

1. Conexión a SQLite con Sequelize

Se emplea Sequelize para conectarse a archivos de base de datos SQLite. Este ORM facilita el acceso a los datos sin escribir sentencias SQL directamente. Se configura indicando el tipo de base de datos (dialect: 'sqlite') y se proporciona la ruta del archivo donde se almacenará la información.

2. Estructuración de modelos de datos

Los modelos definen cómo lucen las tablas dentro de la base de datos. Para cada tabla, se declara una estructura en Sequelize usando DataTypes para establecer el tipo de dato de cada campo. También es posible deshabilitar la inclusión automática de las columnas de fecha con timestamps: false.

Modelos incluidos:

- Artículos: Define atributos como identificador, texto descriptivo, costo y cantidad disponible.
- Proveedores: Establece los campos id, nombre y dirección física.
- Empleados: Incluye identificador, nombre, teléfono, fecha de nacimiento y salario.
- Clientes: Abarca nombre completo, correo electrónico, teléfono y dirección.

3. Desarrollo de API con Express.js para operaciones CRUD

Mediante Express, se construyen servicios backend que permiten realizar acciones básicas sobre los datos: crear, consultar, editar y borrar registros. Se utiliza body-parser para manejar el contenido en formato JSON que se recibe en las peticiones.

Cada conjunto de datos tiene un archivo asociado que ofrece rutas para:

- POST: ingresar nuevos datos.
- GET: listar toda la información almacenada.
- PUT: modificar registros específicos según su ID.
- DELETE: eliminar entradas utilizando el identificador único.

Las operaciones interactúan con la base de datos de manera asíncrona y responden con los resultados en formato JSON.

4. Conexión directa con SQLite3

Algunos módulos del proyecto emplean directamente el módulo sqlite3 para conectarse a la base de datos. Se crea una conexión mediante new sqlite3.Database() y se exporta para reutilización. Este enfoque es más directo que el uso de Sequelize, pero requiere un manejo manual de las consultas SQL.

5. Aplicación para convertir divisas

Se desarrolla un backend Express que permite trabajar con tasas de cambio. Se crean dos rutas:

- Una ruta GET devuelve una lista de monedas disponibles.
- Una ruta POST procesa una conversión monetaria recibiendo como entrada el monto, la divisa de origen y la de destino.

La base contiene las tasas de cambio entre distintas monedas, y la lógica de conversión se basa en esos datos. Se añaden validaciones para asegurar que se ingresen correctamente los datos necesarios.