

NOMBRE Y APELLIDOS: Christopher Da Silva Tan			FECHA: 28/09/2023		
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 hrs.
UF:	1846				
PRÁCTICA Nº:	Actividad 1				
DENOMINACIÓN: API-REST					
<u>DESCRIPCIÓN</u> 1.- Se propone la creación de un servidor para trabajar con bases de datos SQL. Adjuntar Códigos Crear Modelos, Controller y Routes de: <ul style="list-style-type: none"> • Cursos • Usuarios • Mensajes • Tareas <pre>import {Request, Response} from 'express'; import Tarea from '../models/TareaModel'; export const getTareas = async (req: Request, res: Response) =>{ const listTareas = await Tarea.findAll(); res.json(listTareas); } export const getTarea = async (req: Request, res: Response) => { const {id} = req.params; const tarea = await Tarea.findByPk(id); if (tarea) { res.json(tarea) } else {</pre>					

```
    res.status(404).json({
      msg: `No existe un tarea con ese ${id}`
    })
  }
}

export const deleteTarea = async (req: Request, res: Response)
=> {
  const {id} = req.params;
  const tarea = await Tarea.findByPk(id);
  if (!tarea) {
    res.status(404).json({
      msg: `No existe un tarea con ese ${id}`
    })
  } else {
    await tarea.destroy();
    res.json({
      msg: 'El tarea fue eliminado con éxito'
    })
  }
}

export const postTarea = async (req: Request, res: Response)
=> {
  const {body} = req;
  try{
    await Tarea.create(body);
    res.json({
      msg: '¡El tarea fue agregado con éxito!'
    })
  } catch (error){
    console.log(error);
    res.json({
      msg: 'Ha ocurrido un error'
    })
  }
}
```

```
}  
}  
export const updateTarea = async (req: Request, res: Response)  
=> {  
  const {body} = req;  
  const {id} = req.params;  
  try{  
    const tarea = await Tarea.findByPk(id);  
    if(tarea){  
      await tarea.update(body);  
      res.json({  
        msg: 'El tarea fue actualizado con exito'  
      })  
    } else {  
      res.status(404).json({  
        msg: `No existe un tarea con ese ${id}`  
      })  
    }  
  } catch (error){  
    console.log(error);  
    res.json({  
      msg: 'Ha ocurrido un error'  
    })  
  }  
}
```

```
import { DataTypes } from 'sequelize';  
import db from '../config/connectdb';  
  
const Tarea = db.define('Tarea',{  
  nombre: {  
    type: DataTypes.STRING
```

```
    },  
    categoria: {  
      type: DataTypes.STRING  
    },  
    descripcion: {  
      type: DataTypes.STRING  
    },  
    imagen: {  
      type: DataTypes.STRING  
    },  
    fLimit: {  
      type: DataTypes.STRING  
    }  
  }, {  
    createdAt: false,  
    updatedAt: false  
  });  
export default Tarea;
```

```
import { Router } from 'express'  
import { deleteTarea, getTarea, getTareas, postTarea,  
updateTarea } from '../controllers/TareaController'  
  
const routertarea = Router();  
routertarea.get('/tarea/', getTareas);  
routertarea.get('/tarea/:id', getTarea);  
routertarea.delete('/tarea/:id', deleteTarea);  
routertarea.post('/tarea/', postTarea);  
routertarea.put('/tarea/:id', updateTarea);  
  
export default routertarea;
```

La práctica se realizará de manera individual.

MEDIOS PARA SU REALIZACIÓN

- Equipo informático.
- Aplicación Visual Code Studio instalada en el equipo.
- Navegadores actualizados