

FORMULARIOS REACTIVOS

ÍNDICE

- Formularios
- Formularios reactivos
- Validación formularios reactivos

FORMULARIOS

Angular provee dos tipos de formularios. Ambos recogen los eventos y datos desde las etiquetas *input*, validan los datos, crean modelos de formularios y los envían al componente.

- Formularios *Template-driven*, o de Plantilla: para webs con pocos formularios, 1-2 máximo.
- Formularios *Reactive*, o reactivos: para webs con muchos formularios como una zona de administración.

	<i>Reactive</i>	<i>Template-driven</i>
Preparación	Se crea en el componente	Creado por directivas en el HTML
Modelo	Estructural	No estructural
Previsibilidad	Síncrona	Asíncrona
Validación	Funciones	Directivas
Mutabilidad	Inmutable	Mutable
Escalabilidad	Acceso a API a bajo nivel	Abstracción de la API

FORMULARIOS REACTIVOS

- Declaración del formulario como objeto
- Declaración de cada uno de los elementos del formulario y de sus valores
- Importación de componentes de validación
- Asignación de métodos
- Trabajaremos con Observables
- Facilitan el trabajo a la hora de hacer tests unitarios

```
import { RouterModule } from '@angular/router';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

import { DashboardComponent } from './dashboard.component';
import { ReactiveFormsModule } from '@angular/forms';

@NgModule({
  declarations: [DashboardComponent],
  imports: [
    CommonModule,
    ReactiveFormsModule,
    RouterModule.forChild([
      { path: '', component: DashboardComponent}
    ])
  ]
})
export class DashboardModule { }
```

FORMULARIO REACTIVO

Lo primero que debemos hacer es añadir el módulo *ReactiveFormsModule* a nuestro módulo para poder utilizar las funcionalidades de Angular para formulario reactivo.

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

@Component({
  selector: 'app-edit',
  templateUrl: './edit.component.html',
  styleUrls: ['./edit.component.css']
})
export class EditComponent implements OnInit {
  // Atributos
  public usuario: FormControl;

  constructor() {
    this.usuario = new FormControl('');
  }

  ngOnInit(): void {
  }
}
```

LO MÁS BÁSICO

La clase más simple para crear un formulario reactivo es *FormControl*. Con él creamos una instancia de un objeto y podremos utilizarlo en el formulario del HTML

LO MÁS BÁSICO

```
<form>
  <label>Usuario:</label>
  <input type="text" [formControl]="usuario">
</form>
```

HTML

Formulario

HTML

Como recuperar el valor

```
<p>
  Valor atributo: {{ usuario.value }}
</p>
```

Componente

Formas de recuperar el
valor de nuestro atributo

```
this.usuario.valueChanges.subscribe( (value) => {
  console.log(value);
});
// O también
this.usuario.value
```

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

@Component({
  selector: 'app-edit',
  templateUrl: './edit.component.html',
  styleUrls: ['./edit.component.css']
})
export class EditComponent implements OnInit {
  // Atributos
  public usuario: FormGroup;

  constructor() {
    this.usuario = new FormGroup({
      nombre: new FormControl(''),
      apellido: new FormControl('')
    });
  }

  ngOnInit(): void {
  }
}
```

FORMGROUP COMPONENTE

Para formulario de varios campos usaremos *FormGroup*. Esta clase anida *FormControl* en su inicialización.

FORMGROUP HTML

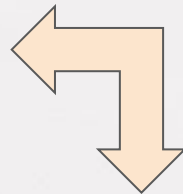
```
<form [formGroup]="usuario">
  <label>Nombre:</label>
  <input type="text" formControlName="nombre">

  <label>Apellido:</label>
  <input type="text" formControlName="apellido">
</form>
```

Para enlazar con el atributo del componente usaremos *formGroup* para el formulario y *formControlName* para cada uno de los valores.

FORMGROUP ANIDADOS

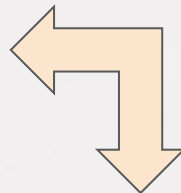
```
this.usuario = new FormGroup({  
  nombre: new FormControl(''),  
  apellido: new FormControl(''),  
  direccion: new FormGroup({  
    calle: new FormControl(''),  
    ciudad: new FormControl('')  
  })  
});
```



```
<form [formGroup]="usuario">  
  <label>Nombre:</label>  
  <input type="text" formControlName="nombre">  
  
  <label>Apellido:</label>  
  <input type="text" formControlName="apellido">  
  
  <div formGroupName="direccion">  
    <label>Calle:</label>  
    <input type="text" formControlName="calle">  
  
    <label>Ciudad:</label>  
    <input type="text" formControlName="ciudad">  
  </div>  
</form>
```

FORMBUILDER

```
this.usuario = this.formBuilder.group({  
  nombre: [''],  
  apellido: [''],  
  direccion: this.formBuilder.group({  
    calle: [''],  
    ciudad: ['']  
  })  
});
```



```
<form [formGroup]="usuario">  
  <label>Nombre:</label>  
  <input type="text" formControlName="nombre">  
  
  <label>Apellido:</label>  
  <input type="text" formControlName="apellido">  
  
  <div formGroupName="direccion">  
    <label>Calle:</label>  
    <input type="text" formControlName="calle">  
  
    <label>Ciudad:</label>  
    <input type="text" formControlName="ciudad">  
  </div>  
</form>
```