

### **conexion empleados.js**

Establece la conexión con la base de datos SQLite llamada empleados.sqlite utilizando Sequelize.

Permite que el backend interactúe con la base de datos sin necesidad de usar consultas SQL manuales.

#### **Código:**

javascript

CopiarEditar

```
const Sequelize = require('sequelize');

const sequelize = new Sequelize({
  dialect: 'sqlite',
  storage: './empleados.sqlite' // archivo de base de datos
});

module.exports = sequelize;
```

### **conexion articulos.js**

Define la conexión con la base de datos articulos.sqlite también usando Sequelize.

#### **Código:**

javascript

CopiarEditar

```
const Sequelize = require('sequelize');

const sequelize = new Sequelize({
  dialect: 'sqlite',
  storage: './articulos.sqlite' // archivo de base de datos
});
```

```
module.exports = sequelize;
```

## **CRUD empleados.js**

Crea una API RESTful con Express para manejar empleados. Permite:

- Crear nuevos empleados (POST)
- Leer todos los empleados (GET)
- Actualizar un empleado (PUT)
- Eliminar un empleado (DELETE)

### **Código:**

javascript

CopiarEditar

```
const express = require('express');
const bodyParser = require('body-parser');
const empleados = require('./empleados');
const puerto = 3000;

const app = express();
app.use(bodyParser.json());

app.listen(puerto, () => {
  console.log('servicio iniciado');
});

// Crear un nuevo empleado
app.post('/empleados', async (req, res) => {
  const { nombre, telefono, nacimiento, sueldo } = req.body;
  const data = await empleados.create({ nombre, telefono, nacimiento, sueldo });
  res.send(data);
});
```

```
});
```

```
// Obtener todos los empleados
```

```
app.get('/empleados', async (req, res) => {  
  const data = await empleados.findAll();  
  res.send(data);  
});
```

```
// Actualizar un empleado
```

```
app.put('/empleados/:id', async (req, res) => {  
  const { nombre, telefono, nacimiento, sueldo } = req.body;  
  const { id } = req.params;  
  const data = await empleados.update({ nombre, telefono, nacimiento, sueldo }, {  
    where: { id }  
  });  
  res.send(data);  
});
```

```
// Eliminar un empleado
```

```
app.delete('/empleados/:id', async (req, res) => {  
  const { id } = req.params;  
  const data = await empleados.destroy({ where: { id } });  
  res.send(data);  
});
```

## **empleados.js**

Define el modelo Sequelize para la tabla empleados. Especifica los campos y tipos de datos.

### **Código:**

javascript

CopiarEditar

```
const Sequelize = require('sequelize');
const db = require('./conexion empleados');

const empleados = db.define('empleados', {
  nombre: Sequelize.STRING,
  telefono: Sequelize.STRING,
  nacimiento: Sequelize.STRING,
  sueldo: Sequelize.FLOAT
});

db.sync();
module.exports = empleados;
```

## **CRUD articulos.js**

Implementa la lógica CRUD para los artículos, similar al CRUD de empleados.

### **Código:**

javascript

CopiarEditar

```
const express = require('express');
const bodyParser = require('body-parser');
const articulos = require('./articulos');
const puerto = 3001;
```

```
const app = express();  
app.use(bodyParser.json());
```

```
app.listen(puerto, () => {  
  console.log('servicio iniciado');  
});
```

```
app.post('/articulos', async (req, res) => {  
  const { nombre, precio, existencia } = req.body;  
  const data = await articulos.create({ nombre, precio, existencia });  
  res.send(data);  
});
```

```
app.get('/articulos', async (req, res) => {  
  const data = await articulos.findAll();  
  res.send(data);  
});
```

```
app.put('/articulos/:id', async (req, res) => {  
  const { nombre, precio, existencia } = req.body;  
  const { id } = req.params;  
  const data = await articulos.update({ nombre, precio, existencia }, {  
    where: { id }  
  });  
  res.send(data);  
});
```

```
app.delete('/articulos/:id', async (req, res) => {
```

```
const { id } = req.params;

const data = await articulos.destroy({ where: { id } });

res.send(data);

});
```

### **proveedores.js**

Define el modelo Sequelize de la tabla proveedores.

#### **Código:**

javascript

CopiarEditar

```
const Sequelize = require('sequelize');
const db = require('./conexion empleados');

const proveedores = db.define('proveedores', {
  nombre: Sequelize.STRING,
  telefono: Sequelize.STRING,
  correo: Sequelize.STRING,
  empresa: Sequelize.STRING
});

db.sync();

module.exports = proveedores;
```

## CRUD clientes.js

Crea una API para manejar operaciones CRUD con clientes.

### Código:

javascript

CopiarEditar

```
const express = require('express');
const bodyParser = require('body-parser');
const clientes = require('./clientes');
const puerto = 3003;

const app = express();
app.use(bodyParser.json());

app.listen(puerto, () => {
  console.log('servicio iniciado');
});

app.post('/clientes', async (req, res) => {
  const { nombre, telefono, correo } = req.body;
  const data = await clientes.create({ nombre, telefono, correo });
  res.send(data);
});

app.get('/clientes', async (req, res) => {
  const data = await clientes.findAll();
  res.send(data);
});
```

```
app.put('/clientes/:id', async (req, res) => {  
  const { nombre, telefono, correo } = req.body;  
  const { id } = req.params;  
  const data = await clientes.update({ nombre, telefono, correo }, {  
    where: { id }  
  });  
  res.send(data);  
});
```

```
app.delete('/clientes/:id', async (req, res) => {  
  const { id } = req.params;  
  const data = await clientes.destroy({ where: { id } });  
  res.send(data);  
});
```