

App Calculadora con @Output



Reto: Modificar la App de Calculadora

En este reto, les dejaremos modificar la aplicación de Calculadora para que agreguen el concepto de @Input y @Output para modularizar la aplicación y aplicar los conceptos de los decoradores estudiados anteriormente.

Aplicación Calculadora

Operando A

Operando B

Sumar

Resultado: 7

En primer lugar, vamos a trabajar con el decorador @Output

1. Crear el componente formulario para procesar los valores de Operando A y Operando B:

ng g c formulario --skip-tests

En este ejemplo el componente de formulario (componente hijo) arroja un resultado, este resultado es procesado por el componente calculadora (componente padre). Por lo tanto, procesamos el evento que arroja el componente hijo.

El componente de formulario pasamos todos los componentes que tienen que ver con el formulario HTML para empezar a modularizar la aplicación de calculadora.

Código formulario.componente.html:

```
<div class="mb-3 d-flex align-items-center">
  <label for="operandoA" class="form-label me-2 text-nowrap">Operando A</label>
  <input type="number" id="operandoA" class="form-control text-center"
    [(ngModel)]="operandoA" placeholder="Operando A">
</div>
<div class="mb-3 d-flex align-items-center">
  <label for="operandoB" class="form-label me-2 text-nowrap">Operando B</label>
  <input type="number" id="operandoB" class="form-control text-center"
    [(ngModel)]="operandoB" placeholder="Operando B">
</div>
<button class="btn btn-primary mb-3 w-100" (click)="sumar()">Sumar</button>
```

Código formulario.componente.ts:

```
import { Component, EventEmitter, Output } from '@angular/core';
import { FormsModule } from '@angular/forms';

@Component({
  selector: 'app-formulario',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './formulario.componente.html',
  styleUrls: ['./formulario.componente.css']
})
export class FormularioComponent {
  operandoA: number | null = null;
  operandoB: number | null = null;

  @Output() resultadoSuma = new EventEmitter<number>();

  sumar(): void {
    if(this.operandoA && this.operandoB){
      const resultado = this.operandoA + this.operandoB;
      this.resultadoSuma.emit(resultado);
    }
  }
}
```

```

    }
    else{
        console.log("Deben introducir valores correctos")
    }
}
}
}

```

Al presionar el botón de sumar, se ejecuta el evento sumar(), y a su vez este evento emite el valor del resultado al componente padre. Veamos el nuevo código del componente de calculadora:

Código calculadora.componente.html:

```

<div class="container mt-5">
  <div class="d-flex justify-content-center">
    <div class="w-25">
      <app-formulario (resultadoSuma)="procesarResultado($event)" />
      <p class="text-center">Resultado: {{ resultado }}</p>
    </div>
  </div>
</div>

```

Al agregar el componente hijo <app-formulario/> estamos monitoreando el evento resultadoSuma, y al emitir el resultado lo recibe el método procesarResultado.

Código calculadora.componente.ts:

```

import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { FormularioComponent } from '../formulario/formulario.component';

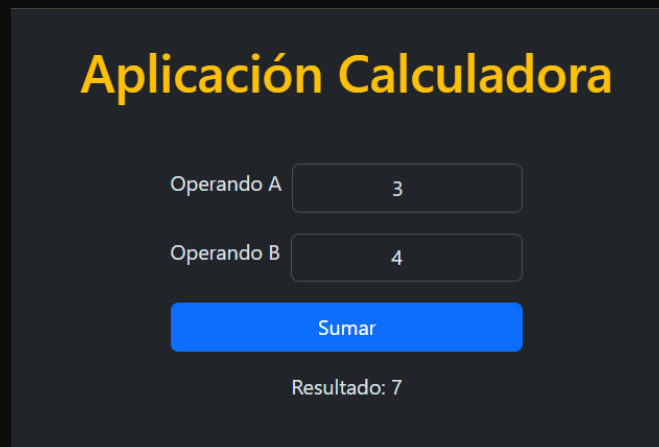
@Component({
  selector: 'app-calculadora',
  standalone: true,
  imports: [FormsModule, FormularioComponent],
  templateUrl: './calculadora.component.html',
  styleUrls: ['./calculadora.component.css']
})
export class CalculadoraComponent {
  resultado: number | null = null;

  procesarResultado(resultado: number) {
    this.resultado = resultado;
  }
}

```

```
}  
}
```

Aquí el método `procesarResultado()` recibe el valor de resultado emitido por el componente hijo de formulario. Y con ese valor podemos actualizar la vista en el componente padre de calculadora.



Aplicación Calculadora

Operando A

Operando B

Sumar

Resultado: 7

Esta es la primera parte de la aplicación. A continuación, el valor de resultado lo vamos a enviar a un nuevo componente hijo que muestre el valor de resultado con ayuda del decorador `@Output`. Esto lo veremos en la siguiente lección.

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)