

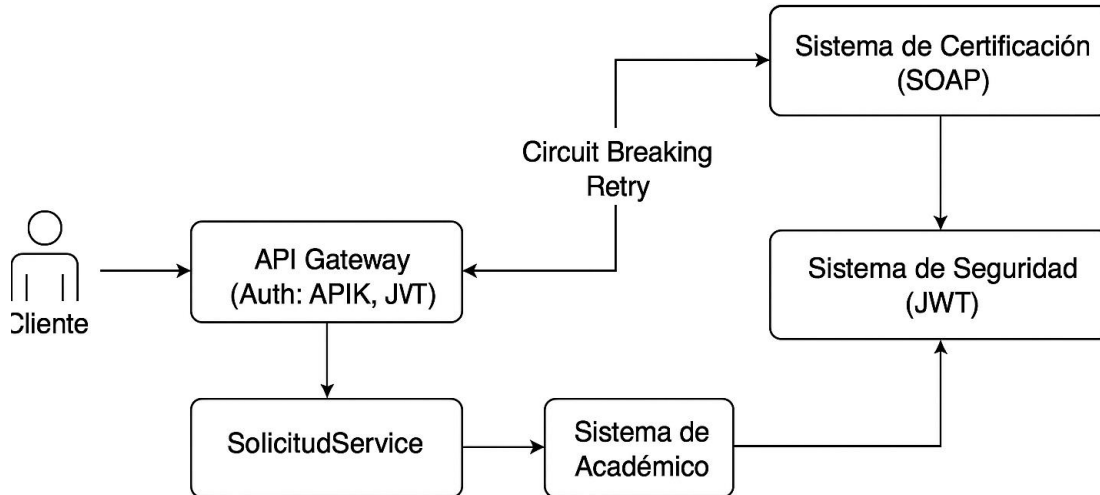
UNIVERSIDAD DE LAS AMÉRICAS
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
INGENIERÍA DE SOFTWARE
INTEGRACIÓN DE SISTEMAS

Nombre: Joe Cordero

Fecha: 29/05/2025

Examen Progreso 2

1. Diseño de arquitectura



El diagrama representa la arquitectura de la Plataforma de Servicios Estudiantiles. El flujo inicia cuando un **cliente** realiza una solicitud a través del **API Gateway (Kong)**, el cual aplica autenticación con **JWT** y **API Key**, y redirige las peticiones al microservicio principal: **SolicitudService**.

Este servicio:

- Valida el token JWT recibido.
- Se comunica con un único **Sistema Académico (REST)** para consultar o registrar datos académicos.
- Realiza llamadas al **Sistema de Certificación (SOAP)** utilizando **Retry** y **Circuit Breaker** para tolerancia a fallos.

2. Microservicio REST

Link de github:

https://github.com/joecordero1/Integracion_examenP2_JoeCordero.git

3. Exposición del servicio a través del API Gateway

UNIVERSIDAD DE LAS AMÉRICAS
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
INGENIERÍA DE SOFTWARE
INTEGRACIÓN DE SISTEMAS

Usando Kong.

Captura de servicios registrados en Kong (solicitud-service)

```
joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$ curl http://localhost:8001/services
{"next":null,"data":[{"enabled":true,"write_timeout":60000,"name":"solicitud-service","port":8081,"client_certificate":null,"path":"/solicitudes","connect_timeout":60000,"read_timeout":60000,"protocol":"http","host":"172.31.42.41","tls_verify":null,"updated_at":"1748570973","tls_verify_depth":null,"tags":null,"ca_certificates":null,"created_at":"1748566673","id":"1d9ceda2-0c78-451b-8e0c-9cb35e24282e","retries":5},{"enabled":true,"write_timeout":60000,"name":"weather-service","port":8082,"client_certificate":null,"path":null,"connect_timeout":60000,"read_timeout":60000,"protocol":"http","host":"192.168.100.117","tls_verify":null,"updated_at":"1748556215","tls_verify_depth":null,"tags":null,"ca_certificates":null,"created_at":"1748556215","id":"8405ec1f-199c-4969-9e79-41c9c8c014db","retries":5},{"enabled":true,"write_timeout":60000,"name":"user-service","port":8081,"client_certificate":null,"path":null,"connect_timeout":60000,"read_timeout":60000,"protocol":"http","host":"192.168.100.117","tls_verify":null,"updated_at":"1748556210","tls_verify":null,"tags":null,"ca_certificates":null,"created_at":"1748556210","id":"8405ec1f-199c-4969-9e79-41c9c8c014db","retries":5}]}joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$
```

Captura de rutas (routes /solicitudes)

```
joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$ curl http://localhost:8001/routes
{"next":null,"data":[{"strip_path":true,"name":null,"service":{"id":"9618c95e-e589-4f7f-937f-befb41ea84c8"},"hosts":null,"protocol":"http","https":true,"preserve_host":false,"headers":null,"tags":null,"destinations":null,"sources":null,"updated_at":"1748556221","id":"0a2771aa-73e7-4f9f-b7bc-021990bdf789","path_handling":"v0","snis":null,"request_buffering":true,"response_buffering":true,"paths":["/api/users"],"methods":null,"created_at":"1748556221","https_redirect_status_code":426,"regex_priority":0,"strip_path":true,"name":null,"service":{"id":"1d9ceda2-0c78-451b-8e0c-9cb35e24282e"},"hosts":null,"protocols":["http","https"],"preserve_host":false,"headers":null,"tags":null,"destinations":null,"sources":null,"updated_at":"1748569876","id":"8afd931e-c4d9-4d6f-8d2e-e9c98eb0e657","path_handling":"v0","snis":null,"request_buffering":true,"response_buffering":true,"paths":["/solicitudes"],"methods":null,"created_at":"1748569876","https_redirect_status_code":426,"regex_priority":0,"strip_path":true,"name":null,"service":{"id":"8405ec1f-199c-4969-9e79-41c9c8c014db"},"hosts":null,"protocols":["http","https"],"preserve_host":false,"headers":null,"tags":null,"destinations":null,"sources":null,"updated_at":"1748556224","id":"a743ef30-31b4-4305-adc3-01abd66917ef","path_handling":"v0","snis":null,"request_buffering":true,"response_buffering":true,"paths":["/api/weather"],"methods":null,"created_at":"1748556224","https_redirect_status_code":426,"regex_priority":0}]}joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$
```

plugin de seguridad

```
joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$ curl http://localhost:8001/services/solicitud-service/plugins
{"next":null,"data":[{"enabled":true,"instance_name":null,"name":"key-auth","service":{"id":"1d9ceda2-0c78-451b-8e0c-9cb35e24282e"},"id":"cce6dd16-1ee3-4f5b-a39a-c83a81d52329","tags":null,"created_at":"1748567203","consumer":null,"route":null,"protocols":["grpc","grpcs","http","https"],"updated_at":"1748567203","config":{"key_in_header":true,"key_in_query":true,"key_in_body":false,"anonymous":null,"run_on_preflight":true,"key_names":["apikey"],"hide_credentials":false}]}joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$
```

Plugin de rate limiting

```
joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero$ curl http://localhost:8001/services/solicitud-service/plugins
{"next":null,"data":[{"enabled":true,"instance_name":null,"name":"rate-limiting","service":{"id":"1d9ceda2-0c78-451b-8e0c-9cb35e24282e"},"id":"12534b16-4b79-413b-ab1c-d0bc3e7248d8","tags":null,"created_at":"1748572571","consumer":null,"route":null,"protocols":["grpc","grpcs","http","https"],"updated_at":"1748572571","config":{"second":null,"minute":5,"policy":"local","day":null,"month":null,"year":null,"redis_port":6379,"hide_client_headers":false,"redis_password":null,"redis_username":null,"redis_ssl":false,"error_code":429,"error_message":"API rate limit exceeded","sync_rate":-1,"redis_host":null,"redis_timeout":2000,"redis_server_name":null,"redis_database":0,"redis_ssl_verify":false,"header_name":null,"hour":null,"path":null,"limit_by":"consumer","redis":{"password":null,"ssl_verify":false,"port":6379,"username":null,"ssl":false,"server_name":null,"database":0,"timeout":2000,"host":null,"fault_tolerant":true}},{"enabled":true,"instance_name":null,"name":"key-auth","service":{"id":"1d9ceda2-0c78-451b-8e0c-9cb35e24282e"},"id":"cce6dd16-1ee3-4f5b-a39a-c83a81d52329","tags":null,"created_at":"1748567203","consumer":null,"route":null,"protocols":["grpc","grpcs","http","https"],"updated_at":"1748567203","config":{"key_in_header":true,"key_in_query":true,"key_in_body":false,"anonymous":null,"run_on_preflight":true,"key_names":["apikey"],"hide_credentials":false}]}joe@joe-Precision-3551: ~/Desktop/Integracion/Integracion_examenP2_JoeCordero/solicitud-service$
```

Con este comando limito a un maximo de 5 solicitudes por minuto:

curl -i -X POST http://localhost:8001/services/solicitud-service/plugins \

--data "name=rate-limiting" \

--data "config.minute=5"

UNIVERSIDAD DE LAS AMÉRICAS
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
INGENIERÍA DE SOFTWARE
INTEGRACIÓN DE SISTEMAS

4. Implementación de Circuit Breaking y Retry

Se aplicó la lógica de **Retry automático** y **Circuit Breaker** en el microservicio SolicitudesService, específicamente en la función que realiza la llamada al servicio SOAP. Esto se hizo usando las librerías tenacity (para retry) y pybreaker (para circuit breaking), dado que el entorno no usó Istio ni Spring Cloud Gateway.

Fragmento de código implementado (solicitud-service.py):

```
python
CopyEdit
from tenacity import retry, stop_after_attempt, wait_fixed
import pybreaker
from zeep import Client
import logging

# Configuración del circuito
breaker = pybreaker.CircuitBreaker(
    fail_max=3,          # Se rompe si hay 3 fallos consecutivos
    reset_timeout=60     # Se reintenta luego de 60 segundos
)

# Configuración de retry: máximo 2 reintentos con 2 segundos entre
cada uno
@retry(stop=stop_after_attempt(3), wait=wait_fixed(2)) # 1 intento
inicial + 2 reintentos
@breaker
```

UNIVERSIDAD DE LAS AMÉRICAS
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
INGENIERÍA DE SOFTWARE
INTEGRACIÓN DE SISTEMAS

```
def llamar_servicio_soap(datos):  
    tipo = datos.get("tipo")  
    wsdl_url = "http://localhost:8002/?wsdl"  
    logging.info("Llamando al servicio SOAP con retry y breaker...")  
    client = Client(wsdl=wsdl_url)  
    respuesta = client.service.procesarSolicitud(tipo)  
    return respuesta
```

- Si el servicio SOAP falla, el sistema intentará **hasta 3 veces** con 2 segundos entre intentos (tenacity).
- Si ocurren **3 fallos consecutivos**, se activará el **circuit breaker** (pybreaker), bloqueando nuevas llamadas durante 60 segundos para proteger el sistema.
- Esto evita saturar el servicio externo y mejora la resiliencia del microservicio.

5. Monitoreo

Para implementar monitoreo y trazabilidad en esta arquitectura basada en microservicios y servicios externos, se propone el uso de herramientas modernas de **observabilidad** que permitan detectar errores, analizar rendimiento y seguir el recorrido de cada solicitud a través del sistema.

Herramientas:

- **Prometheus:** *para la recolección de métricas en tiempo real.*
- **Grafana:** para visualizar gráficas y dashboards personalizados.
- **Jaeger o Zipkin:** para trazabilidad distribuida (tracing), útil para identificar cuellos de botella o fallas en servicios interconectados.
- **Fluentd o ELK Stack** (Elasticsearch, Logstash, Kibana): para centralizar y visualizar logs.
- **Kiali** (si se usara Istio): para entender gráficamente el flujo entre servicios.

UNIVERSIDAD DE LAS AMÉRICAS
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
INGENIERÍA DE SOFTWARE
INTEGRACIÓN DE SISTEMAS

Métricas que se podrían capturar

- Tiempo de respuesta (latencia) de cada microservicio.
- Número de solicitudes exitosas vs. fallidas.
- Uso del circuito breaker (número de veces que se activó).
- Cantidad de reintentos realizados por servicio.
- Uso de recursos (CPU, RAM) de cada contenedor.
- Ratelimiting aplicado por el API Gateway.

Trazas a seguir

- Origen y destino de cada solicitud (desde el cliente hasta el servicio SOAP).
- ID de solicitud y usuario asociado.
- Tiempo total de ejecución de la solicitud.
- Error exacto en caso de fallo (excepción, código, descripción).
- Flujo de eventos dentro del servicio SolicitudService.

Estas herramientas y métricas permitirían **anticiparse a fallos**, detectar problemas en tiempo real y ofrecer **resiliencia** al sistema. Aunque no se implementaron directamente en este ejercicio, la arquitectura está preparada para integrarse fácilmente con estas soluciones en fases futuras del proyecto.