

1er Paso crear un proyecto en Angular

Abrimos la terminal

```
Microsoft Windows [Versión 10.0.18362.418]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Profesortarde>cd Documents

C:\Users\Profesortarde\Documents>ng new prueba
```

Durante la instalación contestamos Y a la creación de angular routing y elegimos CSS como hoja de estilo.

2o Paso Crear un componente

Abrimos el proyecto con Visual Studio Code seleccionando la carpeta del proyecto y pulsando con el botón derecho la opción 'open with Code'.

Desplegamos la terminal de Visual Studio Code con ctrl + ñ.

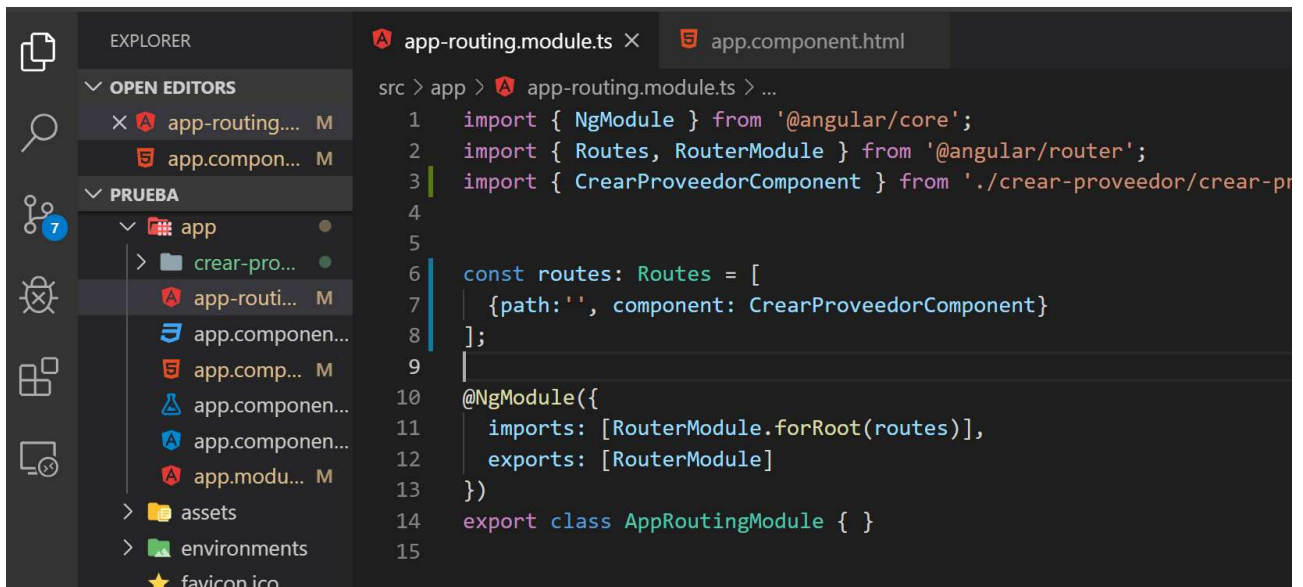
Creamos el componente con el comando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  I: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Profesortarde\Documents\prueba> ng g c crearProveedor
```

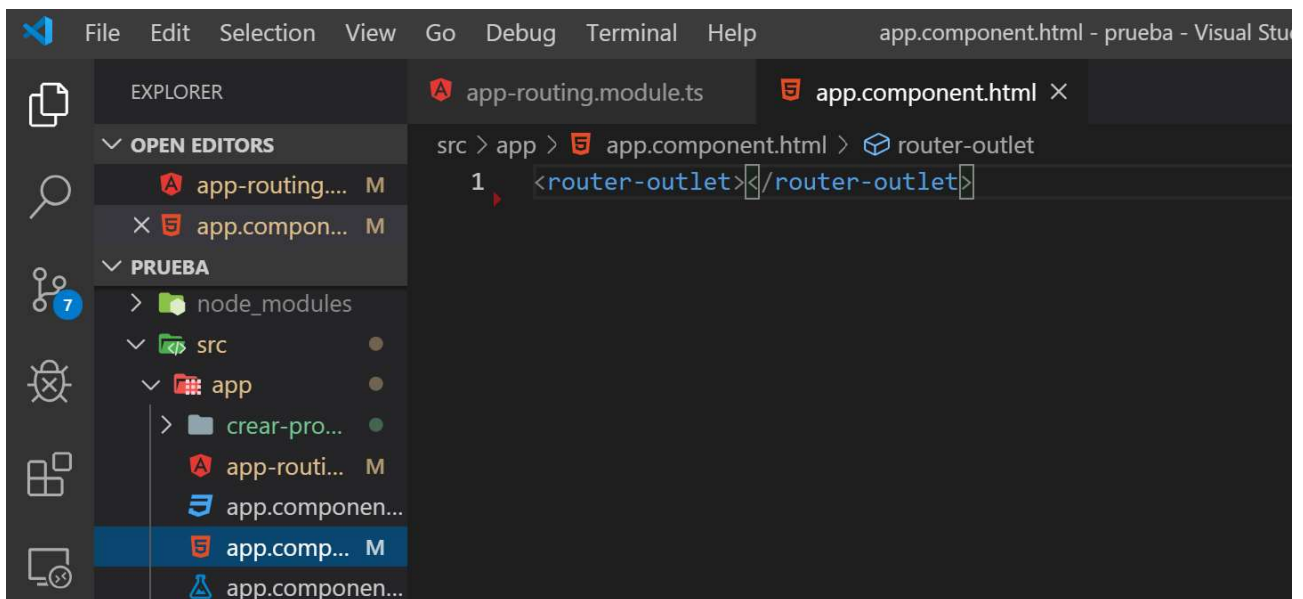
Crear la ruta en el archivo de rutas:



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with a folder named 'PRUEBA' containing a subfolder 'app'. Inside 'app', there is a folder 'crear-pro...' and several files including 'app-routing.module.ts' and 'app.component.html'. The main editor area displays the 'app-routing.module.ts' file. The code defines a route for the root path and exports the 'AppRoutingModule'.

```
src > app > app-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import { CrearProveedorComponent } from './crear-proveedor/crear-pr
4
5
6  const routes: Routes = [
7    {path: '', component: CrearProveedorComponent}
8  ];
9
10 @NgModule({
11   imports: [RouterModule.forRoot(routes)],
12   exports: [RouterModule]
13 })
14 export class AppRoutingModule { }
15
```

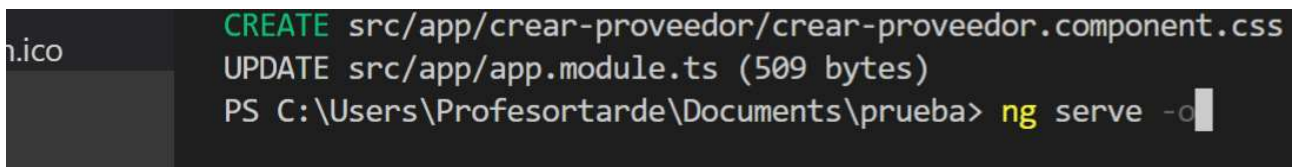
Implementar el acceso al componente desde el componente raíz:



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows the same project structure as the previous image. The main editor area displays the 'app.component.html' file. The code shows a single line with the 'router-outlet' component.

```
src > app > app.component.html > router-outlet
1  <router-outlet></router-outlet>
```

Iniciamos el proyecto en el navegador para comprobar que se carga correctamente el componente:



The screenshot shows a terminal window with the following commands and output:

```
CREATE src/app/crear-proveedor/crear-proveedor.component.css
UPDATE src/app/app.module.ts (509 bytes)
PS C:\Users\Profesortarde\Documents\prueba> ng serve -o
```

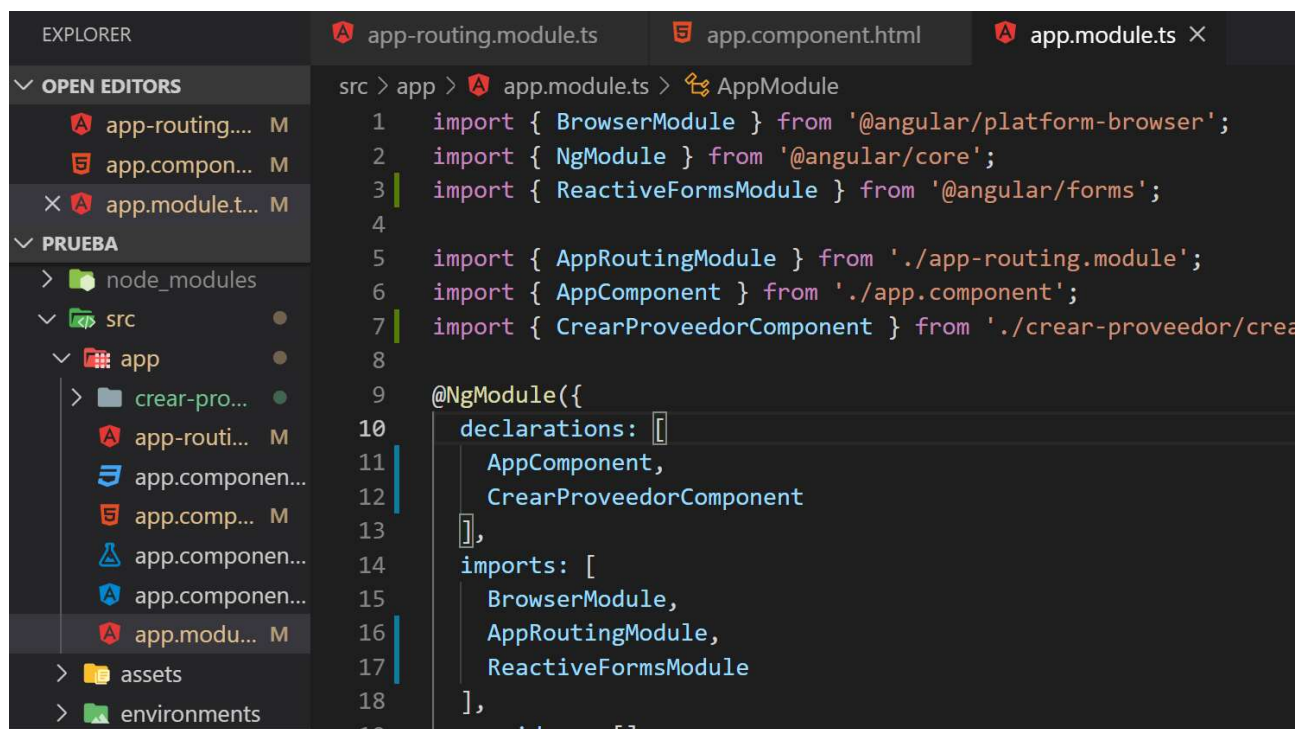


Implementar en el proyecto el módulo de formularios de Angular, en el archivo 'app.module.ts':

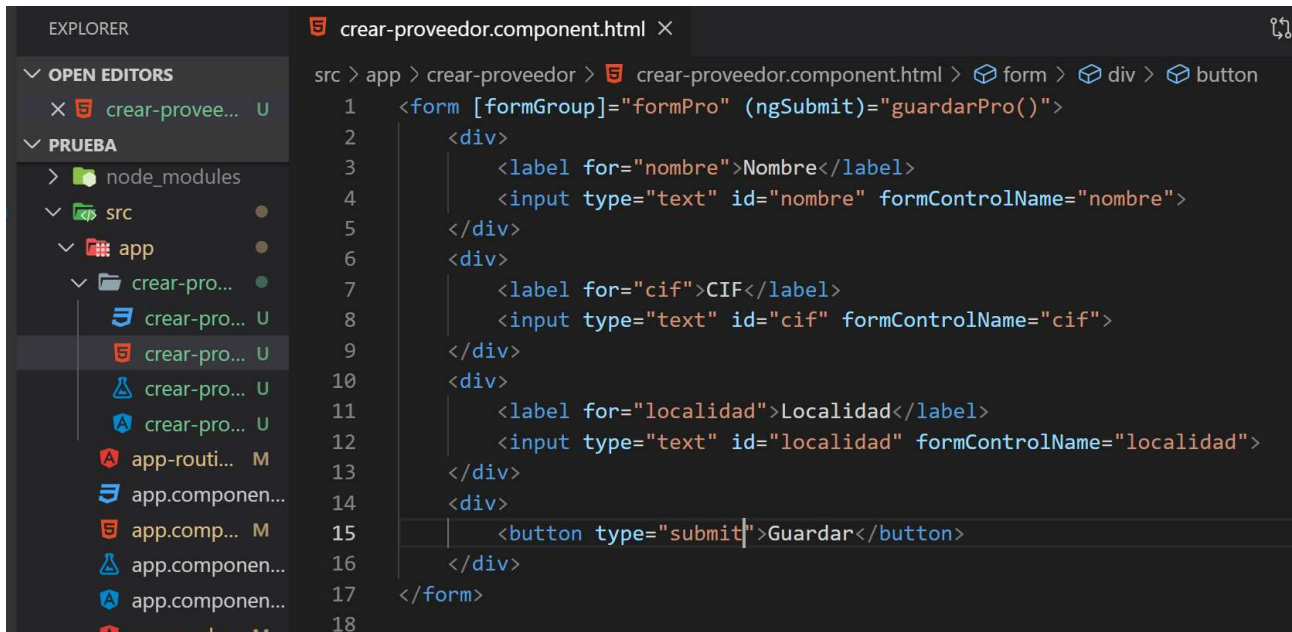
a) Añadir al array de imports la clase ReactiveFormsModule:

```
12 ],
13 imports: [
14     BrowserModule,
15     AppRoutingModule,
16     ReactiveFormsModule
17 ],
18 providers: [],
```

b) Añadimos la importación (similar a la línea 3):



Creamos el formulario en la plantilla del componente (archivo .html):




The screenshot shows the VS Code editor with the Explorer sidebar on the left and the editor window displaying the HTML template for 'crear-proveedor.component.html'. The Explorer sidebar shows the project structure with 'src' > 'app' > 'crear-pro...' expanded. The editor window shows the following HTML code:

```
1 <form [formGroup]="formPro" (ngSubmit)="guardarPro()">
2   <div>
3     <label for="nombre">Nombre</label>
4     <input type="text" id="nombre" formControlName="nombre">
5   </div>
6   <div>
7     <label for="cif">CIF</label>
8     <input type="text" id="cif" formControlName="cif">
9   </div>
10  <div>
11    <label for="localidad">Localidad</label>
12    <input type="text" id="localidad" formControlName="localidad">
13  </div>
14  <div>
15    <button type="submit">Guardar</button>
16  </div>
17 </form>
18
```

Crear el objeto del formulario en el archivo TypeScript del componente (archivo .ts):

a) Importar FormGroup y FormBuilder:



The screenshot shows the VS Code editor with the Explorer sidebar on the left and the editor window displaying the TypeScript file 'crear-proveedor.component.ts'. The Explorer sidebar shows the project structure with 'src' > 'app' > 'crear-proveedor.component.ts' expanded. The editor window shows the following TypeScript code:

```
1 import { Component, OnInit } from '@angular/core';
2 import { FormGroup, FormBuilder } from '@angular/forms';
3
```

b) Declarar el objeto del formulario con el mismo nombre del formulario 'formPro':



The screenshot shows the VS Code editor with the Explorer sidebar on the left and the editor window displaying the TypeScript file 'crear-proveedor.component.ts'. The Explorer sidebar shows the project structure with 'src' > 'app' > 'crear-proveedor.component.ts' expanded. The editor window shows the following TypeScript code:

```
export class CrearProveedorComponent implements OnInit {
  formPro: FormGroup;

  constructor() { }

  ngOnInit() {
  }
}
```

c) Implementar el 'constructor' de formularios en el constructor del componente:

```
12  
13     constructor(private ff: FormBuilder) {}  
14
```

d) Crear el objeto del formulario al iniciar el componente (ngOnInit()):

```
15     ngOnInit() {  
16         this.formPro = this.ff.group({  
17             nombre: '',  
18             cif: '',  
19             localidad: ''  
20         })  
21     }  
22
```


3er Paso Almacenar el contenido del formulario en el localStorage del navegador

En el archivo TypeScript del componente declaramos el método especificado en el formulario y grabamos el contenido del objeto en localStorage:

```
src > app > crear-proveedor > crear-proveedor.component.ts > CrearProveedorComponent >
14
15   ngOnInit() {
16     this.formPro = this.ff.group({
17       nombre: '',
18       cif: '',
19       localidad: ''
20     })
21   }
22
23   guardarPro() {
24     let proveedor = {
25       nombre: this.formPro.get('nombre').value,
26       cif: this.formPro.get('cif').value,
27       localidad: this.formPro.get('localidad').value
28     }
29     localStorage.setItem('proveedor', JSON.stringify(proveedor));
30   }
31
32 }
```

Comprobamos en el navegador que se almacena correctamente:

