1. En este proyecto lo que se quiere es listar varias personas
2. Inicia creando la clase persona, la cual tiene un nombre y un apellido. Esta clase persona por buena práctica, se denota **persona.model.ts**
3. En esta clase persona, colocamos dos atributos, como se había dicho, el atributo nombre y apellido. Se puede colocarlos de una como en POO, o no se colocan y en el constructor se colocan pubkic o private en los parámetros, esto hace que se creen los atributos de la clase.

constructor(public nombre:string, public apellido:string)

Con este constructor, no hay que bni siquiera asignar, pues la instrucción lo indica por debajo, es decir que cada parámetro se asigna automáticamente al atributo creado.

1. Ahora, en la clase app.component.ts, que es donde se especifica la clase principal, colocamos dentro de sus atributos un arreglo que es de tipo Persona, este arreglo es donde se van a guardar las personas y posteriormente a mostrarlas.

personas: Persona[]= [new Persona("Juan", "Perez"), new Persona("Laura", "Juarez")];

Como vemos, se inicia el arreglo con dos nombres por defecto.

1. También dentro de esta clase, app.component.ts, se colocan otros dos atributos, que son los que nos van a permitir pasar datos desde la plantilla al TypeScript, estos atributos son: **nombreInput, y apellidoInput.**
2. Posteriormente lo que se hace es crear el método que nos va a permitir agregar una persona, lo que se hace es que como desde la plantilla ya relacionamos el input con el atributo por medio de [(ngModule)], el método definido en la clase app.component.ts, llamado onAgregarPersona, lo que hace es crear una nueva persona con esos datos nombreInput y apellidoInput para insertar esa persona creada en el arreglo de personas.

onAgregarPersona()

  {

    let persona1 = new Persona(this.nombreInput, this.apellidoInput);

    this.personas.push(persona1);

  }

Aquí, let indica que se va a nombrar una variable, en este caso persona1, y se le va a asignar la creación de una nueva persona con los datos descritos anteriormente. Posteriormente, se indica que al arreglo personas, que pertenece a esa clase, se va a insertar, por medio de push, una nueva persona.

1. Ahora, en la plantilla html, lo que se hace son los formularios que van a tener los input relacionados con nombreInput y apellidoInput de la clase de TypeScript.
2. De igual manera en esta clase, se crea un botón el cual va a tener como evento el onAgregarPersona, que crea la nueva persona con nlos datos relacionados y la inserta en el arreglo.
3. Después de esto en el ultimo div, lo que se hace es recorrer el arreglo, donde let crea una variable del momento, donde se va a guardar una persona del arreglo persona, además en i se va a guardar el índice que se va recorriendo para de igual manera mostrarlo junto con nombre y apellido. Esto se hace también con la variable persona por medio de interpolacipon {{i+1}} y para acceder a persona de igual manera {{persona.nombre}}
4. Para la segunda parte, donde se agrega un componente persona, lo que se hace es que en el padre component, se coloca la etiqueta del selector hijo, persona. Dentro de este selector, lo que se hace es colocar el \*ngFor, para proceder a sacar en una variable cada persona y en otra el índice que se lleva hasta el momento.

<app-ersona

\*ngFor="let personaElemento of personas; let i =index"

     [persona]="personaElemento"

     [indice]="i"

  ></app-persona>

Después de esto, mediante [persona], lo que se hace es relacionar lo que sale del arreglo en el for, con un atributo de tipo Input que se establece en el hijo, en este caso persona.component.ts

export class PersonaComponent  {

@Input() persona: Persona;

@Input() indice: number;

constructor() { }

}

Entonces [persona] se relaciona con @Input() persona: Persona;

[Indice] se relaciona con @Input() índice : number;

El resto de las inserciones de personas, sigue estando en la clase padre.