

# 16 - Formularios reactivos : controles checkbox, radio y select

Vimos en los conceptos anteriores como trabajar con formularios reactivos en Angular, pero siempre utilizando un input de tipo text o un textarea. Ahora veremos como configurar los controles checkbox, radio y select.

## Problema

Confeccionar un formulario que permita al usuario ingresar un valor decimal y luego pueda seleccionar:

- Mediante dos controles radio puede indicar si quiere la conversión a hexadecimal u octal.
- Mediante un control select indica si quiere calcular el valor ingresado elevado a la potencia 2, 3 o 4.
- Finalmente con un control checkbox puede indicar que muestre la cantidad de dígitos del valor ingresado.
- Crearemos primero el proyecto:

```
ng new proyecto011
```

- Modificamos la vista de la componente que muestra el formulario reactivo (app.component.html):

```

<form [formGroup]="formularioConversion" (ngSubmit)="submit()">
  <p>Ingrese número decimal:
    <input type="text" formControlName="numerodecimal">
  </p>
  <p>Convertir a:<br>
    <input type="radio" formControlName="base" value="octal">Octal
    <input type="radio" formControlName="base" value="hexadecimal"
  >Hexadecimal
  </p>
  <p>Elevar a:<br>
    <select formControlName="potencia">
      <option>2</option>
      <option>3</option>
      <option>4</option>
    </select>
  </p>
  <p><input type="checkbox" formControlName="largo">Mostrar cantidad de dígitos
  </p>
  <p><button type="submit">Confirmar</button></p>
</form>
<div>
  <p>Conversion:{{conversion}}</p>
  <p>Potencia:{{potenciaNumero}}</p>
  <p>Cantidad de dígitos:{{cantidad}}</p>
</div>
<router-outlet />

```

Analizaremos este archivo en conjunto luego de presentar 'app.component.ts'

- La clase asociada a la vista es (app.component.ts):

```

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

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, ReactiveFormsModule],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  conversion = '';
  potenciaNumero = '';
  cantidad = '';

  formularioConversion = new FormGroup({
    numerodecimal: new FormControl(''),
    base: new FormControl('octal'),
    potencia: new FormControl('2'),
    largo: new FormControl(true)
  });

  submit() {
    if (this.formularioConversion.value.numerodecimal) {
      if (this.formularioConversion.value.base == "hexadecimal")
        this.conversion = parseInt(this.formularioConversion.value.numerodecimal).toString(16);
      if (this.formularioConversion.value.base == "octal")
        this.conversion = parseInt(this.formularioConversion.value.numerodecimal).toString(8);
      if (this.formularioConversion.value.numerodecimal && this.formularioConversion.value.potencia)
        this.potenciaNumero = Math.pow(parseInt(this.formularioConversion.value.numerodecimal), parseInt(this.formularioConversion.value.potencia)).toString();
      if (this.formularioConversion.value.largo)
        this.cantidad = this.formularioConversion.value.numerodecimal.length.toString();
    }
  }
}

```

Creamos un objeto de la clase FormGroup y le pasamos un objeto literal con la referencia a los 4 controles de formulario que necesitamos:

```
formularioConversion = new FormGroup({
  numerodecimal: new FormControl(''),
  base: new FormControl('octal'),
  potencia: new FormControl('2'),
  largo: new FormControl(true)
});
```

El atributo base se asocia a los controles 'radio' de la vista, he indicamos que aparezca seleccionado por defecto aquel que almacena en value el valor 'octal':

```
<p>Convertir a:<br>
  <input type="radio" formControlName="base" value="octal">Octal
  <input type="radio" formControlName="base" value="hexadecimal">Hexadecimal
</p>
```

El control HTML select se asocia al atributo 'potencia', indicando que se muestre seleccionada la opción con valor '2':

```
potencia: new FormControl('2'),
```

La vista para este FormControl es:

```
<p>Elevar a:<br>
  <select formControlName="potencia">
    <option>2</option>
    <option>3</option>
    <option>4</option>
  </select>
</p>
```

Finalmente para el control checkbox indicamos que se muestre seleccionado por defecto al cargar el formulario:

```
largo: new FormControl(true)
```

Cuando se presiona el botón se ejecuta el método 'submit', donde efectuamos las distintas operaciones según los valores seleccionados en el formulario:

```
submit() {
  if (this.formularioConversion.value.numerodecimal) {
    if (this.formularioConversion.value.base == "hexadecimal")
      this.conversion = parseInt(this.formularioConversion.value.numerodecimal).toString(16);
    if (this.formularioConversion.value.base == "octal")
      this.conversion = parseInt(this.formularioConversion.value.numerodecimal).toString(8);
    if (this.formularioConversion.value.numerodecimal && this.formularioConversion.value.potencia)
      this.potenciaNumero = Math.pow(parseInt(this.formularioConversion.value.numerodecimal), parseInt(this.formularioConversion.value.potencia)).toString();
    if (this.formularioConversion.value.largo)
      this.cantidad = this.formularioConversion.value.numerodecimal.length.toString();
  }
}
```

Si ejecutamos la aplicación tenemos una interfaz similar a:

Proyecto011

localhost:4200

Ingrese número decimal:

Convertir a:

☒ Octal ☐ Hexadecimal

Elevar a:

☒ Mostrar cantidad de dígitos

Conversion:377

Potencia:65025

Cantidad de dígitos:3

Podemos probar esta aplicación en la web [aquí](#).

[Retornar](#)