

<b>NOMBRE Y APELLIDOS:</b>			<b>FECHA: 25-08-2023</b>		
<b>DOCENTE: MANUEL MACÍAS PÉREZ</b>			<b>NOTA:</b>		
<b>(IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB.</b>			<b>Nº CURSO: 22-35/008902</b>		
MF:	0492	UNIDADES DE APRENDIZAJE A LAS QUE RESPONDE:	UA2	Duración:	3 h
UF:	1844				
PRÁCTICA Nº:	E2				
<b>DENOMINACIÓN: Principios de la orientación a objetos</b>					
<p><b><u>DESCRIPCIÓN</u></b></p> <p>1.- El alumno de forma individual deberá en base a la planificación realizada en la prueba E1 de la UF1844, realizar lo siguiente:</p> <ul style="list-style-type: none"> <li>Adaptar la documentación basándose en las herramientas vistas en clase (Framework), tomando como referencia la programación por componentes del Framework de desarrollo.</li> <li>Crear el proyecto de aplicación con sus componentes definido en la documentación.</li> <li>Incluir el documento creado para la planificación de la aplicación en formato .md dentro del directorio raíz del proyecto.</li> </ul> <p>Pegar en este Archivo el resultado de la prueba y convertir en pdf. Enviar o Subir a Github. La práctica se realizará de manera individual.</p> <p><b><u>MEDIOS PARA SU REALIZACIÓN</u></b></p> <ul style="list-style-type: none"> <li>Equipo informático.</li> <li>Aplicación Visual Code Studio instalada en el equipo.</li> <li>Navegadores actualizados</li> </ul> <p><b><u>PAUTAS DE ACTUACIÓN DEL FORMADOR</u></b></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"> <li>Fijará los objetivos de la práctica.</li> <li>Aportará las instrucciones necesarias a los alumnos/as para la realización de la misma, haciendo hincapié en aquellos aspectos más relevantes.</li> <li>Facilitará a cada alumno/a la documentación necesaria para el desarrollo de la práctica.</li> <li>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.</li> </ul> <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>					

ESPECIFICACIONES PARA LA EVALUACIÓN DE LA PRÁCTICA	
Resultados a comprobar	Indicadores de logro
<p>1. Crea objetos, clases y métodos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos.</p> <p>Conforme el criterio de evaluación CE 1.2</p>	1.1 Crea objetos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos
	1.2 Crea clases adecuadas a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos
	1.3 Crea métodos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos

### 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 en base a la planificación realizada en la prueba E1 de la UF1844, realizar lo siguiente:

- Adaptar la documentación basándose en las herramientas vistas en clase (Framework), tomando como referencia la programación por componentes del FrameWork de desarrollo.
- Crear el proyecto de aplicación con sus componentes definido en la documentación.
- Incluir el documento creado para la planificación de la aplicación en formato .md dentro del directorio raíz del proyecto.

Pegar en este Archivo el resultado de la prueba y convertir en pdf. Enviar o Subir a Github.

La práctica se realizará de manera individual.

## Ejemplo:

- Interacción del usuario:
  - o A través de botones en la parte superior – derecha:
    - Botón 1 – Acceder
    - .
    - .
    - .
  - o A través de formularios de pedidos donde se recogerán los siguientes datos
    - Formulario de pedido
      - Nombre
      - Correo electrónico
      - Teléfono
    - Formulario de consulta
    - .
    - .
    - .



## SISTEMAS DE VALORACIÓN MF 0492\_3 – UF1844 – E2

RESULTADOS A COMPROBAR	INDICADORES DE LOGRO	ESCALA DE MEDIDAS		
<p>2. Crea objetos, clases y métodos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos.</p> <p>Conforme el criterio de evaluación CE 1.2</p>	Crea objetos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos	- Crea objetos adecuados a la funcionalidad del componente software entre un 75% y 100%	B	40
		- Crea objetos adecuados a la funcionalidad del componente software entre un 50 % y 75%	R	20
		- Crea objetos adecuados a la funcionalidad del componente software por debajo de un 50 %	M	0
	Crea clases adecuadas a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos	- Crea clases adecuadas a la funcionalidad del componente software entre un 75% y 100%.	B	30
		- Crea clases adecuadas a la funcionalidad del componente software entre un 50% y 75%.	R	15
		- Crea clases adecuadas a la funcionalidad del componente software por debajo de un 50%.	M	0
	Crea métodos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos	- Crea métodos adecuados a la funcionalidad del componente software entre un 75% y 100%.	B	30
		- Crea métodos adecuados a la funcionalidad del componente software entre un 50% y 75%.	R	15
		- Crea métodos adecuados a la funcionalidad del componente software por debajo de un 50%.	M	0
	<b>Valor mínimo exigible: 50</b>	<b>Valor máximo: 100</b>		

```

<header>

  <app-cabecera></app-cabecera>

</header>

<router-outlet></router-outlet>

```

```
<footer>
```

```
  <app-footer></app-footer>
```

```
</footer>
```

```
:root{
```

```
  --cabecera: #79c947;
```

```
  --letraCabecera:#1a0765;
```

```
  --fondo:#091b66d8;
```

```
  --detalle: #650707;
```

```
  --letra: #9bd67de0;
```

```
}
```

```
/* Descripción */
```

```
.Cuerpo{
```

```
display: grid;
```

```
height: 65vh;
```

```
color: var(--letra);
```

```
margin: 3vw;
```

```
}  
  
.imagen{  
  
    grid-column: 1/5;  
  
    grid-row: 1/3;  
  
    width: 30vw;  
  
}  
  
.imagen img{  
  
    border-radius: 40%;  
  
}  
  
.texto{  
  
    color: var(--detalle);  
  
    grid-column: 4/6;  
  
    grid-row: 2/4;  
  
}  
  
.imagen2{  
  
    grid-column: 7/8;  
  
    grid-row: 3/4;  
  
    width: 30vw;  
  
}
```



```
.imagen2 img{
    border-radius: 40%;
}

/* Catalogo */

.catalogo{
    height: 70vh;
    color: var(--letra);
    margin: 3vw;
    display: flex;
    overflow: scroll;
}

.catalogo aside{
    width: 30%;
}

.catalogo div{
    width: 70%;
}
```

```
.lis{  
  
  display: grid;  
  
  grid-template-columns: 1fr 1fr 1fr 1fr;  
  
  gap: .8em;  
  
  height: 70vh;  
  
}  
  
.lis div{  
  
  margin: 2em;  
  
  display: flex;  
  
  flex-direction: column;  
  
  flex-wrap: nowrap;  
  
  justify-content: center;  
  
  align-items: center;  
  
  text-align: center;  
  
  background-color: var(--detalle);  
  
  border-radius: 10px;  
  
  width: 17vw;  
  
  height: 40vh;  
  
  transition: 1s;
```

```
}

.lis div:hover{

    scale: 0.8;

    transition: 1s;

}

.lis div img{

    width: 100%;

    height: auto;

}

/*Login*/

.login{

    display: flex;

    justify-content: center;

    align-items: center;

    height: 70vh;

}

.login form{
```

```
display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

margin: 3em;
}

.login form a{

    color: var(--letra);
}

.login form input{

    margin-top: 10px;
}

/* registro */

.registro{

    height: 70vh;

    display: flex;

    justify-content: center;
```

```
    align-items: center;
}

.registro form{

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

    margin: 3em;
}

.registro form input{

    margin-top: 10px;

    border-radius: 5px;

    width: 200px;
}

.registro form button{

    margin-top: 10px;
}

/* Contacto */
```

```
.contacto{  
  
    height: 70vh;  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
}
```

```
.contacto form{  
  
    display: flex;  
  
    flex-direction: column;  
  
    justify-content: center;  
  
    align-items: center;  
  
    margin: 3em;  
  
}
```

```
.contacto form input{  
  
    margin-top: 10px;  
  
    border-radius: 5px;  
  
    width: 260px;  
  
}
```

```
.contacto form textarea{

    margin-top: 10px;

    border-radius: 5px;

    width: 400px;

}

.contacto form button{

    margin-top: 5px;

}
```

```
import { Component } from '@angular/core';

@Component({

    selector: 'app-root',

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

    styleUrls: ['./app.component.css']

})
```

```
export class AppComponent {  
  
  title = 'E2';  
  
}
```