

Examen Progreso 2

Mateo Encalada

Caso Plataforma de Servicios Estudiantiles

La universidad desea construir una nueva **plataforma integrada** para atender solicitudes académicas de los estudiantes (solicitudes de certificados, legalizaciones, homologaciones y equivalencias). Actualmente, existen **3 sistemas independientes**:

Descripción

SolicitudService es un microservicio REST que permite a estudiantes realizar y consultar solicitudes académicas (certificados, legalizaciones, homologaciones y equivalencias). El servicio integra tres sistemas independientes:

- Sistema de Seguridad y Roles (validación de JWT).
- Sistema Académico (REST API interno).
- Sistema de Certificación (servicio SOAP externo).

Requisitos previos

- **Node.js** v14 o superior.
- **npm** o **yarn** para gestión de paquetes.
- Servicio SOAP en ejecución (puede ser un mock o servidor real).

SOAP_WSDL_URL=http://localhost:5000/CertificationService?wsdl

Validación de JWT

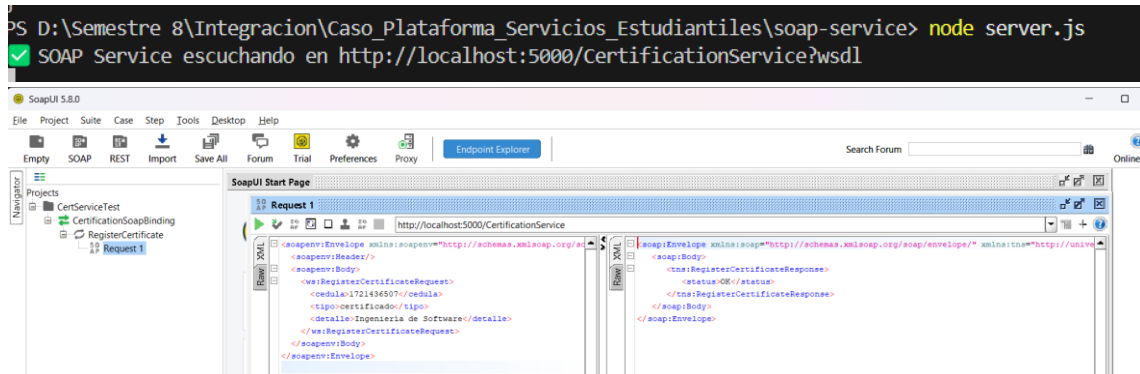
Se utiliza el middleware express-jwt para validar el token JWT recibido en el encabezado Authorization: Bearer <token>. En caso de token inválido o expirado,

todas las rutas retornan un error 401 Unauthorized con mensaje descriptivo.

```
undefined
> const token = jwt.sign(
> const jwt = require('jsonwebtoken');
> const jwt = require('jsonwebtoken');
> const jwt = require('jsonwebtoken');
undefined
> const token = jwt.sign(
...   { iss: 'estudiante', sub: 'estudiante123' },
...   'mi_clave_super_secreta',
...   { expiresIn: '24h' }
... );
undefined
> console.log(token);
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJlc3R1ZGhbnRlIiw7d8Ya7LCYpH40gH8sdBy9QBNr_BVQf4K_g
undefined
> 
```

Integración con el sistema SOAP

El servicio consume el método RegisterCertificate del sistema de certificación SOAP. La URL del WSDL se obtiene desde la variable de entorno SOAP_WSDL_URL. La integración incluye manejo de reintentos (hasta dos intentos en caso de error de conexión) y propagación de errores en caso de falla definitiva.



Endpoints

POST /solicitudes

- **Descripción:** Crea una nueva solicitud académica.
- **Headers:**
 - Content-Type: application/json
 - Authorization: Bearer <token>

- **Body (ejemplo):**
- {
- "tipo": "certificado",
- "cedula": "1721436507",
- "detalle": {
- "carrera": "Ingeniería de Software",
- "semestre": 8
- }
- }
- **Flujo interno:**
 1. Validación del JWT.
 2. Validación de existencia del estudiante en el Sistema Académico (REST).
 3. Registro de la certificación en el sistema SOAP (con retry).
 4. Retorno de un objeto JSON con id y estado: "procesado".
- **Respuesta (201 Created):**
- {
- "id": "<identificador_unico>",
- "estado": "procesado"
- }

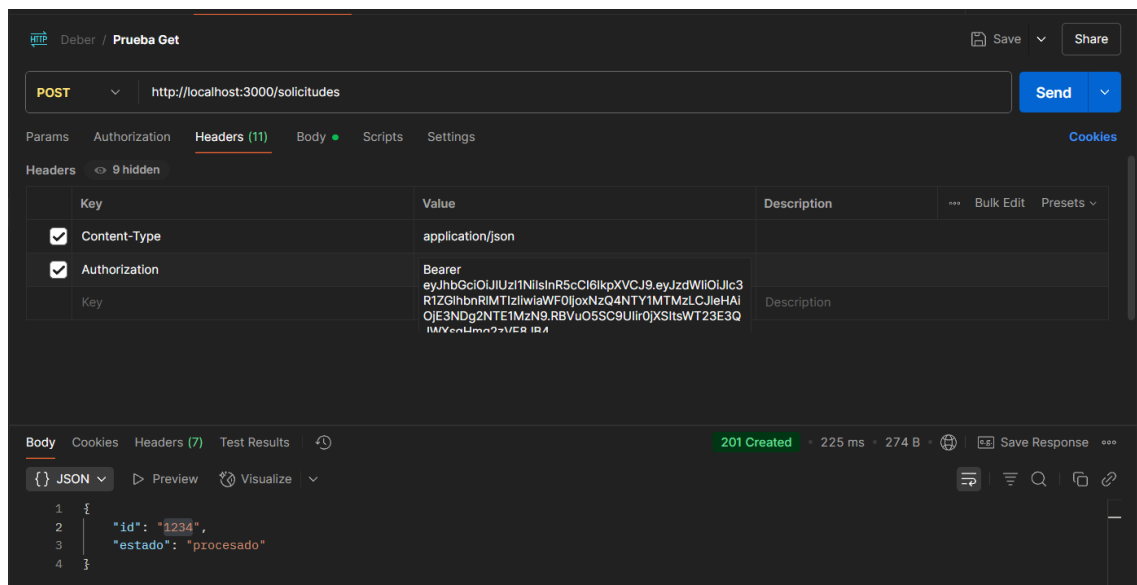
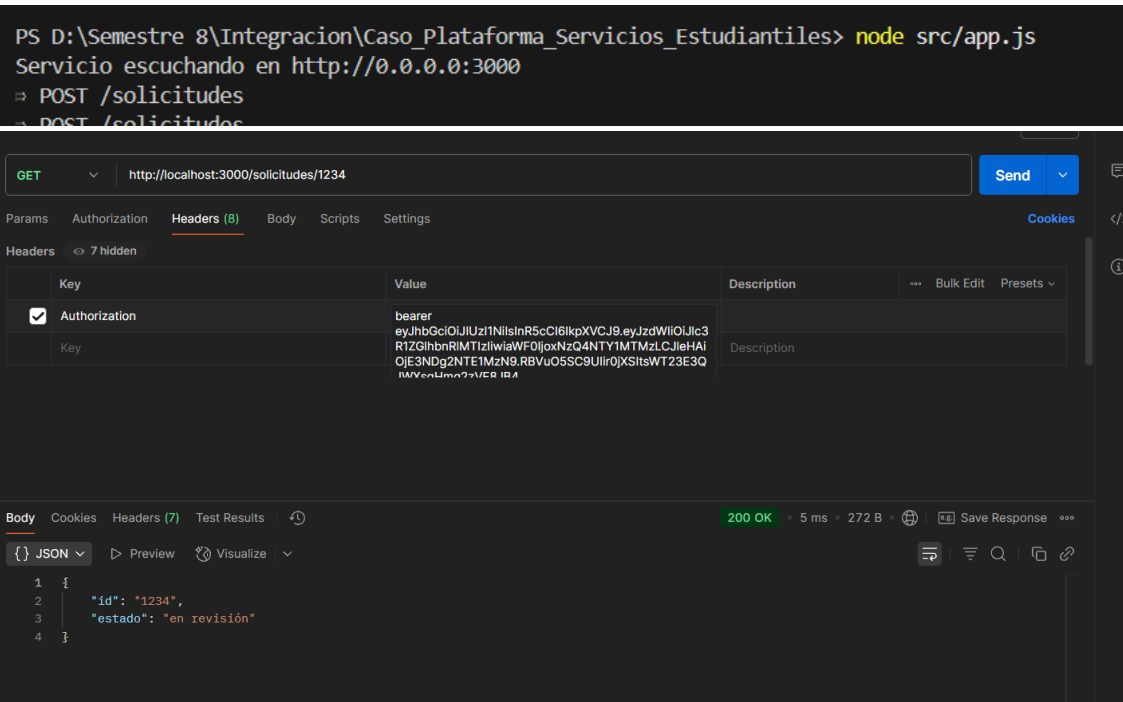
GET /solicitudes/{id}

- **Descripción:** Recupera el estado de una solicitud existente.
- **Headers:**
 - Authorization: Bearer <token>
- **Parámetros de ruta:**
 - id — Identificador de la solicitud.
- **Respuesta (200 OK):**
- {

- "id": "<identificador_unico>",
- "estado": "en revisión"

```
}

```



Kong

| | | | | | | | | | | | | |
|--|--|--|-----------------------------|--------------|--------------------------|---------------------------|------------------------------------|-------|----------------|--|--|--|
| | | | caso_plataforma_servicios - | - | - | 2.26% | 23 seconds ago | | | | | |
| | | | kong-1 | 5dcceb723dd3 | kong:3.0 | 8000:8000 | Show all ports (4) | 2.26% | 23 seconds ago | | | |

Kong funciona al momento de levantarse pero por ciertas razones nunca se logro conectar de todo al microservicio cumplienod su función como api gateway

Pruebas de resiliencia

Para validar la capacidad de recuperación ante fallos del sistema SOAP, se realizaron las siguientes pruebas:

Retry ante fallo de conexión

- Se detuvo el servicio SOAP.
- Se envió un POST /solicitudes al microservicio REST.
- En los logs se observaron dos intentos de reintento, marcados como:
Retry #1 al SOAP debido a: connect ECONNREFUSED ...

- Retry #2 al SOAP debido a: connect ECONNREFUSED

Apertura del Circuit Breaker

- Tras el segundo reintento fallido, el breaker cambió al estado *open* y emitió:

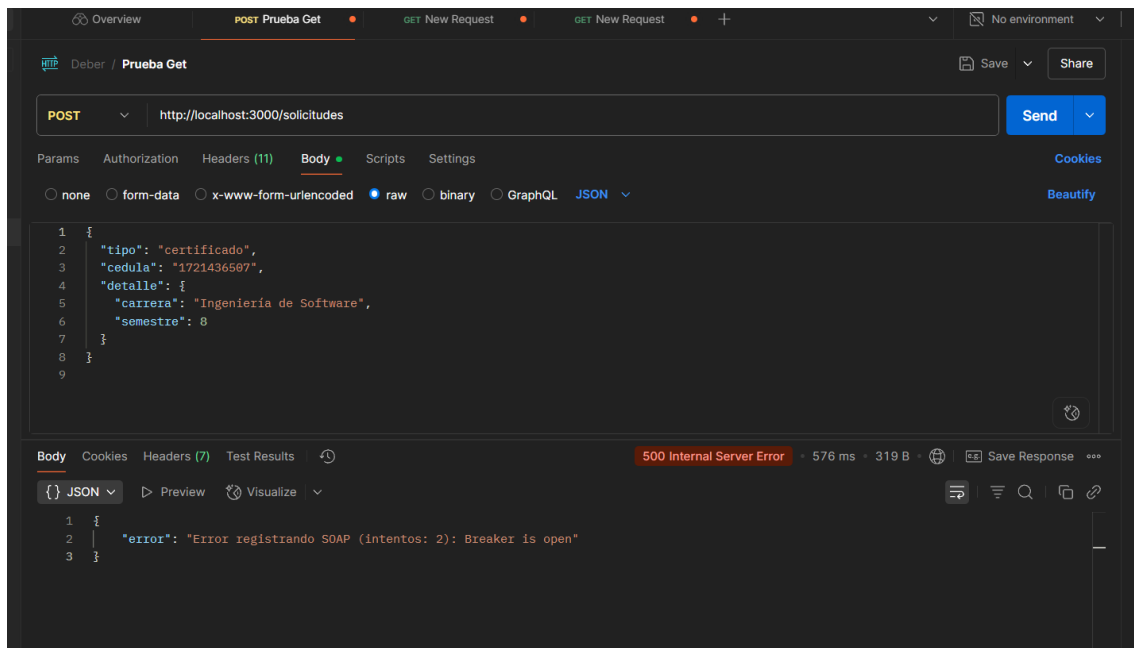
The image shows a REST client interface with a '500 Internal Server Error' status. The JSON body of the response is:

```
{  "error": "Error registrando SOAP (intentos: 2): Breaker is open"}
```

Below the response, a log viewer displays the following messages:

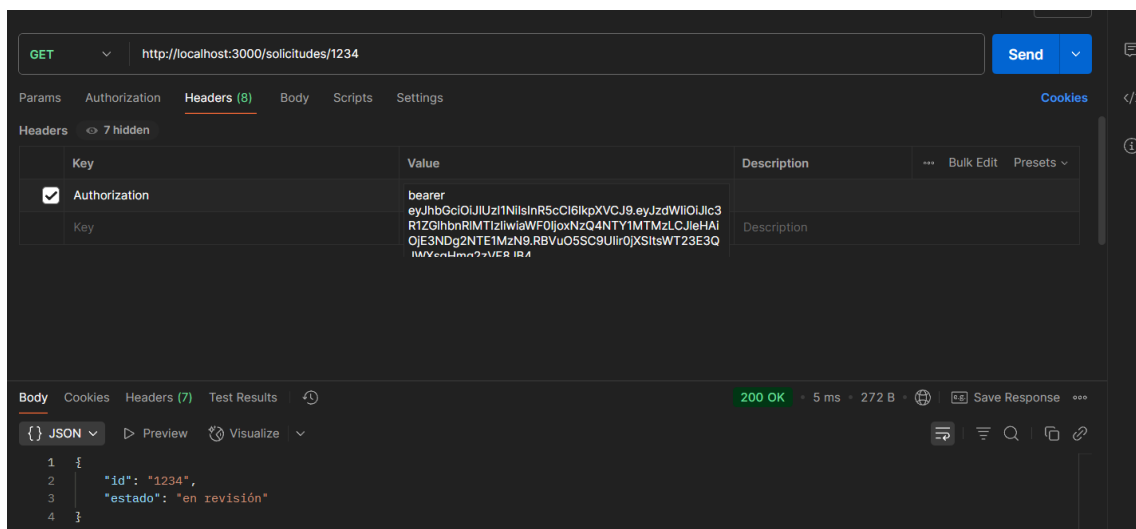
```
at async exports.createSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\controllers\solicitudcontroller.js:8:20)
→ POST /solicitudes
● Breaker OPEN: no más llamadas al SOAP
Retry #1 al SOAP debido a:
▲Breaker REJECT: Breaker is open
Error: Error registrando SOAP (intentos: 2): Breaker is open
    at crearSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\services\solicitudService.js:42:15)
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async exports.createSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\controllers\solicitudcontroller.js:8:20)
→ POST /solicitudes
▲Breaker REJECT: Breaker is open
Retry #1 al SOAP debido a: Breaker is open
▲Breaker REJECT: Breaker is open
Error: Error registrando SOAP (intentos: 2): Breaker is open
    at crearSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\services\solicitudService.js:42:15)
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async exports.createSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\controllers\solicitudcontroller.js:8:20)
→ POST /solicitudes
▲Breaker REJECT: Breaker is open
Retry #1 al SOAP debido a: Breaker is open
▲Breaker REJECT: Breaker is open
Error: Error registrando SOAP (intentos: 2): Breaker is open
    at crearSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\services\solicitudService.js:42:15)
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async exports.createSolicitud (D:\Semestre 8\Integracion\Caso_Plataforma_Servicios_Estudiantiles\src\controllers\solicitudcontroller.js:8:20)
→ POST /solicitudes
```

El cliente REST recibió un 500 Internal Server Error con el cuerpo



Recuperación del Circuit Breaker

- Se volvió a arrancar el servicio SOAP o se esperó a que expirara el `resetTimeout`.
- Al enviar otro POST `/solicitudes`, el breaker pasó a estado *half-open*, probó una llamada de prueba y, al resultar exitosa, cerró el circuito



Monitoreo y trazabilidad

Para garantizar visibilidad y diagnóstico en producción, integraría Prometheus para recolectar métricas de los microservicios y del API Gateway, Grafana para construir dashboards en tiempo real y configurar alertas, y el ELK Stack (Elasticsearch, Logstash, Kibana) para centralizar y consultar logs estructurados. Las métricas clave abarcarían latencia (p50, p95, p99) y throughput de los

endpoints POST y GET de /solicitudes, tasa de errores 4xx/5xx, número y duración de reintentos al sistema SOAP, así como eventos de apertura y cierre del circuit breaker, además de estadísticas de recursos (CPU, memoria, event loop lag de Node.js). Para trazabilidad, cada petición llevaría un identificador único (X-Request-ID) propagado desde el Gateway a lo largo de todo el flujo, instrumentando puntos críticos como la validación JWT, llamadas al sistema académico, invocaciones SOAP (con retries) y cambios de estado del breaker, lo que permitiría reconstruir el recorrido completo de cada solicitud y localizar con precisión cuellos de botella o fallos.

Buenas prácticas y observabilidad

- Registro de cada petición HTTP en consola (método y ruta).
 - Manejo centralizado de errores de validación JWT.
 - Propagación de identificadores de solicitud para trazabilidad.
 - Uso de variables de entorno para configuraciones sensibles.
-