



Comunicación padre e hijo 11/20



RECURSOS

MARCADORES

Establece una **comunicación entre un componente padre con uno hijo** para saber qué productos quiere agregar el usuario a su carrito de compras y poder visualizar el total a pagar.

Comunicación componente hijo a padre

Paso 1: Agrégale al componente hijo un `@Output()` para emitir un mensaje al componente padre cuando el usuario agrega un producto al carrito.

El método `addToShoppingCart()` emitirá el evento cuando el usuario haga clic en el botón.

```
// components/producto/producto.component.ts
@Component({
  selector: 'app-producto',
  templateUrl: './producto.component.html',
  styleUrls: ['./producto.component.scss']
})
export class ProductoComponent {

  @Input() producto!: Producto;
  @Output() addProduct = new EventEmitter<Producto>();
```

```

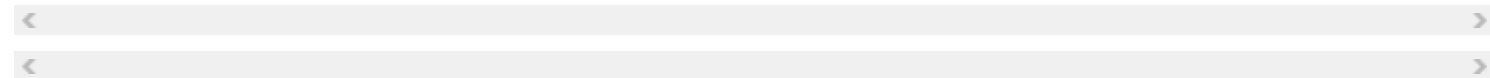
addToShoppingCart(p: Producto): void {
  this.addProduct.emit(p);
}
}

```

```

<!-- components/producto/producto.component.html -->
<div>
  <h2>{{ producto.name }}</h2>
  <img [src]="producto.image" width="200px" height="200px">
  <p>Precio: {{ producto.precio }}</p>
  <button (click)="addToShoppingCart(producto)">Agregar al carrito</button>
</div>

```



Paso 2: En el componente padre crea un método para recibir el mensaje, agregarlo al carrito de compras y calcular el monto total a pagar.

```

// catalogo/catalogo/catalogo.component.ts
@Component({
  selector: 'app-catalogo',
  templateUrl: './catalogo.component.html',
  styleUrls: ['./catalogo.component.scss']
})
export class CatalogoComponent {

```

```

public shoppingCart: Producto[] = [];
public total = 0;

public productos: Producto[] = [
  /* ... */
];

addProductToCart(p: Producto): void {
  this.shoppingCart.push(p);
  this.total += p.precio;
}
}

```

Paso 3: El método `addProductToCart()` recibe el producto, lo guarda en el array `shoppingCart` y calcula el total a pagar por el usuario. Puedes imprimir estos valores fácilmente con una interpolación.

```

<!-- catalogo/catalogo/catalogo.component.ts -->
<div *ngIf="productos" class="catalogo">
  <app-producto *ngFor="let p of productos"
    [producto]="p"
    (addProduct)="addProductToCart($event)"
  ></app-producto>
</div>
<div>
  <h3>Cantidad: {{ shoppingCart.length }}</h3>
  <h3>Total a pagar: {{ total }}</h3>
</div>

```

La comunicación de componente es esencial en cualquier aplicación front-end. Es importante saber manipular el envío de información ya sea de forma descendente o ascendente en la jerarquía de componentes.

Contribución creada con los aportes de Kevin Fiorentino.

Lecturas recomendadas



GitHub - platzi/angular-componentes at d8cfe00c7a118a5cc7097ad3f9a668b748ce9d3d
<https://github.com/platzi/angular-componentes/tree/d8cfe00c7a118a5cc7097ad3f9a668b748ce9d3d>



Curso de Manipulación de Arrays en JavaScript - Platzi
<https://platzi.com/cursos/arrays/>



Clases relacionadas



Comunicación de hijos hacia padres



Curso Profesional de Vue.js 2



Comunicación entre componentes padres e hijos



Curso Profesional de Vue.js 2