

<b>NOMBRE Y APELLIDOS:</b> Christopher Da Silva Tan			<b>FECHA:</b> 31-08-2023		
<b>DOCENTE:</b> MANUEL MACÍAS PÉREZ			<b>NOTA:</b>		
<b>(IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB.</b>			<b>Nº CURSO:</b> 22-35/008902		
MF:	0492	UNIDADES DE APRENDIZAJE A LAS QUE RESPONDE:	UA3	Duración:	3 h
UF:	1844				
PRÁCTICA Nº:	E3				
<b>DENOMINACIÓN:</b> Integración de componentes y librerías.					
<p><b>DESCRIPCIÓN</b></p> <p>1.- El alumno de forma individual deberá en base al proyecto realizado en la prueba E2 de la UF1844, realizar lo siguiente:</p> <ul style="list-style-type: none"> <li>Integrar los componentes que están disponibles para descarga en GitHub. <a href="https://github.com/MacPe-Dev/PruebasIFCD0210.git">https://github.com/MacPe-Dev/PruebasIFCD0210.git</a></li> <li>Revisar los componentes para su correcto funcionamiento.</li> <li>Adaptar los estilos de los componentes integrados a la aplicación creada.</li> <li>Añadir a la documentación los componentes integrados.</li> </ul> <p>Pegar en este Archivo el resultado (pantallazo) de la integración de los componentes según se muestra en el navegador y convertir en pdf. Enviar o Subir a Github el código completo de la integración.</p> <p>La práctica se realizará de manera individual.</p> <p><b>MEDIOS PARA SU REALIZACIÓN</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>PAUTAS DE ACTUACIÓN DEL FORMADOR</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>					

Al finalizar la práctica el formador examinará el desarrollo que han realizado los/as alumnos/as, proponiendo las medidas de corrección, en caso necesario.

#### ESPECIFICACIONES PARA LA EVALUACIÓN DE LA PRÁCTICA

Resultados a comprobar	Indicadores de logro
1. Documenta el componente software desarrollado. Conforme el criterio de evaluación CE 1.4	1.1 Documenta el componente software desarrollado
	1.2 Explica el componente software desarrollado
2. En un supuesto práctico en el que se pide construir componentes de software mediante herramientas de desarrollo y lenguajes de programación orientados a objetos a partir de documentación de diseño detallado: Conforme el criterio de evaluación CE 1.5	2.1 Integra componentes software de control del contenido de los documentos ubicados en el servidor
	2.2 Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor
	2.3 Crea componentes software con la funcionalidad de aplicación de cliente
	2.4 Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software

#### 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 al proyecto realizado en la prueba E2 de la UF1844, realizar lo siguiente:

- Integrar los componentes que están disponibles para descarga en GitHub.
- Revisar los componentes para su correcto funcionamiento.
- Adaptar los estilos de los componentes integrados a la aplicación creada.
- Añadir a la documentación los componentes integrados.

Pegar en este Archivo el resultado (pantallazo) de la integración de los componentes según se muestra en el navegador y convertir en pdf. Enviar o Subir a Github el código completo de la integración.

La práctica se realizará de manera individual.

## Ejemplo:

```
import { Component, OnInit } from '@angular/core';
import { ServiciosBD } from '../modelos/lstServicios'
import { ListaserviciosService } from '../servicios/listaservicios.service'

@Component({
  selector: 'app-lista-servicios',
  templateUrl: '../lista-servicios.component.html',
  styleUrls: ['../lista-servicios.component.css'],
  providers: [ListaserviciosService]
})
export class ListaServiciosComponent implements OnInit {
  public listaServicios: Array<ServiciosBD>
  public codServicios: string[] = []

  constructor(private _listaServicios: ListaserviciosService) {
    this.listaServicios = new Array<ServiciosBD>()
  }

  ngOnInit(): void {
    this.listaServicios = this._listaServicios.getServicios()
  }
}
```



Presupuesto Contacto Ejecución Métodos Registro Login

007 - Guillotina - Una guillotina es un aparato diseñado para aplicar la pena capital por decapitación de manera eficiente. El dispositivo consiste en un marco de madera alto y vertical provisto de una pesada lama de acero con un filo oblicuo, suspendida en la parte superior. DISPONIBLE

069 - Verdugo a domicilio - Un verdugo es la persona encargada de llevar a cabo la ejecución de una persona condenada a la pena de muerte o penas corporales por la justicia civil o eclesiástica. Nuestro verdugo vendrá corriendo y ejecutará en un abrir y cerrar de ojos. DISPONIBLE

420 - Pistoletazo - Una pistola es un arma de fuego corta diseñada para ser apuntada y disparada con una sola mano. Se usa una Colt que de un solo disparo incapacita al 100% el objetivo. DISPONIBLE

666 - Meteorito - Un meteorito es un meteoride que alcanza la superficie de un planeta debido a que no se desintegra por completo en la atmósfera. Es un impacto directo, eficaz que solo afecta al objeto (persona) impactada sin erosionar el entorno. DISPONIBLE

[Ver por detalles](#)

<p>Código: 007 <b>Nombre: Guillotina</b></p> <p>Descripción: Una guillotina es un aparato diseñado para aplicar la pena capital por decapitación de manera eficiente. El dispositivo consiste en un marco de madera alto y vertical provisto de una pesada lama de acero con un filo oblicuo, suspendida en la parte superior. Precio: €675.00</p>  <p>Estado: DISPONIBLE</p> <p><a href="#">Contratar</a></p>	<p>Código: 069 <b>Nombre: Verdugo a domicilio</b></p> <p>Descripción: Un verdugo es la persona encargada de llevar a cabo la ejecución de una persona condenada a la pena de muerte o penas corporales por la justicia civil o eclesiástica. Nuestro verdugo vendrá corriendo y ejecutará en un abrir y cerrar de ojos. Precio: €2,025.00</p>  <p>Estado: DISPONIBLE</p> <p><a href="#">Contratar</a></p>	<p>Código: 420 <b>Nombre: Pistoletazo</b></p> <p>Descripción: Una pistola es un arma de fuego corta diseñada para ser apuntada y disparada con una sola mano. Se usa una Colt que de un solo disparo incapacita al 100% el objetivo. Precio: €4,200.00</p>  <p>Estado: DISPONIBLE</p> <p><a href="#">Contratar</a></p>	<p>Código: 666 <b>Nombre: Meteorito</b></p> <p>Descripción: Un meteorito es un meteoride que alcanza la superficie de un planeta debido a que no se desintegra por completo en la atmósfera. Es un impacto directo, eficaz que solo afecta al objeto (persona) impactada sin erosionar el entorno. Precio: €69,420.00</p>  <p>Estado: DISPONIBLE</p> <p><a href="#">Contratar</a></p>
---	--	---	--

**¡Producto adquirido!**

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

import { RouterModule, Routes } from '@angular/router';

import { ContactoComponent } from
'./contacto/contacto.component';

import { EjecucionComponent } from
'./ejecucion/ejecucion.component';

import { LoginComponent } from './login/login.component';

import { PresupuestoComponent } from
'./presupuesto/presupuesto.component';

import { RegistroComponent } from
'./registro/registro.component';

import { MetodoComponent } from './metodo/metodo.component';

const routes: Routes = [

  {path: "Contacto", component: ContactoComponent},
```

```
{path: "Ejecucion", component: EjecucionComponent},  
  
{path: "Login", component: LoginComponent},  
  
{path: "Presupuesto", component: PresupuestoComponent},  
  
{path: "Registro", component: RegistroComponent},  
  
{path: "Metodo", component: MetodoComponent},  
  
];  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes)],  
  exports: [RouterModule]  
})  
  
export class AppRoutingModule { }
```

```
import { NgModule } from '@angular/core';  
  
import { BrowserModule } from '@angular/platform-browser';  
  
import { AppRoutingModule } from './app-routing.module';  
  
import { AppComponent } from './app.component';  
  
import { RegistroComponent } from  
'./registro/registro.component';
```

```
import { LoginComponent } from './login/login.component';

import { ContactoComponent } from
'./contacto/contacto.component';

import { EjecucionComponent } from
'./ejecucion/ejecucion.component';

import { PresupuestoComponent } from
'./presupuesto/presupuesto.component';

import { HeaderComponent } from './header/header.component';

import { FooterComponent } from './footer/footer.component';

import { MetodoComponent } from './metodo/metodo.component';

import { GestionServiciosComponent } from
'./gestion-servicios/gestion-servicios.component';

import { ListaServiciosComponent } from
'./lista-servicios/lista-servicios.component';

@NgModule({
  declarations: [
    AppComponent,
    RegistroComponent,
    LoginComponent,
    ContactoComponent,
    EjecucionComponent,
    PresupuestoComponent,
```



```
HeaderComponent,

FooterComponent,

MetodoComponent,

GestionServiciosComponent,

ListaServiciosComponent

],

imports: [

  BrowserModule,

  AppRoutingModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }
```

### SISTEMAS DE VALORACIÓN MF 0492\_3 – UF1844 – E3

RESULTADOS A COMPROBAR	INDICADORES DE LOGRO	ESCALA DE MEDIDAS	
1. Documenta el componente software desarrollado. Conforme el criterio de evaluación CE 1.4	1.1 Documenta el componente software desarrollado	- Documenta el componente software desarrollado entre un 75% y 100%	B 10
		- Documenta el componente software desarrollado entre un 50 % y 75%	R 5
		- Documenta el componente software desarrollado por debajo de un 50 %	M 0
	1.2 Explica el componente software desarrollado	- Explica el componente software desarrollado entre un 75% y 100%.	B 10
		- Explica el componente software desarrollado entre un 50% y 75%.	R 5
		- Explica el componente software desarrollado por debajo de un 50%.	M 0
2. En un supuesto práctico en el que se pide construir componentes de software mediante herramientas de desarrollo y lenguajes de programación orientados a objetos a partir de documentación de diseño detallado: Conforme el criterio de evaluación CE 1.5	2.1 Integra componentes software de control del contenido de los documentos ubicados en el servidor	- Integra componentes software de control del contenido de los documentos ubicados en el servidor entre un 75% y 100%.	B 20
		- Integra componentes software de control del contenido de los documentos ubicados en el servidor entre un 50% y 75%.	R 10
		- Integra componentes software de control del contenido de los documentos ubicados en el servidor por debajo de un 50%.	M 0
	2.2 Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor	- Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor entre un 75% y 100%.	B 20
		- Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor entre un 50% y 75%.	R 10
		- Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor por debajo de un 50%.	M 0
	2.3 Crea componentes software con la funcionalidad de aplicación de cliente	- Crea componentes software con la funcionalidad de aplicación de cliente entre un 75% y 100%.	B 20
		- Crea componentes software con la funcionalidad de aplicación de cliente entre un 50% y 75%.	R 10
			M 0

		- Crea componentes software con la funcionalidad de aplicación de cliente por debajo de un 50%.			
	2.4 Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software	- Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software entre un 75% y 100%.	B	20	
		- Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software entre un 50% y 75%.	R	10	
		- Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software por debajo de un 50%.	M	0	
		Valor mínimo exigible: 50	Valor máximo: 100		