

¿En qué requerimiento implementarías una cola de mensajes en una solución orientada a datos? ¿Qué lenguaje utilizarías y por qué?

Implementaría una cola de mensajes en soluciones que utilicen microservicios, puesto que esto haría que los mismos no tengan que interactuar entre sí (comunicación asincrónica). Esto genera varias ventajas:

- Aporta velocidad, ya que un servicio no tiene que esperar a que el otro se conecte para comunicarse
- Aporta fiabilidad, porque no requiere que el servicio esté funcionando para realizar la comunicación. Esto es, si falla un emisor de mensaje luego de dejarlo en el sistema de cola, el receptor podrá interpretarlo y dejar una respuesta para que sea leída por el servicio caído una vez que vuelva a estar en ejecución (o por su servicio redundante si es que lo tiene)
- Facilita la escalabilidad, al dejar mensajes en el sistema de colas, se simplifica la configuración de comunicación del nuevo nodo con el resto de los microservicios

Elegiría un lenguaje de acuerdo con los microservicios puntuales a tratar. Como la situación es sobre una solución de datos, optaría por un lenguaje orientado a objetos o funcional, como puede ser Python, que es reconocido como uno de los más eficientes para el tratamiento de los mismos.

¿Qué experiencia posees sobre Py Spark o Spark Scala? Contar breves experiencias, en caso de contar experiencia con otro Framework de procesamiento distribuido, dar detalles también.

A pesar de conocer sus funciones y objetivos, no poseo experiencia en los temas mencionados en la consigna.

¿Qué funcionalidad podrías expandir desde el área de ingeniería de datos con una API y arquitectónicamente como la modelarías?

En un entorno bancario, implementaría una API privada que pueda ser consultada por todas las sucursales del país. Esta API se utilizará para conocer información necesaria para la toma de las decisiones de negocio de cada sucursal en el área geográfica en la que está situada. Dicha información abarcará datos como el porcentaje de clientes que poseen cierto producto contra otro (por ejemplo, compañías de tarjeta de crédito), descuentos y beneficios más aprovechados por los clientes según su zona de residencia, cantidad de créditos otorgados y pagos al día según ingresos, etc.

Como se trata de una API que no puede ser utilizada por terceros al banco, debe ser una API privada. La misma será montada en un servicio Cloud IaaS (Infrastructure as a Service), con almacenamiento y procesamiento de escalabilidad dinámica debido al volumen de datos y peticiones que se pueden llegar a manejar. También se tendrá una instancia redundante en caso de sufrir fallas.

Se utilizará una arquitectura REST y JSON para la comunicación.

La API se conectará con los diferentes sistemas de las sucursales bancarias con el fin de brindar información, pero también se comunicará con las diversas API de compañías relacionadas con el banco (como de tarjetas de crédito, marcas de ropa con descuento, etc.) y con la de Google Maps, para obtener datos faltantes de clientes y de georreferencia.

Para finalizar, la API obtendrá datos de un sistema de microservicios compuesto por un servicio de almacenamiento, otro de tratamiento de datos y otro de procesamiento de peticiones.

A modo gráfico, se puede tener en cuenta el siguiente diagrama básico:

