

Tabla de Contenidos

- Manejo de Eventos
- Angular NgIf
- Angular ngIf
- Condiciones y Comportamiento
- Angular ngIf y Componentes
- Otros artículos relacionados

Hoy vamos a hablar Angular ngIf como directiva y las opciones que soporta. Para ello en un proyecto de Angular nos vamos a construir un componente sencillo que le denominaremos nota. Este componente albergará la variable nota del examen de un alumno.

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

```
@Component({
  selector: 'app-nota',
  templateUrl: './nota.component.html',
  styleUrls: ['./nota.component.css']
})
export class NotaComponent implements OnInit {

  nota:number;

  constructor() {

    this.nota=5;

  }

  ngOnInit() {
  }
}
```

```
}
```

Una vez tenemos el componente construido nos queda ver como quedaría la plantilla que muestra la nota. En este caso de partida es relativamente sencillo simplemente imprimimos la nota :

```
<p>Nota: {{nota}}</p>
```

El interface de usuario nos mostrará la nota:

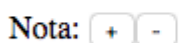
A screenshot of a user interface showing the text "Nota:5" in a simple, sans-serif font. The text is positioned to the right of a vertical grey line that acts as a visual separator.

Manejo de Eventos

Es momento de añadir al interface dos botones que nos permitan modificar los valores de la nota e incrementar o decrementarlo según nuestras necesidades:

```
<p>Nota :{{nota}}  
<input type="button" value="+" (click)="nota=nota+1">  
<input type="button" value="-" (click)="nota=nota-1">  
</p>
```

El interface se actualiza y podemos cambiar los valores de la nota:

A screenshot of a user interface showing the text "Nota:7" in a simple, sans-serif font. The text is positioned to the right of a vertical grey line that acts as a visual separator.A screenshot of a user interface showing the text "Nota:" followed by two small, rounded square buttons. The first button contains a plus sign (+) and the second button contains a minus sign (-). Both buttons are light grey with a thin border.

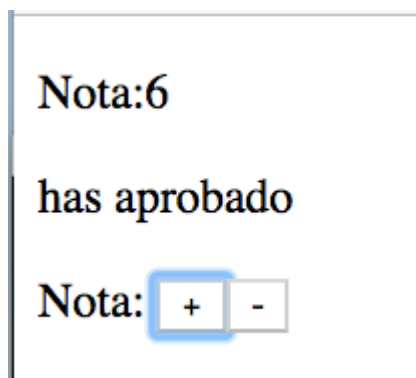
Angular NgIf

Es momento de usar la directiva ngIf para que nos imprima dependiendo del valor numérico

de la nota una calificación. Podríamos tener como punto de partida:

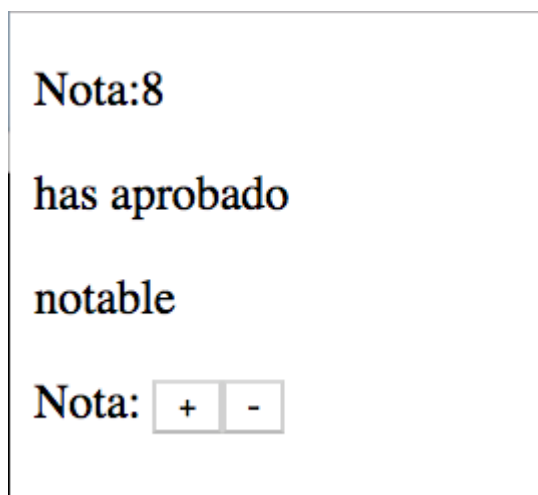
```
<p *ngIf="nota>=5">has aprobado</p>
```

Esto nos imprimirá en la página que hemos aprobado si la nota es superior a 5:



Podemos hacer una operación similar en el caso de que la nota este entre 7 y 9 de tal forma que se imprima notable apoyándonos en un operador and (&&).

```
<p *ngIf="nota>=7 && nota<9">notable</p>
```



Angular ngIf

Modificamos el componente para que incluya dos propiedades adicionales y podamos escribir condiciones más complejas:

```
export class NotaComponent implements OnInit {  
  
    nota:number;  
    asignatura:string;  
    edad:number;  
    constructor() {  
        this.asignatura="matematicas";  
        this.nota=5;  
        this.edad=16;  
    }  
  
    ngOnInit() {  
    }  
  
}
```

De esta forma podemos construir condiciones como las siguientes:

```
<p *ngIf="(nota>=9 && asignatura=='matematicas') ||  
(edad==16)">estudia aeronautica</p>
```

En este caso si la nota es superior a 9 y la asignatura es matemáticas o ha aprobado el examen con 16 años , le recomendamos estudiar Areonaútica.

Nota:9

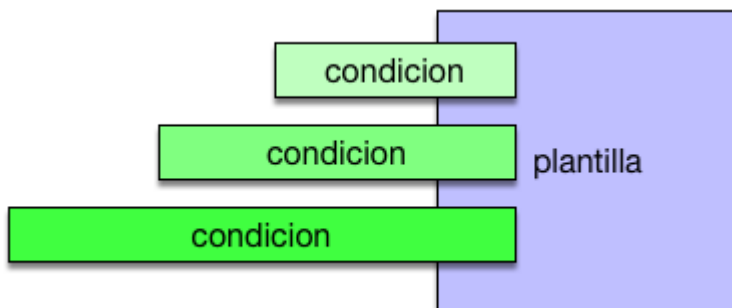
has aprobado

estudia aeronautica

Nota: + -

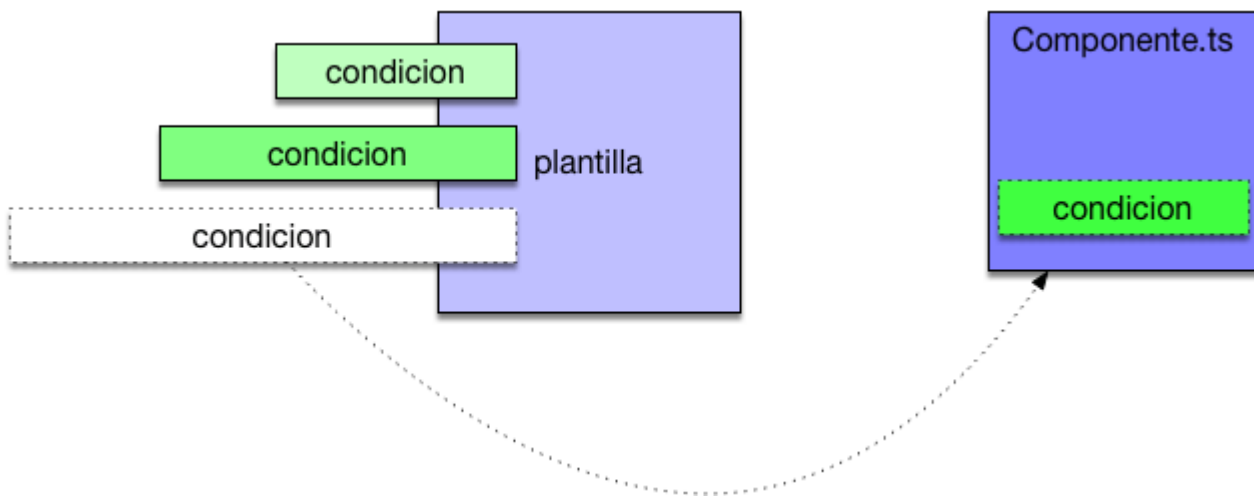
Condiciones y Comportamiento

Sin embargo cuando las condiciones se complican suele ser mejor proceder de otra forma ya que al final las condiciones no quedan claras debido a su tamaño.



Angular ngIf y Componentes

En vez de tener un ngIf inmenso con operadores como `&&` `||` o `!` es preferible definir esa condición a nivel del propio componente de tal forma que el interface de usuario quede mucho más cómodo.



Vamos a verlo:

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

@Component({ selector: 'app-nota',
  templateUrl: './nota.component.html',
  styleUrls: ['./nota.component.css'] })
export class NotaComponent implements OnInit {

  nota:number;
  asignatura:string;
  edad:number;
  constructor() {
    this.asignatura="matematicas";
    this.nota=5;
    this.edad=16;
  }
  esMuyListo():boolean {
    if((this.nota>=9 && this.asignatura=='matematicas') ||
(this.edad==16)) {
      return true;
    }
  }
}
```

```
        else {  
            return false;  
        }  
    }  
}
```

En este caso hemos implementado una función interna al componente que decide si el alumno es muy listo o no. La pasamos a invocar desde la plantilla:

```
<p *ngIf="esMuyListo()">estudia aeronautica</p>
```

Nota:9

has aprobado

estudia aeronautica

Nota:

Siempre que la lógica de las sentencias ngIf se complique suele ser mucho más cómodo moverla a los componentes para un uso más claro.

Otros artículos relacionados

1. [Angular Modules y el uso de servicios](#)
2. [Angular ngFor la directiva y sus opciones](#)
3. [TypeScript Union Type y como usarlos](#)

Externos

1. [Angular](#)