**SERVICIOS Y PROVEEDORES**

* Un servicio, no es más que una clase,que brinda algunos métodos para que sean utilizados por cualquiercomponente relacionado a este. Lo que se hace es crear una nueva clase, la cual tiene definidos algunos métodos.
* El componente que quiera utilizar sus métodos, lo que debe hacer es Aplicar DependencyInjection, es decir crear una instancia del servicio en sus atributos y en su constructor recibir esta instancia de la clase del servicio y se asigna al atributo.

constructor(private logginService:LogginService) { }

* Además de esto, se debe agregar en la sección de componentes el proveedor o los proveedores del servicio así;

@Component({

  selector: 'app-formulario',

  templateUrl: './formulario.component.html',

  styleUrls: ['./formulario.component.css'],

  //Se debe agregar un proveedor del servicio, que es la misma clase del servcio

   providers: [LogginService]>

})

Cada componente que quiera utilizar el servicio lo debe de hacer, pero para evitar esto y definir el proveedor del servicio como por defecto, lo que se hace es agregarlo en app.module.ts, de igual manera se debe **importar** el servicio.

import { LogginService } from './LogginService.service';

@NgModule({

declarations: [

AppComponent,

PersonaComponent,

FormularioComponent

],

imports: [

BrowserModule,

FormsModule

],

providers: [LogginService],

bootstrap: [AppComponent]

})

**DATA SERVICES**

Es la misma logica de una servicio normal, lo que se diferencia es que se tiene para el ejemplo de listado personas, el arreglo ubicado en el servicio, para de la clase propia solo mandar la persona y que se agregue en el servicio, de esta manera también se puede consultar desde otro componente.

**SERVICIO DE UN SERVICIO**

Para hacer esto, se hace normalmente el constructor con el parámetro del servicio que se va a acceder, además de esto se debe:

Agregar el decorador @Injectable, que indica que se va a hacer la injección de dependencia del servicio, pero que es de servicio a servicio.

@Injectable()

export class PersonasService

{

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

    constructor(private loggingService:LogginService){}

    agregarPersona(persona:Persona)

    {

      this.personas.push(persona);

      this.loggingService.enviarMensajeAConsola("Agregando persona : "+persona.nombre);

    }

}

**COMUNICACIÓN ENTRE CLASES USANDO SERVICIOS**

**(ENVÍO DE DATOS ENTRE COMPONENTES- EMIT-SUBSCRIBE)**

Para realizar esta acción, se inicia colocando un botón editar por cada persona que se muestre en pantalla, esto colocándolo en personas.componente.html, ese botón info va a llamar a emitirSaludo, el cual está en el ts de este mismo componente. Dentro de este método lo que se hace es acceder al servicio de PersonasService, con el fin de emitir un valor, en este caso el índice que pertenece la persona. Para esto en PersonasService.service.ts se declara una valiable saludar de tipo EventEmitter, que nos va a permitir emitir ese valor entre componentes. Ahora esta variable es accedida desde personas.component.ts y se va a utilizar esta variable del servicio para emitir el valor.

emitirSaludo()

{

  //saludar está dentro del servicio, se emite el indice

    this.personasService.saludar.emit(this.indice);

Y EN LA CLASE SERVICIO PERSONAS

export class PersonasService

{

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

    saludar= new EventEmitter<number>();

    constructor(private loggingService:LogginService){}

    agregarPersona(persona:Persona)

    {

      this.personas.push(persona);

      this.loggingService.enviarMensajeAConsola("Agregando persona : "+persona.nombre);

    }

}

Como vimos en la imagen donde esta emitirSaludo, se llama a la variable del servicio y se emite el índice. Hasta el momento lo que hemos hecho es pasar el índice de la clase personasComponent a el servicio. Ahora trataremos de pasar del servicio a otro componente, en este caso a formulario, pero para no tener el riesgo de sacar la variable sin que se haya accedido, también se utiliza la variable EventEmitter, pues esta tiene otra función que lo que hace es subscribirse y apenas haya un cambio se dispara.

En el constructor de formulario.component.ts, después de inyectar los servicios, dentro del constructor se accede al servicio de personas y mas específicamente a la variable EventEmitter, y se hace una subscripción, para que apenas el otro componente emita un índice esta lo obtenga inmediatamente.

constructor(private logginService:LogginService, private personasService:PersonasService)

  {

    this.personasService.saludar.subscribe(

      (indice:number)=>alert("EL INDICE ES:  "+indice)

    )

  }

Aquí cuando se obtenga el valor, se hace una función flecha para mostrarlo simplemente en un alert.