Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Diseño de Sistemas Interactivos

Introducción a Angular

Introducción

La presente micro-práctica consiste en un ejercicio guiado que el alumno deberá resolver de manera individual.

Objetivos

Adquirir conocimientos básicos sobre Typescript utilizando el framework de desarrollo Angular.

Duración

La resolución de la micro-práctica comenzará durante la clase práctica y deberá ser actualizada en el repositorio de Github como máximo antes del inicio de la próxima clase práctica.

Preparación para la sesión

Para realizar la micro-práctica se recomienda que el alumno haya revisado la documentación de Typescript (https://www.typescriptlang.org/docs/home.html).

Escenario

Un local de venta de coches de segunda mano quiere automatizar el manejo de inventario de su local utilizando tecnologías Web. Concretamente se quiere desarrollar una página Web que permita llevar un control de los vehículos que tiene en su patio. Los datos básicos que se manejan del vehículo son:

- Foto
- Marca
- Modelo
- Año
- Antigüedad en el patio
- Estado
- Avalúo
- Precio de venta al público.

Requisitos establecidos por el cliente

Apariencia de la vista de la página Web:

Filtrar por: (marca de coche)

Foto	Marca	Modelo	Año	En venta desde	Precio	PVP	Acciones
	RENAULT	scenic	2007	04-2018	5,000€	6,050€	Rebajar Vendido
	SEAT	ibiza	2003	03-2018	1,200€	1,452€	Rebajar Vendido
	RENAULT	megane	2007	03-2018	3,500€	4,235€	Rebajar Vendido
(sin foto!)	TESLA	model 3	2007	03-2018	4,000€	4,840€	Rebajar Vendido

Requisitos funcionales:

1. La pantalla muestra una lista de coches y, para cada uno, los diferentes campos de información.
2. Si el usuario escribe algo en el campo FILTRO, se mostrarán solo los vehículos cuya marca empiece por los caracteres escritos por el usuario. Así, en el ejemplo si escribe “R” sólo se mostrarán los vehículos Renault.
3. Si pulsamos REBAJAR el precio del vehículo se multiplicará por 0,90.
4. Si pulsamos VENDIDO, el vehículo desaparecerá de la lista.
5. El PVP es el Precio x 1,21.
6. Si el estado del vehículo es BUENO, el precio se presentará en fondo VERDE. En caso contrario, en ROJO.

Tareas

1. Crear un nuevo proyecto de angular.
2. Diseñar la vista de la aplicación Web en base a los requisitos establecidos por el cliente.
3. Crear la clase Coche para definir el modelo de datos.
4. Añadir los datos y la lógica del programa en el componente de la aplicación.
5. Añadir los decoradores necesarios a la vista (documento HTML) de la aplicación.

Instrucciones de realización

Iniciar el ambiente de desarrollo

1. En tu ordenador, crea una nueva carpeta llamada microp_2.
2. Abre la nueva carpeta creada utilizando **Visual Studio Code.**
3. crea un nuevo proyecto de angular con el nombre microp02.

```
$ng new microp02
```

4. Comprueba el correcto funcionamiento del proyecto creado.

```
$ng serve --open
```

Crear la estructura de la página Web

1. Crea algunos estilos básicos para la página, para ello modifica el archivo src/app/app.component.css. Coloca el siguiente código.

```
.classform1 .txtInline{text-align:right;width:350px;display:inline-block}
.classform1 input{margin-left:10px;width:200px}

.class_concesionario{background-color: coral;display:block;width:700px;}
.class_concesionario .titulo{text-align:center}

.estadoBuenoClass{background-color:green}
.estadoMaloClass{background-color:red}
```

2. Crea la estructura básica de la página web, para ello modifica el archivo src/app/app.component.html. Coloca el siguiente código.

```
<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
 <h1>
{{ title }}!
 </h1>

</div>



<div class='class_concesionario'>

 <form class='classform1' >
  <br/>
 <br/>
 <div class='txtInline'>Filtrar por:</div><input />
 <br/><br/>

 <table border='1px' width="700px">
 <thead>
 <td  align='center'>Foto</td><td>Marca</td><td>Modelo</td><td>Año</td><td>En
```

```
venta desde</td><td align="right">Precio</td><td align="right">PVP</td><td
align="center">Acciones</td>
 </thead>
  <tr >
     <td align='center'>
     <img />
     <p ></p>
   </td>
   <td></td>
   <td align="left"></td>
   <td align="center"></td>
   <td align="center"></td>
   <td align="right">
     <p >

     </p>
   </td>
   <td align="right"></td>
   <td align="center">
     <button type='button' >Rebajar</button>
     <button type='button' >Vendido</button>
   </td>
 </tr>
  </table>

 </form>
 </div>

<router-outlet></router-outlet>
```

3. guarda los cambios y comprueba en el navegador

Class coche

1. Crear el modelo de datos del coche. Para ello crea una nueva clase llamada coche, utiliza el siguiente comando de angular.

$ng g class coche

2. Define los campos del coche y sus constructores:
    - Marca, modelo y foto (puede ser =””) : String.
    - Fecha_modelo y fecha_venta : date.
    - precio : number
    - estado: EstadoCoche (Enum {Bueno, Malo}

```
export enum EstadoCoche {BUENO, MALO}

export class Coche {

   public marca:string;
    public modelo:string;
    public fecha_modelo:Date;
    public fecha_venta:Date;
    public precio:number;
    public foto:string;
    public estado:EstadoCoche;


constructor(marca:string,modelo:string,fecha_modelo:Date,fecha_venta:Date,precio:
number,foto:string,estado:EstadoCoche)
    {
        this.marca=marca;
        this.modelo=modelo;
        this.fecha_modelo=fecha_modelo;
        this.fecha_venta=fecha_venta;
        this.precio=precio;
```

```
        this.foto=foto;
        this.estado=estado;
    }

}
```

3. Definir 2 métodos para la clase coches: rebajar () y getPVP (). Estos métodos deberán realizar lo especificado en los requisitos funcionales. Estos métodos deberán ser colocados debajo del constructor de la clase.

```
    public getPVP():number{
        return this.precio*1.21;
    }

    public rebajar(){
        this.precio*=0.90;
    }
```

Modificar app.module para utilizar formularios

1. Para utilizar módulo de formularios importar FormsModule. El archivo src/app/app.modules.ts deberá quedar de la siguiente forma:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
 declarations: [
   AppComponent
 ],
 imports: [
```

```
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Añadir la lógica del programa en el archivo app.component.ts

1. Definir la variable *EstadoCoche* permitir que tome cualquier tipo de variable (any).
2. Definir un Array de coches con 5 valores de ejemplo.
3. Crear el método para filtrar los coches de acuerdo a los caracteres ingresados.
4. Crear el método que permita vender un coche *cmd_vender(id)*.
5. Crear el método que permita rebajar el precio de un coche *cmd_rebajar(id).*
6. El archivo *app.component.ts* debería quedar de la siguiente manera.

```
import { Component,Input } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { Coche,EstadoCoche } from './coche';


@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})


export class AppComponent {
  title = 'Micropráctica ANGULAR - 02';
   public EstadoCoche:any = EstadoCoche;
   public coches:Array<Coche>=[  new Coche('renault','scenic',new
Date(2007,10,1),new Date(2018,3,1),5000,'coche01.jpg',EstadoCoche.BUENO),
              new Coche('seat','ibiza',new Date(2003,4,12),new
Date(2018,2,1),1200,'coche02.jpg',EstadoCoche.BUENO),
```

```
              new Coche('renault','megane',new Date(2006,12,23),new
Date(2018,2,1),3500,'coche03.jpg',EstadoCoche.MALO),
              new Coche('tesla','model 3',new Date(2006,12,23),new
Date(2018,2,1),4000,'',EstadoCoche.BUENO)];
  public filtro:string='';

 public getFiltrados():Array<number>{
    var filtrados:Array<number>=[];
    var indice:number=0;
    for(var coche of this.coches)
    {
        if (coche.marca.lastIndexOf(this.filtro, 0) === 0)
filtrados.push(indice);
        indice=indice+1;
    }
  return filtrados;
 }
 public cmd_vender(id)
 {
    this.coches.splice(id,1);
 }
  public cmd_rebajar(id)
 {
    this.coches[id].rebajar();
 }

}
```

Añadir directivas al documento HTML

Ahora tenemos todo lo necesario para decorar el código HTML que permitirá enlazar la vista y la lógica de la aplicación.

1. Añadir el template *#f=ngForm* para utilizar formularios. Añadelo en la etiqueta <form> del documento html.
2. Para actualizar la lista de coches de forma dinámica vamos a utilizar two way data binding. Para ello se debe asociar la directiva [(ngModel)] = ‘filtro’.

3. Utilizar la directiva **ngFor* e interporlación (*{{}}*) para cargar los coches y su precio en la lista. Poblar los diferentes espacio de la tabla con los datos almacenados en el componente.
4. Utilizar la directive *ngIf para mostrar el mensaje (sin foto!) si es que el campo foto del coche está vacío
5. Utilizar la directiva *[ngClass]* para subrayar con rojo o verde el estado del coche tomando en cuenta el campo *EstadoCoche*.
6. Asociar el evento click del botón “Rebajar” con el método *cmd_rebajar(id)* para ello utiliza la directiva (click).
7. Asociar el evento click del botón “Vender” con el método *cmd_vender(id)* para ello utiliza la directiva (click).

Consideraciones adicionales

- Marca se presentará siempre en MAYÚSCULAS.
- Modelo se presentará siempre en MINÚSCULAS.
- Año (solo se presenterá el AÑO de la fecha de compra).
- En venta desde (presentará MES y AÑO de la fecha en que se pone en venta).
- Precio y PVP se presentarán sin decimales, y con “,” de millares. Seguido de €.

Sugerencia: Trabajar con pipes. EG. *coches[id].modelo* | *lowercase*

El documento HTML decorado con las directivas deberá tener la siguiente apariencia.

```
<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
 <h1>
{{ title }}!
 </h1>

</div>



<div class='class_concesionario'>

 <form class='classform1' #f='ngForm'>
  <br/>
 <br/>
 <div class='txtInline'>Filtrar por:</div><input type='search'
placeholder="(marca de coche)" name='filtro2' [(ngModel)]='filtro'/>
 <br/><br/>

 <table border='1px' width="700px">
```

```
 <thead>
 <td  align='center'>Foto</td><td>Marca</td><td>Modelo</td><td>Año</td><td>En
venta desde</td><td align="right">Precio</td><td align="right">PVP</td><td
align="center">Acciones</td>
 </thead>
  <tr *ngFor='let id of getFiltrados(); let i=index'>
     <td align='center'>
     <img src="assets/{{coches[id].foto}}" width='64px' height='64px'
*ngIf='[coches[id].foto]!=""'/>
     <p *ngIf='[coches[id].foto]==""'>(sin foto!)</p>
   </td>
   <td>{{coches[id].marca | uppercase}}</td>
   <td align="left">{{coches[id].modelo | lowercase}}</td>
   <td align="center">{{coches[id].fecha_modelo | date:'yyyy'}}</td>
   <td align="center">{{coches[id].fecha_venta | date:'MM-yyyy'}}</td>
   <td align="right">
     <p [ngClass]="{estadoBuenoClass: [coches[id].estado]==EstadoCoche.BUENO,
estadoMaloClass: [coches[id].estado]!=EstadoCoche.BUENO}">
       {{coches[id].precio | number:'1.0-0'}}€
     </p>
   </td>
   <td align="right">{{coches[id].getPVP() | number:'1.0-0'}}€</td>
   <td align="center">
     <button type='button' (click)='cmd_rebajar(id)'>Rebajar</button>
    <button type='button' (click)='cmd_vender(id)'>Vendido</button>
   </td>
 </tr>
  </table>

 </form>
 </div>
```

Cargar imágenes de los coches en el proyecto

1. Descarga imagenes de internet de acuerdo a los modelos de coche que se inicializaron de ejemplo en el array *coches*.
2. coloca las imágenes en el directorio del proyecto src/assets.

Comprobar la solución y subir el repositorio

1. Guarda los cambios y ejecuta el servidor con el comando ng serve en una terminal. La solución debería quedar como la captura de pantalla que se muestra en la siguiente imagen.

2. Sube la solución a tu cuenta de github, además comprime tu solución y subela al entregador de aula global.

Solución

Nota: Se recomienda ir a la solución al final de la resolución de esta micro-práctica solo para comparar tu solución con la propuesta por los profesores y si no te queda claro algún apartado de la misma.

https://github.com/andresSantos9/microp_02.git