

# Proyecto Node.js - CRUD Clientes con JWT

## Descripción General

Este proyecto es una API REST desarrollada con Node.js, Express y Sequelize, que permite gestionar clientes mediante un CRUD completo. Todas las rutas están protegidas mediante autenticación basada en tokens usando JWT.

## Estructura del Proyecto

```
cliente-api/  
  ??? config/  
  ?   ??? database.js  
  ??? controllers/  
  ?   ??? clienteController.js  
  ??? middleware/  
  ?   ??? auth.js  
  ??? models/  
  ?   ??? cliente.js  
  ?   ??? usuario.js  
  ??? routes/  
  ?   ??? cliente.routes.js  
  ?   ??? auth.routes.js  
  ??? index.js  
  ??? package.json
```

## Dependencias Instaladas

- express
- sequelize
- mysql2
- bcrypt
- jsonwebtoken
- nodemon (desarrollo)

# Proyecto Node.js - CRUD Clientes con JWT

## Base de Datos

La conexión con MySQL se realiza mediante Sequelize, definida en config/database.js. Se sincroniza con `sequelize.sync({ force: false })`.

## Modelo de Cliente

Campos:

- nombre (requerido)
- correo (requerido)
- telefono
- direccion

## Modelo de Usuario

Campos:

- username (único)
- password (encriptado con bcrypt)

## Controlador del Cliente

Incluye funciones para crear, obtener, actualizar y eliminar clientes. Utiliza Sequelize para interactuar con la base de datos.

## Middleware de Autenticación JWT

Verifica la validez del token enviado en los headers. Protege todas las rutas del CRUD.

## Rutas de Clientes (Protegidas)

/clientes (POST, GET, PUT, DELETE)

Requieren un token JWT válido en el header Authorization.

## Rutas de Autenticación

# Proyecto Node.js - CRUD Clientes con JWT

/auth/register ? Registrar nuevo usuario

/auth/login ? Autenticación y generación de token JWT

## Inicialización del Servidor

El archivo index.js configura Express, importa rutas y sincroniza la base de datos. El servidor se inicia en el puerto 3000.

## Pruebas

1. Registrar usuario en /auth/register
2. Iniciar sesión en /auth/login y obtener token
3. Usar token en el header Authorization para acceder a /clientes

## Resumen

Este proyecto implementa un CRUD seguro para clientes usando Node.js, Sequelize y JWT, proporcionando una base sólida para aplicaciones empresariales con autenticación robusta.