17 - Formularios reactivos : FormGroup anidados

Cuando trabajamos con formularios reactivos en Angular normalmente creamos un FormGroup para agrupar los controles visuales que se les asocian un objeto de la clase FormControl a cada uno.

Podemos crear dentro de un FormGroup otro FormGroup que agrupe un conjunto de controles relacionados.

Problema

Confeccionar un formulario que permita cargar los siguientes datos de un alumno: dni, nombre, materia y sus tres notas. Agrupar las tres notas dentro de un FormGroup anidado al principal.

Al presionar un botón mostrar un mensaje si se encuentra aprobado (todas las notas mayores o iguales a 4)

- Crearemos primero el proyecto:

```
ng new proyecto012
```

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

```
<form [formGroup]="formAlumno" (ngSubmit)="submit()">
  <p>Dni:
    <input type="text" formControlName="dni">
  </p>
  <p>Nombre:
    <input type="text" formControlName="nombre">
  </p>
  <div formGroupName="notas">
    <p>Primer nota:
      <input type="text" formControlName="nota1">
    </p>
    <p>Segunda nota:
      <input type="text" formControlName="nota2">
    </p>
    <p>Tercer nota:
      <input type="text" formControlName="nota3">
    </p>
  </div>
  <p><button type="submit">Confirmar</button></p>
</form>
<div>
  <p>Resultado:{{resultado}}</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 '@angu
lar/forms';

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

  formAlumno = new FormGroup({
    dni: new FormControl(''),
    nombre: new FormControl(''),
    notas: new FormGroup({
      nota1: new FormControl(''),
      nota2: new FormControl(''),
      nota3: new FormControl('')
    })
  });

  submit() {
    if (this.formAlumno.value.notas) {
      if (this.formAlumno.value.notas.nota1 &&
        this.formAlumno.value.notas.nota2 &&
        this.formAlumno.value.notas.nota3) {
        let nota1 = parseInt(this.formAlumno.value.notas.nota1);
        let nota2 = parseInt(this.formAlumno.value.notas.nota2);
        let nota3 = parseInt(this.formAlumno.value.notas.nota3);
        if (nota1 >= 4 && nota2 >= 4 && nota3 >= 4)
          this.resultado = "El alumno queda aprobado por esas nota
s";
        else
          this.resultado = "El alumno no aprueba por esas notas";
      }
    }
  }
}
```

Creamos un objeto de la clase FormGroup y dentro de este otro objeto de la clase FormGroup que agrupa las tres notas:

```
formAlumno = new FormGroup({
  dni: new FormControl(''),
  nombre: new FormControl(''),
  notas: new FormGroup({
    nota1: new FormControl(''),
    nota2: new FormControl(''),
    nota3: new FormControl('')
  })
});
```

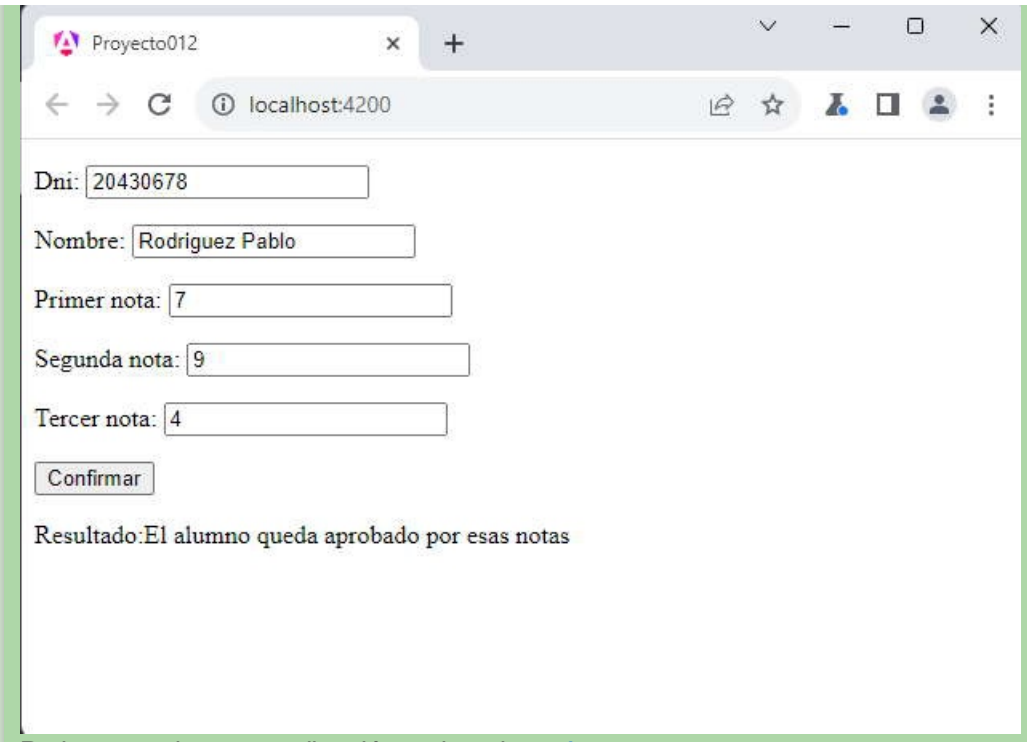
Esto provoca en la vista que cuando asociemos las tres notas debemos agruparlas dentro de un etiqueta HTML que defina una propiedad 'formGroupName':

```
<div formGroupName="notas">
  <p>Primer nota:
    <input type="text" formControlName="nota1">
  </p>
  <p>Segunda nota:
    <input type="text" formControlName="nota2">
  </p>
  <p>Tercer nota:
    <input type="text" formControlName="nota3">
  </p>
</div>
```

Cuando se presiona el botón se verifica si las tres notas ingresadas son mayores o iguales a 4 para mostrar por pantalla si el alumno se encuentra aprobado o no (mediante los if verificamos que se hayan cargado valores en los controles input):

```
submit() {
  if (this.formAlumno.value.notas) {
    if (this.formAlumno.value.notas.nota1 &&
      this.formAlumno.value.notas.nota2 &&
      this.formAlumno.value.notas.nota3) {
      let nota1 = parseInt(this.formAlumno.value.notas.nota1);
      let nota2 = parseInt(this.formAlumno.value.notas.nota2);
      let nota3 = parseInt(this.formAlumno.value.notas.nota3);
      if (nota1 >= 4 && nota2 >= 4 && nota3 >= 4)
        this.resultado = "El alumno queda aprobado por esas notas";
      else
        this.resultado = "El alumno no aprueba por esas notas";
    }
  }
}
```

Si ejecutamos la aplicación tenemos una interfaz similar a:



Podemos probar esta aplicación en la web aquí.

Retornar