

Cuestionario KAFKA

1.- Indica si las siguientes afirmaciones son verdaderas (V) o falsas (F).

☐ F ___ Kafka es una base de datos relacional utilizada para almacenar grandes volúmenes de datos.

☒ V ___ Un "topic" en Kafka es como un canal donde se publican y leen mensajes.

☐ F ___ Los productores en Kafka son los encargados de consumir los mensajes.

☒ V ___ Kafka permite procesar datos en tiempo real.

☒ V ___ Un clúster de Kafka puede tener múltiples brokers.

2.- ¿Qué es Apache Kafka?

- a) Un sistema de archivos distribuido
- ☒ b) Un sistema de mensajería distribuido y tolerante a fallos
- c) Una base de datos NoSQL
- d) Un framework de procesamiento batch

3.- ¿Qué componente de Kafka publica mensajes en un topic?

- a) Consumer
- b) Broker
- ☒ c) Producer
- d) Zookeeper

4.- ¿Cuál de los siguientes es responsable de almacenar y distribuir los mensajes en Kafka?

- a) Zookeeper
- ☒ b) Broker
- c) Topic
- d) Consumer

5.- ¿Qué hace un consumidor (consumer) en Kafka?

- a) Crea topics
- b) Publica mensajes
- ☒ c) Lee mensajes desde un topic
- d) Configura el clúster

6.- ¿Cuál es el formato típico de los mensajes que viajan por Kafka?

- a) Archivos XML
- b) Objetos serializados
- ☒ c) Mensajes en formato binario o texto (JSON, Avro, etc.)
- d) Logs del sistema

7.- ¿Qué es un *topic* en Kafka?

- a) Un archivo de configuración de los brokers
- b) Una cola FIFO compartida entre todos los consumidores
- ☒ c) Un flujo de mensajes categorizado donde los productores publican datos
- d) Un componente que almacena offsets de los consumidores

8.- ¿Qué componente es responsable de publicar mensajes en Kafka?

- a) Broker
- b) Producer**
- c) Consumer
- d) Zookeeper

9.- ¿Cuál de los siguientes modos de envío de mensajes tiene el menor nivel de garantía pero el mayor *throughput*?

- a) Síncrono (acks=all)
- b) Asíncrono (acks=1)
- c) Fire-and-forget (acks=0)**
- d) Replicated write (acks=replica)

10 ¿Qué ocurre cuando un *consumer group* se reconfigura y las particiones cambian de consumidor?

- a) Se descartan los mensajes no leídos
- b) Se reinician todos los offsets
- c) Se produce un *rebalance***
- d) Se detiene el broker

11. ¿Cuál es la principal ventaja de usar claves al producir mensajes a un topic particionado?

- a) Reducir el uso de memoria en el broker
- b) Garantizar que todos los mensajes vayan a distintas particiones
- c) Asegurar que los mensajes con la misma clave vayan a la misma partición**
- d) Eliminar la necesidad de consumidores

12.- ¿Qué hace un consumidor con respecto a los *offsets* en Kafka?

- a) Los genera de forma aleatoria
- b) Los ignora si no hay fallos
- c) Los comitea en Zookeeper manualmente
- d) Los guarda (comitea) en Kafka para reanudar la lectura desde ese punto**

13.- ¿Qué ventajas ofrece Kafka frente a otros sistemas de mensajería?

Paralelización, compatibilidad con herramientas big data, escalabilidad, tolerancia a fallos, datos en tiempo real

14.- ¿Qué función cumple Zookeeper en un clúster de Kafka?

Configura y gestiona el cluster que almacena los brokers de kafka

15.- ¿Qué entiendes por procesamiento en tiempo real con Kafka?

La capacidad del software para manipular los datos que llegan de múltiples fuentes al mismo tiempo

16.- ¿Qué significa que Kafka es “tolerante a fallos”?

Significa que si un broker cae o algún componente falla kafka seguirá realizando trabajo

17.- Con la configuración predeterminada de un *Kafka Producer*, ¿qué parte de un registro se utiliza para determinar su partición de destino dentro de un *topic*?

- a) El valor (*value*) del registro
- b) El tamaño del mensaje
- c) La clave (*key*) del registro**
- d) La hora de producción del mensaje

18.- Si necesitamos enviar en flujo (*stream*) cada registro de un *topic* de Kafka a Elasticsearch para construir un índice de búsqueda, ¿qué tecnología proporcionaría una solución sin necesidad de escribir código personalizado?

- a) Kafka Streams
- b) Producer API
- c) Kafka Connect**
- d) Consumer API

19.- El *consumer* c1 acaba de consumir el mensaje m1. El *consumer* c2 quiere consumir m1 poco después. Si c1 y c2 pertenecen a grupos de *consumer* distintos, ¿cuál de las siguientes afirmaciones es verdadera?

- a) El mensaje **m1** ya fue consumido, por lo tanto **c2** no puede consumirlo.
- b) El *consumer* c2 puede consumir el mensaje m1 igual que lo hizo c1.**
- c) Ninguna de las anteriores.
- d) A y B son correctas.

20.- Supongamos que un *Producer* ha escrito un mensaje en Kafka. Ese mensaje puede ser modificado:

- a) En cualquier momento, por cualquier *Producer*.
- b) Solo por el *Producer* que lo envió a Kafka.
- c) Solo para modificar sus metadatos.
- d) Nunca.**

21.- ¿Qué garantiza la semántica de entrega "Exactly once" en Kafka?

- a) Que todos los mensajes se entregan con baja latencia.
- b) Que cada mensaje se entrega como mínimo una vez.
- c) Que cada mensaje se entrega una única vez y sin duplicados.**
- d) Que todos los consumidores reciben los mismos mensajes.

22.- ¿Qué componente de Kafka permite almacenar el estado localmente para operaciones como agregaciones o joins en Kafka Streams?

- a) Kafka Broker
- b) State Store**
- c) Kafka Producer
- d) Zookeeper

23.- ¿Cuál es la unidad mínima de paralelismo en Kafka?

- a) El topic
- b) El broker
- c) La partición**
- d) El cluster

24.- ¿Qué sucede si múltiples *consumers* pertenecen al mismo *consumer group*?

- a) Cada partición es asignada a un único *consumer* dentro del grupo.**
- b) Cada mensaje se entrega a todos los *consumers* del grupo.
- c) Todos los *consumers* compiten por el mismo mensaje.
- d) Todos procesan el mismo mensaje simultáneamente.

25.- ¿Qué ocurre cuando se supera el tiempo de retención configurado para un topic en Kafka?

- a) Los mensajes se reenvían automáticamente.
- b) Los mensajes se comprimen.
- c) Los mensajes se archivan en HDFS.
- d) Los mensajes antiguos se eliminan.**

26.- Indica si es verdadera o falsa cada una de las siguientes afirmaciones:

1. F Kafka garantiza la entrega "exactamente una vez" sin necesidad de configuración adicional.
2. V Cada partición de un *topic* en Kafka solo puede ser leída por un único *consumer* dentro de un mismo *consumer group*.
3. F Kafka almacena los mensajes para siempre, independientemente de la configuración.
4. V Kafka Connect permite integrar Kafka con sistemas externos sin necesidad de escribir código.
5. V El orden de los mensajes está garantizado dentro de una partición, pero no entre particiones.
6. V Kafka Streams permite el procesamiento de flujos de datos en tiempo real utilizando una API declarativa.
- 7 F En Kafka, un *consumer group* puede consumir múltiples veces el mismo mensaje si sus *consumers* fallan.

27. ¿Qué componente se utiliza para coordinar y gestionar la configuración del clúster en Kafka?

- a) Broker
- b) Producer
- c) Consumer
- d) Zookeeper**

28. ¿Cuál es una característica clave de Kafka frente a otros sistemas de mensajería?

- a) Almacenamiento de datos estructurados
- b) Transacciones ACID
- c) Alta escalabilidad horizontal**
- d) Procesamiento batch exclusivo

29. ¿Cuál es el propósito del parámetro acks en un Producer de Kafka?

- a) Especificar el tamaño del mensaje
- b) Controlar la compresión
- c) Determinar el número de réplicas que deben confirmar el mensaje**
- d) Definir la prioridad del mensaje

30. ¿Qué ventaja ofrece particionar un topic?

- a) Reduce la necesidad de brokers
- b) Incrementa la latencia
- c) Mejora la paralelización y escalabilidad**
- d) Simplifica la configuración del clúster

31. ¿Qué ocurre si un mensaje no puede ser entregado por un Producer y no se maneja el error?

- a) Kafka lo reintenta automáticamente
- b) Se pierde el mensaje**
- c) Se envía a todos los topics
- d) Kafka lo almacena para más tarde

32. ¿Qué API usarías para construir pipelines ETL sin escribir mucho código en Kafka?

- a) Kafka Streams
- b) Kafka CLI
- c) Kafka Connect**
- d) AdminClient

33. ¿Qué ocurre si se cae un broker que contiene una partición líder?

- a) Todos los mensajes se pierden
- b) Kafka lo ignora
- c) Se elige una nueva réplica como líder**
- d) Se detiene todo el clúster

34. ¿Cuál es la principal función del log de Kafka?

- a) Configurar brokers
- b) Almacenar el historial de consumo
- c) Almacenar mensajes de forma ordenada**
- d) Ejecutar tareas batch

35. ¿Qué sucede si un consumidor no comitea su offset?

- a) Kafka lo hace por defecto
- b) Volverá a leer los mensajes antiguos al reiniciarse**
- c) No podrá leer nuevos mensajes
- d) El broker se reinicia

36. ¿Qué permite la replicación en Kafka?

- a) Multiplicar los brokers
- b) Garantizar la disponibilidad de datos**
- c) Enviar mensajes más rápido
- d) Eliminar particiones duplicadas

37. ¿Cómo se define la durabilidad de los mensajes en Kafka?

- a) Por la política de retención**
- b) Por el tamaño del mensaje
- c) Por el número de consumidores
- d) Por el tipo de clave

38. ¿Qué es el “offset” en Kafka?

- a) Una dirección IP
- b) Un identificador de consumidor
- c) La posición de lectura de un consumidor en una partición**
- d) El tamaño de una partición

39. ¿Cuál es el propósito de los consumer groups?

- a) Compartir configuración
- b) Balancear la carga de consumo**
- c) Crear nuevos topics
- d) Recolectar logs

40. ¿Qué función cumple el commit de offset automático?

- a) Reinicia los consumidores
- b) Guarda automáticamente la posición de lectura**
- c) Aumenta la seguridad
- d) Borra los mensajes

41. ¿Qué pasa si hay más consumidores que particiones en un grupo?

- a) Todos consumen todos los mensajes
- b) Kafka lanza un error
- c) Se duplican las particiones
- d) Algunos consumidores estarán inactivos**

42. ¿Qué permite hacer la API Kafka Streams?

- a) Consumir mensajes sin transformación
- b) Configurar clusters
- c) Procesar flujos de datos con transformaciones**
- d) Guardar mensajes en bases de datos

43. ¿Qué es una partición en Kafka?

- a) Un proceso
- b) Una división de un topic para distribuir mensajes**
- c) Un archivo temporal
- d) Un tipo de consumidor

44. ¿Cuál es la latencia típica de Kafka en procesamiento de mensajes?

- a) Milisegundos**
- b) Minutos
- c) Horas
- d) Días

45. ¿Qué significa “at least once” en Kafka?

- a) Que un mensaje se entrega una vez como máximo**
- b) Que se puede perder un mensaje
- c) Que un mensaje puede entregarse más de una vez**
- d) Que no se guarda ningún mensaje

46. ¿Dónde se almacenan físicamente los mensajes en Kafka?

- a) En memoria
- b) En el filesystem del broker**
- c) En la nube
- d) En Redis

47. ¿Qué ocurre si un broker tiene muchas particiones y poca memoria?

- a) Todo funciona normalmente
- b) Kafka aumenta el número de topics
- c) Puede afectar el rendimiento**
- d) Kafka redirecciona el tráfico**

48. ¿Kafka garantiza el orden de los mensajes?

- a) Nunca
- b) Sí, en todo el topic
- c) Solo dentro de una partición**
- d) Solo si hay un consumidor

49. ¿Qué significa que Kafka es “distributed”?

- a) Que se ejecuta en un solo nodo
- b) Que usa múltiples máquinas para balancear carga y alta disponibilidad**
- c) Que depende de otros sistemas externos
- d) Que requiere Internet

50. ¿Para qué sirve el parámetro `retention.ms` en un topic de Kafka?

- a) Limitar el número de particiones
- b) Restringir acceso a consumidores
- c) Determinar cuánto tiempo se almacenan los mensajes**
- d) Cambiar el nombre del topic

51. ¿Qué tipo de datos puede manejar Kafka?

- a) Cualquier tipo de datos serializables**
- b) Imágenes únicamente
- c) Solo texto plano
- d) Solo JSON

52. ¿Cuál de los siguientes NO es un uso típico de Kafka?

- a) Monitorización de logs
- b) Reemplazo directo de bases de datos relacionales**
- c) Procesamiento en tiempo real
- d) Integración de sistemas

53. ¿Qué propiedad asegura que un mensaje no se duplique ni se pierda en Kafka (usando configuración adecuada)?

- a) Compresión
- b) Replicación
- c) Exactly-once delivery**
- d) Retención infinita

54. ¿Qué opción describe mejor a un "broker" en Kafka?

- a) Un archivo temporal
- b) Un consumidor especializado
- c) Un clúster virtual
- d) Un servidor que gestiona el almacenamiento y la transmisión de mensajes**

55. ¿Qué hace Kafka si un consumidor se desconecta inesperadamente?

- a) Borra su offset
- b) Envía sus mensajes a otro topic
- c) Reasigna las particiones a otros consumidores del grupo**
- d) Detiene el broker

56. ¿Cuál es el resultado de tener un número de particiones muy alto para un topic?

- a) Mejora sin límite el rendimiento
- b) Puede generar sobrecarga en los brokers y problemas de rendimiento**
- c) Duplica todos los mensajes
- d) Desactiva la replicación

57. ¿Qué componente permite transformar datos en tránsito sin consumir ni producir directamente?

- a) Topic
- b) Connector
- c) Middleware externo
- d) Kafka Streams Processo

58. ¿Qué significa la configuración `min.insync.replicas` en un topic de Kafka?

- a) Número mínimo de productores activos
- b) Cantidad mínima de réplicas que deben estar sincronizadas para permitir escrituras
- c) Número de brokers disponibles para lectura
- d) Mínimo de particiones activas por clúster

59. ¿Qué comando usarías con Kafka para crear un nuevo topic desde la terminal?

- a) kafka-start-topic
- b) kafka-create
- c) kafka-topics --create
- d) kafka-init --topic

60. ¿Qué ocurre si un topic no tiene configurada ninguna política de retención?

- a) Los mensajes se conservan para siempre
- b) Los mensajes se eliminan inmediatamente
- c) Kafka lanza una excepción
- d) El broker se apaga automáticamente