Tarea - Semana 2

Desarrollo de Software 3 - Agustin De Luca

Enfoques de Integración

1. Punto a Punto (Point-to-Point)

Este enfoque establece conexiones directas entre sistemas individuales. Aunque es sencillo en entornos pequeños, su escalabilidad es limitada: a medida que crece el número de sistemas, las conexiones se multiplican exponencialmente, generando una red compleja y difícil de mantener.

- **Ejemplo**: Un sistema de gestión de relaciones con clientes (CRM) que se conecta directamente a un servicio de correo electrónico para enviar notificaciones.
- **Visualización con mapa mental**: Un diagrama mostraría líneas directas entre cada par de sistemas, resaltando la interdependencia y la fragilidad ante cambios.
- **Importancia**: Útil para prototipos o entornos con pocas integraciones, pero riesgoso en escalabilidad.

2. Enterprise Service Bus (ESB)

El middleware actúa como intermediario centralizado, traduciendo y enrutando datos entre sistemas. Un ESB, por ejemplo, ofrece capacidades avanzadas como transformación de mensajes y gestión de errores.

- **Ejemplo**: Una empresa manufacturera utiliza un ESB para conectar su sistema ERP con el de inventario y el CRM, asegurando que los datos de pedidos se sincronicen en tiempo real.
- **Visualización**: Un mapa mental ubicaría al ESB en el centro, con ramas hacia cada sistema, enfatizando su rol como "columna vertebral".
- **Importancia**: Reduce el acoplamiento entre sistemas, facilitando la modularidad y el mantenimiento.

3. Integración Basada en APIs

Las APIs (Interfaces de Programación de Aplicaciones) permiten la comunicación estandarizada entre sistemas, siguiendo capas como *system* (acceso a datos), *process* (orquestación) y *experience* (interacción con usuarios).

- **Ejemplo**: Una aplicación móvil de banca utiliza APIs REST para acceder a servicios transaccionales, autenticación y notificaciones, integrando backend y frontend sin exponer lógica interna.
- **Visualización**: Un mapa mental estratificaría las APIs en tres niveles, mostrando cómo cada capa sirve a propósitos específicos.
- **Importancia**: Fomenta la reutilización, agiliza el desarrollo y permite ecosistemas tecnológicos extensibles.

4. Integración Orientada a Eventos

Los sistemas se comunican mediante eventos (p. ej., cambios de estado), usando brokers como Kafka o RabbitMQ para publicar y suscribirse a mensajes.

- **Ejemplo**: En un e-commerce, cuando un usuario finaliza una compra, se publica un evento que actualiza inventario, genera una factura y notifica al equipo de logística.
- **Visualización**: Un mapa mental ilustraría un bus de eventos con productores y consumidores, destacando la reactividad en tiempo real.
- **Importancia**: Ideal para escenarios que requieren baja latencia y alta escalabilidad, como IoT o microservicios.

5. Integración de Datos (ETL/ELT)

Se enfoca en consolidar información de múltiples fuentes mediante procesos de Extracción, Transformación y Carga (ETL) o su variante ELT.

- **Ejemplo**: Una empresa retail centraliza datos de ventas físicas, online y redes sociales en un data warehouse, usando herramientas como Informatica para limpieza y análisis.
- **Visualización**: Un flujo en el mapa mental mostraría fuentes dispersas convergiendo en un repositorio unificado tras pasar por transformaciones.
- **Importancia**: Clave para business intelligence y machine learning, donde la calidad y consistencia de los datos son críticas.

6. Microservicios: Descentralización y Autonomía

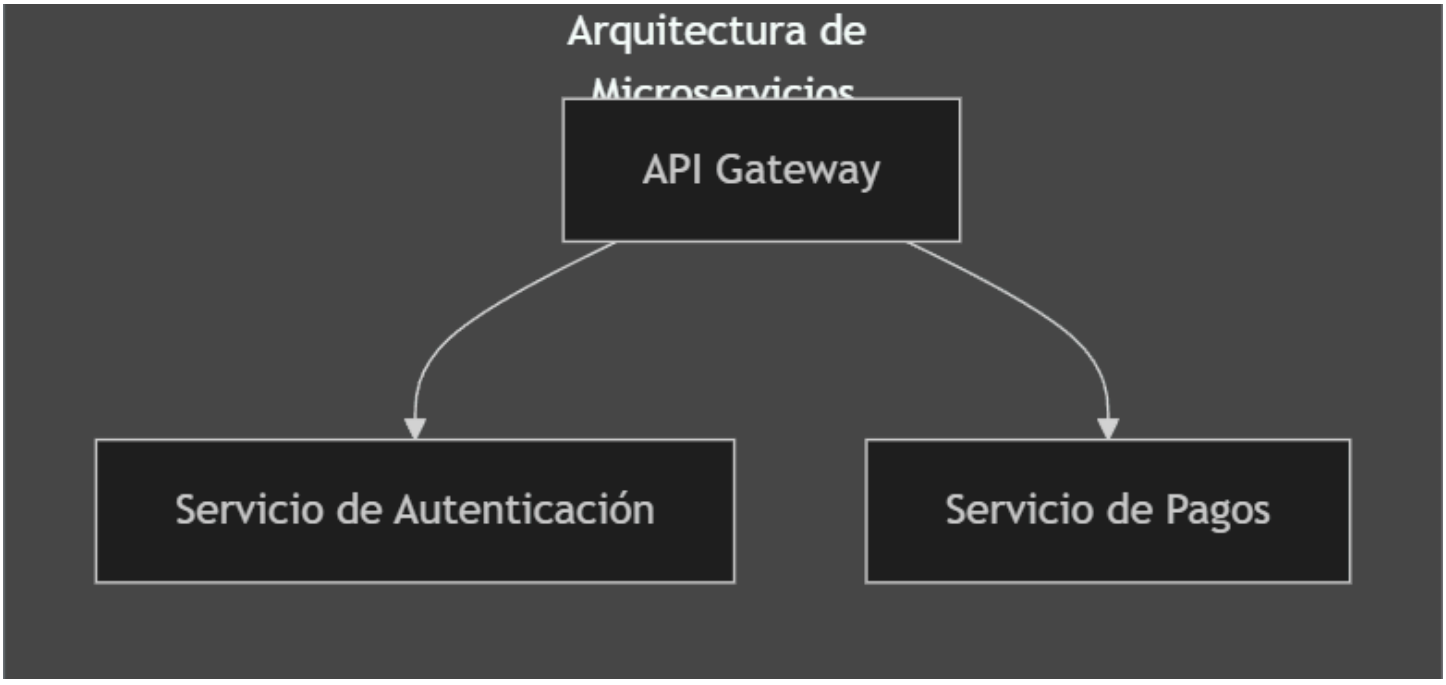
Los microservicios son un estilo arquitectónico donde una aplicación se divide en servicios pequeños, independientes y especializados, cada uno con su propia lógica y base de datos. La comunicación entre ellos se realiza mediante APIs ligeras (REST/gRPC) o eventos (mensajería).

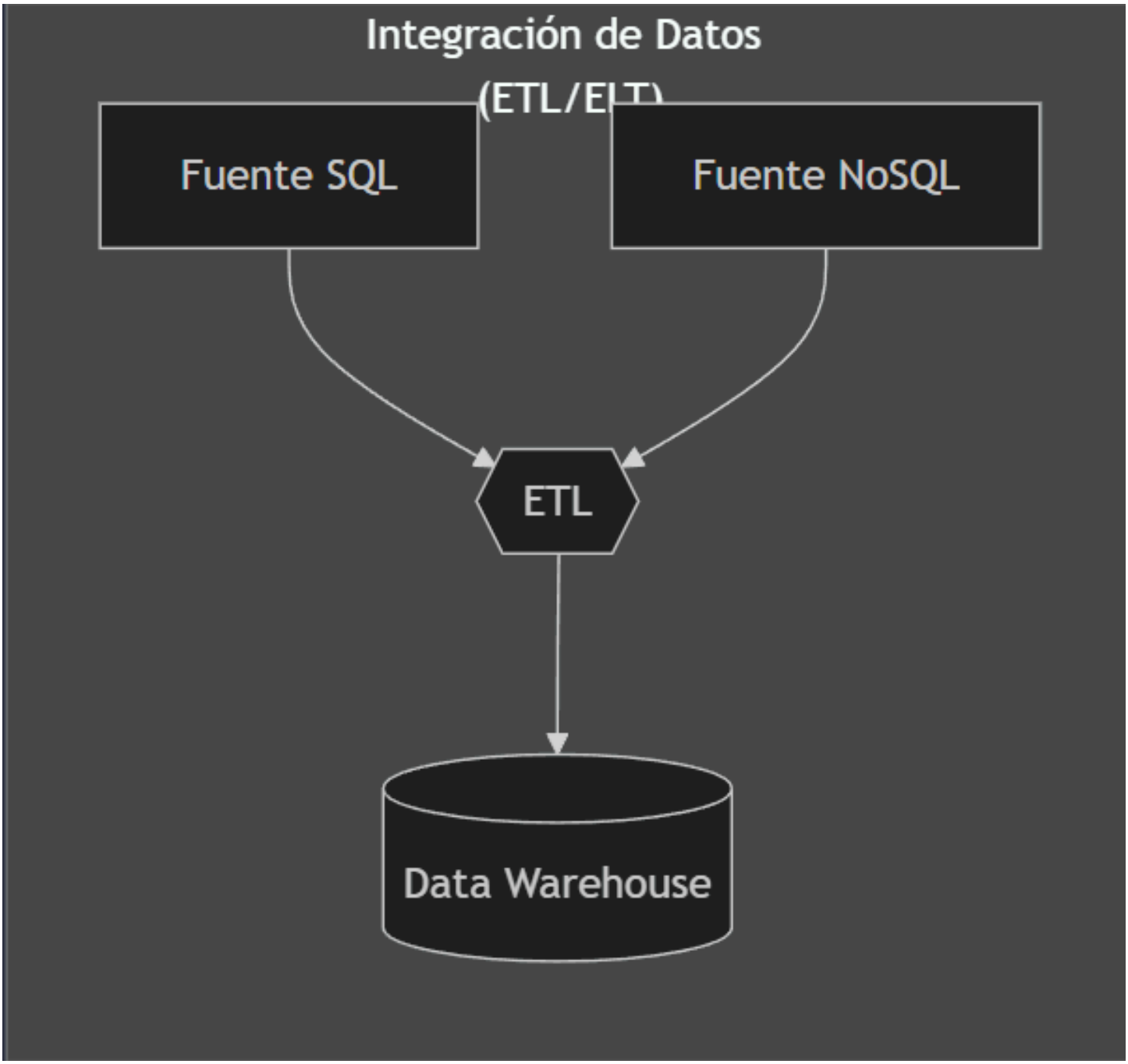
- **Características clave**:
  - **Autonomía**: Cada servicio se despliega y escala de forma independiente.
  - **Enfoque domain-driven**: Alineados a dominios de negocio específicos (ejemplo: servicio de pagos, servicio de autenticación).
  - **Resistencia a fallos**: Un error en un servicio no colapsa el sistema completo.
- **Ejemplo**: Una plataforma de streaming como Netflix usa microservicios para gestionar perfiles de usuario, recomendaciones y reproducción de videos, permitiendo actualizaciones continuas sin interrupciones globales.
- **Integración en microservicios**:
  - **APIs REST/GraphQL**: Para interacciones síncronas (ejemplo: un frontend que consume datos de un catálogo).
  - **Eventos y mensajería (Kafka, RabbitMQ)**: Para flujos asíncronos (ejemplo: notificar al servicio de logística cuando un pedido es confirmado).
  - **API Gateway**: Actúa como punto único de entrada, enrutando solicitudes y gestionando seguridad (autenticación, rate limiting).

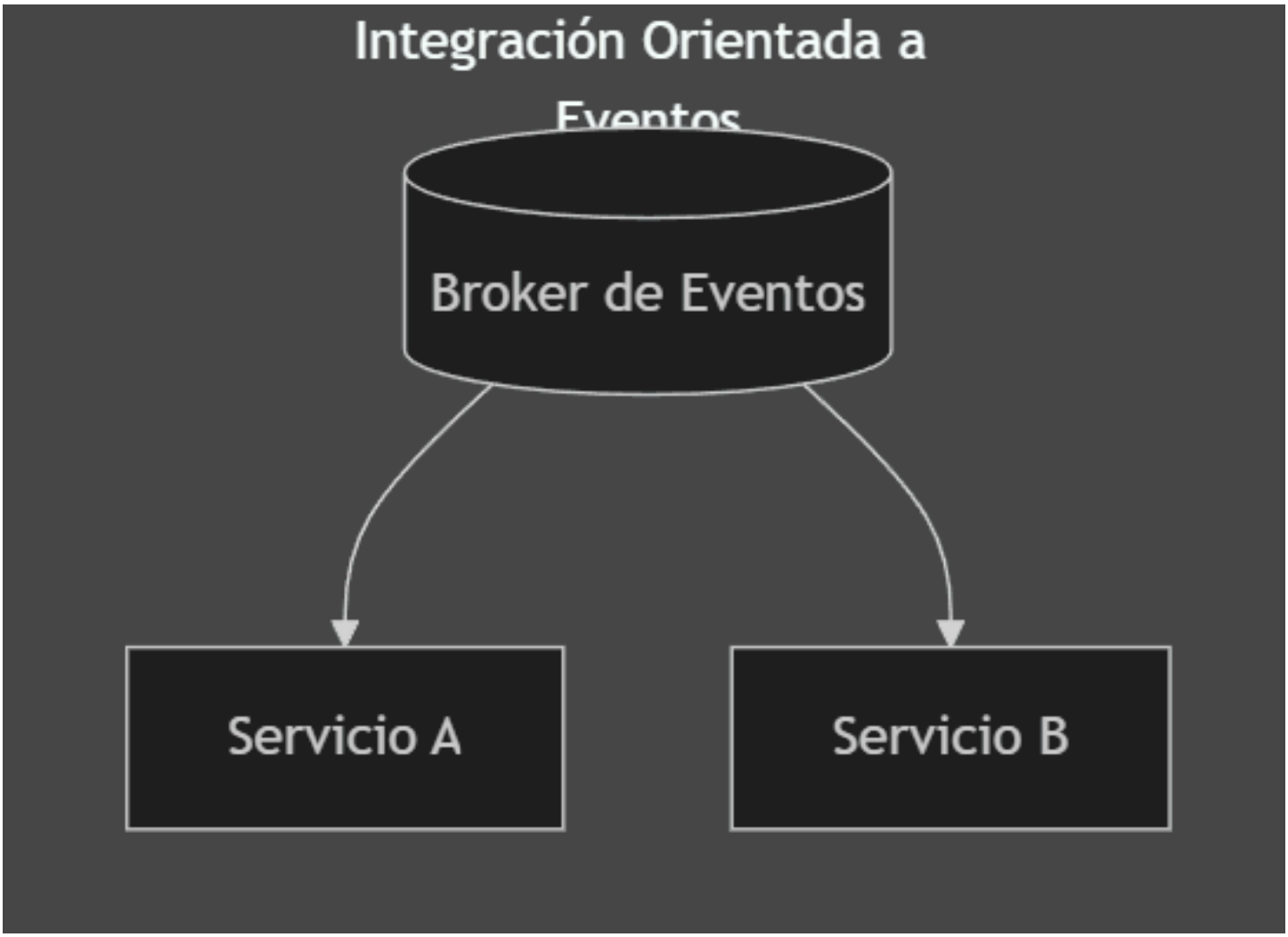
7. Middleware: Centralización y Mediación

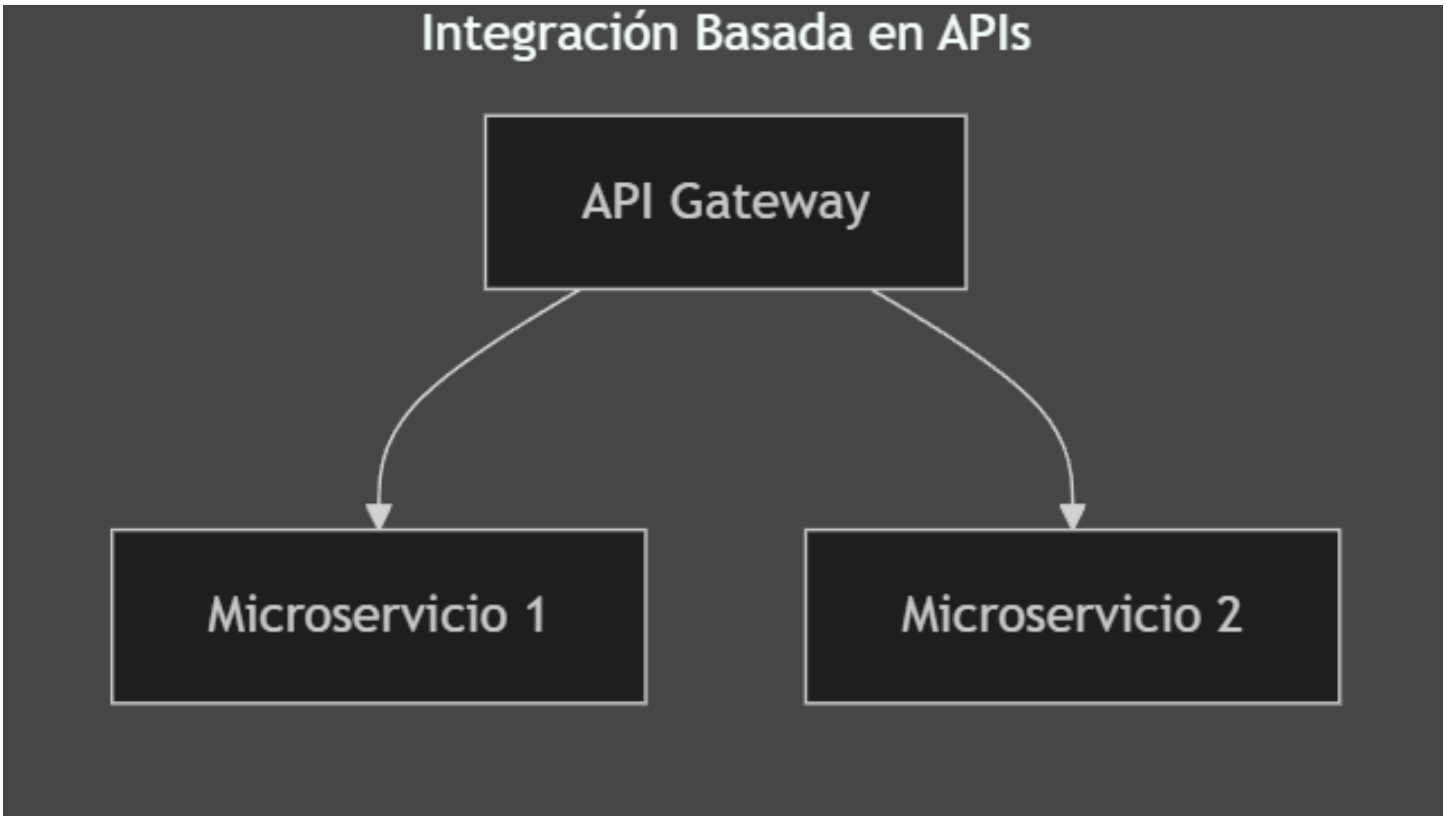
El middleware es un software intermediario que facilita la comunicación entre sistemas heterogéneos. Enfoques tradicionales como el **Enterprise Service Bus (ESB)** son ejemplos clásicos, centralizando la lógica de integración (transformación de datos, enrutamiento).

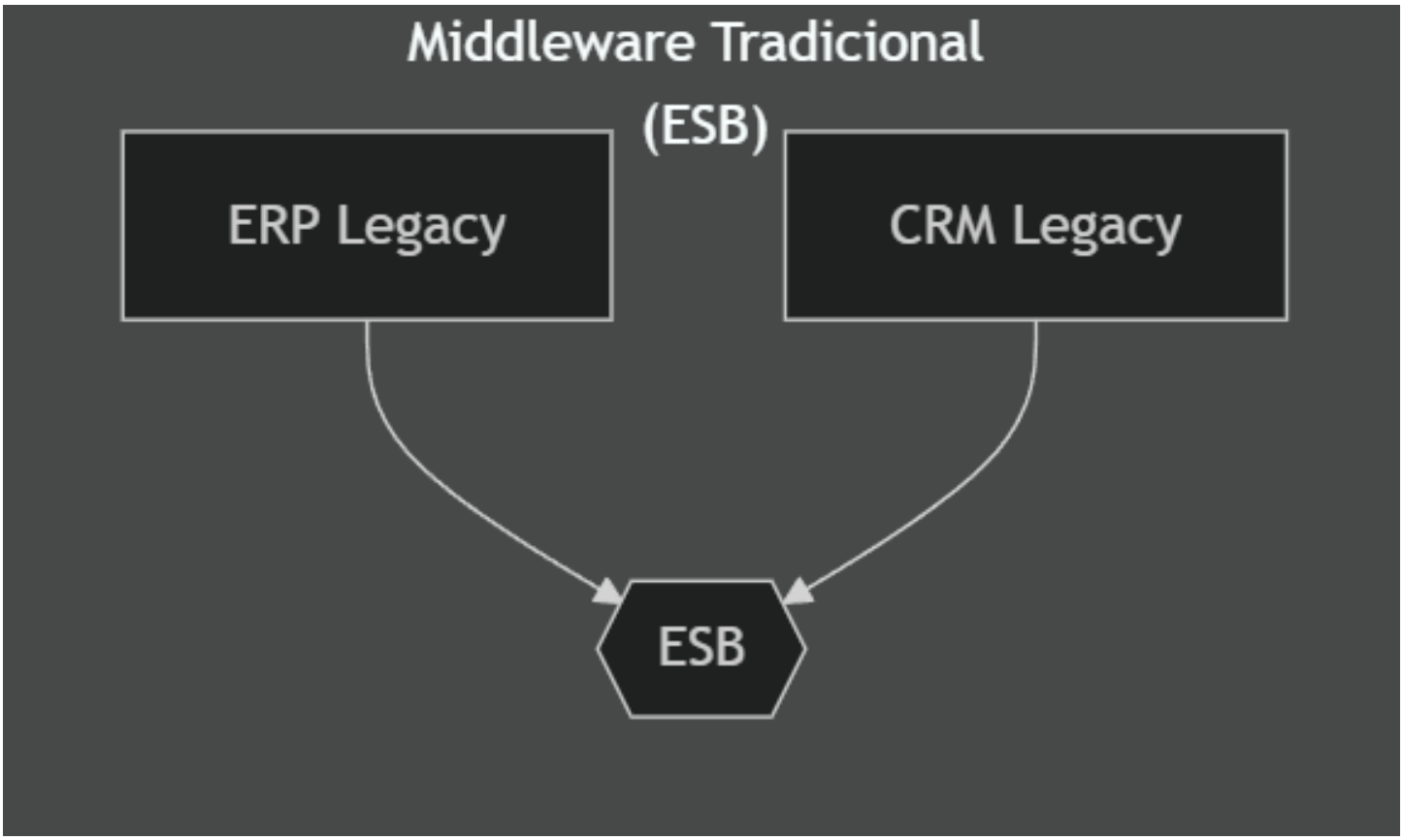
- **Características clave**:
  - **Acoplamiento flexible**: Los sistemas no se comunican directamente, sino a través del middleware.
  - **Capacidades avanzadas**: Transformación de formatos (XML a JSON), gestión de errores, monitoreo.
  - **Costo de complejidad**: Puede convertirse en un cuello de botella si no se diseña adecuadamente.
- **Ejemplo**: Un banco que integra su sistema core (COBOL) con una aplicación móvil moderna usando un ESB para traducir mensajes entre protocolos legacy (SOAP) y modernos (REST).
- **Middleware en la era de microservicios**:
  - **Adaptación a arquitecturas modernas**: Herramientas como **API Management** (ejemplo: Apigee) o **Service Mesh** (ejemplo: Istio) emergen como "middleware ligero", ofreciendo funcionalidades similares a un ESB pero adaptadas a microservicios (ejemplo: enrutamiento basado en políticas, observabilidad).
  - **Integración híbrida**: Combina ESB para sistemas legacy y APIs/eventos para microservicios.

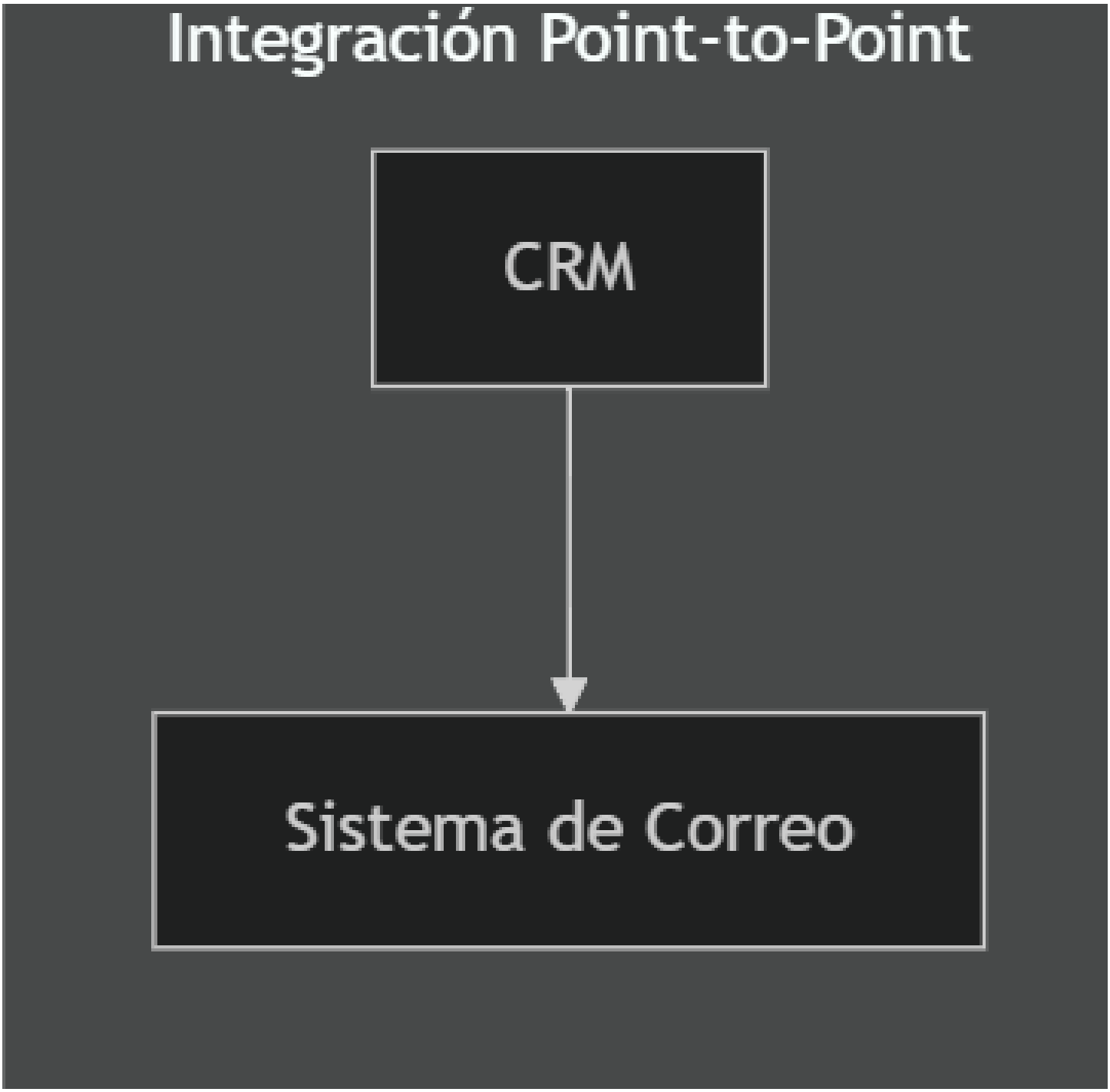



Integración de Datos
(ETL/ELT)
Fuente SQL
Fuente NoSQL
ETL
Data Warehouse


Integración Orientada a
Eventos
Broker de Eventos
Servicio A
Servicio B


Integración Basada en APIs
API Gateway
Microservicio 1
Microservicio 2


Middleware Tradicional
(ESB)
ERP Legacy
CRM Legacy
ESB


Integración Point-to-Point
CRM
Sistema de Correo