|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NOMBRE Y APELLIDOS:**  **Miguel Ramírez Ramos** | | | **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** | | | | | |
| **DESCRIPCIÓN**  1.- Se propone la creación de un servidor para trabajar con bases de datos SQL. Adjuntar Códigos  Crear Modelos, Controller y Routes de:   * Cursos * Usuarios * Mensajes * Tareas   La práctica se realizará de manera individual.  **MEDIOS PARA SU REALIZACIÓN**   * Equipo informático. * Aplicación Visual Code Studio instalada en el equipo. * Navegadores actualizados   Modelos   * Cursos   import { DataTypes } from 'sequelize';  import db from '../config/connectdb';  const Curso = db.define('Curso',{      nombre: {          type: DataTypes.STRING      },      descripcion: {          type: DataTypes.STRING      },      categoria: {          type: DataTypes.STRING      },      imagen: {          type: DataTypes.STRING      },      duracion: {          type: DataTypes.NUMBER      },      temas: {          type: DataTypes.STRING      }  }, {      createdAt: false,      updatedAt: false  });  export default Curso;   * Usuarios   import { DataTypes } from 'sequelize';  import db from '../config/connectdb';  const Usuario = db.define('Usuario',{      nombre: {          type: DataTypes.STRING,      },      apellidos: {          type: DataTypes.STRING,      },      Fnacimiento: {          type: DataTypes.STRING,      },      email: {          type: DataTypes.STRING,      },      telefono: {          type: DataTypes.STRING,      },      foto: {          type: DataTypes.STRING,      },      pass: {          type: DataTypes.STRING,      },      tipo: {          type: DataTypes.STRING,      },      fecha\_registro: {          type: DataTypes.STRING,      },      Anotaciones: {          type: DataTypes.STRING,      }  }, {      createdAt: false,      updatedAt: false,  });  export default Usuario   * Mensajes   import { DataTypes } from "sequelize";  import db from '../config/connectdb';  const Mensaje = db.define('Mensaje', {      remite: {          type: DataTypes.STRING,      },      mail: {          type: DataTypes.STRING,      },      asunto: {          type: DataTypes.STRING,      },      mensaje: {          type: DataTypes.STRING,      },      fecha\_envio: {          type: DataTypes.STRING,      },  }, {      createdAt: false,      updatedAt: false,  });  export default Mensaje   * Tareas   import { DataTypes } from 'sequelize';  import db from '../config/connectdb';  const Tarea = db.define('Tarea', {      nombre: {          type: DataTypes.STRING,      },      descripcion: {          type: DataTypes.STRING,      },      puntuacion: {          type: DataTypes.STRING,      },      estado: {          type: DataTypes.STRING,      },      importancia: {          type: DataTypes.STRING,      },    }, {      createdAt: false,      updatedAt: false,  });  export default Tarea  Controller   * Cursos   import {Request, Response} from 'express';  import Curso from '../models/CursoModel';  export const getCursos = async (req: Request, res: Response) =>{      const listCursos = await Curso.findAll();      res.json(listCursos);  }  export const getCurso = async (req: Request, res: Response) => {      const {id} = req.params;      const curso = await Curso.findByPk(id);      if (curso) {          res.json(curso)      } else {          res.status(404).json({              msg: `No existe un curso con ese ${id}`          })      }  }  export const deleteCurso = async (req: Request, res: Response) => {      const {id} = req.params;      const curso = await Curso.findByPk(id);      if (!curso) {          res.status(404).json({              msg: `No existe un curso con ese ${id}`          })      } else {          await curso.destroy();          res.json ({              msg: 'El curso fue eliminado con exito'          })      }  }  export const postCurso = async (req: Request, res: Response) => {      const {body} = req;      try{          await Curso.create(body);          res.json({              msg: '¡El curso fue agregado con exito!'          })      } catch (error){          console.log(error);          res.json({              msg: 'Ha ocurrido un error'          })      }  }  export const updateCurso = async (req: Request, res: Response) => {      const {body} = req;      const {id} = req.params;      try{          const curso = await Curso.findByPk(id);          if(curso){              await curso.update(body);              res.json({                  msg: 'El curso fue actualizado con exito'              })          } else {              res.status(404).json({                  msg: `No existe un curso con ese ${id}`              })          }      } catch (error){          console.log(error);          res.json({              msg: 'Ha ocurrido un error'          })      }  }   * Usuarios   import {Request, Response} from 'express';  import  Usuario from '../models/UsuarioModel';  export const getUsuarios = async(req: Request, res: Response) => {      const listUsuarios = await Usuario.findAll();      res.json(listUsuarios);  }  export const getUsuario = async (req: Request, res: Response) => {      const {id} = req.params;      const usuario = await Usuario.findByPk(id);      if(usuario){          res.json(usuario)      }else {          res.status(404).json({              msg: `No existe un usuario con ese ${id}`,          })      }  }  export const deleteUsuario= async (req: Request, res: Response) => {      const {id} = req.params;      const usuario = await Usuario.findByPk(id);      if(!usuario){          res.json(404).json({              msg: `No existe un usuario con ese ${id}`          })      }else {          await usuario.destroy();          res.json({              msg: 'El usuario fue eliminado con exito.',          })      }  }  export const postUsuario = async (req: Request, res: Response) => {      const {body} = req;      try{          await Usuario.create(body);          res.json({              msg: '¡El usuario fue agregado con exito!'          })      }catch (error){          console.log(error);          res.json({              msg: 'Ha ocurriod un error'          })      }  }  export const updateUsuario = async (req: Request, res: Response) =>{      const {body} = req;      const {id} = req.params;      try{          const usuario = await Usuario.findByPk(id);          if(usuario){              await usuario.update(body);              res.json({                  msg: 'El usuario fue actualizado con exito'              })          }else{              res.status(404).json({                  msg: `No existe un usuario con ese ${id}`              })          }      } catch (error){           console.log(error);           res.json({              msg: 'Ha ocurrido un error'           })      }  }   * Mensajes   import {Request, Response} from 'express';  import Mensaje from '../models/MensajeModel';  export const getMensajes =  async (req: Request, res: Response) => {      const listMensajes = await Mensaje.findAll();      res.json(listMensajes);  }  export const getMensaje = async (req: Request, res: Response) => {      const {id} = req.params;      const mensaje = await Mensaje.findByPk(id);      if(mensaje){          res.json(mensaje)      }else {          res.status(404).json({              msg: `No existe un mensaje con ese ${id}`          })      }  }  export const deleteMensaje = async (req: Request, res: Response) => {      const {id} = req.params;      const mensaje = await Mensaje.findByPk(id);      if(!mensaje){          res.json(404).json({              msg: `No existe un mensaje con ese ${id}`          })      }else {          await mensaje.destroy();          res.json({              msg: 'El mensjae fue eliminado con exito.',          })      }  }  export const postMensaje = async (req: Request, res: Response) => {      const {body} = req;      try{          await Mensaje.create(body);          res.json({              msg: '¡El mensaje fue agregado con exito!'          })      }catch (error){          console.log(error);          res.json({              msg: 'Ha ocurriod un error'          })      }  }  export const updateMensaje = async (req: Request, res: Response) =>{      const {body} = req;      const {id} = req.params;      try{          const mensaje = await Mensaje.findByPk(id);          if(mensaje){              await mensaje.update(body);              res.json({                  msg: 'El mensaje fue actualizado con exito'              })          }else{              res.status(404).json({                  msg: `No existe un mensaje con ese ${id}`              })          }      } catch (error){           console.log(error);           res.json({              msg: 'Ha ocurrido un error'           })      }  }   * Tareas   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 {          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.json(404).json({              msg: `No existe una tarea con ese ${id}`          })      }else {          await tarea.destroy();          res.json({              msg: 'La tarea fue eliminada con exito.',          })      }  }  export const postTarea = async (req: Request, res: Response) => {      const {body} = req;      try{          await Tarea.create(body);          res.json({              msg: '¡La tarea fue agregado con exito!'          })      }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: 'La tarea fue actualizado con exito'              })          }else{              res.status(404).json({                  msg: `No existe una tarea con ese ${id}`              })          }      } catch (error){           console.log(error);           res.json({              msg: 'Ha ocurrido un error'           })      }  }  Routes   * Cursos     import { Router } from 'express'  import { deleteCurso, getCurso, getCursos, postCurso, updateCurso } from '../controllers/CursoController'  const routercursos = Router();  routercursos.get('/cursos/', getCursos);  routercursos.get('/cursos/:id', getCurso);  routercursos.delete('/cursos/:id', deleteCurso);  routercursos.post('/cursos/', postCurso);  routercursos.put('/cursos/:id', updateCurso);  export default routercursos;   * Usuarios   import { Router } from 'express';  import {deleteUsuario, getUsuario, getUsuarios, postUsuario, updateUsuario} from '../controllers/UsuarioController';  const routerusuarios = Router();  routerusuarios.get('/usuarios/', getUsuarios);  routerusuarios.get('/usuarios/:id', getUsuario);  routerusuarios.delete('/usuarios/:id', deleteUsuario);  routerusuarios.post('/usuarios/', postUsuario);  routerusuarios.put('/usuarios/:id', updateUsuario);  export default routerusuarios;   * Mensajes   import { Router } from 'express';  import { deleteMensaje, getMensaje, getMensajes, postMensaje, updateMensaje} from '../controllers/MensajeController';  const routermensajes = Router();  routermensajes.get('/mensajes/', getMensajes);  routermensajes.get('/mensajes/:id', getMensaje);  routermensajes.delete('/mensajes/:id', deleteMensaje);  routermensajes.post('/mensajes/', postMensaje);  routermensajes.put('/mensajes/:id', updateMensaje);  export default routermensajes;   * Tareas | | | | | |
|  | | | | | |
|  | | | | | |