

NOMBRE Y APELLIDOS:			FECHA: 06-10-2023		
DOCENTE: MANUEL MACÍAS PÉREZ			NOTA:		
(IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB.			Nº CURSO: 22-35/008902		
MF:	0492	UNIDADES DE APRENDIZAJE A LAS QUE RESPONDE:	UA1	Duración:	3 h
UF:	1846				
PRÁCTICA Nº:	E2				
DENOMINACIÓN: framework					
<p><u>DESCRIPCIÓN</u></p> <p>1.- El alumno de forma individual deberá realizar:</p> <ul style="list-style-type: none"> o Un componente utilizando el framework angular y visual studio code como IDE para la que un usuario registrado pueda adjuntar un archivo-imagen para completar su perfil. o Hay que tener en cuenta la estructura creada en la prueba evaluable E1. o Tiene que contener un formulario para la inserción de datos. <p>Adjuntar los códigos creados a este archivo y convertir este documento en pdf. Enviar o Subir a Github. La práctica se realizará de manera individual.</p> <p><u>MEDIOS PARA SU REALIZACIÓN</u></p> <ul style="list-style-type: none"> - Equipo informático. - Aplicación Visual Code Studio instalada en el equipo. - Navegadores actualizados <p><u>PAUTAS DE ACTUACIÓN DEL FORMADOR</u></p> <p><i>Al inicio de la práctica, que se desarrollará de manera individual por cada uno de los alumnos, el formador/a realizará las siguientes actuaciones:</i></p> <ul style="list-style-type: none"> - Fijará los objetivos de la práctica. - Aportará las instrucciones necesarias a los alumnos/as para la realización de la misma, haciendo hincapié en aquellos aspectos más relevantes. - Facilitará a cada alumno/a la documentación necesaria para el desarrollo de la práctica. - Resolverá las dudas que se planteen durante el transcurso de la práctica, con objeto de que el alumnado aprenda y pueda concluir la realización de la misma. <p>Durante la realización de la práctica el formador/a supervisará el desarrollo de esta para evaluar tanto los procedimientos como el resultado final.</p> <p>Al finalizar la práctica el formador examinará el desarrollo que han realizado los/as alumnos/as, proponiendo las medidas de corrección, en caso necesario.</p> <p style="text-align: center;">ESPECIFICACIONES PARA LA EVALUACIÓN DE LA PRÁCTICA</p>					
Resultados a comprobar			Indicadores de logro		

<p>1. Especificar las características de los protocolos estándares del mercado para poder utilizar servicios web en la aplicación a desarrollar</p> <p>Conforme el criterio de evaluación CE 1.2</p>	1.1 Especifica las características de los protocolos estándares del mercado
	1.2 Utiliza servicios web en la aplicación a desarrollar.
<p>2. Seleccionar y emplear los servicios web más adecuados para ser utilizados en la aplicación web en función del diseño especificado.</p> <p>Conforme el criterio de evaluación CE 1.3</p>	2.1 Selecciona los servicios web más adecuados para ser utilizados en la aplicación web
	2.2 Emplea los servicios web más adecuados para ser utilizados en la aplicación web
	2.3 Documenta los servicios web más adecuados para ser utilizados en la aplicación web

Sistema de valoración

Definición de indicadores y escalas de medida

Los indicadores que se van a establecer, será una hoja de chequeo, sistema de valoración, que complementa a este documento, donde se evalúan todos los resultados a comprobar (tareas). En este documento, se establecerán a su vez los indicadores de logro que se han de tener en cuenta, para conseguir los resultados a comprobar.

Mínimo exigible

El mínimo exigible para la superación de la práctica es de 50 puntos sobre 100 puntos

SUPUESTO PRÁCTICO

1.- El alumno de forma individual deberá realizar:

- Un componente utilizando el framework angular y visual studio code como IDE para la que un usuario registrado pueda adjuntar un archivo-imagen para completar su perfil.
- Hay que tener en cuenta la estructura creada en la prueba evaluable E2.
- Tiene que contener un formulario para la inserción de datos.

Adjuntar los códigos creados a este archivo y convertir este documento en pdf. Enviar o Subir a Github.
La práctica se realizará de manera individual.

Ejemplo:

```
<div class="input-group mb-3">  
  <input type="file" class="form-control" id="inputGroupFile02">  
  <label class="input-group-text" for="inputGroupFile02">Upload</label>  
</div>
```

```
// Módulos y Librerías  
import { Component, OnInit } from '@angular/core';  
import { FormBuilder, FormGroup, Validators } from '@angular/forms';  
import { ActivatedRoute, Router } from '@angular/router';  
import { ToastrService } from 'ngx-toastr';  
  
// Servicio - Modelo  
import { GregistroService } from 'src/app/config/Servicios/gregistro';  
  
@Component({  
  selector: 'app-perfil',  
  templateUrl: './perfil.component.html',  
  styleUrls: ['./perfil.component.css']  
})  
export class PerfilComponent implements OnInit {  
  registroForm: FormGroup;  
  verRegistro: any  
  titulo = 'Editar registro';  
  id: string | null;  
  constructor(private fb: FormBuilder,  
              private router: Router,  
              private toastr: ToastrService,
```

```
private _registroService: RegistroService,  
private aRouter: ActivatedRoute) {  
  this.registroForm = this.fb.group({  
    nombre: ['', Validators.required],  
    apellidos: ['', Validators.required],  
    Fnacimiento: ['', Validators.required],  
    email: ['', Validators.required],  
    telefono: ['', Validators.required],  
    foto: ['', Validators.required],  
    pass: [''],  
    tipo: [''],  
  })  
  this.id = this.aRouter.snapshot.paramMap.get('id');  
}
```

```
ngOnInit(): void {  
  this.esEditar();  
}
```

```
esEditar() {  
  if(this.id !== null) {  
    this.titulo = 'Editar registro';  
    this._registroService.obtenerRegistro(this.id).subscribe(data => {
```

```
esEditar() {  
  if(this.id !== null) {  
    this.titulo = 'Editar registro';  
    this._registroService.obtenerRegistro(this.id).subscribe(data => {  
      this.registroForm.setValue({  
        nombre: data.nombre,  
        apellidos: data.apellidos,  
        Fnacimiento: data.Fnacimiento,  
        email: data.email,  
        telefono: data.telefono,  
        foto: data.foto,  
        pass: data.pass,  
        tipo: data.tipo,  
      })  
    })  
  }  
}
```

```
<div class= "container mt-15">
  <div class="row">
    <div class="col-lg-12">
      <div class="card">
        <div class="card-body text-center">
          <span class="titulo">{{ titulo | uppercase }}</span>
          <form class="mb-3" [formGroup]="registroForm" (ngSubmit)="esEditar()">

            <!-- Nombre -->

            <div class="mb-3">
              <input type="text" formControlName="nombre" class="form-control"
                placeholder="Nombre">
              <div class="text-danger" *ngIf="registroForm.get('nombre')?.hasError('required') &&
                registroForm.get('nombre')?.touched">
                <span>El nombre es obligatorio</span>
              </div>/.text-danger
            </div>/.mb-3

            <!-- apellidos -->

            <div class="mb-3">
              <input type="text" formControlName="apellidos" class="form-control"
                placeholder="apellidos">
            </div>/.text-danger
          </div>/.mb-3

          <!-- Fnacimiento -->

          <div class="mb-3">
            <input type="text" formControlName="Fnacimiento" class="form-control"
              placeholder="Fecha de nacimiento">
            <div class="text-danger" *ngIf="registroForm.get('Fnacimiento')?.hasError(
              'required') && registroForm.get('Fnacimiento')?.touched">
              <span>La fecha es obligatoria</span>
            </div>/.text-danger
          </div>/.mb-3

          <!-- email -->

          <div class="mb-3">
            <input type="email" formControlName="email" class="form-control"
              placeholder="correo electrónico">
            <div class="text-danger" *ngIf="registroForm.get('email')?.hasError('required') &&
              registroForm.get('email')?.touched">
              <span>El correo electrónico es obligatorio</span>
            </div>/.text-danger
          </div>/.mb-3
        </div>
      </div>
    </div>
  </div>
</div>
```

```

    </div>/.text-danger
  </div>/.mb-3

  <!-- Fnacimiento -->

  <div class="mb-3">
    <input type="text" formControlName="Fnacimiento" class="form-control"
      placeholder="Fecha de nacimiento">
    <div class="text-danger" *ngIf="registroForm.get('Fnacimiento')?.hasError(
      'required') && registroForm.get('Fnacimiento')?.touched">
      <span>La fecha es obligatoria</span>
    </div>/.text-danger
  </div>/.mb-3

  <!-- email -->

  <div class="mb-3">
    <input type="email" formControlName="email" class="form-control"
      placeholder="correo electrónico">
    <div class="text-danger" *ngIf="registroForm.get('email')?.hasError('required') &&
      registroForm.get('email')?.touched">
      <span>El correo electrónico es obligatorio</span>
    </div>/.text-danger
  </div>/.mb-3
</div>
```

```
</div>/.mb-3

<!-- telefono -->

<div class="mb-3">
  <input type="number" formControlName="telefono" class="form-control"
    placeholder="telefono">
  <div class="text-danger" *ngIf="registroForm.get('telefono')?.hasError
    ('required') && registroForm.get('telefono')?.touched">
    <span>El telefono es obligatorio</span>
  </div>/text-danger
</div>/.mb-3

<!-- pass -->

<div class="mb-3">
  <input type="password" formControlName="pass" class="form-control"
    placeholder="contraseña">
  <!-- <div class="text-danger" *ngIf="registroForm.get('pass')?.hasError
    ('required') && registroForm.get('pass')?.touched">
    <span>El pass es obligatorio</span>
  </div> -->
</div>/.mb-3
```

```
<!-- tipo -->

<div class="mb-3">
  <input type="text" formControlName="tipo" class="form-control" placeholder="tipo">
  <!-- <div class="text-danger" *ngIf="registroForm.get('tipo')?.hasError
    ('required') && registroForm.get('tipo')?.touched">
    <span>El tipo es obligatorio</span>
  </div> -->
</div>/.mb-3

<!-- Explorador de archivo -->

<div class="mb-3">
  <label for="Imagen">Imagen</label>
  <input type="file" formControlName="imagen" class="form-control"
    placeholder="Imagen">
  <!-- <div class="text-danger" *ngIf="registroForm.get('imagen')?.hasError
    ('required') && registroForm.get('imagen')?.touched">
    <span>La Imagen es obligatoria</span>
  </div> -->
</div>/.mb-3
```

```
<!-- Botonera -->

<div class="mb-3">
  <button class="btn btn-secondary float-start" routerLink="/">VOLVER</button>
  <button type="submit" [disabled]="registroForm.invalid" class="btn btn-success
    float-end">ACEPTAR</button>
</div> /.mb-3

</form> /.mb-3
</div> /.card-body.text-center
</div> /.card
</div> /.col-lg-12
</div> /.row
</div> /.container.mt-15
```




SISTEMAS DE VALORACIÓN MF 0492_3 – UF1846 – E2

RESULTADOS A COMPROBAR	INDICADORES DE LOGRO	ESCALA DE MEDIDAS		
3. Especificar las características de los protocolos estándares del mercado para poder utilizar servicios web en la aplicación a desarrollar Conforme el criterio de evaluación CE 1.2	Especifica las características de los protocolos estándares del mercado	- Especifica las características de los protocolos estándares del mercado entre un 75% y 100%	B	20
		- Especifica las características de los protocolos estándares del mercado entre un 50 % y 75%	R	10
		- Especifica las características de los protocolos estándares del mercado por debajo de un 50 %	M	0
	Utiliza servicios web en la aplicación a desarrollar.	- Utiliza servicios web en la aplicación a desarrollar entre un 75% y 100%.	B	20
		- Utiliza servicios web en la aplicación a desarrollar entre un 50% y 75%.	R	10
		- Utiliza servicios web en la aplicación a desarrollar por debajo de un 50%.	M	0
4. Seleccionar y emplear los servicios web más adecuados para ser utilizados en la aplicación web en función del diseño especificado. Conforme el criterio de evaluación CE 1.3	2.1 Selecciona los servicios web más adecuados para ser utilizados en la aplicación web	- Selecciona los servicios web más adecuados para ser utilizados en la aplicación web entre un 75% y 100%.	B	20
		- Selecciona los servicios web más adecuados para ser utilizados en la aplicación web entre un 50% y 75%.	R	10
		- Selecciona los servicios web más adecuados para ser utilizados en la aplicación web por debajo de un 50%.	M	0
	2.2 Emplea los servicios web más adecuados para ser utilizados en la aplicación web	- Emplea los servicios web más adecuados para ser utilizados en la aplicación web entre un 75% y 100%.	B	30
		- Emplea los servicios web más adecuados para ser utilizados en la aplicación web entre un 50% y 75%.	R	15
		- Emplea los servicios web más adecuados para ser utilizados en la aplicación web por debajo de un 50%.	M	0
	2.3 Documenta los servicios web más adecuados para ser utilizados en la aplicación web	- Documenta los servicios web más adecuados para ser utilizados en la aplicación web entre un 75% y 100%.	B	10
		- Documenta los servicios web más adecuados para ser utilizados en la aplicación web entre un 50% y 75%.	R	05
		- Documenta los servicios web más adecuados para ser utilizados en la aplicación web por debajo de un 50%.	M	0
	Valor mínimo exigible: 50	Valor máximo: 100		