Universidad

Universidad Autónoma de Sinaloa

Carrera

Lic. en Informática

Materia

Desarrollo web del lado del servidor

Actividad

Seguridad basada en tokens (JWT)

Grupo

2-3

Fecha

08/06/2025 Culiacán, Sinaloa

Maestro

José Manuel Cazarez Alderete

Alumna

Núñez Sarabia Jessica Anahí

ÍNDICE

JUSTIFICACIÓN

Seguridad con JWT (auth.js y middlewareJWT.js)

- Autenticación segura: Se usa jsonwebtoken para generar y verificar tokens de usuario.
- Protección de rutas: Todas las rutas CRUD requieren un token válido para evitar accesos no autorizados.
- Expiración del token: Los tokens expiran en 1 hora, mejorando la seguridad y reduciendo riesgos de accesos prolongados sin control.
- Middleware JWT (verificarToken): Separa la lógica de autenticación y facilita la reutilización en distintas rutas.

app.js - Configuración del servidor

- Modularidad: Mantiene la separación de responsabilidad entre las rutas CRUD y las de autenticación.
- Carga ordenada de rutas: Primero rutasAuth (login), luego rutasCRUD (protegidas), asegurando control de acceso lógico.
- Sincronización de base de datos: sequelize.sync() garantiza que las tablas estén listas antes de iniciar el servidor.

crudRutas.js - Generación dinámica de rutas

- Estructura eficiente: crearCRUD() evita duplicar código al generar rutas para cualquier modelo.
- Protección con verificarToken: Cada operación requiere autenticación, evitando accesos sin permisos.
- Uso de async/await: Asegura operaciones asincrónicas limpias sin bloqueos en la ejecución.

Modelos y estructura de datos (Cliente.js)

- Validaciones esenciales: isEmail en correo previene inserciones inválidas.
- Definición clara: Cada modelo sigue la estructura requerida (id, nombre, correo, etc.).
- Optimización: Uso de claves primarias (id) y autoIncrement asegura gestión eficiente de registros.