

<b>NOMBRE Y APELLIDOS:</b>			<b>FECHA: 28/09/2023</b>		
<b>DOCENTE: MANUEL MACÍAS PÉREZ</b>			<b>NOTA:</b>		
<b>(IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB</b>			<b>Nº CURSO: 22-35/008902</b>		
MF:	0492	UNIDADES DE APRENDIZAJE A LAS QUE RESPONDE:	UA1	Duración:	3 hrs.
UF:	1846				
PRÁCTICA Nº:	Actividad 1				
<b>DENOMINACIÓN: API-REST</b>					
<p><b><u>DESCRIPCIÓN</u></b></p> <p>1.- Se propone la creación de un servidor para trabajar con bases de datos SQL. Adjuntar Códigos Crear Modelos, Controller y Routes de:</p> <ul style="list-style-type: none"> <li>• Cursos</li> <li>• Usuarios</li> <li>• Mensajes</li> <li>• Tareas</li> </ul> <p>La práctica se realizará de manera individual.</p> <p><b><u>MEDIOS PARA SU REALIZACIÓN</u></b></p> <ul style="list-style-type: none"> <li>- Equipo informático.</li> <li>- Aplicación Visual Code Studio instalada en el equipo.</li> <li>- Navegadores actualizados</li> </ul>					

[Link de la carpeta src Donde esta Routers, Controller y Modelos](#)

```
1 import { Router } from 'express'
2 import { deleteUsuario, getUsuario, getUsuarios, postUsuario, updateUsuario } from '../controllers/UsuarioControler'
3
4 const router = Router();
5 router.get('/', getUsuarios);
6 router.get('/:id', getUsuario);
7 router.delete('/:id', deleteUsuario);
8 router.post('/', postUsuario);
9 router.put('/:id', updateUsuario);
10
11 export default router;
```

```
1 import { DataTypes } from "sequelize";
2 import db from '../config/connectdb';
3
4 const Usuario = db.define('Usuario', {
5   nombre: {
6     type: DataTypes.STRING
7   },
8   apellido: {
9     type: DataTypes.STRING
10  },
11  email: {
12    type: DataTypes.STRING
13  },
14  Telefono: {
15    type: DataTypes.STRING
16  },
17  foto: {
18    type: DataTypes.STRING
19  },
20  Fnacimiento: {
21    type: DataTypes.STRING
22  },
23  pass: {
24    type: DataTypes.STRING
25  },
26  tipo: {
27    type: DataTypes.STRING
28  },
29  anotacion: {
30    type: DataTypes.STRING
31  },
32 }, {
33   createdAt: false,
34   updatedAt: false
35 });
36 export default Usuario;
```

```
1 import {Request, Response} from 'express';
2 import Mensaje from '../models/MensajeModel';
3
4 export const getMensajes = async (req: Request, res: Response) => {
5   const listMensajes = await Mensaje.findAll();
6   res.json(listMensajes);
7 }
8
9 export const getMensaje = async (req: Request, res: Response) => {
10   const {id} = req.params;
11   const mensaje = await Mensaje.findByPk(id);
12   if (mensaje) {
13     res.json(mensaje)
14   } else {
15     res.status(404).json({
16       msg: 'No existe un mensaje con ese ${id}'
17     })
18   }
19 }
20 export const deleteMensaje = async (req: Request, res: Response) => {
21   const {id} = req.params;
22   const mensaje = await Mensaje.findByPk(id);
23   if (!mensaje) {
24     res.status(404).json({
25       msg: 'No existe un mensaje con ese ${id}'
26     })
27   } else {
28     await mensaje.destroy();
29     res.json({
30       msg: 'El mensaje fue eliminado con exito'
31     })
32   }
33 }
34 export const postMensaje = async (req: Request, res: Response) => {
35   const {body} = req;
36   try{
37     await Mensaje.create(body);
38     res.json({
39       msg: '¡El mensaje fue agregado con exito!'
40     })
41   } catch (error){
42     console.log(error);
43     res.json({
44       msg: 'Ha ocurrido un error'
45     })
46   }
47 }
48 export const updateMensaje = async (req: Request, res: Response) => {
49   const {body} = req;
50   const {id} = req.params;
51   try{
52     const mensaje = await Mensaje.findByPk(id);
53     if(mensaje){
54       await mensaje.update(body);
55       res.json({
56         msg: 'El mensaje fue actualizado con exito'
57       })
58     } else {
59       res.status(404).json({
60         msg: 'No existe un mensaje con ese ${id}'
61       })
62     }
63   } catch (error){
64     console.log(error);
65     res.json({
66       msg: 'Ha ocurrido un error'
67     })
68   }
69 }
```