

## Actividad 4 - Módulo 4

**Nombre:** Carlos Saldivia Susperreguy

### 1. Preguntas de desarrollo

**a) ¿Qué diferencia principal tiene la arquitectura de Cassandra respecto a las bases de datos relacionales tradicionales?**

La principal diferencia es que Cassandra tiene una arquitectura distribuida peer-to-peer sin nodo maestro, donde todos los nodos son iguales, eliminando puntos únicos de fallo. En contraste, las bases de datos relacionales tradicionales suelen tener una arquitectura centralizada con un servidor principal. Además, Cassandra está orientada a columnas y diseñada para escalabilidad horizontal automática, mientras que las bases relacionales están diseñadas principalmente para escalabilidad vertical.

**b) Explica con tus palabras qué es un keyspace y su función dentro de Cassandra.**

Un keyspace es el equivalente a una base de datos en Cassandra. Es un contenedor lógico donde se crean y almacenan las tablas. Su función principal es definir la estrategia de replicación y el factor de replicación, determinando cuántas copias de cada dato se almacenarán en el clúster y cómo se distribuirán entre los nodos.

**c) Menciona y describe dos ventajas y dos desventajas del uso de Cassandra.**

**Ventajas:**

1. **Alta disponibilidad:** Si un nodo falla, el clúster sigue funcionando sin interrupciones, manteniendo la disponibilidad de los datos en todo momento.
2. **Escalabilidad horizontal:** Cassandra puede escalar fácilmente agregando nodos sin interrupciones, lo que permite manejar grandes volúmenes de datos.

**Desventajas:**

1. **Curva de aprendizaje:** Más compleja que las bases de datos relacionales tradicionales, requiriendo un cambio de mentalidad en el diseño y modelado de datos.
2. **No soporta JOINS ni subconsultas:** Las relaciones entre tablas se modelan de otra forma, generalmente mediante la desnormalización y duplicación estratégica de datos.

#### d) ¿Por qué es importante el diseño de la clave primaria en Cassandra?

El diseño de la clave primaria en Cassandra es crítico para el rendimiento, ya que define cómo se almacenan y distribuyen los datos en el clúster. Un mal diseño puede llevar a particiones desbalanceadas, hotspots o dificultades en las consultas por rango.

#### e) ¿Qué significa que Cassandra sea "orientada a columnas"? ¿Cómo impacta esto en el almacenamiento y las consultas?

Que Cassandra sea orientada a columnas significa que los datos se organizan en familias de columnas en lugar de filas tradicionales. Cada fila puede tener columnas distintas, proporcionando gran flexibilidad.

Esto impacta en el almacenamiento optimizando el acceso a columnas específicas y en las consultas permitiendo una lectura más eficiente cuando solo se necesitan ciertas columnas, especialmente útil para grandes volúmenes de datos analíticos.

## 2. Análisis de escenario

### a) Diseña una tabla adecuada en CQL para almacenar esta información:

```
CREATE TABLE lecturas_temperatura (  
    sensor_id text,  
    fecha_hora timestamp,  
    ciudad text,  
    temperatura float,  
    PRIMARY KEY (sensor_id, fecha_hora)  
);
```

### b) ¿Qué tipo de clave primaria usarías y por qué?

Usaría una clave primaria compuesta:

**Clave de partición:** sensor\_id

**Clave de clustering:** fecha\_hora

Esta estructura permite:

- Particionar los datos por sensor, distribuyendo la carga entre nodos
- Ordenar automáticamente las lecturas por fecha dentro de cada partición
- Consultar eficientemente por sensor específico y rangos de fechas

**c) ¿Qué recomendaciones de modelado de Cassandra aplicarías para este caso?**

1. **Modelar según consultas:** La tabla está diseñada específicamente para las consultas requeridas (por sensor\_id y rango de fechas).
2. **Evitar particiones muy grandes:** Si un sensor genera demasiados datos, podría considerarse incluir la fecha (año/mes) en la clave de partición.
3. **Desnormalización:** Incluir la ciudad en cada registro para evitar JOINS, aunque signifique duplicar esta información.