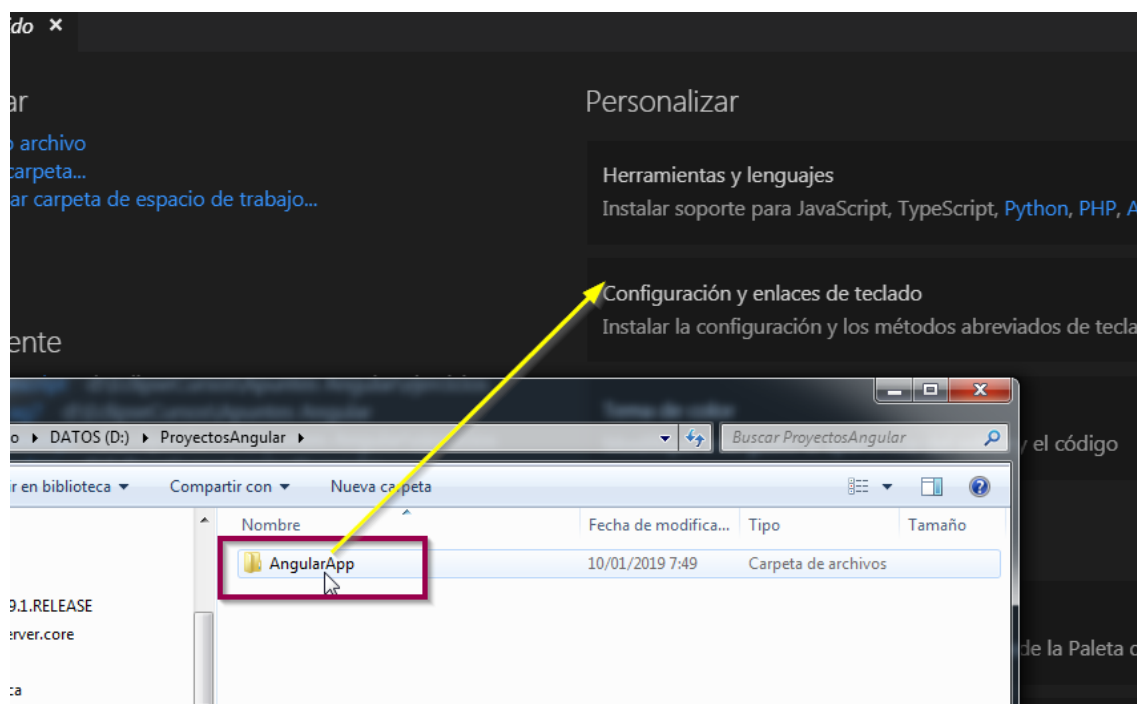
Siempre que estemos en el entorno de desarrollo, tendremos abiertas las herramientas de desarrollo, en la pestaña console. Esas herramientas se abren con la tecla F12.

**Editamos el proyecto en visual studio code.**

**En VSC, se puede o arrastrar la carpeta del proyecto o abrir la carpeta con la opción Archivo-> Abrir carpeta**



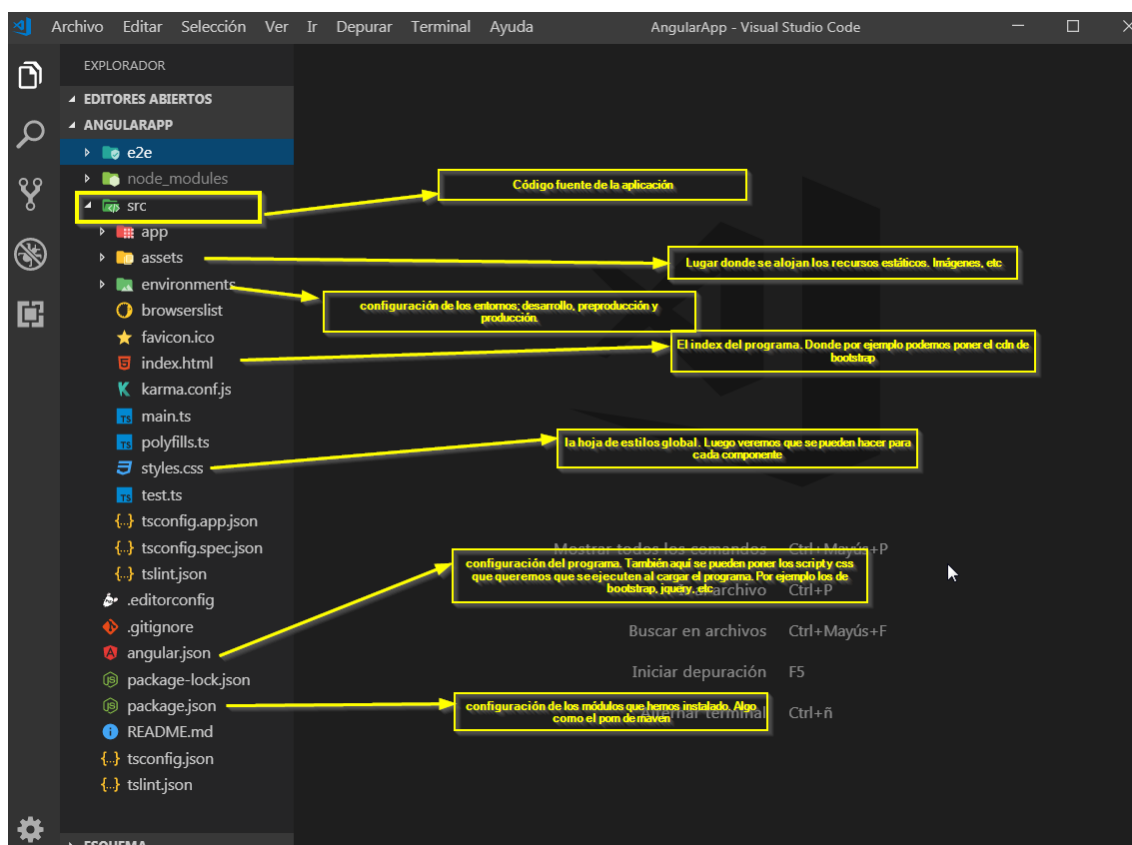
**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



Esto es lo que tendremos en el VSC



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



No he contado lo que hay en la carpeta Node\_modules, pero esos módulos son los que se instalan cuando se genera el programa. Además, estos módulos se ignoran en el .gitignore para no subirse a los repositorios, ya que el package.json tiene configurados los paquetes necesarios para funcionar y al descargarse este repositorio y poner npm install, instalaría todos esos paquetes.

Además, cuando se genera el paquete para producción, eso no va.

*Para más información sobre esto, me podéis dar un toque.*

Esto es lo que tenemos y los directorios son así .

Sin cerrar el servidor, en el fichero index.html vamos a incorporar bootstrap, para ello, en la página de bootstrap

<https://getbootstrap.com/>

en GetStarted

copiamos los css y los js los incluimos en el index.html



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Quedará así

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>AngularApp</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.mi
n.css" integrity="sha384-
GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMP0DgiMDm4iYMj70gZWKYbI706tWS"
crossorigin="anonymous">
</head>

<body>
  <app-root></app-root>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.m
in.js" integrity="sha384-
wHAIffRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6Z0blgut"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.
js" integrity="sha384-
B0UglyR+jN6CkvvICOB2joaf5I413gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script>
</body>
</html>
```

Y ya la página de nuestro programa tendrá otra apariencia.



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

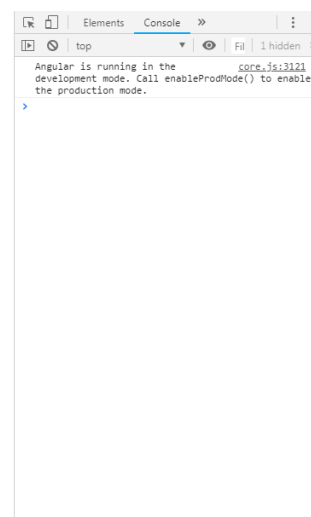
Si además no hay errores en la consola, es que vamos bien.

Welcome to AngularApp!



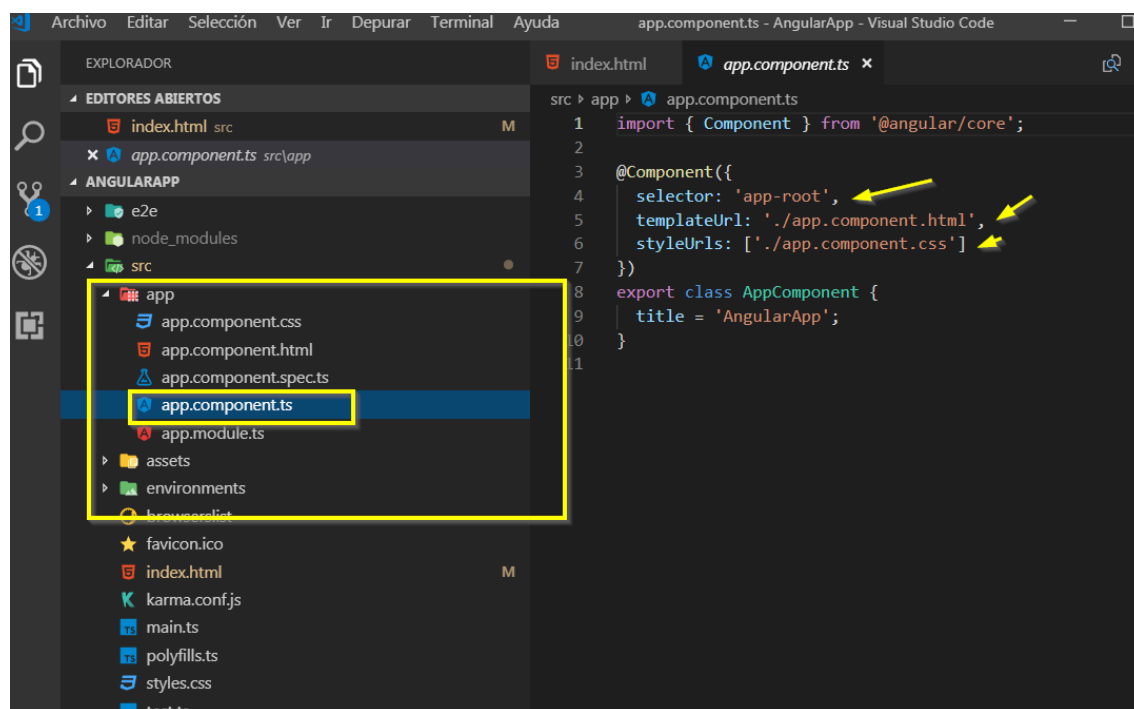
Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)



Vamos a editar nuestro componente principal.

Este componente es el app. Concretamente es el app.component.ts





**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

### **Pero, ¿Qué es un componente?**

Un componente en Angular, no es mas que una clase TypeScript decorada con la anotación `@Component`. Al decorarlo con esta anotación, hay que importar desde `@angular/core`.

Además, al crear este componente, se va a poder “inyectar” en cualquier parte que se requiera en la aplicación.

Como todos los componentes de angular, tiene un selector, que este caso se llama `app-root`, un template, que hace referencia a una página html y donde irá todo “bindido” con los valores de la clase principal del componente. Además incluye o puede incluir un css propio que solo estará disponible para el componente con el que se está trabajando.

El selector, es el nombre que le damos para poder ponerlo en el html que necesitamos. En el programa base, este componente se le llama desde el `index.html` poniendo:

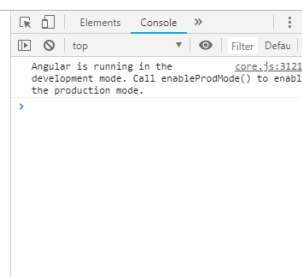
```
<app-root></app-root>
```

Ahora, si vamos al `app.component.html` vamos a quitar unas pocas cosas y lo dejamos únicamente con el título.

```
<h1>
  Welcome to {{ title }}!
</h1>
```

En la pantalla estará así:

**Welcome to AngularApp!**



Pero por qué sale `Welcome to AngularApp!`,

Como vemos en el `app.component.html`, hay algo que está de la forma

`{{ title }}`





**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Lo que está haciendo es que en la clase del componente le configura cual es su html y cada vez que ve algo entre llaves, va a buscar en la clase .ts el valor de esa propiedad.

Esto de las llaves, se llama interpolación de string y puede ponerse entre las llaves cualquier expresión javascript, por ejemplo una suma, una función, etc. Y en este sitio aparecería renderizado el valor.

Por eso si en app.component.ts, donde pone

```
title = 'AngularApp';
```

ponemos otra cosa, por ejemplo

```
title = 'Primera aplicación con Angular';
```

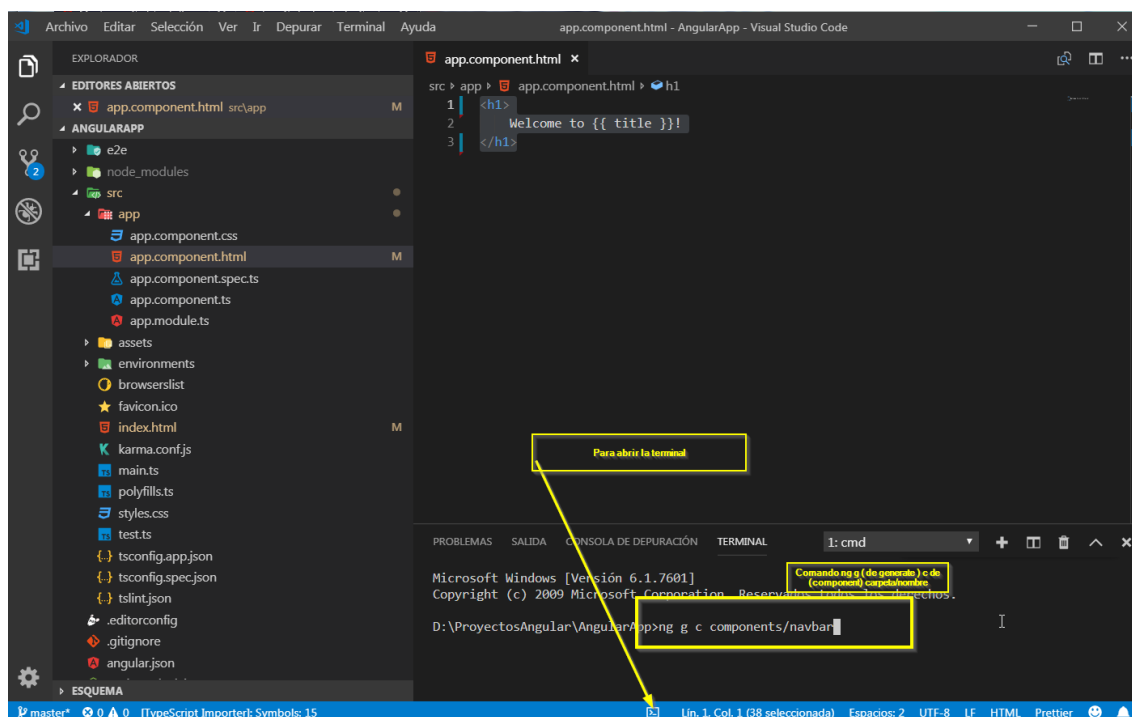
Veremos el cambio automáticamente.

Vamos a crear ahora un componente para poner un navbar de bootstrap.

Lo vamos a crear de forma automática con el “cli” de angular.

Para ello, abrimos la consola de comandos dentro del directorio del proyecto o abrimos la terminal del VSC y escribimos:

ng g c components/navbar





*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

veremos que nos informa de lo que ha hecho

```
D:\ProyectosAngular\AngularApp>ng g c components/navbar
CREATE src/app/components/navbar/navbar.component.html (25 bytes)
CREATE src/app/components/navbar/navbar.component.spec.ts (628 bytes)
CREATE src/app/components/navbar/navbar.component.ts (269 bytes)
CREATE src/app/components/navbar/navbar.component.css (0 bytes)
UPDATE src/app/app.module.ts (407 bytes)

D:\ProyectosAngular\AngularApp>
```

Lo que nos informa es que ha creado el html del componente, el fichero spec.ts que es el de las pruebas, el fichero ts, que es la clase typescript del componente y el css específico de este componente. Normalmente, no se suele utilizar un css exclusivo para cada componente, es mas normal, utilizar el style.css que es global.

También, **y esto es importante, nos informa que hay modificado el fichero app.module.ts.**

Este archivo, es el que contiene las referencias a todos los módulos que se van a utilizar en la aplicación. Tanto los módulos que hemos creado, como los servicios, etc.

Así es como tiene que estar el app.module.ts.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';
import { NavbarComponent } from './components/navbar/navbar.component';

@NgModule({
  declarations: [
    AppComponent,
    NavbarComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```



*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

Ahora, si abrimos el fichero navbar.component.ts, veremos que contiene las referencias a su html, a su css y su nombre de selector.

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

@Component({
  selector: 'app-navbar',
  templateUrl: './navbar.component.html',
  styleUrls: ['./navbar.component.css']
})
export class NavbarComponent implements OnInit {

  constructor() { }

  ngOnInit() {
  }

}
```

Abrimos entonces el html, para ver lo que contiene....

```
<p>
  navbar works!
</p>
```

Y es esto lo único que tiene.

Para probar:

Pues en app.component.html, vamos a poner la referencia a este componente de la forma:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

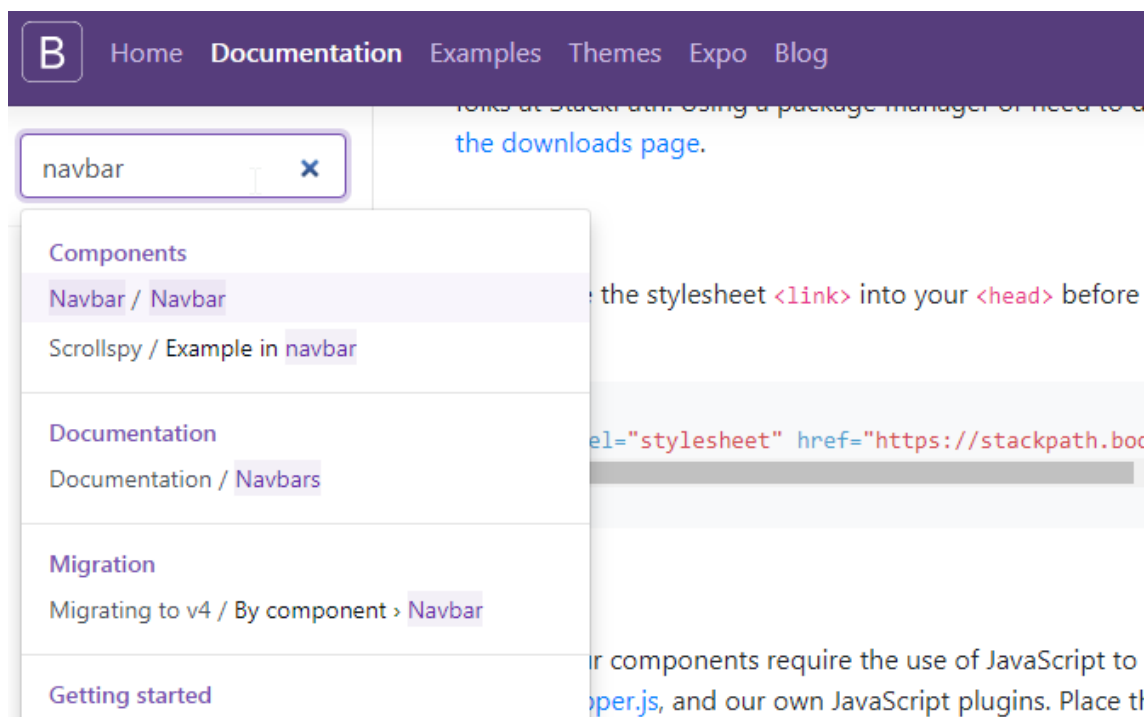
```
<app-navbar></app-navbar>

<h1>
  Welcome to {{ title }}!
</h1>
```

Y el programa ahora tendrá esta pinta:



En bootstrap, vamos a buscar un navbar para incluirlo en nuestro componente navbar.component.html





## Apuntes de Angular. Juan Carlos Fernández García. 03 – Angular , bootstrap y rutas

- `.navbar-text` for adding vertically centered strings of text.
- `.collapse.navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint.

Here's an example of all the sub-components included in a responsive light-themed navbar that automatically collapses at the `lg` (large) breakpoint.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
```

Esta queda chula.

Vamos a copiar y pegar esto en nuestro `navbar.component.html`, sustituyendo lo que hay

```
<p>
  navbar works!
</p>
```

Por lo que copiamos de bootstrap.

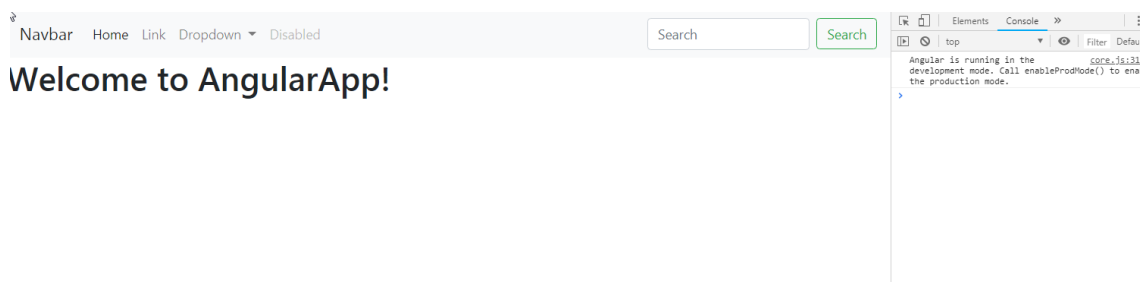
Quedará así:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
4 <span class="navbar-toggler-icon"></span>
5 </button>
6
7 <div class="collapse navbar-collapse" id="navbarSupportedContent">
8   <ul class="navbar-nav mr-auto">
9     <li class="nav-item active">
10       <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
11     </li>
12     <li class="nav-item">
13       <a class="nav-link" href="#">Link</a>
14     </li>
15     <li class="nav-item dropdown">
16       <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown">Dropdown
17     </a>
18     <div class="dropdown-menu" aria-labelledby="navbarDropdown">
19       <a class="dropdown-item" href="#">Action</a>
20       <a class="dropdown-item" href="#">Another action</a>
21       <div class="dropdown-divider"></div>
22       <a class="dropdown-item" href="#">Something else here</a>
23     </div>
24   </li>
25   <li class="nav-item">
26     <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled
27   </li>
28 </ul>
29
30 <form class="form-inline my-2 my-lg-0">
31   <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
32   <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search
33 </form>
34 </div>
```

y nuestra aplicación irá así:



Una vez que lo tenemos así, vamos a cambiar un poco nuestro Navbar para adaptarlo a lo que vamos a hacer en este ejercicio.

Cambiaremos los títulos en navbar.component.html para dejarlo así:

Y he quitado donde pone el Dropdown, y he puesto lo de Search en español, por si luego hacemos una búsqueda en un servicio.

Nos tiene que ir quedando así:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Angular Directivas Servicios Disabled

Buscar

Buscar

# Welcome to AngularApp!

Vamos a cambiar un poco los colores de nuestro navbar .

En el navbar.component.html en la línea

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
```

La dejamos así:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
```

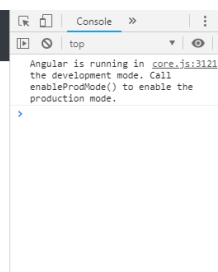
Nos debería quedar así:

Angular Directivas Servicios Disabled

Buscar

Buscar

# Welcome to AngularApp!

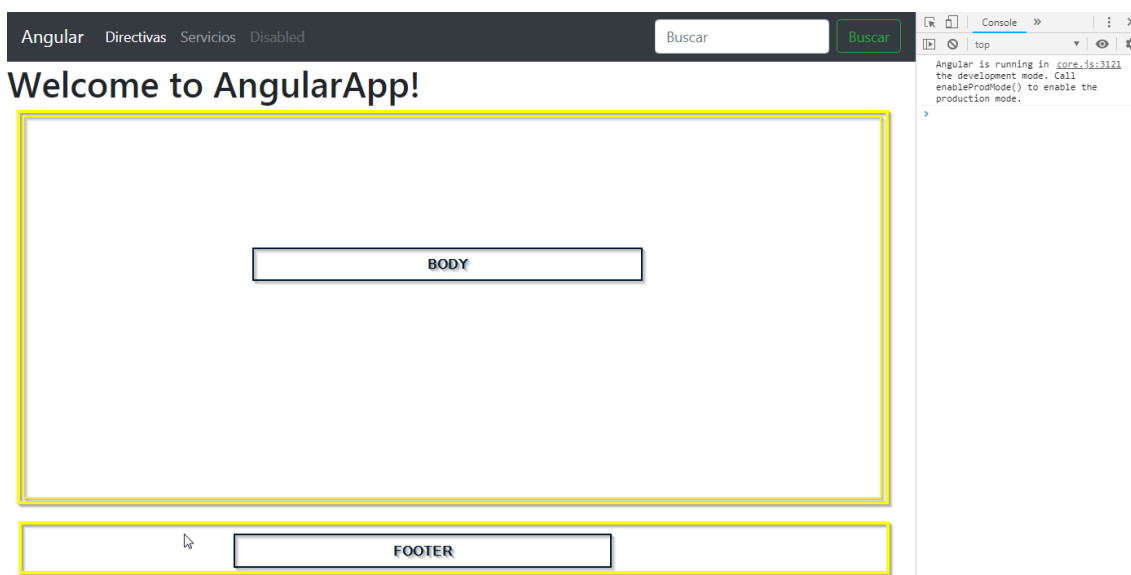


Ahora vamos a crear otro par de componentes.

Uno para un footer y otro para el body del programa, esto es lo que vamos a hacer ahora.



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



Creando el body component

En la terminal, decimos y escribimos “ng g de generate , c de componente carpeta components barra el nombre del componente body y además no queremos que nos genere el archivo de pruebas” De forma que quedará así:

```
ng g c components/body --spec=false
```

Una vez terminado, veremos que nos saca esta información de lo que ha creado y actualizado:

```
PROBLEMAS  TERMINAL  ...  1: cmd

Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

D:\ProyectosAngular\AngularApp>ng g c components/body --spec=false
CREATE src/app/components/body/body.component.html (23 bytes)
CREATE src/app/components/body/body.component.ts (261 bytes)
CREATE src/app/components/body/body.component.css (0 bytes)
UPDATE src/app/app.module.ts (492 bytes)

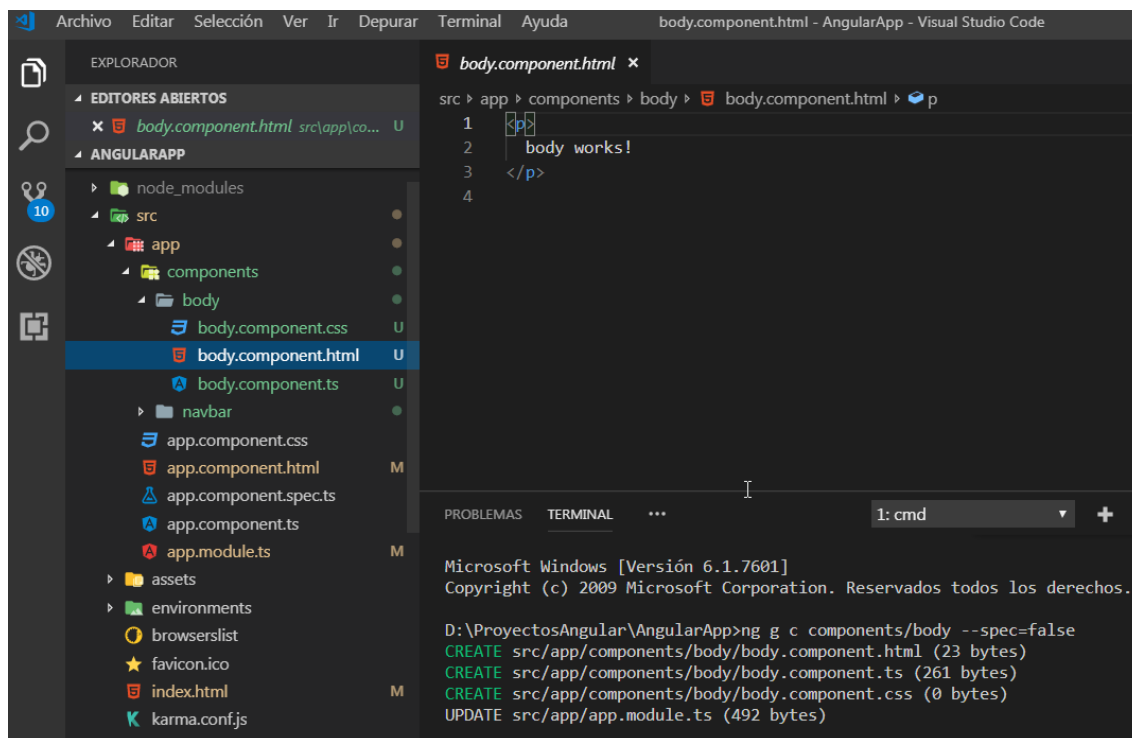
D:\ProyectosAngular\AngularApp>
```

Abrimos el body.component.html





**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



Y reemplazamos el código de este body por:

```
<h2>Pantalla Principal</h2>
```

Y además, en el app.component.html, que tiene:

```
<app-navbar></app-navbar>

<h1>
  Welcome to {{ title }}!
</h1>
```

Lo vamos a cambiar por

```
<app-navbar></app-navbar>

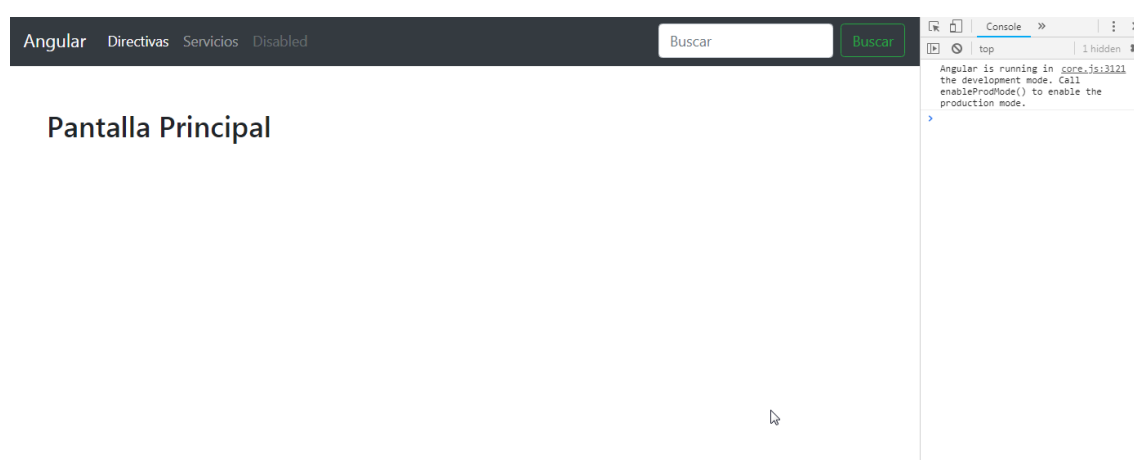
<div class="container mt-5">
  <app-body></app-body>
</div>
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Lo que estamos poniendo es que el componente del body, lo ponga en un container de bootstrap y que además lo separa de la parte superior 5 medidas.

Debería quedarnos así:



Volvemos a angular CLI para crear el componente del footer y ponemos:

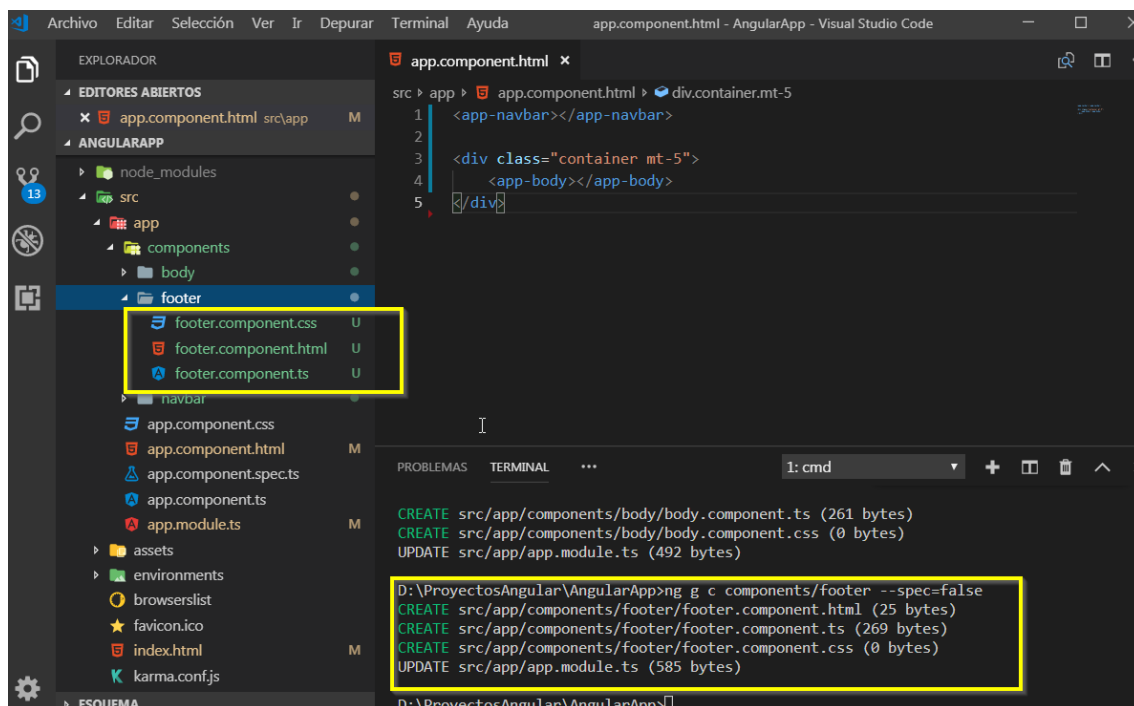
(siempre en la raíz del proyecto)

`ng g c components/footer --spec=false`

Una vez hecho, nos tiene que quedar así:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



Ahora, abrimos el footer.component.html y vamos a poner este código.

```
<footer class="footer bg-dark">
  <div class="container">
    <p>Consejería de Educación - JCCM </p>
  </div>
</footer>
```

Todas esas clases son del bootstrap.

En la pantalla, de momento no sale nada porque no hemos puesto el selector del componente en ningún sitio.

Lo ponemos en el app.component.html, que tiene este contenido

```
<app-navbar></app-navbar>
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
<div class="container mt-5">
  <app-body></app-body>
</div>
```

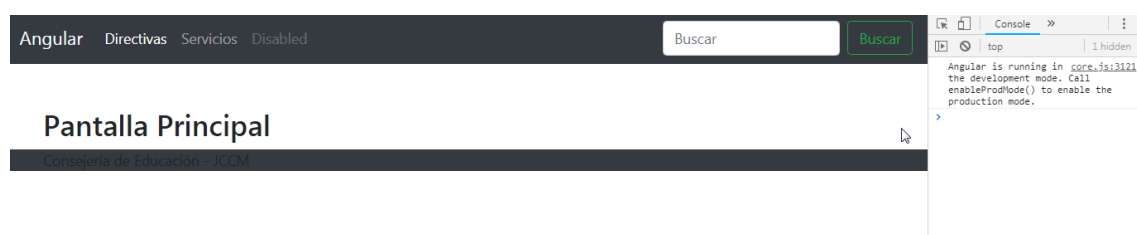
Y lo dejaremos con este :

```
<app-navbar></app-navbar>

<div class="container mt-5">
  <app-body></app-body>
</div>

<app-footer></app-footer>
```

Así nos saldría :



Claro que..... no sale bien. ¿Será por los estilos?

Pongamos en el style.css el siguiente código:

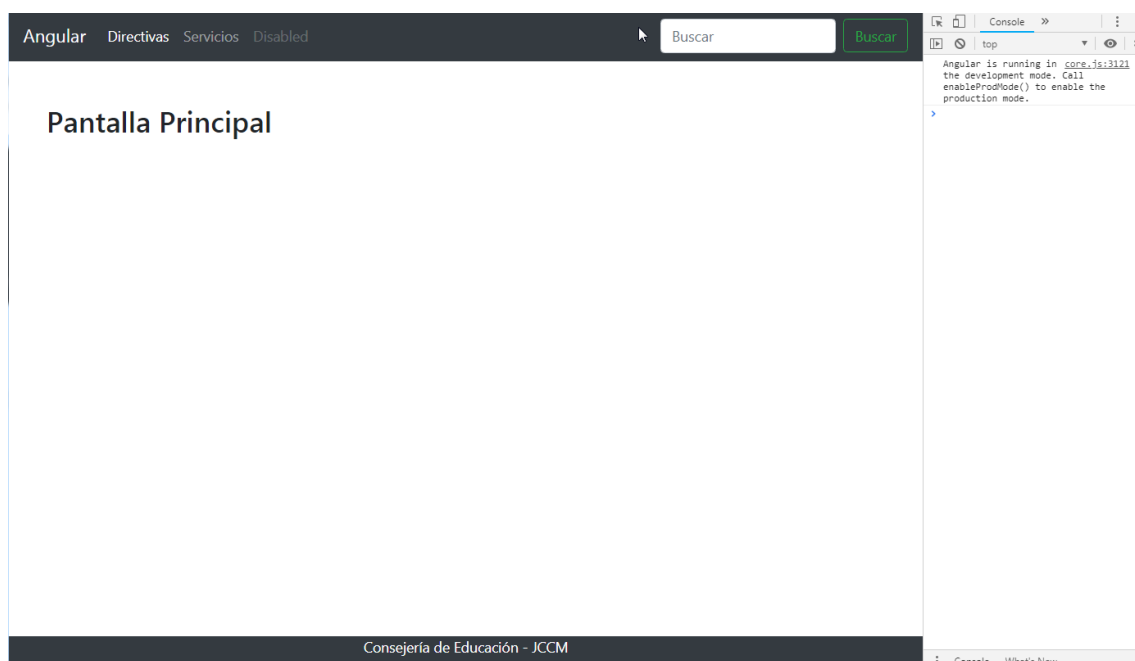
```
/* You can add global styles to this file, and also import other style
files */

footer {
  color: white;
  position: fixed;
  bottom: 0px;
  width: 100%;
  text-align: center;
}
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Ahora si que sale bien:



Pero no vamos a poner el nombre en el footer directamente en el html.

Lo pondremos en la clase del componente, todo por si luego esos nombres, vienen de un servicio o de otro sitio.

Para ello, en el fichero footer.component.ts, vamos a crearnos una propiedad con el valor Consejería de Educación – JCCM de forma que el footer.component.ts quedará así:

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

@Component({
  selector: 'app-footer',
  templateUrl: './footer.component.html',
  styleUrls: ['./footer.component.css']
})
export class FooterComponent implements OnInit {

  nombreConsejeria = 'Consejería de Educación - JCCM';
```



```
constructor() { }  
  
ngOnInit() {  
}  
  
}
```

Por cierto, que lo del ngOnInit(), forma parte del ciclo de vida de los componentes de angular.

En otros documentos, como lo vamos a utilizar, pues lo cuento.

Y en el html cambiaremos el texto por la interpolación de string y quedará así:

```
<footer class="footer bg-dark">  
  <div class="container">  
    <p>{{ nombreConsejeria }} </p>  
  </div>  
</footer>
```

Como se puede ver, sigue funcionando sin problemas.

Aunque, en el .ts, podemos hacer algo para ver la funcionabilidad del constructor, por ejemplo:

```
import { Component, OnInit } from '@angular/core';  
  
@Component({  
  selector: 'app-footer',  
  templateUrl: './footer.component.html',  
  styleUrls: ['./footer.component.css']  
})  
export class FooterComponent implements OnInit {  
  
  nombreConsejeria: string;  
  curso = 'Curso 2019/2020';  
  
  constructor() {  
    this.nombreConsejeria = 'Consejería de Educación - ' + this.curso;  
  }  
}
```



```
ngOnInit() {  
  }  
}
```

El this, lo estamos poniendo porque nos referimos a los atributos de la misma clase.

**Nos vamos a centrar ahora en el body component.**

**Donde vamos a poner los ejercicios de las directivas.**

Por lo tanto abrimos el body.component.html y lo vamos a trabajar.

Vamos a poner una fila y dos columnas del bootstrap.

forma.

```
<div class="row">  
  <div class="col">  
    <h3>columna 1</h3>  
  </div>  
  
  <div class="col">  
    <h3>columna 2</h3>  
  </div>  
</div>
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Que mostrará algo así



Aquí lo que estamos poniendo es una clase de bootstrap “row”, para decirle que queremos una fila y luego en la fila, dos columnas. Ya sabéis que bootstrap divide cada fila en 12 columnas y por eso si no se pone nada, deja 6 y 6.

Si ponemos por ejemplo :

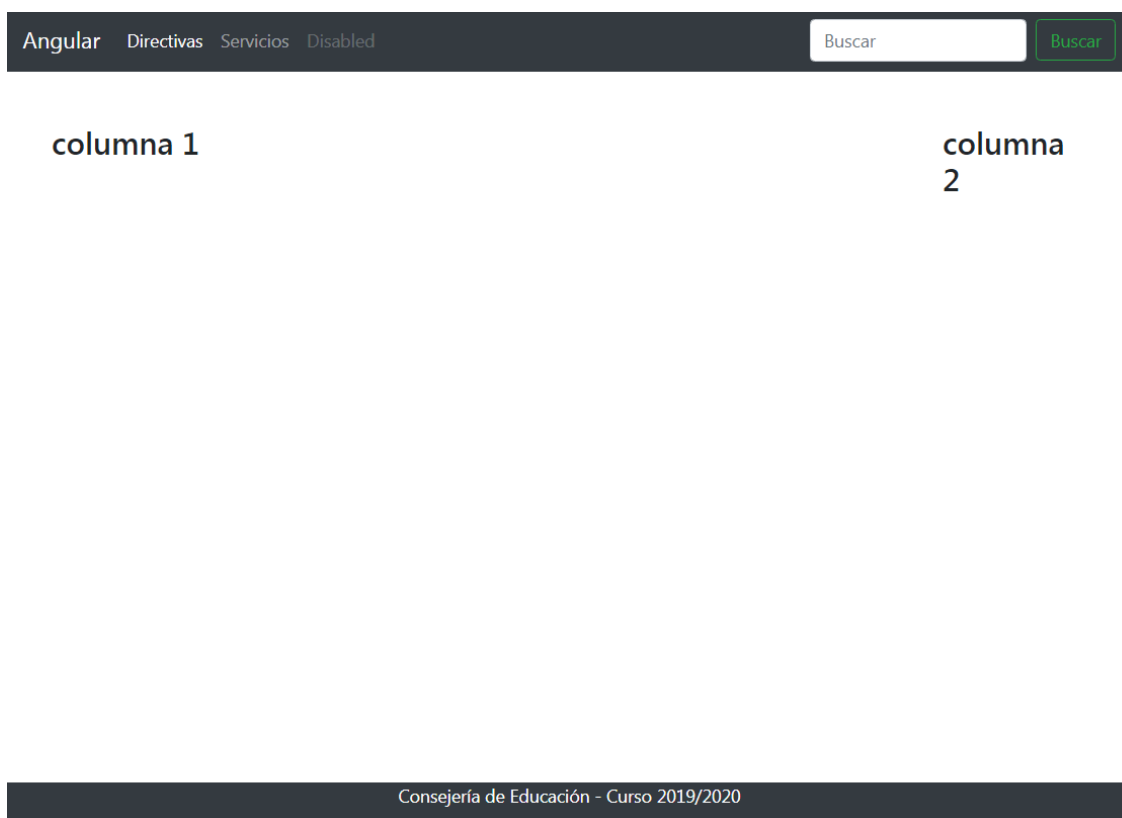
```
<div class="row">
  <div class="col-10">
    <h3>columna 1</h3>
  </div>

  <div class="col">
    <h3>columna 2</h3>
  </div>
</div>
```





*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*



Vemos que deja la primera con 10 columnas y la segunda mas pequeña.

Pero lo vamos a dejar como estaba:

```
<div class="row">
  <div class="col">
    <h3>columna 1</h3>
  </div>

  <div class="col">
    <h3>columna 2</h3>
  </div>
</div>
```

Para que quede un poco más bonito, vamos a coger de los cards del bootstrap alguno que nos guste, por ejemplo:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
<div class="card" style="width: 18rem;">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <h6 class="card-subtitle mb-2 text-muted">Card subtitle</h6>
    <p class="card-text">Some quick example text to build on the card title
and make up the bulk of the card's content.</p>
    <a href="#" class="card-link">Card link</a>
    <a href="#" class="card-link">Another link</a>
  </div>
</div>
```

Quitamos algunas cosas y lo dejamos así:

```
<div class="row">
  <div class="col">

    <div class="card">
      <div class="card-body">
        <h5 class="card-title">*NgIf</h5>
        <p class="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>
      </div>
    </div>

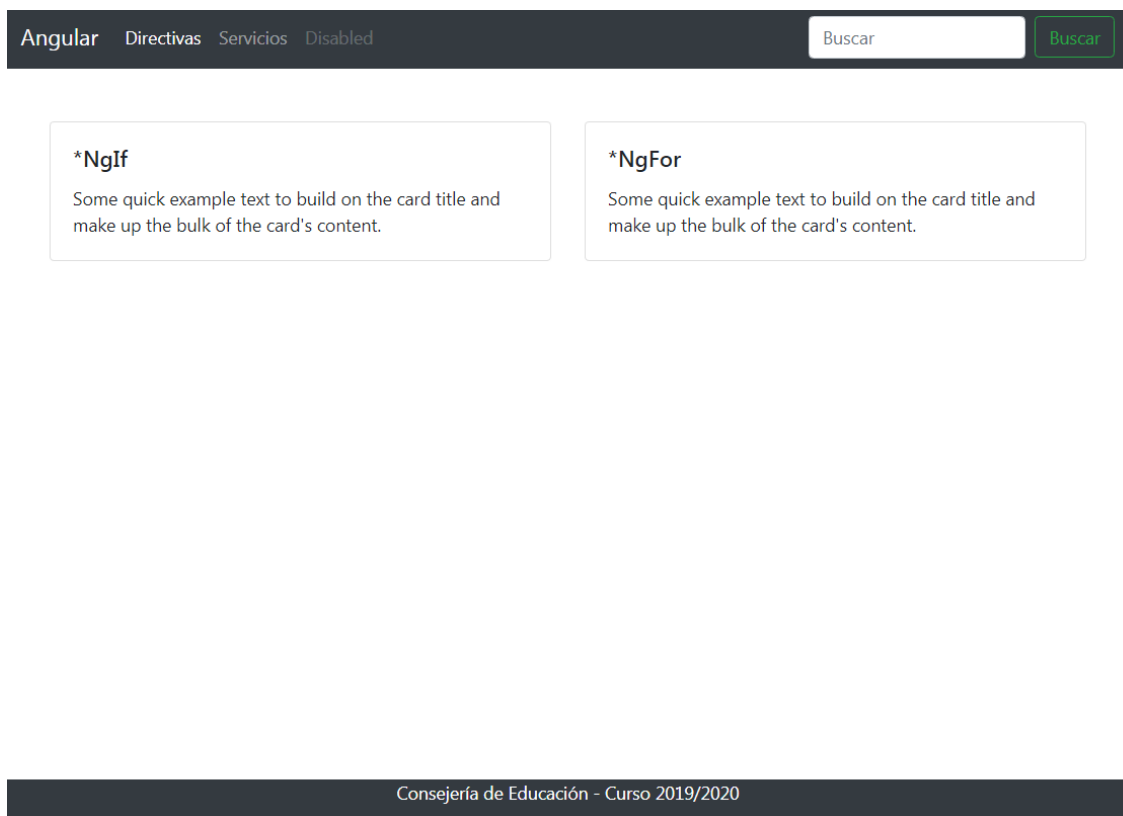
  </div>

  <div class="col">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">*NgFor</h5>
        <p class="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>
      </div>
    </div>
  </div>
</div>
```

La pantalla, debe ir quedando así:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



Vamos ahora a poner un botón en la tarjeta del ngif.

```
<div class="card mb-3">
  <div class="card-body">
    <h5 class="card-title">*NgIf</h5>
    <p class="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>
  </div>
</div>

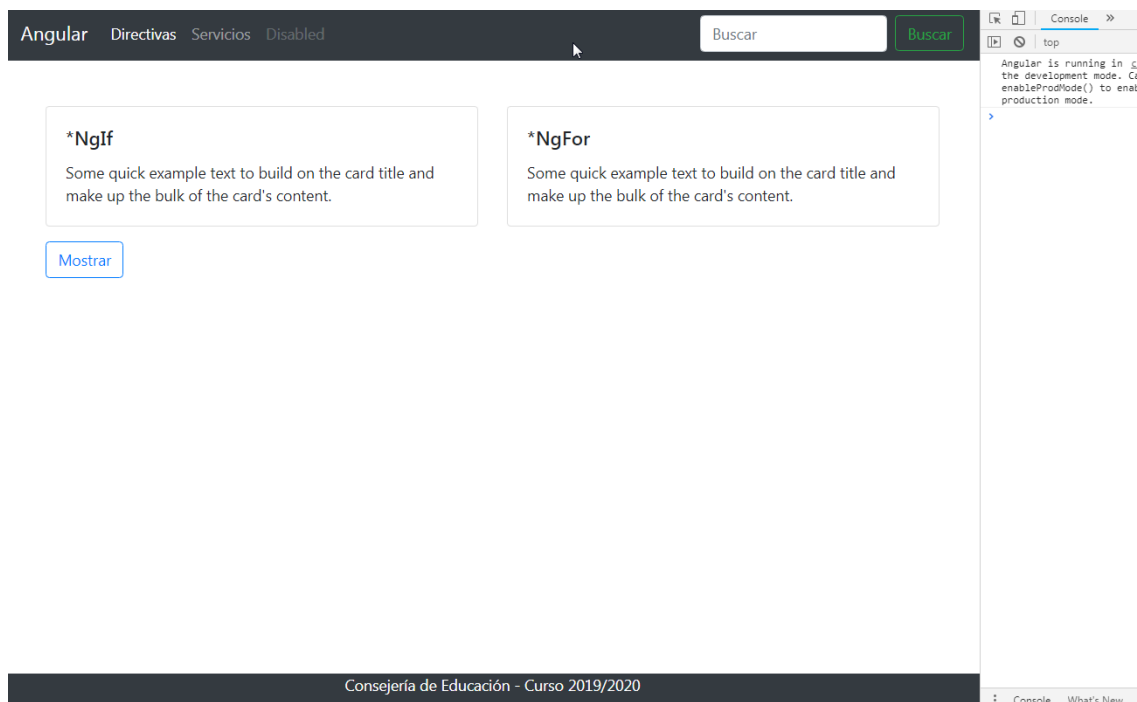
<button class="btn btn-outline-primary">Mostrar</button>
```

He utilizado las clases del bootstrap y además, en la clase del card, he puesto mb-3, para que separe un poco la card del button.

Irà quedando así:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**



También vamos a necesitar una lista ordenada, por lo que en el for, vamos a poner una de las que vienen en bootstrap.

Así nos va de momento:

Código:

```
<div class="row">
  <div class="col">

    <div class="card mb-3">
      <div class="card-body">
        <h5 class="card-title">*NgIf</h5>
        <p class="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>
      </div>
    </div>

    <button class="btn btn-outline-primary">Mostrar</button>

  </div>

  <div class="col">
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
<div class="card">
  <div class="card-body">
    <h5 class="card-title">*NgFor</h5>
    <ul class="list-group">
      <li class="list-group-item">Cras justo odio</li>
      <li class="list-group-item">Dapibus ac facilisis
in</li>

      <li class="list-group-item">Morbi leo risus</li>
      <li class="list-group-item">Porta ac consectetur
ac</li>

      <li class="list-group-item">Vestibulum at eros</li>
    </ul>
  </div>
</div>
</div>
</div>
```

Pantalla:

Angular Directivas Servicios Disabled

Buscar

Buscar

\*NgIf

Some quick example text to build on the card title and make up the bulk of the card's content.

Mostrar

\*NgFor

Cras justo odio

Dapibus ac facilisis in

Morbi leo risus

Porta ac consectetur ac

Vestibulum at eros

Angular is in the development mode. To enable production mode, run: ng build --prod

Consejería de Educación - Curso 2019/2020

Console



*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

Ahora, en el body.component.ts, vamos a crear un objeto de tipo any (cualquier cosa) y le vamos a poner dos propiedades.

```
export class BodyComponent implements OnInit {  
  
  cita: any = {  
    cita: 'No pienso nunca en el futuro porque llega muy pronto.',  
    autor: 'Albert Einstein'  
  };  
  
  constructor() { }  
  
  ngOnInit() {  
  }  
  
}
```

Además, he cambiado un poco el html, dejándolo así

```
<div class="row">  
  <div class="col">  
  
    <div class="card mb-3">  
  
      <div class="card-body text-white bg-secondary">  
        <h5 class="card-title">*NgIf</h5>  
        <p class="card-text">Ejemplo del uso de la directiva  
*ngIf</p>  
      </div>  
      <div class="card-body">  
        <h5 class="card-title">{{ cita.autor }}</h5>  
        <p class="card-text">{{ cita.mensaje }}</p>  
      </div>  
  
    </div>  
  
    <button class="btn btn-outline-primary">Mostrar</button>  
  

```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
</div>

<div class="col">
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">*NgFor</h5>
      <ul class="list-group">
        <li class="list-group-item">Cras justo odio</li>
        <li class="list-group-item">Dapibus ac facilisis
in</li>

        <li class="list-group-item">Morbi leo risus</li>
        <li class="list-group-item">Porta ac consectetur
ac</li>

        <li class="list-group-item">Vestibulum at eros</li>
      </ul>
    </div>
  </div>
</div>
</div>
```

Para que salga esto:



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Ahora, vamos a dar vida al botón , para que salga o no salga la cita de Albert Einstein.

Para hacerlo, le ponemos al elemento que queremos ocultar un ngif y una condición lógica.

Esta condición lógica, puede ser el resultado de una función, el valor que tenga una variable en el ts, etc.

Yo voy a utilizar el valor de un booleano que esté en el ts, de esta forma. Y además, que dando al botón se cambie ese valor, incluso el texto del botón.

Para eso he creado unos parámetros en el .ts y una función que se llama cuando se hace click en el botón, quedando así el html

```
<div class="row">
  <div class="col">

    <div class="card mb-3">

      <div class="card-body text-white bg-secondary">
        <h5 class="card-title">*NgIf</h5>
        <p class="card-text">Ejemplo del uso de la directiva
*ngIf</p>
      </div>
      <div *ngIf="citaVisible" class="card-body">
        <h5 class="card-title">{{ cita.autor }}</h5>
        <p class="card-text">{{ cita.mensaje }}</p>
      </div>

    </div>

    <button (click)="ocultaMuestraCita()" class="btn btn-outline-
primary">{{ textoDelBoton }}</button>

  </div>

  <div class="col">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">*NgFor</h5>
        <ul class="list-group">
          <li class="list-group-item">Cras justo odio</li>
          <li class="list-group-item">Dapibus ac facilisis
in</li>
```





*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

```
        <li class="list-group-item">Morbi leo risus</li>
        <li class="list-group-item">Porta ac consectetur
ac</li>
        <li class="list-group-item">Vestibulum at eros</li>
    </ul>
</div>
</div>
</div>
</div>
</div>
```

Y así el .ts

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

@Component({
  selector: 'app-body',
  templateUrl: './body.component.html',
  styleUrls: ['./body.component.css']
})
export class BodyComponent implements OnInit {

  cita: any = {
    mensaje: 'No pienso nunca en el futuro porque llega muy pronto.',
    autor: 'Albert Einstein'
  };

  textoDelBoton = 'Ocultar Cita';

  citaVisible = true;

  constructor() { }

  ngOnInit() {
  }

  ocultaMuestraCita() {
    this.citaVisible = !this.citaVisible;

    if ( this.citaVisible ){
      this.textoDelBoton = 'Ocultar Cita';
    } else {
      this.textoDelBoton = 'Mostrar Cita';
    }
  }
}
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

$$\left\{ \begin{array}{l} \text{if } x = y \\ \text{then } \text{true} \\ \text{else } \text{false} \end{array} \right.$$

Se puede ver la funcionabilidad cuando se pulsa el botón.

Angular

Directivas

Servicios

Disabled

Buscar

\*NgIf

Ejemplo del uso de la directiva \*ngIf

Albert Einstein

No pienso nunca en el futuro porque llega muy pronto.

Ocultar Cita

\*NgFor

Cras justo odio

Dapibus ac facilisis in

Morbi leo risus

Porta ac consectetur ac

Vestibulum at eros

Pulsamos en el botón y



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

[Angular](#) [Directivas](#) [Servicios](#) [Disabled](#)

### \*NgIf

Ejemplo del uso de la directiva \*ngIf

[Mostrar Cita](#)

### \*NgFor

Cras justo odio

Dapibus ac facilisis in

Morbi leo risus

Porta ac consectetur ac

Vestibulum at eros

Esto es como funciona el \*ngIf.

Recomiendo inspeccionar estos elementos con las herramientas de google.

### La directiva \*ngfor.

Los For, trabajan con arrays. Para ello, vamos a crear un array de elementos tipo string que van a ser autores de citas célebres, por ejemplo podemos crear lo siguiente.

```
autores: string[] = ['Aristóteles', 'Einstein', 'San Agustín', 'Platón'];
```

Ahora lo que queremos que se repita es donde ponemos el ngfor, por lo que vamos a sustituir esto en el body.component.html

```
<ul class="list-group">  
  <li class="list-group-item">Cras justo odio</li>
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
in</li>
<li class="list-group-item">Morbi leo risus</li>
<li class="list-group-item">Porta ac consectetur
ac</li>
<li class="list-group-item">Vestibulum at eros</li>
</ul>
```

Por:

```
<h5 class="card-title">*NgFor</h5>
  <ul class="list-group" *ngFor="let autor of autores">
    <li class="list-group-item">{{ autor }}</li>
  </ul>
```

Veremos que ahora sale la lista de nuestros autores.

The screenshot shows a web application with a dark header containing the text "Activas Servicios Disabled" and a search bar with the placeholder "Buscar" and a green "Buscar" button. Below the header, there is a card with a dark header "El uso de la directiva \*ngIf" and a light body containing the text "Einstein" and "nunca en el futuro porque llega muy pronto." To the right of this card is a box labeled "\*NgFor" containing a list of authors: "Aristóteles", "Einstein", "San Agustín", and "Platón". On the far right, a Chrome DevTools console is visible, showing a message "Angular: the d... enabl... produ..." with a blue arrow pointing to the right.

Incluso, podemos poner un prefijo con el número de iteración en el for. Me explico haciéndolo porque además esto es luego muy necesario cuando se pulse en uno de ellos y tengamos que ir a una página de detalle.

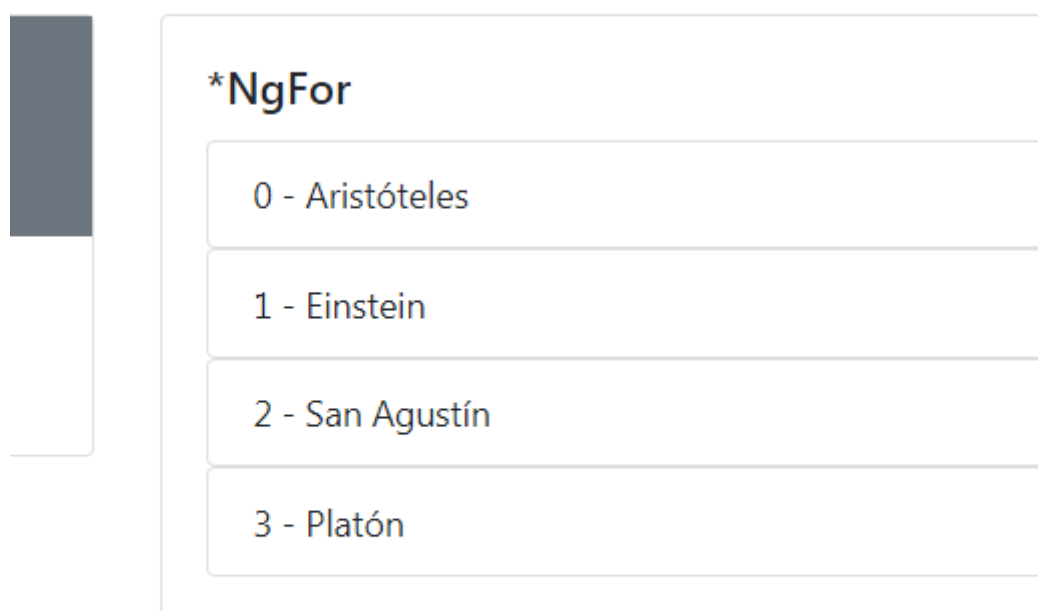


*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

Hay que poner el la iteración del \*ngFor, lo siguiente:

```
<ul class="list-group" *ngFor="let autor of autores; let i = index">  
<li class="list-group-item">{{ i }} - {{ autor }}</li>
```

Y sale así:



No me gustaría hacer este documento tan largo, pero antes de comenzar con otro, me gustaría incorporar las rutas.

Las **RUTAS** en angular, son para podernos cambiar de una página a otra, para ello, en la carpeta app, vamos a crear un fichero que le vamos a llamar app.routes.ts.

También vamos a poner en nuestro navbar, donde teníamos el nombre de servicios, vamos a poner citas y vamos a crear un componente llamado "citas" .

Vamos por partes.

1. Cambiar en el navbar la palabra servicios por Citas.



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

2. Creamos el componente citas con la instrucción `ng g c components/citas --spec=false`

Y ahora comenzamos con el fichero `app.routes.ts`

Estando vacío, si se comienza a escribir Routes y están instalados en el VSC los snippets que comenté en otro documento, te muestra esto

The screenshot shows the VS Code editor with the file `src > app > app.routes.ts` open. The first line contains the text `1 rou` with a yellow lightbulb icon indicating a snippet suggestion. The suggestion list on the left includes `RouterModule`, `Routes`, `ROUTES`, and several Angular-specific routing guards and components. The `Routes` snippet is selected. The preview on the right shows the following code:

```
Angular root routes snippet (Angular 7 and TypeScript/HTML VS Code Snippets)

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

const routes: Routes = [
  { path: 'routePath', component: Component },
  { path: '**', pathMatch: 'full', redirectTo: 'routePath' }
];
```

Y dando a la tecla tabulador te lo escribe automáticamente.

En cualquier caso, el fichero tiene que quedar así:

```
import { RouterModule, Routes } from '@angular/router';
import { BodyComponent } from '../components/body/body.component';
import { CitasComponent } from '../components/citas/citas.component';

const APP_ROUTES: Routes = [
  { path: 'body', component: BodyComponent },
  { path: 'citas', component: CitasComponent },
  { path: '**', pathMatch: 'full', redirectTo: 'body' }
];
```



*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

```
export const APPROUTING = RouterModule.forRoot(APP_ROUTES);
```

Y que es lo que contiene.

Pues se puede ver que contiene una constante llamada APP\_ROUTES con la configuración.

En la configuración hace referencia a que componente tiene que llamar/renderizar cuando se pide el path citas o body.

Ahora, como hemos creado una constante con esta configuración llamada APPROUTING, lo informamos en el app.module.ts que es donde están todos los componentes configurados.

De forma , que en el app.module.ts hemos puesto:

En el import

```
imports: [  
  BrowserModule,  
  APPROUTING
```

Y hemos tenido que importar donde está ese APPROUTING de la forma

```
import { APPROUTING } from './app.routes';
```

quedando el app.module así:

```
import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
  
import { AppComponent } from './app.component';  
import { NavbarComponent } from './components/navbar/navbar.component';  
import { BodyComponent } from './components/body/body.component';  
import { FooterComponent } from './components/footer/footer.component';  
import { CitasComponent } from './components/citas/citas.component';  
import { APPROUTING } from './app.routes';  
  
@NgModule({
```



*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

```
declarations: [  
  AppComponent,  
  NavbarComponent,  
  BodyComponent,  
  FooterComponent,  
  CitasComponent  
],  
imports: [  
  BrowserModule,  
  APPROUTING  
],  
providers: [],  
bootstrap: [AppComponent]  
})  
export class AppModule { }
```

claro, que ¿como vemos lo que hemos hecho?

Ponemos ahora en el app.component.html el selector

```
<router-outlet></router-outlet>
```

Quedando así:

```
<app-navbar></app-navbar>  
  
<div class="container mt-5">  
  <router-outlet></router-outlet>  
</div>  
  
<app-footer></app-footer>
```

Aquí lo que estamos diciendo es que se renderize aquí el componente que se pida según el path,

Puedes probar lo que pasa ahora cuando se pone en la url

<http://localhost:4200/citas>

o





**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

<http://localhost:4200/body>

e incluso otra url, que en el app.routes.ts, le hemos puesto que cuando no sea ninguna de las que están en la configuración tienen que ir a la del body.

http:// <http://localhost:4200/noloencuentro>

Ahora ya para ir terminando ,en el navbar vamos a poner los link a estos dos componentes.

Quedará así utilizando [routerLink] de angular.

Entonces el navbar.component.html, lo dejamos así

```
<li class="nav-item active">
    <a class="nav-link" [routerLink]="['body']">Directivas
<span class="sr-only">(current)</span></a>
</li>
<li class="nav-item">
    <a class="nav-link" [routerLink]="['citas']">Citas</a>
</li>
```

Veremos entonces que nuestra aplicación cambia de un componente a otro sin problema.

Lo único es que no se ilumina la que está activa. Pues eso es quitar la clase active del link ponemos en cada ítem lo siguiente. Ojo que no lleva llaves,

Quedará así:

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
        <li class="nav-item">
            <a class="nav-link" routerLinkActive="active"
[routerLink]="['body']">Directivas <span class="sr-
only">(current)</span></a>
        </li>
        <li class="nav-item">
            <a class="nav-link" routerLinkActive="active"
[routerLink]="['citas']">Citas</a>
        </li>
```



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

```
<li class="nav-item">
  <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a>
</li>
</ul>
```

Bueno, hasta aquí este documento que ha sido un poco extenso.

**Pero antes de terminar hay algo que es una obligación y es subirlo a un repositorio.**

Yo utilizo GitLab, pero se puede utilizar cualquiera.

En GitLab, voy a crear el proyecto:

The screenshot shows the 'Create project' form in GitLab. The form is titled 'Blank project' and has four tabs: 'Blank project', 'Create from template', 'Import project', and 'CI/CD for external repo'. The 'Blank project' tab is selected. The form contains the following fields and options:

- Project name:** appangular
- Project URL:** https://gitlab.com/carlos\_mio/
- Project slug:** appangular
- Project description (optional):** Aplicación de ejemplo de angular
- Visibility Level:** Private (selected), Internal, Public
- Initialize repository with a README:** ☐ (unchecked)
- Create project:** Button
- Cancel:** Button

Ahora, seguimos las instrucciones para subirlo al repositorio.

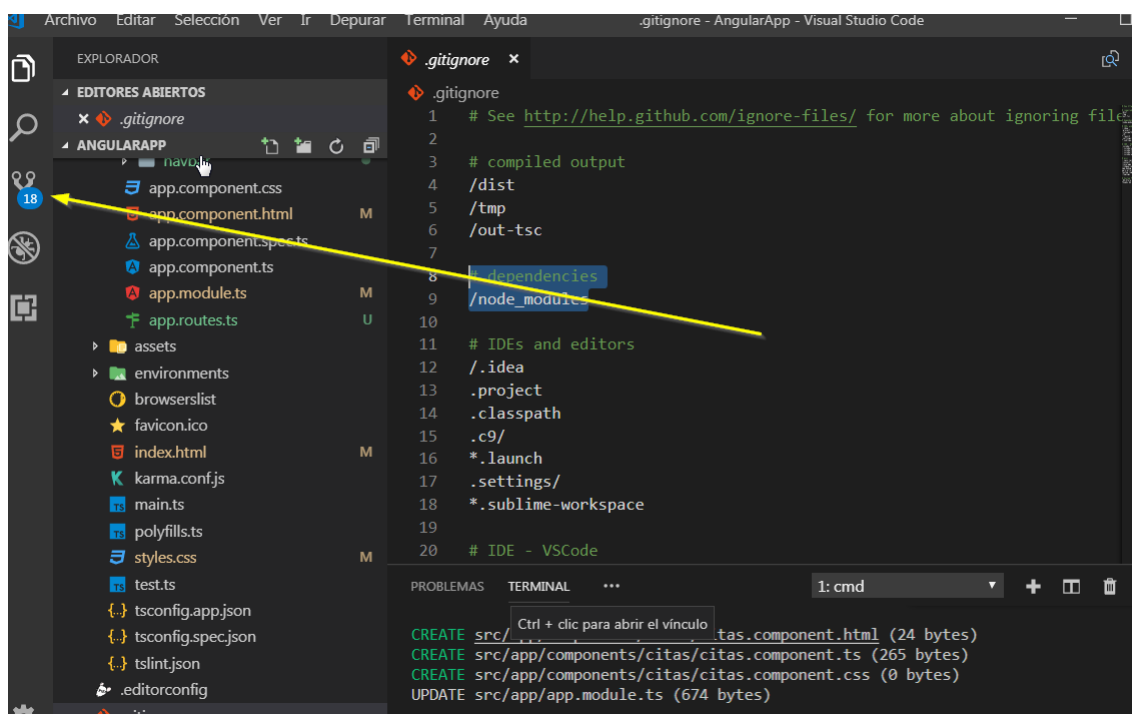
Primero, revisamos nuestro gitignore a ver si tiene configurado que no se suban los módulos.



**Apuntes de Angular. Juan Carlos Fernández García.**  
**03 – Angular , bootstrap y rutas**

Efectivamente lo tiene.

Además en mi caso, me está diciendo que hay 18 recursos de los cuales no se ha hecho commit.



Yo utilizo la consola de comandos, aunque con VSC también se puede.

Entonces, sigo las instrucciones que vienen en GitLab

Iniciamos el repositorio ( esto se hace una sola vez)

```
git init
```

añadimos el repositorio remoto ( solo se hace esta vez)

```
git remote add origin https://gitlab.com/carlos_mio/appangular.git
```

Vemos los ficheros que tienen que subirse con

```
git status
```

(saldrán un montón rojos)



*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

```
\ProyectosAngular\AngularApp>git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/app/app.component.html
        modified:   src/app/app.module.ts
        modified:   src/index.html
        modified:   src/styles.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        src/app/app.routes.ts
        src/app/components/

no changes added to commit (use "git add" and/or "git commit -a")
\ProyectosAngular\AngularApp>git add .
warning: LF will be replaced by CRLF in src/app/app.component.html
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.module.ts.
The file will have its original line endings in your working directory
```

Los añadimos al status con git add .

Ahora si ponemos git status, saldrán en verde

```
\ProyectosAngular\AngularApp>git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   src/app/app.component.html
        modified:   src/app/app.module.ts
        new file:   src/app/app.routes.ts
        new file:   src/app/components/body/body.component.css
        new file:   src/app/components/body/body.component.html
        new file:   src/app/components/body/body.component.ts
        new file:   src/app/components/citas/citas.component.css
        new file:   src/app/components/citas/citas.component.html
        new file:   src/app/components/citas/citas.component.ts
        new file:   src/app/components/footer/footer.component.css
        new file:   src/app/components/footer/footer.component.html
        new file:   src/app/components/footer/footer.component.ts
        new file:   src/app/components/navbar/navbar.component.css
        new file:   src/app/components/navbar/navbar.component.html
        new file:   src/app/components/navbar/navbar.component.spec.ts
        new file:   src/app/components/navbar/navbar.component.ts
```

Hacemos el primer commit con un mensaje

```
git commit -m "Primer commit con bootstrap, components, directivas y rutas"
```



*Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas*

```
D:\ProyectosAngular\AngularApp>git commit -m "Primer commit con bootstrap, compo
nents, directivas y rutas"
[master ffd6b3a1 Primer commit con bootstrap, components, directivas y rutas
18 files changed, 240 insertions(+), 36 deletions(-)
rewrite src/app/app.component.html <99%>
create mode 100644 src/app/app.routes.ts
create mode 100644 src/app/components/body/body.component.css
create mode 100644 src/app/components/body/body.component.html
create mode 100644 src/app/components/body/body.component.ts
create mode 100644 src/app/components/citas/citas.component.css
create mode 100644 src/app/components/citas/citas.component.html
create mode 100644 src/app/components/citas/citas.component.ts
create mode 100644 src/app/components/footer/footer.component.css
create mode 100644 src/app/components/footer/footer.component.html
create mode 100644 src/app/components/footer/footer.component.ts
create mode 100644 src/app/components/navbar/navbar.component.css
create mode 100644 src/app/components/navbar/navbar.component.html
create mode 100644 src/app/components/navbar/navbar.component.spec.ts
create mode 100644 src/app/components/navbar/navbar.component.ts
rewrite src/index.html <78%>

D:\ProyectosAngular\AngularApp>git push -u origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.
```

y por ultimo hacemos el push

```
git push -u origin master
```

```
ca. Símbolo del sistema
D:\ProyectosAngular\AngularApp>git remote add origin https://gitlab.com/carlos_m
io/appangular.git

D:\ProyectosAngular\AngularApp>git push origin master
Enumerating objects: 61, done.
Counting objects: 100% (61/61), done.
Delta compression using up to 4 threads
Compressing objects: 100% (58/58), done.
Writing objects: 100% (61/61), 103.09 KiB | 3.44 MiB/s, done.
Total 61 (delta 8), reused 0 (delta 0)
To https://gitlab.com/carlos_mio/appangular.git
 * [new branch]      master -> master

D:\ProyectosAngular\AngularApp>
```

Podemos ver en GitLab como ha quedado.



## Apuntes de Angular. Juan Carlos Fernández García.

### 03 – Angular , bootstrap y rutas

The screenshot shows the GitHub repository page for 'appangular'. At the top, there's a header with the repository name, Project ID (10317199), and buttons for cloning, starring, forking, and adding a license. Below this, it says 'Aplicación de ejemplo de angular'. The main content area shows the commit history, with the latest commit being 'Primer commit con bootstrap, components, directivas y rutas' by Juan Carlos, made 4 minutes ago. A table below lists the files in the repository, including 'e2e', 'src', '.editorconfig', '.gitignore', 'README.md', 'angular.json', and 'tsconfig.json', along with their last commit and update times.

Name	Last commit	Last update
e2e	initial commit	1 day ago
src	Primer commit con bootstrap, components, directivas y rutas	4 minutes ago
.editorconfig	initial commit	1 day ago
.gitignore	initial commit	1 day ago
README.md	initial commit	1 day ago
angular.json	initial commit	1 day ago
tsconfig.json	initial commit	1 day ago

Incluso, el VSC, ya nos informa que todos nuestros recursos están en el repositorio

The screenshot shows the Visual Studio Code interface. On the left, the Explorer view displays the project structure for 'ANGULARAPP'. A yellow circle highlights the Git icon in the Explorer sidebar, and a yellow arrow points from it to the 'src/app/components/citas/citas.component.html' file in the Explorer. The main editor area shows the content of this file. At the bottom, the Terminal view displays the output of a command, showing the creation and update of files in the project.

```
CREATE src/app/components/citas/citas.component.html (24 bytes)
CREATE src/app/components/citas/citas.component.ts (265 bytes)
CREATE src/app/components/citas/citas.component.css (0 bytes)
UPDATE src/app/app.module.ts (674 bytes)
D:\ProyectosAngular\AngularApp>
```



***Apuntes de Angular. Juan Carlos Fernández García.  
03 – Angular , bootstrap y rutas***

Espero que os guste.

Si tenéis alguna duda, pues ya sabéis.

Hasta otro documento.