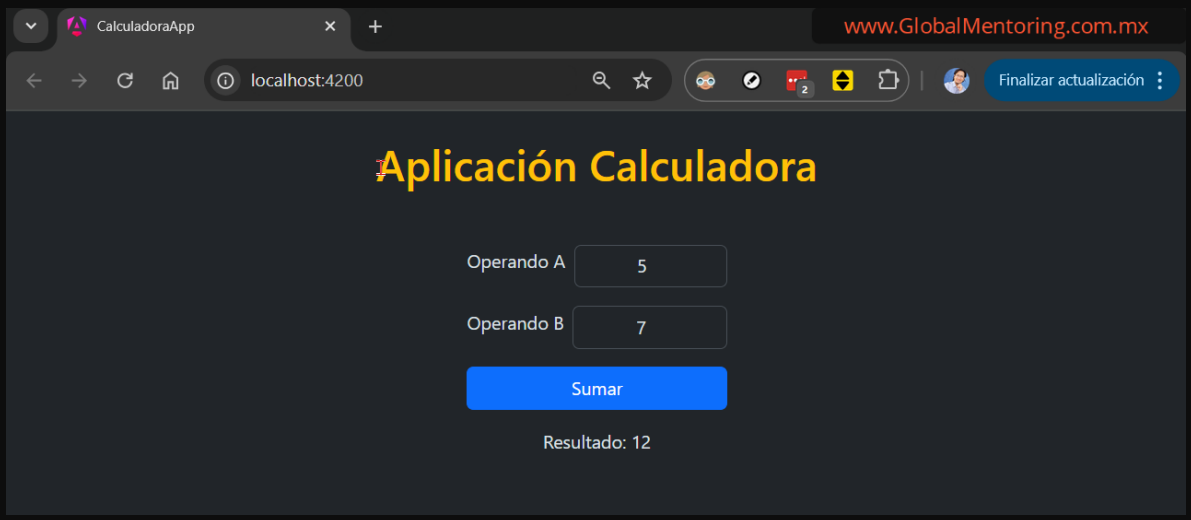
App Calculadora con @Input

Reto: Modificar la App de Calculadora con el decorador @Input

Continuamos con este reto. Ahora deben modificar la aplicación de Calculadora para que agreguen el concepto de @Input y así puedan mostrar el resultado en un componente nuevo (componente hijo)



Así que, ahora trabajaremos con el decorador @Input

1. Crear el componente de resultado:

ng g c resultado --skip-tests

En este ejemplo el componente de resultado (componente hijo) va a recibir el resultado que fue enviado desde el componente padre de calculadora. Para ello utilizaremos el decorador de @Input en el componente hijo de resultado y así recibir el valor de resultadoPadre de la suma. Los nombres pueden ser cualquier otro, pero hemos usado nombres muy explícitos para que quede claro cada concepto, pero incluso el nombre de las variables puede ser el mismo en todos los componentes, ej. resultado.

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)" />
            <app-resultado [resultadoHijo]="resultadoPadre"/>
        </div>
    </div>
</div>
```

Al agregar el componente hijo <app-resultado/> estamos pasando el resultado al componente hijo. Recordar que ya antes hemos recibido el resultado al monitorear el evento emitido de resultadoSuma.

Código calculadora.componente.ts:

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

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

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

El método procesarResultado es el encargado de actualizar la variable resultadoPadre, y al hacer este cambio, el componente hijo de resultado es el que recibe esta información a través del atributo definido con el decorador de @Input.

Código resultado.componente.html:

```
@if (resultadoHijo !== null) {
<p class="text-center">Resultado: {{ resultadoHijo }}</p>
}
```

Validamos que sólo se muestre el resultado si la variable resultadoHijo tiene valor distinto de null.

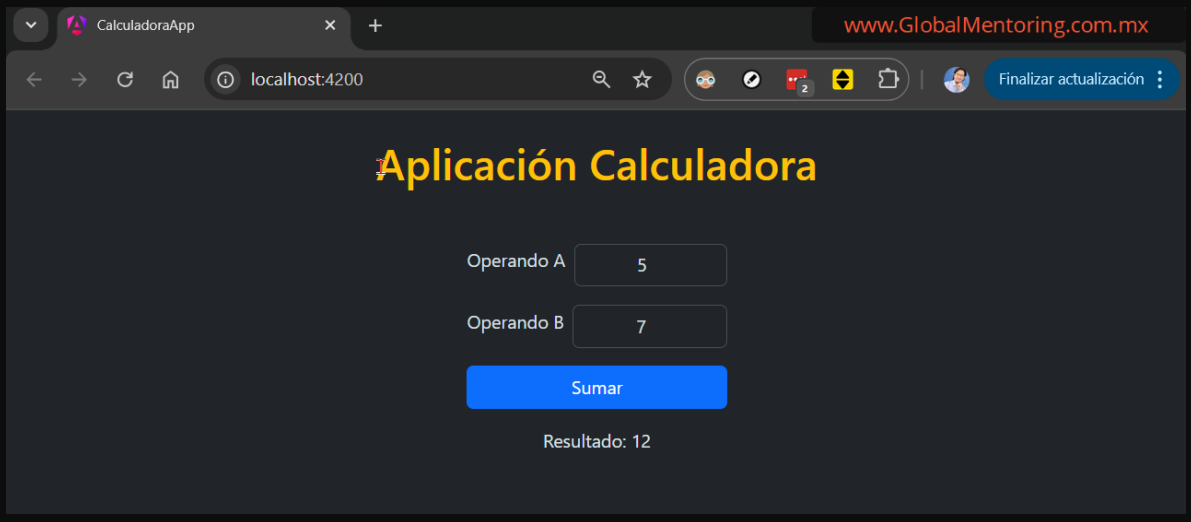
Código resultado.componente.ts:

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

@Component({
  selector: 'app-resultado',
  standalone: true,
  imports: [],
  templateUrl: './resultado.component.html',
  styleUrl: './resultado.component.css'
})
export class ResultadoComponent {
  @Input() resultadoHijo: number | null = null;
}
```

Aquí el atributo resultadoHijo es el que recibe el valor enviado por el componente padre de calculadora. Y con ese valor podemos actualizar la vista en el componente hijo de resultado.

De esta manera hemos modularizado nuestra aplicación aplicando el concepto de comunicación entre componentes con los decoradores @Input y @Output.



Saludos!

Ing. Ubaldo Acosta

Fundador de GlobalMentoring.com.mx