

**Sistema Backend seguro**

**Ingeniería en Informática**

Programación Back End

**Alejandro Segundo Carez Armijo**

Sede Virtual, 16 de junio de 2025

# 1.-Introducción

Este documento describe la arquitectura, componentes, flujo de trabajo y medidas de seguridad implementadas en un sistema backend desarrollado con Node.js y Express.js. El objetivo es proveer una API Restful segura, con autenticación basada en JWT, conexión a una base de datos MongoDB y buenas prácticas de seguridad para proteger los datos sensibles.

# 2.-Estructura



# 3.-Tecnologías utilizadas

|  |  |
| --- | --- |
| **Tecnología** | **Uso** |
| Node.js | Entorno de ejecución Javascript |
| Express.js | Framework backend |
| MongoDB | Base de datos NoSQL |
| Mongoose | ODM para MongoDB |
| bcryptjs | Encriptación de contraseñas |
| jsonwebtoken | Generación y verificación de JWT |
| dotenv | Manejo de variables de entorno |
| Postman/Thunder | Pruebas de endpoints |

# 4.-Seguridad implementada

**-Cifrado de contraseñas:** las contraseñas se cifran con bcryptjs antes de ser almacenadas en MongoDB

**-JWT:** autenticación basada en tokens firmados, con expiración configurable.

**-Middleware de autenticación:** valida el token en rutas protegidas.

**-.env:** variables sensibles almacenadas fuera del código fuente.

**-Validación:** se valida el input del usuario en endpoints críticos.

# 5.-Endpoint principales

POST /api/auth/register

-Registra un usuario nuevo.

-Usuario creado con token.

POST /apit/auth/login

-inicia sesión

-Devuelve Token JWT

GET /api/protected

-Ruta protegida, accesible sólo con token válido

# 6.-Flujo de autenticación

-El usuario se registra – contraseña cifrada y guardada.

-Al iniciar sesión – se verifica la contraseña y se genera un JWT.

-El cliente almacena el token y lo envía en cada solicitud.

-El middleware verifica el token antes de permitir el acceso.

# 7.-Pruebas

Se utilizaron herramientas como Postman y Thunder Client:

-Registro y login con respuestas esperadas.

-Acceso denegado sin token o token inválido.

# 8.-Mejora y escalabilidad

Rate limiting: agregar protección contra ataques de fuerza bruta.

Roles y permisos: control de acceso más granular.

Helmet y otros middleware de seguridad.