

NOMBRE Y APELLIDOS: Christopher Da Silva Tan			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					
<p><u>DESCRIPCIÓN</u></p> <p>1.- El alumno de forma individual deberá realizar:</p> <ol style="list-style-type: none"> La configuración un servidor en Nodejs y Typescript para servir como servicio de archivos. <ul style="list-style-type: none"> Tener en cuenta: <ul style="list-style-type: none"> Modelo (Filename, iduser y path) Controller (getFile) Routes (post) Interface Conexión con Base de Datos SQL Carpeta para alojar archivos <p>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.</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> <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. Identificar las posibilidades que ofrecen los servicios distribuidos web para su integración en la aplicación a desarrollar.</p> <p>Conforme el criterio de evaluación CE 1.1</p>	1.1 Identifica las posibilidades que ofrecen los servicios distribuidos web
	1.2 Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web.
	1.3 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:

2. La configuración un servidor en Nodejs y Typescript para servir como servicio de archivos.
 - Tener en cuenta:
 - o Modelo (Filename, iduser y path)
 - o Controller (getFile)
 - o Routes (post)
 - o Interface
 - o Conexión con Base de Datos SQL
 - o 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 { DataTypes } from 'sequelize';

import db from '../config/connectdb';

const Upload = db.define('Upload',{

  filename: {

    type: DataTypes.STRING

  },

  iduser: {

    type: DataTypes.STRING

  },

  path: {

    type: DataTypes.STRING

  }

}, {

  createdAt: false,

  updatedAt: false

});

export default Upload;
```

```
import {Request, Response} from 'express';

import Upload from '../models/UploadModel';

export const getUploads = async (req: Request, res: Response)
=>{

    const listUploads = await Upload.findAll();

    res.json(listUploads);

}

export const getUpload = async (req: Request, res: Response) =>
{

    const {id} = req.params;

    const upload = await Upload.findByPk(id);

    if (upload) {

        res.json(upload)

    } else {

        res.status(404).json({

            msg: `No existe un archivo con ese ${id}`

        })

    }

}
```

```
export const deleteUpload = async (req: Request, res: Response)
=> {

  const {id} = req.params;

  const upload = await Upload.findByPk(id);

  if (!upload) {

    res.status(404).json({

      msg: `No existe un archivo con ese ${id}`

    })

  } else {

    await upload.destroy();

    res.json ({

      msg: 'El archivo fue eliminado con éxito'

    })

  }

}

export const postUpload = async (req: Request, res: Response) =>
{

  const {body} = req;

  try{

    await Upload.create(body);

    res.json({

      msg: '¡El archivo fue agregado con éxito!'

    })

  } catch (error) {

    res.status(500).json({

      msg: 'Error al crear el archivo'

    })

  }

}
```

```
    })

    } catch (error){

        console.log(error);

        res.json({

            msg: 'Ha ocurrido un error'

        })

    }

}

export const updateUpload = async (req: Request, res: Response)
=> {

    const {body} = req;

    const {id} = req.params;

    try{

        const upload = await Upload.findByPk(id);

        if(upload){

            await upload.update(body);

            res.json({

                msg: 'El archivo fue actualizado con éxito'

            })

        } else {

            res.status(404).json({

                msg: `No existe un archivo con ese ${id}`

            })

        }

    } catch (error){

        console.log(error);

        res.json({

            msg: 'Ha ocurrido un error'

        })

    }

}
```

```
    })  
  
  }  
  
  } catch (error){  
  
    console.log(error);  
  
    res.json({  
  
      msg: 'Ha ocurrido un error'  
  
    })  
  
  }  
  
}
```

```
export interface Upload{  
  
  filename: string;  
  
  iduser: string;  
  
  path: string;  
  
}
```

▼ upload

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		
<p>1. Identificar las posibilidades que ofrecen los servicios distribuidos web para su integración en la aplicación a desarrollar.</p> <p>Conforme el criterio de evaluación CE 1.1</p>	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%	B	30
		- Identifica las posibilidades que ofrecen los servicios distribuidos web entre un 50 % y 75%	R	15
		- Identifica las posibilidades que ofrecen los servicios distribuidos web por debajo de un 50 %	M	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%.	B	40
		- Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web entre un 50% y 75%.	R	20
		- Integra las posibilidades que ofrecen los servicios distribuidos web en la aplicación web por debajo de un 50%.	M	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%.	B	20
		- Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web entre un 50% y 75%.	R	10
		- Conoce las posibilidades que ofrecen los servicios distribuidos web en la aplicación web por debajo de un 50%.	M	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%.	B	10
		- Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de la aplicación web entre un 50% y 75%.	R	05
		- Documenta las posibilidades que ofrecen los servicios distribuidos web en la integración de la aplicación web por debajo de un 50%.	M	0
	Valor mínimo exigible: 50	Valor máximo: 100		