Florida
Universitària

Formularios reactivos

Curso 2025/2026

Paco Segura

Formularios reactivos

- Angular dispone de dos formas de trabajar con formularios: Template Forms y Reactive Forms.

- **Template forms:** trabajan con directivas para acceder a los datos introducidos y validarlos. Útiles para añadir un formulario sencillo a una aplicación. No escalan tan bien como los formularios reactivos.

- **Reactive forms:** trabajan a partir de funciones, son más robustos, más escalables, reutilizables y comprobables.

Formularios reactivos

- En el componente donde vayamos a implementar el formulario, importamos **ReactiveFormsModule, FormGroup** y **FormControl**:

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

@Component({
  selector: 'app-view-one',
  imports: [ReactiveFormsModule],
  templateUrl: './view-one.html',
  styleUrl: './view-one.css',
})
```

Formularios reactivos

- **Archivo HTML:**

```
<form (ngSubmit)="onSubmit()" [formGroup]="reactiveForm">
    <input type="text" name="name" placeholder="Nombre" formControlName="name"><br>
    <input type="text" name="age" placeholder="Edad" formControlName="age"><br>
    <input type="text" name="email" placeholder="e-mail" formControlName="email"><br>
    <div id='render'>
        <button type="submit">MOSTRAR</button>
    </div>
</form>
```

El evento asociado a un formulario es **ngSubmit**

Empleamos **formGroup** para asignarle un objeto donde agruparemos todos los campos del formulario. Cada campo del formulario se define con **FormControlName.**

Formularios reactivos

- **Archivo TypeScript:**

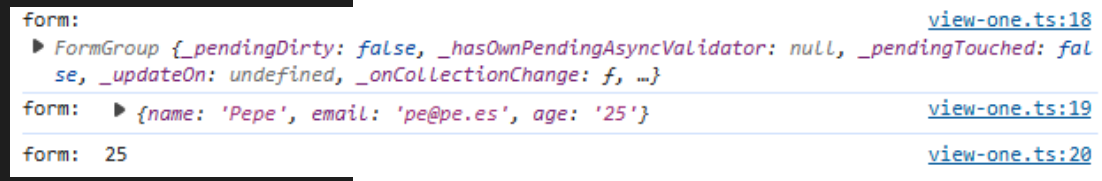
```typescript
import { Component } from '@angular/core';
import { FormControl, FormGroup, ReactiveFormsModule } from '@angular/forms';

@Component({
  selector: 'app-view-one',
  imports: [ReactiveFormsModule],
  templateUrl: './view-one.html',
  styleUrl: './view-one.css',
})
export class ViewOne {
  reactiveForm = new FormGroup({
    name: new FormControl(''),
    email: new FormControl(''),
    age: new FormControl(''),
  });

  public onSubmit(): void {
    console.log('form: ', this.reactiveForm);
    console.log('form: ', this.reactiveForm.value);
    console.log('form: ', this.reactiveForm.value.age);
  }
}
```

FormGroup permite agrupar en un objeto todos los campos del formulario. Cada campo del formulario se define con **FormControl**.

Cuando el usuario activa el evento submit, los resultados introducidos en los campos del formulario se guardan en la variable en la que hemos definido el **FormGroup**. Con ***.value** podemos acceder a los valores guardados.



Formularios reactivos

- Al trabajar con formularios disponemos de varias funcionalidades que podemos emplear:
    - **reset():** permite borrar del archivo HTML los datos introducidos por el usuario:

```
public onSubmit(): void {
  console.log('form: ', this.reactiveForm);
  console.log('form: ', this.reactiveForm.value);
  console.log('form: ', this.reactiveForm.value.age);
  this.reactiveForm.reset();
}
```

Formularios reactivos

- **getRawValue():** permite obtener directamente los datos introducidos por el usuario –para por ejemplo guardarlos en una variable-:

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

@Component({
  selector: 'app-view-one',
  imports: [ReactiveFormsModule],
  templateUrl: './view-one.html',
  styleUrl: './view-one.css',
})
export class ViewOne {
  reactiveForm = new FormGroup({
    name: new FormControl(''),
    email: new FormControl(''),
    age: new FormControl(''),
  });

  public onSubmit(): void {
    let data = this.reactiveForm.getRawValue();
    console.log(data);
    console.log(data.name);
  }
}
```

Formularios reactivos

- Aunque inicialicemos cada campo del formulario con un valor, por defecto el formulario le añadirá el tipo null. Por ejemplo si definimos un campo como string, el tipo será **string | null**:

Formularios reactivos

- Para evitarlo, añadimos la propiedad ***nonNullable***:

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

@Component({
  selector: 'app-view-one',
  imports: [ReactiveFormsModule],
  templateUrl: './view-one.html',
  styleUrl: './view-one.css',
})
export class ViewOne {
  reactiveForm = new FormGroup({
    name: new FormControl('', { nonNullable: true }),
    email: new FormControl('', { nonNullable: true }),
    age: new FormControl('', { nonNullable: true }),
  });
```