

NOMBRE Y APELLIDOS:			FECHA: 31-08-2023		
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:	UA3	Duración:	3 h
UF:	1844				
PRÁCTICA Nº:	E3				
DENOMINACIÓN: Integración de componentes y librerías.					
<p><u>DESCRIPCIÓN</u></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"> • Integrar los componentes que están disponibles para descarga en GitHub. https://github.com/MacPe-Dev/PruebasIFCD0210.git • 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. <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><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>					

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/1stServicios'
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()
  }
}
```

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';
import { CabeceraComponent } from './componentes/cabecera/cabecera.component';
import { InfoComponent } from './componentes/info/info.component';
import { TiendaComponent } from './componentes/tienda/tienda.component';
import { AccesoComponent } from './componentes/acceso/acceso.component';
import { RegistroComponent } from './componentes/registro/registro.component';
import { FooterComponent } from './componentes/footer/footer.component';
import { CarritoComponent } from './componentes/carrito/carrito.component';
```

```
import { ContactoComponent } from './componentes/contacto/contacto.component';
import { ListaServiciosComponent } from
'./componentes/lista-servicios/lista-servicios.component';
import { GestionServiciosComponent } from
'./componentes/gestion-servicios/gestion-servicios.component';

@NgModule({
  declarations: [
    AppComponent,
    CabeceraComponent,
    InfoComponent,
    TiendaComponent,
    AccesoComponent,
    RegistroComponent,
    FooterComponent,
    CarritoComponent,
    ContactoComponent,
    ListaServiciosComponent,
    GestionServiciosComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AppComponent } from './app.component';
import { AccesoComponent } from './componentes/acceso/acceso.component';
import { InfoComponent } from './componentes/info/info.component';
import { TiendaComponent } from './componentes/tienda/tienda.component';
```

```
import { ContactoComponent } from './componentes/contacto/contacto.component';
import { RegistroComponent } from './componentes/registro/registro.component';
import { CarritoComponent } from './componentes/carrito/carrito.component';
import { CabeceraComponent } from './componentes/cabecera/cabecera.component';
import { FooterComponent } from './componentes/footer/footer.component';
import { ListaServiciosComponent } from
'./componentes/lista-servicios/lista-servicios.component';
import { GestionServiciosComponent } from
'./componentes/gestion-servicios/gestion-servicios.component';

const routes: Routes = [
  {path: '', component: InfoComponent},
  {path: 'Header', component: CabeceraComponent},
  {path: 'MiTienda', component: InfoComponent},
  {path: 'Catalogo', component: TiendaComponent},
  {path: 'Acceso', component: AccesoComponent},
  {path: 'Contacto', component: ContactoComponent},
  {path: 'Registro', component: RegistroComponent},
  {path: 'Carrito', component: CarritoComponent},
  {path: 'Footer', component: FooterComponent},
  {path: 'ListaServicios', component: ListaServiciosComponent,
  children:[
    {path:'GestionServicios', component: GestionServiciosComponent}
  ]},
  {path: '**', component: AppComponent}
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

[MiTienda](#)[Catálogo](#)[Acceso](#)[Contacto](#)[Servicios](#)

Gestión de Servicios

100 - Sastrería - Servicio de sastrería por prenda DISPONIBLE

101 - Estampado de ropa - Servicio de estampado en ropa con plantilla de cliente DISPONIBLE

102 - Estampado de ropa - Servicio de estampado en ropa, realizando nosotros el diseño DISPONIBLE

103 - Lavandería - Servicio de lavandería DISPONIBLE

[Ver por detalles](#)

Código: 100

Nombre:
Sastrería

Descripción: Servicio de sastrería por

Código: 101

Nombre:
Estampado de ropa

Descripción: Servicio de estampado en ropa con plantilla de cliente

Código: 102

Nombre:
Estampado de ropa

Descripción: Servicio de estampado en ropa, realizando nosotros el diseño

Código: 103

Nombre:
Lavandería

Descripción: Servicio de lavandería

[Política de privacidad de cookies](#)

[Términos y condiciones de uso](#)

[MiTienda](#)[Catálogo](#)[Acceso](#)[Contacto](#)[Servicios](#)

Nombre:
Sastrería

Descripción: Servicio de sastrería por prenda

Precio: €20.00



Estado: DISPONIBLE

[Contratar](#)

Nombre:
Estampado de ropa

Descripción: Servicio de estampado en ropa con plantilla de cliente

Precio: €30.00



Estado: DISPONIBLE

[Contratar](#)

Nombre:
Estampado de ropa

Descripción: Servicio de estampado en ropa, realizando nosotros el diseño

Precio: €60.00



Estado: DISPONIBLE

[Contratar](#)

Nombre:
Lavandería

Descripción: Servicio de lavandería

Precio: €20.00



Estado: DISPONIBLE

[Contratar](#)

[Política de privacidad de cookies](#)

[Términos y condiciones de uso](#)





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		