

# Métodos de la plate del padre

car nos de la componente padre a la componente  
e llamar a métodos de la componente hija  
e en el template HTML donde declaramos la

odos la componente debe definir una variable en

ore. Luego podemos llamar a métodos indicando

el mínimo y el máximo valor que puede mostrar.  
itar o decrementar en 1 el valor actual.  
merico y una variable local para poder llamar

az:

el proyecto005:

```
c:\angulardevya> ng new proyecto005
```

2. Primero descendemos a la carpeta proyecto005 y nuevamente desde la línea de comandos procedemos a crear la componente 'selectornumerico' escribiendo:

```
c:\angulardevya\proyecto005> ng generate component selectornumerico
```

Recordemos que al ejecutar este comando se crean 4 archivos.

Además dentro de la carpeta 'app' se crea una carpeta llamada 'selectornumerico' y dentro de ella se localizan los cuatro archivos creados.

3. En nuestro tercer paso vamos a implementar la vista de la componente 'selectornumerico' y su modelo. Abrimos el archivo 'selectornumerico.component.ts' y codificamos:

```

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

@Component({
  selector: 'app-selectornumerico',
  imports: [],
  templateUrl: './selectornumerico.component.html',
  styleUrls: ['./selectornumerico.component.css']
})
export class SelectornumericoComponent {
  @Input() minimo: number = 1;
  @Input() maximo: number = 1;
  actual: number = 1;

  ngOnInit() {
    this.actual = this.minimo;
  }

  incrementar() {
    if (this.actual < this.maximo)
      this.actual++;
  }

  decrementar() {
    if (this.actual > this.minimo)
      this.actual--;
  }

  fijar(v: number) {
    if (v >= this.minimo && v <= this.maximo)
      this.actual = v;
  }
}

```

En la clase SelectornumericoComponent podemos identificar el método que llamaremos desde el HTML template de la componente AppComponent:

```

fijar(v: number) {
  if (v >= this.minimo && v <= this.maximo)
    this.actual = v;
}

```

4. Codificamos ahora el archivo 'selectornumerico.component.html':

```

<div class="selector">
  <button (click)="decrementar()">-</button>
  <div class="valor">{{actual}}</div>
  <button (click)="incrementar()">+</button>
</div>

```

5. Para definir la hoja de estilo del 'selectornumerico' abrimos el archivo 'selectornumerico.component.css' y codificamos:

```
.selector {
  display:inline-flex;
  margin:0.2rem;
}
.valor {
  display:inline-flex;
  justify-content: center;
  align-items: center;
  width: 3rem;
  height: 3rem;
  background:#ff0;
  font-size:2rem;
}

button {
  height: 3rem;
}
```

6. Abrimos ahora el archivo 'app.component.html' y remplazamos su contenido con la definición de un selectornumerico que define una variable local y tres botones que llaman a partir de esa variable al método definido dentro del selectornumerico:

```
<div>
  <app-selectornumerico [minimo]="1" [maximo]="10" #selector1></app-selectornumerico>
  <br>
  <button (click)="selector1.fijar(1)">Fijar en 1</button><br>
  <button (click)="selector1.fijar(5)">Fijar en 5</button><br>
  <button (click)="selector1.fijar(10)">Fijar en 10</button>
</div>

<router-outlet />
```

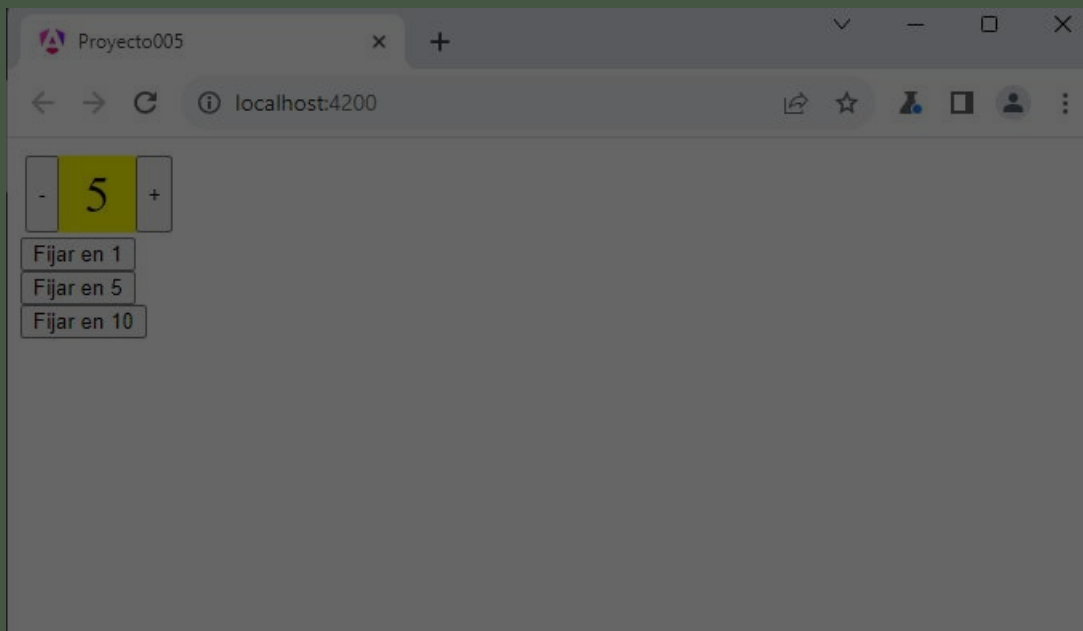
7. En la clase AppComponent nos queda importar la clase 'SelectornumericoComponent' que se encuentra en el archivo 'selectornumerico.component':

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { SelectornumericoComponent } from '../selectornumerico/selectornumerico.component';

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

Si ejecutamos ahora el proyecto:

```
ng server -o
```



Es decir que mediante dos botones definidos en la componente AppComponent podemos llamar al método fijar de la clase SelectornumericoComponent gracias a que definimos la variable #selector1:

```
<app-selectornumerico [minimo]="1" [maximo]="10" #selector1></app-selectornumerico>
<br>
<button (click)="selector1.fijar(1)">Fijar en 1</button><br>
<button (click)="selector1.fijar(5)">Fijar en 5</button><br>
<button (click)="selector1.fijar(10)">Fijar en 10</button>
```

Lo que debe quedar claro que debemos definir una variable en la etiqueta que define la componente y a partir de esta podemos llamar a métodos o inclusive acceder a propiedades de la componente.

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

## Retornar

