

Angular

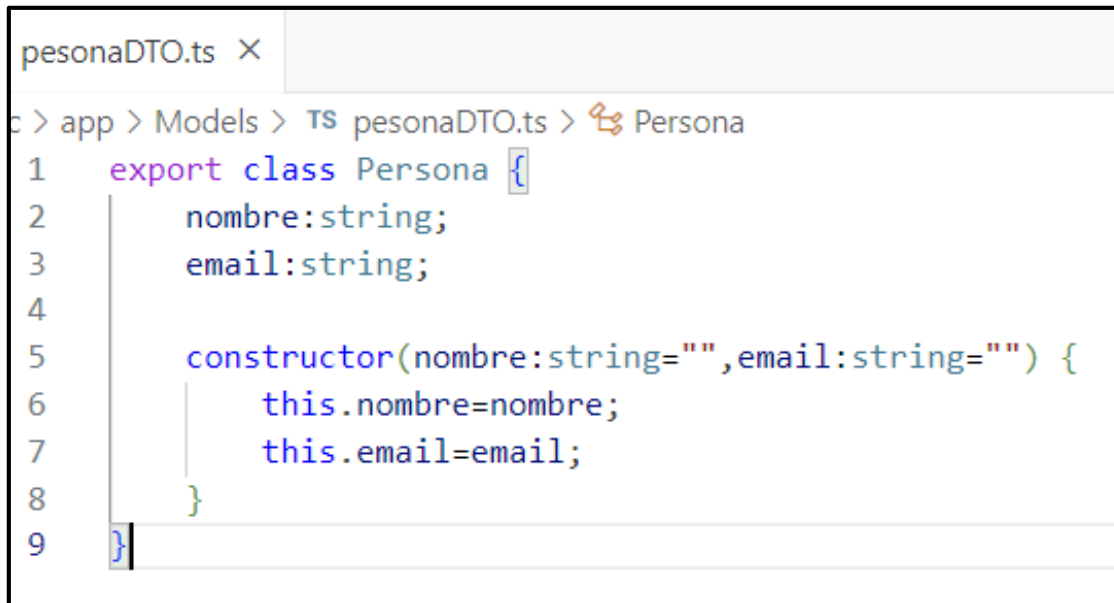
5.2. Directiva @Output

1. Directiva @Output

- En este caso vamos a enviar información de un componente hijo a un componente padre.
- Para ello nos vamos a valer de la directiva @Output

2. Caso de prueba

- Hemos creado una carpeta **Models**, donde hemos creado un modelo nuevo llamado: ***personaDTO.ts***. Lo usaremos para guardar el nombre y el email de dichas personas.



```
personaDTO.ts ×
c > 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  }
```

2. Caso de prueba

Go to component

```
<div class="container">
  <label class="form-group">Nombre</label>
  <input type="text" name="nombre" id="nombre" #nombre class="form-control">
  <label class="form-group">Email</label>
  <input type="email" name="email" id="email" #email class="form-control">
  <button (click)="insertar(nombre,email)" class="btn btn-secondary">Insertar</button>
  <hr>

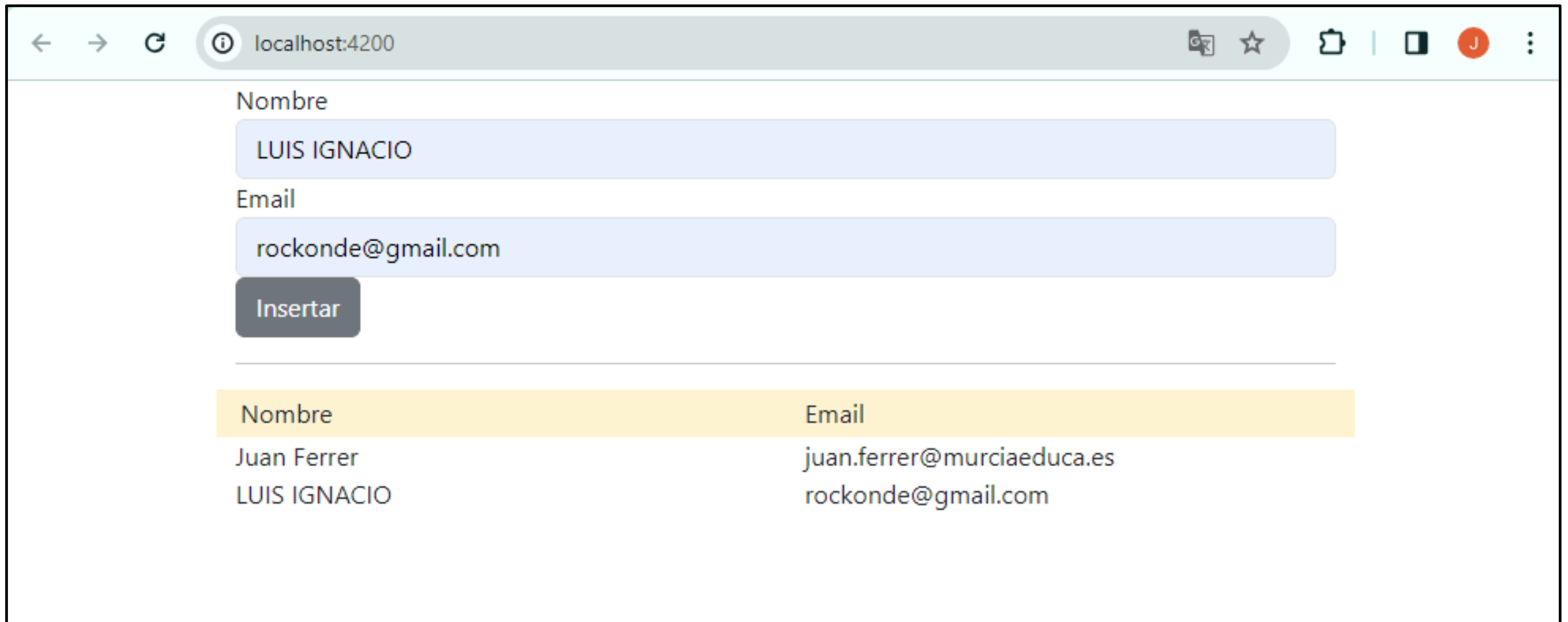
  <div class="row mark">
    <div class="col">Nombre </div>
    <div class="col">Email </div>
  </div>

  <div *ngFor="let elemento of ArrPersonas" class="row">
    <div class="col">{{elemento.nombre}} </div>
    <div class="col">{{elemento.email}} </div>
  </div>
</div>
<router-outlet></router-outlet>
```

2. Caso de prueba

```
export class AppComponent {  
  ArrPersonas: Persona[] = [];  
  
  insertar(nombre: HTMLInputElement, email: HTMLInputElement) {  
    const obj = new Persona(nombre.value, email.value);  
    this.ArrPersonas.push(obj);  
    console.log(this.ArrPersonas);  
  }  
}
```

2. Caso de prueba



A screenshot of a web browser window showing a form for adding a user and a table of existing users. The browser's address bar shows 'localhost:4200'. The form has two input fields: 'Nombre' (Name) with the value 'LUIS IGNACIO' and 'Email' with the value 'rockonde@gmail.com'. Below the inputs is a dark grey button labeled 'Insertar'. Below the form is a horizontal line, followed by a table with two columns: 'Nombre' and 'Email'. The table contains two rows of data.

Nombre	Email
Juan Ferrer	juan.ferrer@murciaeduca.es
LUIS IGNACIO	rockonde@gmail.com

3. @Output

- Vamos a generar un componente nuevo llamado formulario, en el que pediremos el nombre y el email.
- Estos datos los enviará al padre que guardará dichos datos y mostrará dichos elementos.
- Generamos el componente formulario.

3. @Output

```
c:\angular\output>ng g c formulario  
CREATE src/app/formulario/formulario.component.html (26 bytes)  
CREATE src/app/formulario/formulario.component.spec.ts (652 bytes)  
CREATE src/app/formulario/formulario.component.ts (225 bytes)  
CREATE src/app/formulario/formulario.component.css (0 bytes)  
UPDATE src/app/app.module.ts (491 bytes)
```


3. @Output

- Vamos a separar (de forma similar a @Input) el formulario original (app.component.html) en dos partes.
- Por un lado vamos a tener el componente formulario que va a incluir el formulario de inserción de personas.

3. @Output

<> formulario.component.html X

src > app > formulario > <> formulario.component.html >  hr

Go to component

```
1  <label class="form-group">Nombre</label>
2  <input type="text" name="nombre" id="nombre" #nombre
   class="form-control">
3  <label class="form-group">Email</label>
4  <input type="email" name="email" id="email" #email
   class="form-control">
5  <button (click)="insertar(nombre,email)" class="btn
   btn-secondary">Insertar</button>
6  <hr>
```

3. @Output

- En la parte TS del formulario vamos a definir una variable que será la que emita al padre los datos que hemos insertado.

```
export class FormularioComponent {  
  @Output() emitir=new EventEmitter<Persona>();  
  
  insertar(nombre: HTMLInputElement, email: HTMLInputElement) {  
    const obj = new Persona(nombre.value, email.value);  
    this.emitir.emit(obj);  
  }  
}
```

3. @Output

- Aquí aparecen las palabras claves @Output y EventEmitter, que nos van a permitir enviar al padre los datos que hemos recogido.
- En el padre tan solo hay que recoger dichos valores en la llamada al componente hijo.

3. @Output

<> app.component.html X

src > app > <> app.component.html >  div.container

Go to component

1 <div class="container">

2 <app-formulario (emitir)="insertar(\$event)"></app-formulario>

3

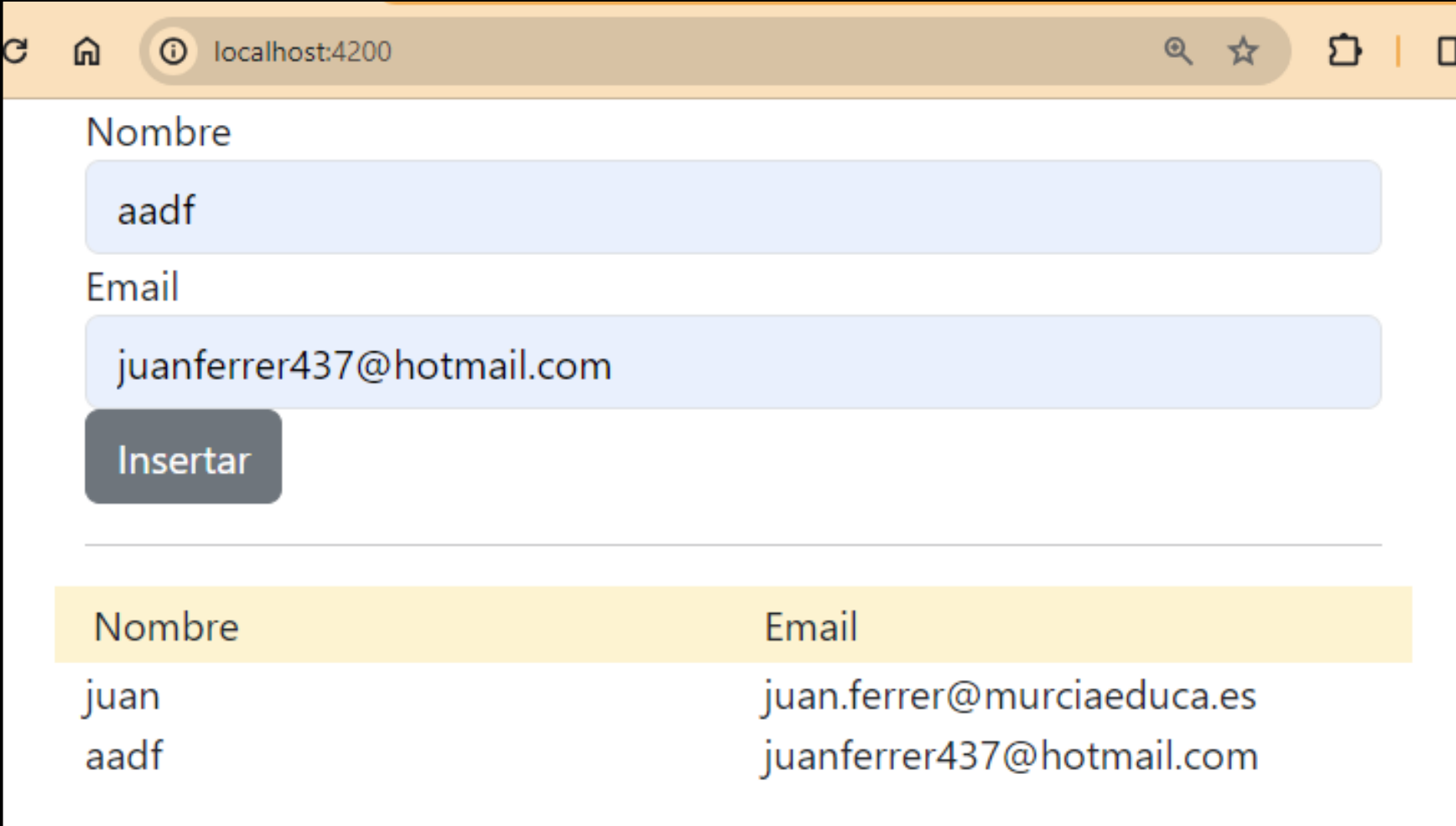
3. @Output

- Hay que poner el nombre de la variable que hemos definido en el hijo con @Output, y pondremos el nombre del método que vamos a utilizar en el padre para insertar los valores recogidos.

```
export class AppComponent {  
  ArrPersonas: Persona[] = [];  
  
  insertar(obj:Persona):void {  
    this.ArrPersonas.push(obj);  
    console.log(this.ArrPersonas);  
  }  
}
```

3. @Output

- Por supuesto, el resultado es el mismo que en el ejemplo de partida.



A screenshot of a web browser window showing a form and a table. The browser's address bar displays 'localhost:4200'. The form contains two input fields: 'Nombre' with the value 'aadf' and 'Email' with the value 'juanferrer437@hotmail.com'. Below the fields is a dark grey button labeled 'Insertar'. A horizontal line separates the form from a table below. The table has two columns, 'Nombre' and 'Email', with a yellow header row. The table contains two rows of data: one with 'juan' and 'juan.ferrer@murciaeduca.es', and another with 'aadf' and 'juanferrer437@hotmail.com'.

Nombre	Email
juan	juan.ferrer@murciaeduca.es
aadf	juanferrer437@hotmail.com