



Universidad

Universidad Autónoma de Sinaloa

Carrera

Lic. en Informática

Materia

Desarrollo web del lado del servidor

Actividad

Actividad en línea - CRUD

Grupo

2-3

Fecha

08/06/2025 Culiacán, Sinaloa

Maestro

José Manuel Cazarez Alderete

Alumna

Núñez Sarabia Jessica Anahí



ÍNDICE

03

APP.JS

04

BASEDATOS.JS

05

CRUDRUTAS.JS

06

MODELOS

APP.JS

Se encarga de inicializar Express, configurar el middleware para recibir JSON y definir la conexión con la base de datos. Además, sincroniza Sequelize con la base de datos antes de arrancar el servidor.

Importación de módulos

- express: Framework para manejar el servidor.
- sequelize: Instancia de conexión a la base de datos.
- rutas: Archivo que contiene las rutas CRUD de la API.

Configuración de Express

`app.use(express.json())`: Permite manejar datos en formato JSON, asegurando que las solicitudes con POST y PUT funcionen correctamente.

Carga de rutas

`app.use('/api', rutas)`: Define que todas las rutas estarán bajo el prefijo /api, evitando conflictos con otras posibles rutas en la aplicación.

Conexión y sincronización con la base de datos

`app.use('/api', rutas)`: Define que todas las rutas estarán bajo el prefijo /api, evitando conflictos con otras posibles rutas en la aplicación.

Inicio del servidor

`app.listen(PORT, () => console.log(...))`: Indica que el servidor está corriendo en el puerto 3000, lo que facilita el acceso local.

BASEDATOS.JS

Este archivo define la conexión a la base de datos usando Sequelize y el dialecto MySQL. Tenerlo separado facilita la administración de credenciales y configuración de la conexión.

Instalación de Sequelize

Se trae Sequelize desde el paquete para inicializar la conexión.

Configuración de la conexión

- Define la base de datos, el usuario y la contraseña en el objeto Sequelize().
- Se establece host: 'localhost' y dialect: 'mysql', indicando que la base de datos corre en el mismo servidor y usa MySQL.

Exportación de la instancia

Se exporta sequelize para que pueda ser utilizada en los modelos y en la aplicación.

CRUDRUTAS.JS

Este archivo crea rutas CRUD automáticamente mediante la función `crearCRUD()`, evitando código repetitivo y facilitando la escalabilidad del proyecto.

Uso de `express.Router()`

- Separar las rutas en un archivo independiente mejora la organización y mantiene `app.js` limpio.
- `router.get()`, `router.post()`, `router.put()`, `router.delete()` manejan las solicitudes adecuadas para cada operación CRUD.

Generación dinámica de CRUD

- La función `crearCRUD(modelo, nombre)` evita repetir código, generando automáticamente las rutas para cada modelo.
- Permite escalar la API fácilmente, solo agregando nuevos modelos sin tener que escribir cada ruta manualmente.

Manejo de errores

- Se usa `try-catch` en cada endpoint para capturar errores y enviar respuestas JSON con mensajes adecuados.
- Se incluyen respuestas con códigos HTTP apropiados (404 si el dato no existe, 500 en caso de error de servidor, 201 para creación exitosa).

Uso de `findByPk()` y `findAll()`

- `findByPk(id)`: Busca registros por ID, asegurando una consulta eficiente.
- `findAll()`: Devuelve todos los registros disponibles en la tabla del modelo.

MODELOS

Este archivo crea rutas CRUD automáticamente mediante la función `crearCRUD()`, evitando código repetitivo y facilitando la escalabilidad del proyecto.

Cliente (`mCliente.js`)

correo: Validado con `isEmail` para asegurar un formato correcto.

telefono: Almacena el número de contacto, útil para notificaciones o servicio al cliente.

direccion: Permite futuras funcionalidades como envíos o segmentación de clientes.

Proveedor (`mProveedor.js`)

nombre: Identifica el proveedor.

direccion: Esencial si se necesita gestionar logística o envíos.

Artículo (`mArticulo.js`)

descripcion: Define el producto.

precio: Controla el costo y permite cálculos de descuentos.

existencia: Ayuda en la gestión de inventarios.

Empleado (`mEmpleado.js`)

fecha_nacimiento: Útil para cálculos de edad y beneficios.

suelo: Permite administrar pagos y finanzas de la empresa.