

## Solución Práctica 3.4 Formulario Reactive

Rama git de partida: **p\_form\_reactive**

Rama git finalizada: **p\_form\_reactive\_resuelta**

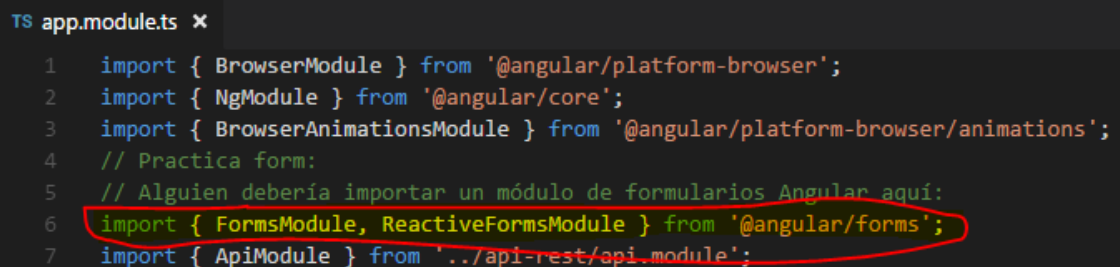
**Vamos a hacer un formulario para las nuevas Salas con 4 campos:**

- **nombre**: obligatorio, con al menos 4 caracteres
- **descripcion**: como máximo 50 caracteres
- **capacidad**: máximo 2 caracteres y que cumpla el patrón definido en `formatoCapacidad`
- **localizacion**: como máximo 50 caracteres

SOLUCIÓN:

1. Ya está hecho (para que sigan funcionando los otros formularios de la aplicación), pero si no lo tuviéramos de antes, en **app.module.ts** incluimos el **import** del módulo de **ReactiveForms**:

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
```



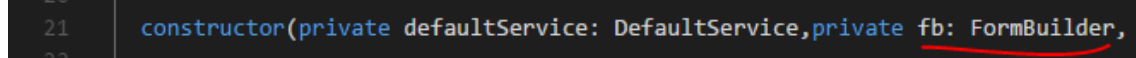
```
TS app.module.ts x
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3  import { BrowserModuleAnimationsModule } from '@angular/platform-browser/animations';
4  // Practica form:
5  // Alguien debería importar un módulo de formularios Angular aquí:
6  import { FormsModule, ReactiveFormsModule } from '@angular/forms';
7  import { ApiModule } from '../api-rest/api.module';
```

En **nueva-sala.component.ts**

2. Ya está hecho, pero si no lo tuviéramos de antes, incluimos el **import** de **FormGroup**, **FormControl**, **FormBuilder**, **Validators**:

```
import { FormGroup, FormControl, FormBuilder, Validators } from
 '@angular/forms';
```

3. Ya está hecho, pero si no lo tuviéramos de antes, incluimos el **FormBuilder** en el **constructor**:



```
21  constructor(private defaultService: DefaultService, private fb: FormBuilder,
```

4. Ya está hecho. En **ngOnInit** hay una llamada a un método **this.createForm** donde creamos el formulario:

```
23   ngOnInit() {
24       this.idEdificio = this.route.snapshot.params['idEdificio'];
25       this.createForm();
26   }
```

5. En el método **createForm()** declaramos los **campos** del formulario dentro de **fb.group** ({:
- El primer parámetro, el valor por defecto, lo dejamos vacío en todos.
  - El segundo parámetro, los **Validators**, si hay varios van entre [ ]
  - En el caso de Validators.**pattern** ponemos la constante **formatoCapacidad**

```
nombre: ['', [Validators.required, Validators.minLength(4)]],
descripcion: ['', Validators.maxLength(50)],
capacidad: ['', [Validators.maxLength(2),
Validators.pattern(formatoCapacidad)]],
localizacion: ['', Validators.maxLength(50)],
```

```
48   createForm() {
49       const formatoCapacidad = '[1-9]+[0-9]*$'
50       this.salaForm = this.fb.group({
51           // Práctica 4: Formulario Reactive:
52           // Alguien debería declarar los campos del formulario aquí:
53           nombre: ['', [Validators.required, Validators.minLength(4)]],
54           descripcion: ['', Validators.maxLength(50)],
55           capacidad: ['', [Validators.maxLength(2), Validators.pattern(formatoCapacidad)]],
56           localizacion: ['', Validators.maxLength(50)],
```

6. En **validationMessages** declaramos los mensajes de error:

```
'nombre': {
    'required': 'Nombre es obligatorio',
    'minlength': 'Nombre debe tener mínimo 4 letras',
},
'descripcion': {
    'maxlength': 'La descripción no puede tener más de 50
caracteres',
},
'capacidad': {
    'maxlength': 'Capacidad no puede tener más de 2 números',
    'pattern': 'Capacidad sólo admite números',
},
'localizacion': {
    'maxlength': 'La localización no puede tener más de 50
caracteres',
}
```

```

97  validationMessages = {
98      // Práctica 4: Formulario Reactive:
99      // Alguien debería declarar los mensajes de validación de cada campo del formulario aquí:
100  'nombre': {
101      'required': 'Nombre es obligatorio',
102      'minlength': 'Nombre debe tener mínimo 4 letras',
103  },
104  'descripcion': {
105      'maxlength': 'La descripción no puede tener más de 50 caracteres',
106  },
107  'capacidad': {
108      'maxlength': 'Capacidad no puede tener más de 2 números',
109      'pattern': 'Capacidad sólo admite números',
110  },
111  'localizacion': {
112      'maxlength': 'La localización no puede tener más de 50 caracteres',
113  }
114  };

```

En **nueva-sala.component.html**

7. Ya está hecho. En la etiqueta **<form>** añadimos el **formGroup**:

```

6  <form [formGroup]="salaForm" (ngSubmit)="onSubmit(f)">

```

8. En las etiquetas **<input>** añadimos los **formControlNames** con los nombres que hemos definido en el **fb.group**:

```

<input matInput formControlName="nombre" placeholder="Nombre" required>
<input matInput formControlName="descripcion" placeholder="Descripción">
<input matInput formControlName="capacidad" placeholder="Capacidad" >
<input matInput formControlName="localizacion" placeholder="Localizacion">

```