

# 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 el funcionamiento de angular que servirá como base para la utilización del framework Ionic 3.

## 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 angular (<https://angular.io/guide/quickstart>).

## Escenario

El nuevo restaurante “Mesón Gallego” quiere una forma moderna y eficiente de manejar la facturación en su local. Para ello el propietario desea una página Web que le permita a sus empleados ingresar consumiciones y el precio de dicha consumición. Para ello ha decidido contratar tus servicios para que elabores de forma rápida dicha página Web en el menor tiempo posible. Dada la premura de tiempo tú has decidido utilizar angular para tener una página responsiva y dinámica.

## Tareas

1. Crear un nuevo proyecto de angular.
2. Diseñar una página web sencilla que conste de dos bloques: en el primer bloque la página tendrá un selector con el nombre de la consumición y el precio, el segundo bloque permitirá visualizar la factura de forma dinámica.
3. Añadir los módulos necesarios para poder trabajar con formularios.
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\_1.
2. Abre la nueva carpeta creada utilizando **Visual Studio Code**.
3. crea un nuevo proyecto de angular con el nombre microp01.

```
$ng new microp1
```

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 label{width:100px;display:inline-block}
.classform1 input{width:200px}
.classform1 select{width:300px}
.class_comanda{background-color: coral;display:block;width:450px;}
.class_comanda .titulo{text-align:center}
```

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_comanda'>
<h2 class="titulo">Mesón Gallego</h2>
<form class="clasform1">
  <br/>
<label>Consumicion:</label>
<select name='lstConsumicion'>
</select>
<button type="button">+</button>
```

```
</form>
<br/>
<h3>Consumiciones:</h3>
<table width="450px">
  <thead>
    <td>
      <u>Producto</u>
    </td>
    <td align="right">
      <u>Precio</u>
    </td>
  </thead>
  <tr>
    <td></td>
    <td align="right"></td>
    <td align="center">
      <button type="button">-</button>
    </td>
  </tr>
  <tr>
    <td>
      <b>TOTAL</b>
    </td>
    <td align="right">
      <b></b>
    </td>
  </tr>
</table>

</div>
```

3. guarda los cambios y comprueba en el navegador

## App Module

1. Recuerda lo visto en clase, toda aplicación de angular tiene un root module que provee un mecanismo bootstrap para ejecutar la aplicación.
2. Mira la estructura de este archivo en la dirección `src/app/app.module.ts`. Identifica los elementos de este archivo (Declaration, Imports, providers).
3. En la aplicación vamos a utilizar el módulo *FormsModule* que permitirá trabajar con formularios. Añádelo en el bloque *imports* del App Module. El archivo debería quedar como se muestra a continuación.

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

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule, FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

## App Component

1. Recuerda lo visto en clase, toda aplicación de angular tiene al menos un componente que contiene la lógica y los datos de la aplicación y está asociado a un documento HTML que contiene una vista de la aplicación.
2. Abre el archivo App Component (se encuentra en la dirección: `src/app/app.component.ts`).

- Definir las variables a utilizar para la lógica de la aplicación. En este caso definiremos: Lista de productos(`productos[]`), Importe de la factura (`importe`), productos añadidos (`lineas[]`) y una variable auxiliar para identificar el código de producto (`codProducto`). Coloca el siguiente código en la clase `AppComponent` (`src/app/app.component.ts`).

```
export class AppComponent {  
  title = 'Micro-práctica 1';  
  
  private productos:Array<[string,number]>=[['Arroz tres delicias',3.50],['Cerdo  
agridulce',5.50],['Ternera con bambu y setas',7.00]];  
  
  public codProducto=-1;  
  public importe:number=0.0;  
  public lineas:Array<[string,number]>=[];  
  
}
```

- Añadir el método `addLinea`: este método permitirá añadir nuevas consumiciones a la factura. Coloca el siguiente código debajo de la variable `importe` definida en el numeral anterior.

```
public addLinea()  
{  
  var producto=this.productos[this.codProducto][0];  
  var precio:number=this.productos[this.codProducto][1];  
  this.lineas.push([producto,precio]);  
  this.importe=this.importe+precio;  
}
```

- Añadir el método `delLinea`: este método permitirá eliminar consumiciones añadidas a la factura. Coloca el siguiente código debajo de el método `addLinea` añadido en el numeral anterior.

```
public delLinea(i)
{
    this.importe-=this.lineas[i][1];
    this.lineas.splice(i,1);
}
```

## 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 el selector de producto de forma dinámica vamos a utilizar two way data binding. Para ello se debe asociar la directiva `[(ngModel)] = 'codProducto'`.
3. Utilizar la directiva `*ngFor` para cargar los productos y su precio en el selector.
4. Asociar el evento click del botón + con el método `addLinea` para ello utiliza la directiva (click).
5. Para poblar la lista de consumiciones, vamos a utilizar la directiva `*nform`.
6. Utilizar interpolación para poblar la tabla `<tr>` con el array `Líneas`.
7. Asociar el evento click del botón - con el método `delLinea` para ello utiliza la directiva (click).
8. Utilizar interpolación para poblar el importe total de la factura.

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_comanda'>
<h2 class="titulo">COMANDA</h2>
<form class='classform1' #f='ngForm'>
<br/>
<label>Consumicion:</label>
<select name='lstColor' [(ngModel)]='codProducto'>
  <option *ngFor='let prod of productos; let i=index' value='{{i}}'>{{prod[0]}} (
  {{prod[1]}} €)</option>
</select>
<button type='button' (click)='addLinea()'>+</button>
</form>
<br/>
<h3>Consumiciones:</h3>
<table width="450px">
<thead><td><u>Producto</u></td><td align="right"><u>Precio</u></td></thead>
<tr *ngFor='let linea of lineas; let i=index'>
  <td>{{linea[0]}}</td>
  <td align="right">{{linea[1]}}€</td>
  <td align="center"><button type='button' (click)='delLinea(i)'>-</button></td>
</tr>
<tr><td><b>TOTAL</b></td><td align="right"><b>{{importe}}€</b></td></tr>
</table>
</div>
```

### 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.



## Mesón Gallego

Consumición: Cerdo agri dulce (5.5 €) +

**Consumiciones:**

<u>Producto</u>	<u>Precio</u>	
Ternera con bambu y setas	7€	-
Cerdo agri dulce	5.5€	-
<b>TOTAL</b>	<b>12.5</b>	

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\\_01](https://github.com/andresSantos9/microp_01)