Comparación y Decisión: Kafka vs RabbitMQ

# ¿Qué es Event Streaming?

Es el procesamiento en tiempo real de eventos que ocurren de forma continua. Por ejemplo, plataformas como Netflix o YouTube publican eventos cada vez que un usuario interactúa con el sistema, y estos eventos se procesan para analítica, detección de fraude, recomendaciones, etc.

# ¿Qué es Procesamiento Masivo de Eventos?

Es cuando necesitas procesar millones de eventos, como ocurre en sistemas financieros o de monitoreo global. Kafka es ideal para esto por su capacidad de almacenar eventos como logs distribuidos y su alto rendimiento.

# Comparación técnica realista

|  |  |  |
| --- | --- | --- |
| Característica | RabbitMQ | Kafka |
| Modelo | Mensajería tradicional (AMQP) | Log distribuido (append-only) |
| Latencia | Muy baja para flujos ligeros (<5ms) | Baja, pero más optimizada para throughput |
| Garantía de entrega | Soporta ACK, reintento, DLQ, TTL | El consumidor es responsable de confirmar |
| Temporales y TTL por mensaje | ✅ Soportado directamente | ❌ No nativo, necesitas lógica adicional |
| Routing avanzado | Exchanges (fanout, topic, direct, header) | ❌ No existe. El productor elige el topic |
| Simplicidad de despliegue | ✅ Fácil (una instancia, contenedor) | ❌ Complejo: necesita ZooKeeper o KRaft |
| Casos ideales | Microservicios, RPC, workflows, IoT | Big Data, Analytics, Event Sourcing, Logs |

# Tabla de decisión: ¿Kafka o RabbitMQ?

|  |  |  |  |
| --- | --- | --- | --- |
| Componente / Caso de Uso | RabbitMQ 🐇 | Kafka 🦄 | Justificación técnica |
| Orquestación de SAGA | ✅ | ❌ | Requiere entrega puntual, reintento, TTL, DLQ y routing fino |
| Coreografía de eventos (tipo Event-Driven) | ⚠️ | ✅ | Kafka permite múltiples consumidores y relectura |
| Workflows BPMN (Camunda, Activiti, etc.) | ✅ | ❌ | Se necesita control por mensaje, orden y TTL |
| Procesamiento en tiempo real de logs, métricas, sensores | ❌ | ✅ | Kafka maneja millones de eventos por segundo |
| Eventos de auditoría e históricos | ❌ | ✅ | Kafka guarda los eventos como un log persistente |
| Sistema de pagos y antifraude | ✅ | ⚠️ | Rabbit permite control preciso de entrega, orden y reintentos |
| Carga inicial de catálogo (batch de productos) | ⚠️ | ✅ | Kafka puede procesar millones de eventos en lotes |
| Monitoreo o analítica de comportamiento de usuario | ❌ | ✅ | Kafka + Streams permite análisis en tiempo real |
| Comunicaciones internas entre microservicios (RPC-like) | ✅ | ❌ | RabbitMQ permite request/response y expiración |
| Notificaciones push a usuarios | ✅ | ❌ | Rabbit + TTL + routing + DLQ es ideal |
| Integración con ELK, BigQuery, Hadoop | ❌ | ✅ | Kafka es ideal para pipelines de datos |
| Sistema distribuido con consumo paralelo sin pérdida | ⚠️ | ✅ | Kafka permite múltiples grupos consumidores |