

- 1) ¿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 cuando el sistema debe seguir notificando al cliente sobre una determinada operación independientemente de si un componente se cae y deja de funcionar. Este atributo se llama **disponibilidad**. De esta forma, si, por ejemplo, el componente consumidor se cae, los mensajes se van encolando y no se perderá información. Incluso si el consumidor se cae mientras se encuentra procesando un mensaje de la cola, este tipo de arquitectura puede soportar el reintento de procesamiento del mensaje, asegurando el procesamiento de los mensajes.

Por otro lado, este tipo de arquitectura se podría implementar también cuando las acciones de los consumidores son asincrónicas, es decir, no necesitan confirmar la recepción de un mensaje en el mismo momento en que se generan, sino que pueden realizar su operatoria en otro momento.

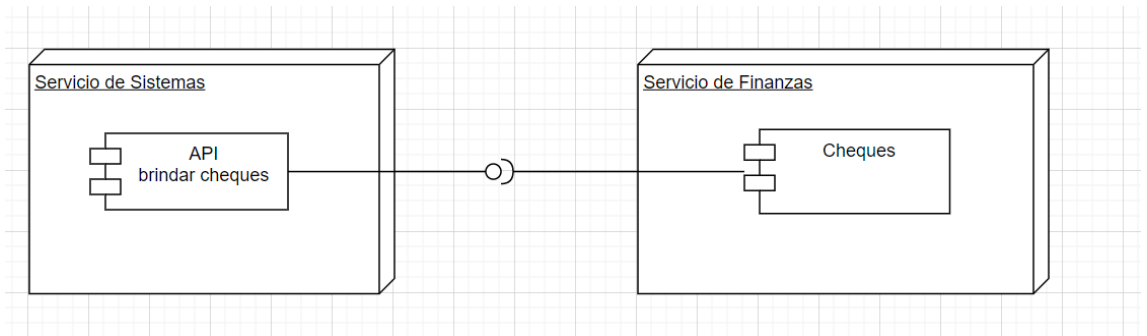
- 2) ¿Qué experiencia posees sobre py spark o spark scala? Contar breves experiencias, en caso de contar experiencia con otro framework de procesamiento distribuido, dar detalles también.

No poseo experiencia en py spark ni en spark scala, pero me gustaría aprender.

- 3) ¿Qué funcionalidad podrías expandir desde el área de ingeniería de datos con una API y arquitectónicamente como lo modelarías?

Con una API, se podría expandir la flexibilidad en un modelo de datos porque permite que se comuniquen entre sí un productor y un consumidor sin la necesidad de saber cómo están implementados. Por ejemplo, el área de Finanzas de la empresa Santander necesita tener acceso a las características de los cheques que posee, como la descripción, el código, los tipos, etc. Para ello, el área de Sistemas dispone de una API para que el área de Finanzas pueda obtener esa información sin necesidad de darle acceso a una base de datos. Esto hace que aumente la performance (más velocidad al obtener los datos) y además brinda seguridad a la base de datos.

Esto se vería implementado de la siguiente forma:



El Servicio de Sistemas disponibiliza una interfaz (API) que "consume" el Servicio de Finanzas.