Cuestionario Flink

1. ¿Qué componente de Flink coordina la ejecución de un job?
a) TaskManager
b) ResourceManager
c) JobManager
d) Dispatcher

Respuesta correcta: c) JobManager

2. ¿Cuál es la función principal del TaskManager?
a) Coordinar el envío de jobs
b) Ejecutar tareas del flujo de datos
c) Asignar recursos del sistema
d) Gestionar el estado de los checkpoints
Respuesta correcta: b) Ejecutar tareas del flujo de datos

3. ¿Qué representa un Task Slot en Flink?
a) Un buffer de datos intermedios
b) Un hilo de ejecución y conjunto de recursos asignados
c) Una instancia del JobManager
d) Un nodo de red del clúster
Respuesta correcta: b) Un hilo de ejecución y conjunto de recursos asignados

4. ¿Qué componente expone la interfaz REST para enviar trabajos en Flink?
a) JobManager
b) Dispatcher
c) Client
d) ResourceManager
Respuesta correcta: b) Dispatcher

5. ¿Qué componente es responsable de solicitar recursos al sistema subyacente (por ejemplo, Kubernetes, YARN)?
a) JobManager
b) TaskManager
c) Dispatcher
d) ResourceManager
Respuesta correcta: d) ResourceManager

7. ¿Qué representa el DataFlow Graph en Flink?
a) La arquitectura física del clúster
b) El uso de memoria
c) El plan lógico de ejecución del programa
d) La topología de red
Respuesta correcta: c) El plan lógico de ejecución del programa

8. ¿Qué hace el cliente (Client) en Flink?
a) Ejecuta tareas de procesamiento
b) Administra recursos del clúster

c) Genera y envía el JobGraph al JobManager
d) Monitorea los TaskManagers
Respuesta correcta: c) Genera y envía el JobGraph al JobManager

9.- ¿Cuál es el rol primario de JobManager en la arquitectura de Apache Flink?
a) ejecutar las tareas de un dataflow
b) asignar recursos informáticos
c) coordinar la ejecución de aplicaciones Flink
d) gestionar el almacenamiento temporal (buffering) e intercambio de flujos de datos

Correcta: c

10.- ¿Cuántos TaskManagers se requieren como mínimo en un clúster de Flink?
a) cero
b) uno
c) dos
d) depende del tamaño de los datos

Correcta: uno, al menos necesita uno.

11.- ¿Cuál es la función de los slots de tareas (*task slots*) en los TaskManagers de Flink?
a) almacenar temporalmente los flujos de datos
b) representar un subconjunto de recursos para la ejecución de tareas
c) ejecutar las órdenes del JobManager
d) optimizar el análisis y la optimización de datos

Correcta: b

12.- En la ejecución distribuida de Flink, ¿cuál es el propósito de encadenar subtareas de operadores entre sí?
a) reducir el tráfico de red
b) aumentar la complejidad del grafo de flujo de datos
c) optimizar el rendimiento al reducir la sobrecarga del traspaso entre hilos (*thread-to-thread handover*)
d) simplificar el modelo de programación

Correcta: c

13.- ¿Qué representa un Task Slot en la arquitectura de Apache Flink?
a) una unidad de asignación de memoria
b) una unidad de trabajo ejecutada en una CPU
c) un mecanismo de almacenamiento temporal de datos
d) un canal de comunicación entre nodos

Correcta: b

14.- ¿Qué componente de la arquitectura de Flink gestiona la asignación de recursos en el clúster?
a) TaskManager

b) JobManager
c) ResourceManager
d) Dispatcher

Respuesta correcta: c) ResourceManager

15.- ¿ Cuál es el rol del Dispatcher en Apache Flink?
a) ejecutar tareas
b) proporcionar una interfaz REST para el envío de trabajos
c) asignar recursos para tareas
d) gestionar el almacenamiento temporal (buffering) de flujos de datos

Respuesta correcta: b) proporcionar una interfaz REST para el envío de trabajos

16.-¿Cómo contribuye el JobMaster de Flink al clúster?
a) ejecuta tareas de flujo de datos
b) almacena temporalmente los flujos de datos
c) gestiona la ejecución de un único JobGraph
d) asigna memoria para las tareas
Respuesta correcta: c) gestiona la ejecución de un único JobGraph

17.- ¿Cuál es el rol del Cliente (*Client*) en la arquitectura de Flink?
a) gestionar los *task slots*
b) ejecutar tareas
c) preparar y enviar un flujo de datos (*dataflow*) al JobManager
d) asignar recursos
Respuesta correcta: c) preparar y enviar un flujo de datos al JobManager

18.- ¿Qué representa el Grafo de Flujo de Datos (*DataFlow Graph*) de Flink?
a) la disposición física del clúster
b) el esquema que guía la ejecución de un programa
c) la asignación de memoria para las tareas
d) la topología de red del clúster
Respuesta correcta: b) el esquema que guía la ejecución de un programa

---

19.- ¿Qué tipo de entorno se configura en el código con EnvironmentSettings.in_batch_mode()?

A) Un entorno de streaming
B) Un entorno en tiempo real
C) Un entorno por lotes (batch)
D) Un entorno de pruebas unitarias

Respuesta: C) Un entorno por lotes (batch)

20.- ¿Qué salida produce el siguiente fragmento si se ejecuta correctamente?

```
orders = table_env.from_elements(
   [('Jack', 'FRANCE', 10), ('Rose', 'ENGLAND', 30), ('Jack', 'FRANCE', 20)],
   ['name', 'country', 'revenue']
)
orders.execute().print()
```

A) Una tabla con solo los nombres de los clientes
B) Una tabla con los nombres, países e ingresos de cada fila
C) Una tabla con la suma total de ingresos
D) Un error porque falta una función de agregación

Respuesta: B) Una tabla con los nombres, países e ingresos de cada fila

21.- ¿Qué hace este fragmento de código?

```
orders \
   .where(col("country") == 'FRANCE') \
   .group_by(col("name")) \
   .select(col("name"), col("revenue").sum.alias('rev_sum'))
```

A) Agrupa a todos los clientes sin filtrar
B) Agrupa a los clientes por país y suma sus ingresos
C) Filtra los clientes de Francia, agrupa por nombre y suma ingresos
D) Ordena los clientes franceses por nombre

Respuesta: C) Filtra los clientes de Francia, agrupa por nombre y suma ingresos

22.- ¿Qué cambio se debe hacer para agrupar por país en lugar de por nombre?

```
.group_by(col("name"))
.select(col("name"), col("revenue").sum.alias('rev_sum'))
```

A) Cambiar "name" por "revenue"
B) Usar .distinct(col("country"))
C) Reemplazar col("name") por col("country") en ambas líneas
D) Agregar .filter_by(col("country"))

Respuesta: C) Reemplazar col("name") por col("country") en ambas líneas

23.- ¿Qué ocurre si ejecutamos solo este fragmento?

```
from pyflink.table import EnvironmentSettings, TableEnvironment
env_settings = EnvironmentSettings.in_batch_mode()
```

```
table_env = TableEnvironment.create(env_settings)
```

A) Crea un flujo de datos para streaming
B) Inicia un entorno de ejecución en modo batch
C) Conecta automáticamente con una base de datos externa
D) Lanza un error porque falta una fuente de datos

Respuesta: B) Inicia un entorno de ejecución en modo batch

24.- ¿Qué sucede si se omite el método .execute().print() en el siguiente código?

```
revenue = orders \
   .where(col("country") == 'FRANCE') \
   .group_by(col("name")) \
   .select(col("name"), col("revenue").sum.alias('rev_sum'))
# Falta: revenue.execute().print()
```

A) El código lanza una excepción inmediatamente
B) No se realiza ninguna ejecución y no se ve salida
C) La tabla se guarda automáticamente en un archivo
D) Se imprime una tabla vacía

Respuesta: B) No se realiza ninguna ejecución y no se ve salida

25.- ¿Qué efecto tiene el siguiente código en la tabla orders?

```
map_function = udf(
   lambda x: pd.concat([x.name, x.revenue * 10], axis=1),
   result_type=DataTypes.ROW([
      DataTypes.FIELD("name", DataTypes.STRING()),
      DataTypes.FIELD("revenue", DataTypes.BIGINT())
   ]),
   func_type="pandas"
)
```

A) Crea una función que agrupa los datos por nombre y calcula su media
B) Define una función UDF que multiplica el campo revenue por 10 usando pandas
C) Aplica una transformación SQL sobre la tabla
D) Convierte la tabla en un DataFrame de Pandas

Respuesta: B) Define una función UDF que multiplica el campo revenue por 10 usando pandas

26.- ¿Por qué se especifica func_type="pandas" en la definición de la UDF?

func_type="pandas"

A) Porque solo se puede usar UDFs con Pandas en Flink
B) Para que la función use operaciones vectorizadas más eficientes en batch
C) Para convertir automáticamente los datos a strings
D) Porque revenue solo puede manipularse con Pandas DataFrames

Respuesta: B) Para que la función use operaciones vectorizadas más eficientes en batch

27.- ¿Qué hace este fragmento?

orders.map(map_function).execute().print()

A) Aplica un filtro a los datos usando la función definida
B) Llama a la función map_function y muestra el resultado agrupado
C) Aplica la función UDF definida a cada fila de la tabla y muestra los resultados
D) Ordena los resultados antes de imprimirlos

Respuesta: C) Aplica la función UDF definida a cada fila de la tabla y muestra los resultados

28.- ¿Para qué sirve este fragmento en el contexto del entorno de ejecución?

```
env_settings = EnvironmentSettings.in_batch_mode()
table_env = TableEnvironment.create(env_settings)
```

A) Permite trabajar con archivos CSV como entrada
B) Inicia un entorno de ejecución en tiempo real
C) Configura el entorno de tabla para procesar datos finitos en modo batch
D) Inicializa el entorno para procesamiento de gráficos en red

Respuesta: C) Configura el entorno de tabla para procesar datos finitos en modo batch

29.- ¿Qué error potencial puede tener esta UDF?

lambda x: pd.concat([x.name, x.revenue * 10], axis=1)

A) x.name y x.revenue no están en el mismo DataFrame
B) No se puede multiplicar texto por un número en Pandas
C) concat no es compatible con Flink
D) axis=1 no es válido en esta operación

Respuesta: A) x.name y x.revenue no están en el mismo DataFrame

30.- ¿Qué ocurre al ejecutar este fragmento?

```
CREATE TABLE random_source (
   id BIGINT,
   data TINYINT
) WITH (
   'connector' = 'datagen',
   'fields.id.kind' = 'sequence',
   'fields.id.start' = '1',
   'fields.id.end' = '8'
)
```

A) Se genera una tabla de entrada que obtiene datos de una base de datos
B) Se crea una fuente estática que solo genera valores aleatorios
C) Se simula una fuente de datos en streaming con IDs secuenciales del 1 al 8
D) Se lee un archivo CSV en modo batch

Respuesta: C) Se simula una fuente de datos en streaming con IDs secuenciales del 1 al 8

31.- ¿Qué propósito cumple este fragmento?

```
SELECT id / 2 AS id_res, data FROM random_source
```

A) Agrupa los datos por pares
B) Divide el ID por 2 y lo guarda como nuevo campo llamado id_res
C) Aplica un filtro solo a los IDs pares
D) Elimina las filas que tienen datos duplicados

Respuesta: B) Divide el ID por 2 y lo guarda como nuevo campo llamado id_res

32.- ¿Qué hace el siguiente bloque SQL en conjunto?

```
INSERT INTO print_sink
   SELECT id_res, SUM(data) AS data_sum FROM
      (SELECT id / 2 AS id_res, data FROM random_source)
   WHERE id_res > 1
   GROUP BY id_res
```

A) Inserta todos los datos sin transformación en el print_sink
B) Agrupa los datos por ID y cuenta cuántos hay
C) Filtra los datos donde el ID dividido por 2 es mayor que 1 y suma sus valores
D) Multiplica el campo data y lo imprime en consola

Respuesta: C) Filtra los datos donde el ID dividido por 2 es mayor que 1 y suma sus valores

33.- ¿Cuál es el rol del siguiente fragmento?

table_env.execute_sql(""" CREATE TABLE print_sink (...) WITH ('connector' = 'print') """)

A) Imprime errores de la tabla
B) Muestra los datos agrupados en un archivo
C) Define una tabla de salida que imprime resultados en consola
D) Exporta la tabla como imagen

Respuesta: C) Define una tabla de salida que imprime resultados en consola

34.- ¿Por qué se utiliza .wait() al final del script?

table_env.execute_sql("INSERT INTO ...").wait()

A) Para bloquear el flujo de datos hasta que se cierre el entorno
B) Para esperar a que los datos estén completamente cargados desde un archivo
C) Para asegurar que la operación SQL se complete antes de terminar el programa
D) Porque en streaming se deben procesar todos los registros de forma sincrónica

Respuesta: C) Para asegurar que la operación SQL se complete antes de terminar el programa