

Módulo: Artículos

Archivos:

articulos.js: Modelo Sequelize para la tabla articulos.

conexion.js: Configuración de conexión a base de datos.

articulos.db: Base de datos SQLite.

app.js: Script para insertar un artículo.

Estructura del modelo:

id: INTEGER (clave primaria)

descripcion: STRING

precio: DOUBLE

existencia: INTEGER

¿Qué hace app.js?

Conecta a la base de datos.

Crea la tabla si no existe.

Inserta un artículo con `create({ ... })`.

⚡ No incluye API (sin Express). Solo hace una inserción.

Módulo: Clientes

Archivos:

clientes.js: Modelo Sequelize.

conexion.js: Conexión a SQLite.

clientes.sqlite: Base de datos.

app.js: Servidor Express + API CRUD.

Estructura del modelo:

id: INTEGER (clave primaria)

nombre: STRING

correo: STRING

telefono: STRING

direccion: STRING

API implementada en app.js (similar al de proveedores):

Ruta	Método	Función
/clientes	POST	Crear cliente
/clientes	GET	Obtener todos
/clientes/:id	PUT	Actualizar por ID
/clientes/:id	DELETE	Eliminar por ID

◆ Funcionalidad completa CRUD con Express y Sequelize.

Módulo: Empleados

Archivos:

empleados.js: Modelo Sequelize.

conexion.js: Conexión a SQLite.

empleados.sqlite: Base de datos.

app.js: API Express para CRUD.

Estructura del modelo:

id: INTEGER (clave primaria)

nombre: STRING

telefono: STRING

fecha_nacimiento: STRING

sueldo: DOUBLE

API Express en app.js (se espera que sea igual que proveedores y clientes):

Ruta	Método	Función
/empleados	POST	Crear empleado
/empleados	GET	Obtener todos
/empleados/:id	PUT	Actualizar por ID
/empleados/:id	DELETE	Eliminar por ID

Funciona como un módulo independiente para gestionar empleados.

Módulo: Proveedores (escrito como "proveedores")

Archivos:

proveedores.js: Modelo Sequelize.

conexion.js: Conexión a SQLite.

proveedores.sqlite: Base de datos.

app.js: API REST completa con Express.

Estructura del modelo:

id: INTEGER (clave primaria)

nombre: STRING

direccion: STRING

Código del servidor en app.js:

Crea un servidor Express (puerto 3003).

Usa body-parser para interpretar JSON.

Define rutas para CRUD:

Ruta	Método	Función
/proveedores	POST	Crear proveedor
/proveedores	GET	Leer todos
/proveedores/:id	DELETE	Eliminar proveedor
/proveedores/:id	PUT	Actualizar proveedor

API REST completa y funcional.

Resumen General del Proyecto

Módulo	API CRUD	Base de datos	Observaciones
Artículos	✗	✓ .db	Solo inserta un registro inicial
Cientes	✓	✓ .sqlite	API REST bien estructurada
Empleados	✓	✓ .sqlite	API REST esperada (estructura similar)
Proveedores	✓	✓ .sqlite	API REST completa, error en el nombre