**Introducción**

Este documento describe los pasos necesarios para realizar la integración entre dos sistemas de facturación: API A, que es el sistema del cual se obtendrán los datos, y API B, que es el sistema al cual se enviarán los datos transformados. El objetivo es asegurar que los equipos de Services Managers y desarrolladores comprendan claramente el flujo y los requisitos para implementar correctamente la integración en la API B.

**Visión General del Proceso**

La integración consta de varias etapas críticas, desde la autenticación en API A hasta la transformación de los datos y su envío a API B. A continuación, se desglosan estos pasos en detalle.

**Diagrama de Flujo**

A diagram of a software process

Description automatically generated

**Descripción de las APIs**

API A: Sistema de Facturación A Endpoint de Consulta de Facturas:

URL: https://api.sistemaA.com/facturas

Método: GET

Parámetros de Consulta:

fecha\_inicio: Fecha de inicio para el rango de consulta (Formato: YYYY-MM-DD).

fecha\_fin: Fecha de fin para el rango de consulta (Formato: YYYY-MM-DD).

Autenticación: Requiere un token OAuth 2.0, obtenido a través de client credentials.

Respuesta: La API A devuelve un JSON con una lista de facturas, cada una con detalles como ID, cliente, monto, fecha de emisión, y estado de pago.

API B: Sistema de Facturación B

Endpoint de Envío de Facturas:

URL: https://api.sistemaB.com/bills

Método: POST

Autenticación: Utiliza una clave API que debe ser enviada en el encabezado.

Datos de Entrada: Se espera un JSON que contenga una lista de facturas en un formato específico que API B pueda procesar.

**Flujo de Integración**

1. Autenticación en API A

Para interactuar con la API A, es necesario obtener un token OAuth 2.0. Este token se obtiene realizando una solicitud a un endpoint de autenticación, utilizando las credenciales del cliente (client ID y client secret). Este token luego se usa en el encabezado de las solicitudes subsiguientes para autenticar las peticiones.

2. Consulta de Facturas en API A

Una vez autenticados, se realiza una solicitud GET al endpoint de facturas de API A, proporcionando las fechas de inicio y fin como parámetros. La API A devolverá un conjunto de facturas que coinciden con los criterios trabajadores en ese rango de fechas.

3. Transformación de Datos

Los datos recibidos de la API A deben ser transformados al formato esperado por la API B. Esto implica renombrar campos, ajustar estructuras de datos, y asegurarse de que los datos cumplan con los requisitos de la API B. Por ejemplo:

id → invoice\_id

cliente → customer

monto → amount\_due

fecha\_emision → date\_issued

estado → status

4. Envío de Datos a API B

Finalmente, los datos transformados se envían a API B mediante una solicitud POST. Es fundamental asegurar que los datos cumplen con el formato requerido y que la autenticación mediante la clave API es correcta.

**Consideraciones Técnicas**

Manejo de Errores

Es crucial implementar un manejo robusto de errores en cada etapa del proceso:

Autenticación: Validar que el token se ha obtenido correctamente y manejar los casos en los que la autenticación falle.

Consulta de Facturas: Implementar reintentos en caso de fallos de red y manejar respuestas vacías o errores específicos del servicio.

Transformación de Datos: Validar la integridad de los datos transformados antes de enviarlos a API B.

Envío a API B: Gestionar las respuestas de éxito y error, asegurando que las facturas se envían de manera idempotente para evitar duplicaciones.

Seguridad

Almacenamiento Seguro de Credenciales: Las credenciales de acceso y las claves API deben almacenarse de manera segura, utilizando prácticas recomendadas como variables de entorno o servicios de gestión de secretos.

Transmisión Segura: Todas las comunicaciones entre los sistemas deben realizarse sobre HTTPS para asegurar la confidencialidad e integridad de los datos transmitidos.

**Conclusión**

Este documento ha detallado los pasos y consideraciones necesarios para integrar las APIs A y B. Siguiendo este flujo, el equipo de desarrollo podrá construir un proceso confiable que garantice la correcta transferencia de datos entre los sistemas de facturación, asegurando que los datos se manejan de manera segura y eficiente.