

# PE-2.3: Seguridad Avanzada en APIs

Laboratorio de Programación de Microservicios Basados en Datos

Estudiante: JHANDRY DAVID BECERRA LIMA

---

## Configuración Auth0

Domain: No proporcionado

Client ID: No proporcionado

## Checkpoints Completados

- ' 1.1 Env preparado y copiado
- ' 1.2 package.json type:module
- ' 1.3 tsconfig.json ESNext
- ' 1.4 Imports con extensión .js
- ' 2.1 Deps de seguridad instaladas
- ' 2.2 Hardening (Helmet/Rate-Limit) implementado
- ' 2.3 Configuración .env, Cookies y JWT
- ' 3.1 Decorador authenticate implementado
- ' 3.2 Auth0 y Rutas registradas

## Evidencias de Pruebas

Test 401 Unauthorized:

```
{  
  "statusCode": 401,  
  "error": "Unauthorized",  
  "message": "Token JWT inválido o no proporcionado. Por favor, autentícate en /login"  
}
```

Test 200 OK con JWT:

```
{  
  "result": 8,  
  "operation": "add"  
}
```

## Análisis y Conclusiones

Usar Auth0 para el inicio de sesión es importante porque no guardamos contraseñas en nuestra base de datos, dejamos que ellos manejen esa parte complicada. Así es más difícil que se metan a nuestras cuentas ya que no tenemos esa información.

Los JWT funcionan como pases digitales para cada usuario al iniciar sesión. Con ese pase solo acceden a lo que les toca. Si alguien intenta entrar sin un pase válido, la API no responde y no lo deja pasar, protegiendo la información importante.

También agregamos Helmet, una capa extra que tapa huecos de seguridad comunes y evita que metan código malicioso o engañen con páginas falsas.

Todo esto hace la aplicación más segura y fácil de mantener

# Evidencia de Login (Token)

Impresión con formato estilístico   
{"message": "Autenticación exitosa. Usa este token en el Header Authorization.", "jwt\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJleUpoYkdjau9pIiwiAiF0IjoxNzY1OTgzMzkwfQ.DF68-W-2npXBLH\_qK-EzKPG9T4kZ3oPE-g2GHFIkAyM"}

---

Generado el 17/12/2025 - PE-2.3 v2.0

Universidad Internacional del Ecuador