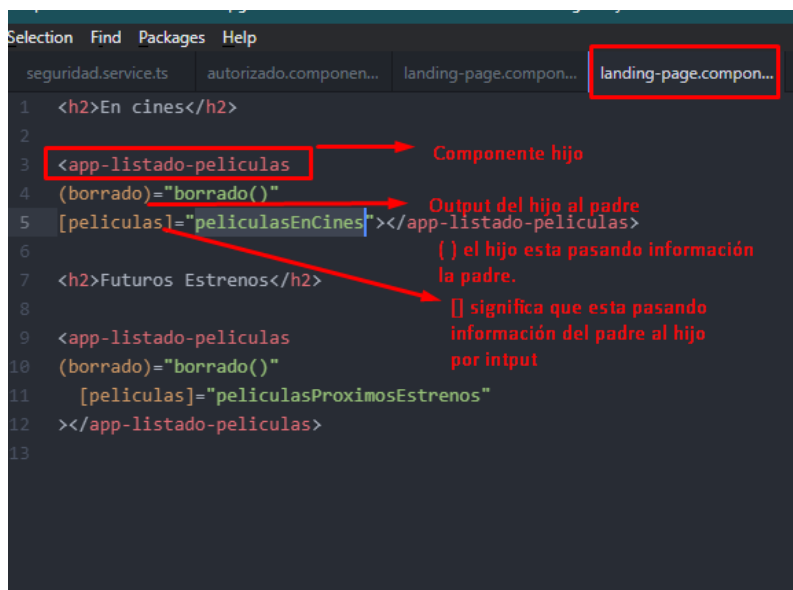


Componentes padre e hijo comunicación por medio de input y output.

El componente padre es Landing-pages y el componente hijo es listado-peliculas. Se observa como el componente padre pasa información al componente hijo por input en [peliculas] el cual lo carga de información y como el hijo pasa información al padre por (borrador)



```
Selection Find Packages Help
seguridad.service.ts autorizado.componen... landing-page.compon... landing-page.compon...
1 <h2>En cines</h2>
2
3 <app-listado-peliculas
4   (borrado)="borrado()"
5   [peliculas]="peliculasEnCines"></app-listado-peliculas>
6
7 <h2>Futuros Estrenos</h2>
8
9 <app-listado-peliculas
10   (borrado)="borrado()"
11   [peliculas]="peliculasProximosEstrenos"
12 ></app-listado-peliculas>
13
```

Componente hijo

Output del hijo al padre
() el hijo esta pasando información la padre.

[] significa que esta pasando información del padre al hijo por input

En el componente padre se define peliculaEnCines que trae los datos de cines desde la base de datos a través del servicio peliculaSevices.obtenerLanginPage y este carga la variable peliculaEnCines con cines, que se pasa al componente hijo

```

Selection Find Packages Help
seguridad.service.ts autorizado.component... landing-page.compon... landing-page.compon... listado-peliculas
9  })
10 export class LandingPageComponent implements OnInit {
11
12     constructor(private peliculasService: PeliculasService){}
13
14     ngOnInit(): void {
15         this.cargarDatos();
16     }
17     peliculasEnCines: PeliculaDTO[];
18     peliculasProximosEstrenos: PeliculaDTO[];
19
20     cargarDatos(){
21         this.peliculasService.obtenerLandingPage().subscribe(landingPage => {
22             this.peliculasEnCines = landingPage.enCines;
23             this.peliculasProximosEstrenos = landingPage.proximosEstrenos;
24         });
25     }
26
27     borrado(){
28         this.cargarDatos();
29     }

```

El componente hijo se definió los input y output para la comunicación entre padre e hijo.

```

styleUrls: ['./listado-peliculas.component.css']
})]
export class ListadoPeliculasComponent implements OnInit {

    constructor(private peliculasService: PeliculasService) { }

    @Input()
    peliculas: PeliculaDTO[];

    @Output()
    borrado: EventEmitter<void> = new EventEmitter<void>();

    ngOnInit(): void {

    }

    borrar(peliculaId: number): void {
        this.peliculasService.borrar(peliculaId)
        .subscribe(() => this.borrado.emit());
    }

}

```

Input se define la variables peliculas que recibe los datos del padre para que puedan mostrar en la vista del componente hijo listado peliculas

output el componente hijo quiere llamar un evento que esta en el padre a través un método que esta en el padre definido como borrado y lo llama con el emit desde el metodo que esta definido en el hijo borrar

En la vista del componente hijo lista-peliculas se observa como se muestra la información que es pasada por el padre y como el hijo llama al método borrar que a su vez llama al método del padre borrado.

```

<app-listado-generico [listado]="películas">
  <ng-container contenido>
    <div class="películas-contenedor">
      <div
        *ngFor="let pelicula of películas; let index = index"
        class="pelicula-contenedor"
      >
        <a [routerLink]="'/pelicula/' + pelicula.id">
          <img [src]="pelicula.poster" alt="Poster" />
        </a>
        <p>
          <a [routerLink]="'/pelicula/' + pelicula.id">{{pelicula.titulo}}</a>
        </p>
        <app-autorizado [rol]='admin'>
          <ng-container autorizado>
            <div>
              <a mat-flat-button color="primary" [routerLink]="'/peliculas/editar/' + pelicula.id">
                Editar
              </a>
              <button
                mat-flat-button
                color="warn"
                type="button"
                [swal]="{title: 'Confirmación',
                  text: '¿Está seguro que desea borrar el registro?',
                  showCancelButton: true}"
                (confirm)="borrar(pelicula.id)"
              >
                Borrar
              </button>
            </div>
          </ng-container>
        </app-autorizado>
      </div>
    </div>
  </ng-container>
</app-listado-generico>

```

películas es la lista que contiene todas las películas que se consulto en el componente padre y se las paso al hijo por input y este las muestra en la vista del componente hijo

Llama al metodo borrar que esta definido en el el componente hijo listado-peliculas, y este metodo a su vez llama al metodo borrado, que esta definido en el componente padre landig-page que lo que hace es consultar nuevamente las peliculaas para mostrar la información actualizada de películas que nuevamente se pasa al hijo.