|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE Y APELLIDOS:**  **Daoulé MALLE** | | | | **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.** | | | | | | |
| **DESCRIPCIÓN**  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.   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.   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.  **MEDIOS PARA SU REALIZACIÓN**   * Equipo informático. * Aplicación Visual Code Studio instalada en el equipo. * Navegadores actualizados   **PAUTAS DE ACTUACIÓN DEL FORMADOR**  *Al inicio de la práctica, que se desarrollará de manera individual por cada uno de los alumnos, el formador/a realizará las siguientes actuaciones:*  - 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.  Durante la realización de la práctica el formador/a supervisará el desarrollo de esta para evaluar tanto los procedimientos como el resultado final.  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. Documenta el componente software desarrollado | | | |
| * 1. Explica el componente software desarrollado | | | |
| 1. 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()

 }

}

**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%  - Documenta el componente software desarrollado entre un 50 % y 75%  - Documenta el componente software desarrollado por debajo de un 50 % | B  R  M | 10  5  0 |
| 1.2 Explica el componente software desarrollado | | - Explica el componente software desarrollado entre un 75% y 100%.  - Explica el componente software desarrollado entre un 50% y 75%.  - Explica el componente software desarrollado por debajo de un 50%. | B  R  M | 10  5  0 |
| 1. 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%.  - Integra componentes software de control del contenido de los documentos ubicados en el servidor entre un 50% y 75%.  - Integra componentes software de control del contenido de los documentos ubicados en el servidor por debajo de un 50%. | B  R  M | 20  10  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%.  - Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor entre un 50% y 75%.  - Utiliza variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor por debajo de un 50%. | B  R  M | 20  10  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%.  - Crea componentes software con la funcionalidad de aplicación de cliente entre un 50% y 75%.  - Crea componentes software con la funcionalidad de aplicación de cliente por debajo de un 50%. | B  R  M | 20  10  0 |
| 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%.  - Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software entre un 50% y 75%.  - Crea componentes software que puedan ofrecer su funcionalidad a otros componentes software por debajo de un 50%. | B  R  M | 20  10  0 |
|  | | **Valor mínimo exigible: 50** | **Valor máximo: 100** | |  |