**Preguntas Taller RabitMQ**

**Nombres:** Alex Duvan Hernandez Buitrago

Yesid Alejandro Martinez Guerrero

**1. ¿Qué beneficio aporta RabbitMQ en comparación con el modelo de solicitud directa HTTP?**

**RTA:** Aporta los siguientes beneficios:

* Los servicios no necesitan conocerse entre sí, solo necesitan conocer la cola, lo que facilita su implementación al reducir las dependencias directas y mejora su escalabilidad.
* Los mensajes se pueden procesar cuando el receptor esté disponible y a la vez pueden ser almacenados hasta que haya un receptor disponible que pueda procesarlos, lo que significa que los mensajes no se pierden incluso si un servicio está caído temporalmente.
* Facilita la entrega y distribución de mensajes a múltiples receptores.

**2. ¿Qué problemas podrían surgir si se caen algunos servicios?**

**RTA:** En programas que no implementan RabitMQ si un servicio se cae mientras se le envía algún tipo de mensaje o solicitud, dichos mensajes se perderían o el programa fallaría, caso diferente que con la implementación de RabitMQ las solicitudes se almacenan en la cola hasta que el servicio correspondiente u otro servicio que pueda recibir el mensaje lo haga

**3. ¿Cómo ayuda RabbitMQ a mejorar la resiliencia del sistema?**

**RTA:** RabbitMQ mejora la resiliencia mediante:

* **Buffer de mensajes:** Almacena temporalmente los mensajes si los servicios están caídos
* **Reconexión automática:** Los servicios intentan reconectarse automáticamente
* **Persistencia de mensajes:** Los mensajes no se pierden durante las caídas
* **Balanceo de carga:** Distribuye la carga entre múltiples instancias

**4. ¿Cómo cambiaría la lógica de escalabilidad con esta nueva arquitectura?**

**RTA:** La nueva arquitectura con RabbitMQ permite:

* Fácil adición de nuevos servicios
* Balanceo automático que distribuye mensajes entre los servicios que los requiera
* Ya que los servicios están desacoplados entre ellos permite que cada servicio escale individualmente sin ver mayor afectación de los demás servicios

**5. ¿Qué formato de mensaje es más conveniente y por qué (JSON, texto plano, etc.)?**

JSON es el más conveniente porque posee una estructura clara y de fácil comprensión que agrupa los objetos de forma jerárquica facilitando su leída y escritura, los nombres de los objetos y atributos tienen nombres muy descriptivos lo que facilita su comprensión y es soportado ampliamente por una cantidad significativa de lenguajes