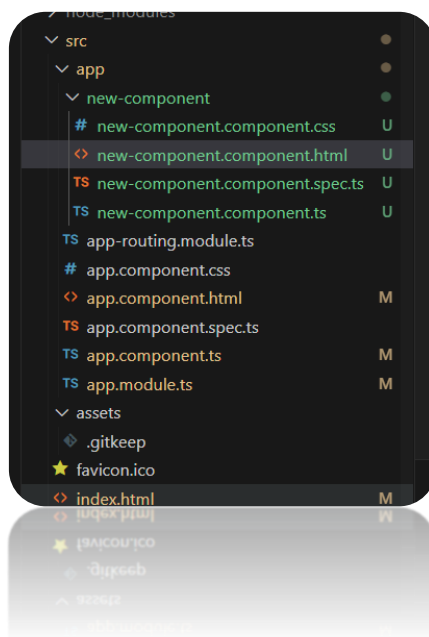


# Sharing data between Components.

- Creamos un nuevo componente en nuestro proyecto anterior con el siguiente comando desde la carpeta del proyecto: `ng generate component("Nombre del componente")`:



- **Primero vamos a enviar información al componente hijo.**
  - Para ello primero tendremos que configurar el componente hijo:
  - Primero exportamos input de angular en el componente del hijo y modificamos el export poniendo el atributo que se envía como parámetro:

```
src > app > new-component > TS new-component.component.ts > NewComponentComponent
1  import { Component, Input } from '@angular/core';
2
3
4  @Component({
5    selector: 'app-new-component',
6    templateUrl: './new-component.component.html',
7    styleUrls: ['./new-component.component.css']
8  })
9  export class NewComponentComponent {
10    @Input() item = '';
11  }
12
```

- Ahora vamos al html del nuevo componente y lo modificamos, ponemos el atributo entre corchetes para mostrar el atributo que le hemos pasado como parámetro:

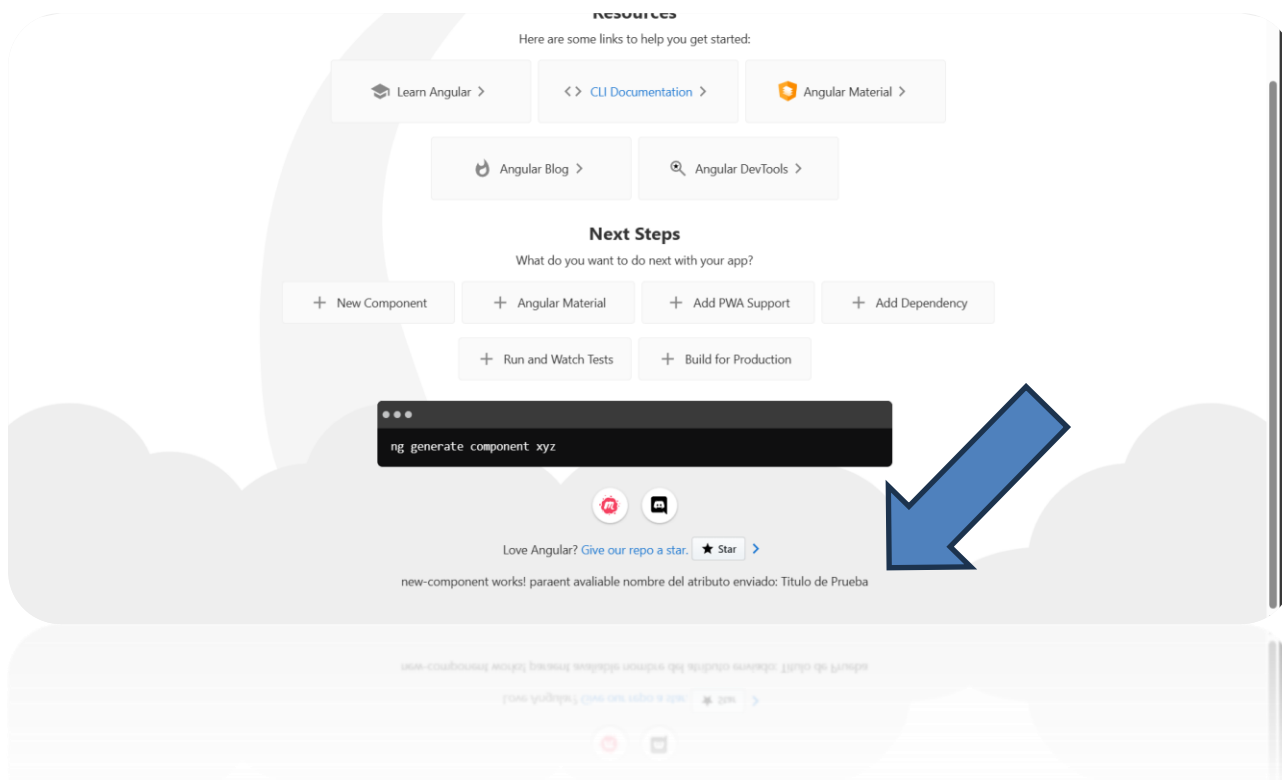
Go to component

```
1 <p>new-component works! paraent avaliable nombre del atributo enviado: {{item}}</p>
2
```

- Ahora vamos a configurar el html del padre para que aparezca el componente hijo:
- En esta imagen invocamos al componente hijo y le madamos el atributo que nosotros queremos que aparezca reflejado en el hijo (title)

```
472 </svg>
473
474 <!-- Aqui va el componente hijo-->
475 <app-new-component [item]="title"> </app-new-component>
476
477 </div>
478
```

- Si todo ha ido bien tendríamos que ver esto por pantalla, el componente hijo ha sido mostrado en la parte inferior de la pagina web:



- **Mandar información a un componente padre:**

- Importamos output y eventemitter de las librerías y configuramos a nuestro componente hijo.

```
src > app > new-component > TS new-component.component.ts > ...  
1  import { Component, Input, Output, EventEmitter } from '@angular/core';  
2  
3
```

- Configuramos nuestro componente:

```
11  
12  @Output() newItemEvent = new EventEmitter<string>();  
13  addNewItem(value: string) {  
14    this.newItemEvent.emit(value);  
15  }  
16  
17
```

- Ahora configuramos el template del componente hijo:

```
src > app > new-component > new-component.component.html > button  
Go to component  
1  <p>new-component works! paraent avaliable nombre del atributo enviado: {{item}}</p>  
2  <label for="item-input">Add an item:</label>  
3  <input type="text" id="item-input" #newItem>  
4  <button type="button" (click)="addNewItem(newItem.value)">Add to parent's list</button>  
5
```

- Configuramos el componente padre:

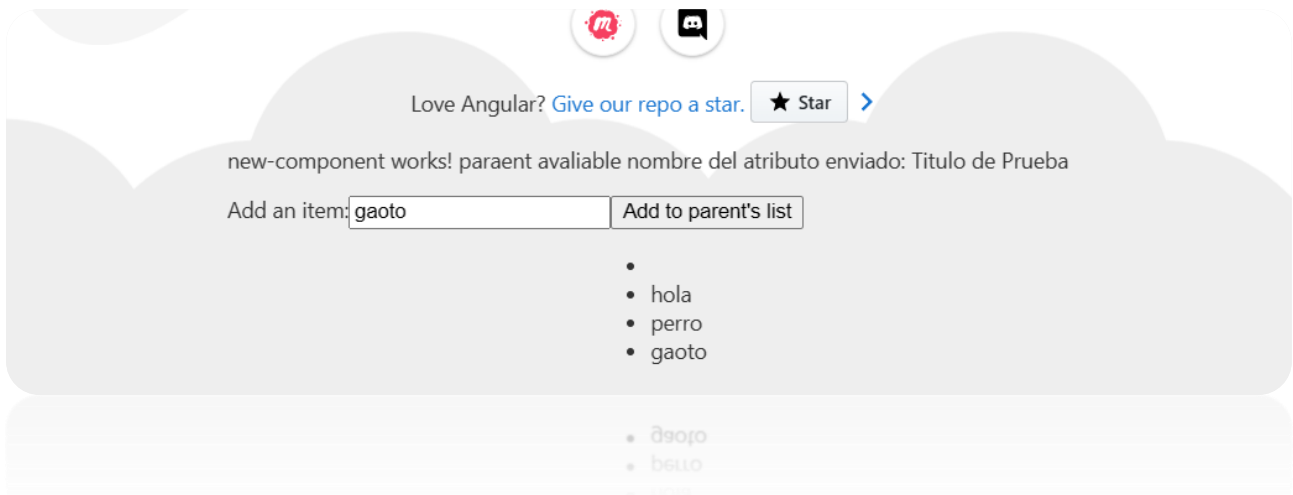
```
7    }  
8    export class AppComponent {  
9        title = 'Titulo de Prueba';  
10       items = ['item1', 'item2', 'item3', 'item4'];  
11  
12       addItem(newItem: string) {  
13           this.items.push(newItem);  
14       }  
15   }  
16
```

- Configuramos el template del componente padre:

```
<!-- Aqui va el componente hijo-->  
<app-new-component [item]="title" (newItemEvent)="addItem($event)" > </app-new-component>  
  
<!-- Aqui va el input para el componente padre-->  
  
<ul>  
  <li *ngFor="let item of items">{{item}}</li>  
</ul>  
  
</div>
```



- Si hemos hecho todo de manera correcta nos debería dar un resultado parecido a este:



## Bibliografía

- <https://angular.io/guide/setup-local>