|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE Y APELLIDOS:**  **Daoulé MALLE** | | | | **FECHA: 02-10-2023** | | |
| **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: | E1 |
| DENOMINACIÓN: **api-rest** | | | | | | |
| **DESCRIPCIÓN**  1.- El alumno de forma individual deberá realizar:   1. La configuración un servidor en Nodejs y Typescript para servir como servicio de archivos.  * Tener en cuenta:   + Modelo (Filename, iduser y path)   + Controller (getFile)   + Routes (post)   + Interface   + Conexión con Base de Datos SQL   + Carpeta para alojar archivos   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.  import { DataType, DataTypes } from "sequelize";  import db from '../config/connectdb';  import { STRING } from "sequelize";  const File = db.define('File', {      nombre: {          type: DataTypes.STRING      },      iduser: {          type: DataTypes.STRING      },      path: {          type: DataTypes.STRING      }  },{      createdAt: false,      updatedAt: false  });  export default File;  import {Request, Response} from 'express';  import File from '../models/FileModel';  export const getFile = async (req: Request, res: Response) => {      const {id} = req.params;      const file = await File.findByPk(id);      if (file) {          res.json(file)      }else{          res.status(404).json({              msg: 'No existe un fichero con ese &{id}'          })      }  }  export const postFile =  async (req: Request, res: Response) => {      const {body} = req;      try{          await File.create(body);          res.json({              msg: '!El fichero fue agregado con exito!'          })      }catch (error){          console.log(error);          res.json({              msg: 'Ha occurido un error'          })      }  }  import { Router } from "express";  import { postFile  } from "../controllers/FileController";  const router = Router();  router.post('/', postFile);  export default router;  export interface File{      nombre: string;      iduser: string;      path: string;  }  import { Sequelize } from "sequelize";  const sequelize = new Sequelize( 'appfile' , 'root' , '\*As03Ch07Fz13#',{      host: 'localhost',      dialect:'mysql'  });  export default sequelize;  CREATE TABLE `appfile`.`file` (  `nombre` VARCHAR(45) NOT NULL,  `iduser` VARCHAR(45) NULL,  `path` VARCHAR(45) NULL,  PRIMARY KEY (`nombre`));  **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. Identificar las posibilidades que ofrecen los servicios distribuidos web para su integración en la aplicación a desarrollar.   Conforme el criterio de evaluación CE 1.1 | | | * 1. Identifica las posibilidades que ofrecen los servicios distribuidos web | | | |
| * 1. Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web. | | | |
| * 1. Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web. | | | |
| 1.4 Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de 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:

1. La configuración un servidor en Nodejs y Typescript para servir como servicio de archivos.

* Tener en cuenta:
  + Modelo (Filename, iduser y path)
  + Controller (getFile)
  + Routes (post)
  + Interface
  + Conexión con Base de Datos SQL
  + Carpeta para alojar archivos

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:

const mongoose =  require('mongoose')

const UserSchema = new mongoose.Schema(

    {

    cod: Number,

    nombre: String,

    apellidos: String,

    edad: Number,

    mail: String

    }

)

module.exports = mongoose.model('user', UserSchema)

**SISTEMAS DE VALORACIÓN MF 0492\_3 – UF1846 – E1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RESULTADOS A COMPROBAR** | **INDICADORES DE LOGRO** | **ESCALA DE MEDIDAS** | |  |
| 1. Identificar las posibilidades que ofrecen los servicios distribuidos web para su integración en la aplicación a desarrollar.   Conforme el criterio de evaluación CE 1.1 | 1.1 Identifica las posibilidades que ofrecen los servicios distribuidos web | - Identifica las posibilidades que ofrecen los servicios distribuidos web entre un 75% y 100%  - Identifica las posibilidades que ofrecen los servicios distribuidos web entre un 50 % y 75%  - Identifica las posibilidades que ofrecen los servicios distribuidos web por debajo de un 50 % | B  R  M | 30  15  0 |
| 1.2 Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web. | - Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web entre un 75% y 100%.  - Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web entre un 50% y 75%.  - Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web por debajo de un 50%. | B  R  M | 40  20  0 |
| 1.3 Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web. | - Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web entre un 75% y 100%.  - Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web entre un 50% y 75%.  - Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web por debajo de un 50%. | B  R  M | 20  10  0 |
| 1.4 Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de la aplicación web. | - Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de la aplicación web entre un 75% y 100%.  - Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de la aplicación web entre un 50% y 75%.  - Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de la aplicación web por debajo de un 50%. | B  R  M | 10  05  0 |
|  | **Valor mínimo exigible: 50** | **Valor máximo: 100** | |  |