

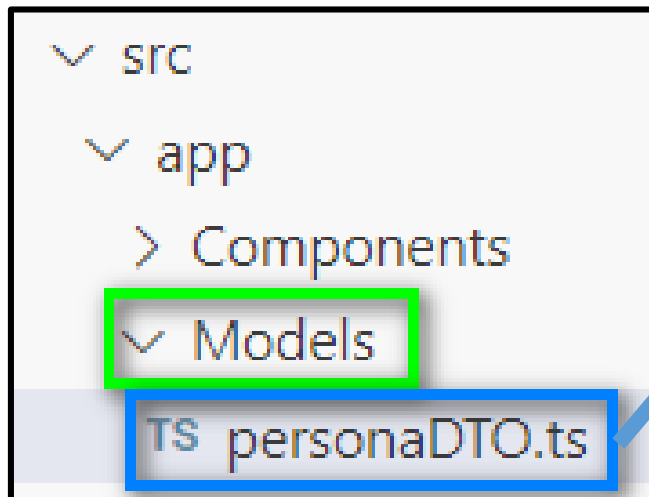


Angular

6.2. Data Services

1. Caso de partida

- Creamos una aplicación nueva.
- La aplicación contendrá un modelo Persona, con 2 atributos: nombre y email.



```
TS personaDTO.ts X
src > app > Models > TS personaDTO.ts > Persona
1  export class Persona {
2      nombre:string;
3      email:string;
4
5      constructor(nombre:string="",email:string="") {
6          this.nombre=nombre;
7          this.email=email;
8      }
9  }
```

1. Caso de partida

- Habrá un formulario para insertar personas y una zona donde mostraremos dichos datos.
- Vamos a tener un componente formulario, donde insertaremos las personas y otro componente listado donde mostraremos los datos.
- Creamos ambos componentes:

1. Caso de partida

```
c:\angular\servicio>ng g c Components\formulario
CREATE src/app/Components/formulario/formulario.component.html (26 bytes)
CREATE src/app/Components/formulario/formulario.component.spec.ts (652 bytes)
CREATE src/app/Components/formulario/formulario.component.ts (225 bytes)
CREATE src/app/Components/formulario/formulario.component.css (0 bytes)
UPDATE src/app/app.module.ts (621 bytes)
```

▼ Components

- > formulario
- > listado

▼ Components

- ▼ formulario
 - # formulario.component.css
 - <> formulario.component.html
 - TS formulario.component.ts
- ▼ listado
 - # listado.component.css
 - <> listado.component.html
 - TS listado.component.ts

```
c:\angular\servicio>ng g c Components\listado
CREATE src/app/Components/listado/listado.component.html (23 bytes)
CREATE src/app/Components/listado/listado.component.spec.ts (631 bytes)
CREATE src/app/Components/listado/listado.component.ts (213 bytes)
CREATE src/app/Components/listado/listado.component.css (0 bytes)
UPDATE src/app/app.module.ts (718 bytes)
```

2. Crear un servicio

- Vamos a crear un servicio para que el acceso a los datos no se produzca a través del componente directamente, y se realice usando un servicio.
- Para crear un servicio desde la terminal ejecutamos:
 - **ng generate service Services\persona**
 - **ng g c Services\persona**

2. Crear un servicio

```
c:\angular\servicio> ng g s Services\persona  
CREATE src/app/Services/persona.service.spec.ts (378 bytes)  
CREATE src/app/Services/persona.service.ts (145 bytes)
```

2. Crear un servicio

- Llamamos a este servicio **persona**.
- El código generado será el siguiente:

```
TS persona.service.ts X
src > app > Services > TS persona.service.ts > ...
1  import { Injectable } from '@angular/core';
2
3  @Injectable({
4    providedIn: 'root'
5  })
6  export class PersonaService {
7
8    constructor() { }
9  }
10
```

2. Crear un servicio

- Los datos y las operaciones que afecten al modelo los vamos a implementar en el servicio.
- Crearemos el array vacío y el método insertar dentro de este servicio.
- Acordarse de incluir el include al modelo dentro del servicio para poder utilizar el mismo.

2. Crear un servicio

```
export class PersonaService {  
  Arrpersonas:Persona[]=[];  
  constructor() { }  
  
  insertaPersona(nombre:string,email:string) {  
    const persona=new Persona(nombre,email);  
    this.Arrpersonas.push(persona);  
    console.log(this.Arrpersonas);  
  }  
}
```

2. Crear un servicio

- Ahora incluimos el servicio en el modulo (app.module.ts)

```
AppModule
],
providers: [
  PersonaService
],
bootstrap: [AppComponent]
})
```

```
import { PersonaService } from '../Services/persona.service';
```

2. Crear un servicio

- En el componente formulario, debemos insertar el servicio como un inyectable (injectable).
- Se hace cargando dicho servicio en el constructor de la clase (app.component.ts).

```
pp > Components > formulario > TS formulario.component.ts > FormularioComponent
import { Component } from '@angular/core';
import { PersonaService } from '../Services/persona.service';
```

```
export class FormularioComponent {
  constructor(public servicio:PersonaService) {
  }

  insertar(nombre:HTMLInputElement,email:HTMLInputElement){
    this.servicio.insertaPersona(nombre.value,email.value);
  }
}
```

2. Crear un servicio

Go to component

```
<h1>Formulario de Personas</h1>
```

```
<label>Nombre</label><br>
```

```
<input type="text" name="nombre" id="nombre" #nombre>
```

```
<label>Email</label><br>
```

```
<input type="email" name="email" id="email" #email>
```

```
<button (click)="insertar(nombre,email)">Insertar</button>
```

2. Crear un servicio

- Ahora el inserta persona utilizará también el servicio.

```
insertar(nombre:HTMLInputElement,email:HTMLInputElement){  
|   this.servicio.insertaPersona(nombre.value,email.value);  
}
```

2. Crear un servicio



The screenshot shows a web browser at `localhost:4200` displaying a form titled "Formulario de Personas". The form has two input fields: "Nombre" with the value "laberit" and "Email" with the value "sdafasdf". Below the fields is an "Insertar" button. The browser's developer console is open, showing the "Consola" tab. It displays a message from Angular: "Angular is running in development mode." followed by a log statement from `persona.service.ts:14`. The log shows an array with one object: `Persona {nombre: 'laberit', email: 'sdafasdf'}`.

Formulario de Personas

Nombre
laberit

Email
sdafasdf

Insertar

Consola

Angular is running in development mode. [core.mjs:30074](#)

[persona.service.ts:14](#)

▼ Array(1) ⓘ

- ▶ 0: Persona {nombre: 'laberit', email: 'sdafasdf'}
length: 1
- ▶ [[Prototype]]: Array(0)

2. Crear un servicio

- Al guardarse los datos en el servicio no es necesario están enviándolos de un componente a otro.
- En el componente listado, podremos acceder a los datos tan solo inyectando el servicio en el constructor.

```
export class ListadoComponent {
```

```
  constructor(public servicio:PersonaService) {
```

```
  }
```

```
}
```

```
import { PersonaService } from '../..../Services/persona.service';
```

2. Crear un servicio

- En el template podemos acceder al array de personas como si estuviese definido en nuestra estructura.

```
<> listado.component.html X
src > app > Components > listado > <> listado.component.html > ...
Go to component
1  <table>
2    <thead>
3      <th>Nombre</th>
4      <th>Email</th>
5    </thead>
6    <tbody>
7      <tr *ngFor="let elemento of servicio.Arrpersonas">
8        <td>{{ elemento.nombre }}</td>
9        <td>{{ elemento.email }}</td>
10     </tr>
11   </tbody>
12 </table>
13
```


2. Crear un servicio

- Ahora, en `app.component.html` podemos cargar ambos componentes y utilizarlos.

```
<> app.component.html X
src > app > <> app.component.html > ...
Go to component
1  <div class="container">
2    <app-formulario></app-formulario>
3    <hr />
4    <app-listado></app-listado>
5  </div>
6
7  <router-outlet></router-outlet>
8  |
```

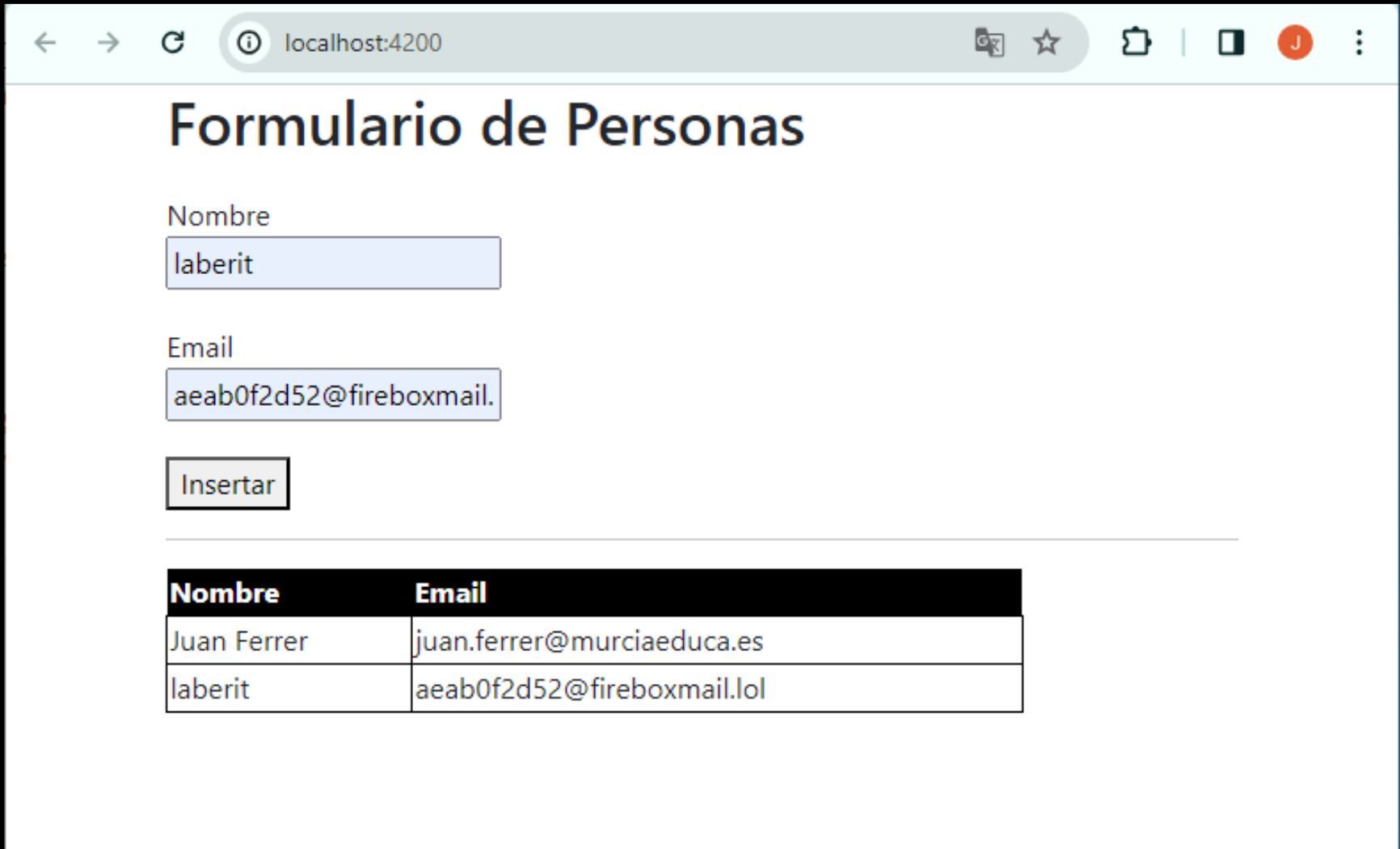
2. Crear un servicio

- Con un poco de CSSeo por nuestra parte se nos queda un proyecto guay.

```
# formulario.component.css X
src > app > Components > formulario >
1  button, label {
2      margin-top: 20px;
3      display: block;
4  }
5  |
```

```
# listado.component.css X
src > app > Components > listado > # listado.compone
1  table {
2      width: 80%;
3  }
4
5  td {
6      border: 1px solid black;
7  }
8
9  th {
10     background-color: black;
11     color: white;
12 }
13 |
```

2. Crear un servicio



← → ↻ ⓘ localhost:4200

Formulario de Personas

Nombre
laberit

Email
aeab0f2d52@fireboxmail.

Insertar

Nombre	Email
Juan Ferrer	juan.ferrer@murciaeduca.es
laberit	aeab0f2d52@fireboxmail.lol