



INF-325 Bases de Datos Avanzadas: Tarea 1 Apache Cassandra

Ernesto Barría Andrade - ernesto.barria@usm.cl

Camilo Díaz Galaz – camilo.diazg@usm.cl

Sebastián Gutiérrez Milla –
sebastian.gutierrezmi@usm.cl

Carlos Lagos Cortes – carlos.lagosc@usm.cl

Luis Zegarra Stuardo – Luis.zegarra@usm.cl

Tarea en GitHub: <https://github.com/luphin/Tarea-Cassandra-Grupo4-BaseDatosAvanzada>

Link video presentación: <https://youtu.be/XcMnFvB96tM>

GRUPO 4

viernes, 13 de septiembre de
2024

Resumen

El documento describe la implementación de consultas en Cassandra y su conexión con Power BI para la visualización de datos. Se realizaron pruebas en Excel y Cassandra para validar la precisión de las consultas, obteniendo resultados consistentes en la mayoría de los casos. En el desarrollo del Requisito 4, se configuró el ODBC "CData" para conectar Power BI con Cassandra, y se validó la correcta visualización de los datos.

En el Requisito 5, se probó el nivel de consistencia QUORUM en Cassandra para garantizar la aceptación de los datos por varios nodos. Se demostró la alta disponibilidad al simular la caída de nodos: Power BI mantuvo la conexión siempre que al menos un nodo estaba activo, perdiéndola solo cuando el nodo principal fallaba.

En conclusión, la implementación del clúster Cassandra con tres nodos y el uso de Power BI fue exitosa, demostrando consistencia, alta disponibilidad y una visualización eficiente de los datos.

1 Introducción

Con el paso del tiempo, el almacenamiento de estructuras de datos ha experimentado una evolución significativa, principalmente debido a la creciente cantidad de datos que es necesario procesar y gestionar en las bases de datos. Existen tres tipos principales de bases de datos:

1. SQL: Son bases de datos relacionales que permiten la gestión de datos mediante relaciones entre ellos, facilitando así las consultas. Este tipo de bases de datos tiene un escalado vertical, lo que significa que su principal limitación está en la capacidad de mejorar la máquina en la que se ejecutan.

2. NoSQL: Las bases de datos no relacionales, conocidas como NoSQL, se comenzaron a utilizar por primera vez en 1998 y fueron formalizadas en el año 2000. Estas bases de datos son más eficientes en el manejo de grandes volúmenes de datos debido a su escalado horizontal, lo que permite almacenar la información en particiones o réplicas distribuidas en diferentes nodos. Actualmente, son más utilizadas y preferidas que las bases de datos SQL, ya que, además de mejorar en aspectos como la frecuencia y latencia de las consultas, ofrecen diversas formas de almacenar la información, como pares clave-valor, bases orientadas a documentos, grafos y familias de columnas. [1,2,3,4]

3. NewSQL: Este es un nuevo sistema de gestión de bases de datos (DBMS), introducido por primera vez en 2011. NewSQL combina el enfoque relacional de SQL con la escalabilidad horizontal de NoSQL, al tiempo que mejora la consistencia, un aspecto en el que las bases de datos NoSQL suelen presentar limitaciones (según el teorema CAP). Esto se logra mediante la implementación de los enfoques BASE y ACID. [4,5]

En este informe se utilizará una base de datos tipo NoSQL, específicamente [Apache Cassandra](#), implementando una arquitectura de clúster con un centro de datos Cassandra compuesto por 3 nodos operativos sobre Docker y un Factor de Replicación de 3. [6]

¿Qué significa esto? La arquitectura de clúster se basa en la integración de un conjunto de nodos o instancias, que en Cassandra se visualizan como un anillo. En estos nodos se realiza la replicación de la información, lo que permite que el sistema siga funcionando incluso si uno de los nodos deja de estar disponible, ya que cada nodo es independiente de los demás. Cabe destacar que, a medida que se incrementa el número de nodos, también aumentan los requisitos de memoria RAM y CPU. [7]

El contexto del problema a resolver se enfoca en el diseño, poblamiento y consulta de una base de datos creada en Cassandra [6], utilizando un dataset que contiene información sobre los registros de postulaciones y matrículas efectivas en un determinado periodo de tiempo en una institución educativa. Después de realizar un análisis OLAP, se identificaron 16 campos en el dataset, los cuales se dividen en categóricos y numéricos, descritos a continuación:

Campos categóricos:

- CEDULA: RUT identificador del postulante.



- PERIODO: Año del registro (2015, 2016, 2017).
- SEXO: Género del postulante (MASCULINO, FEMENINO).
- PREFERENCIA: Orden de preferencia en la postulación (1 a 10).
- CARRERA: Lista de carreras ofrecidas por la UCM.
- ESTADO: Estado de la postulación (MATRICULADO, NO MATRICULADO).
- FACULTAD: Facultades de la UCM.
- GRUPO_DEPEN: Dependencia del establecimiento (MUNICIPAL, PARTICULAR SUBVENCIONADO, PARTICULAR PAGADO).
- REGION: Nombre de la región en Chile.
- PACE: Participación en el programa PACE (PACE, Blanco).
- GRATUIDAD: Indicación de gratuidad (SI, NO).

Campos numéricos:

- PUNTAJE: Puntaje ponderado PSU.
- LATITUD: Latitud de la región.
- LONGITUD: Longitud de la región.
- PTJE_NEM: Puntaje de Enseñanza Media.
- PSU_PROMLM: Puntaje promedio de Lenguaje y Matemáticas.

Una vez cargados los datos en los 3 nodos, se deben resolver los siguientes requerimientos:

1. Implementar el clúster.
2. Repartir los datos en el clúster y disminuir la cantidad de particiones a leer.
3. Realizar consultas sobre los datos con CQL.
4. Establecer la conexión con [Power BI Desktop](#) para la visualización de las consultas.
5. Demostrar consistencia y alta disponibilidad.

Cada uno de estos requerimientos será explicado, detallando las instrucciones y su implementación en el siguiente punto.

2. Desarrollo

2.1 Desarrollo Requisito 1

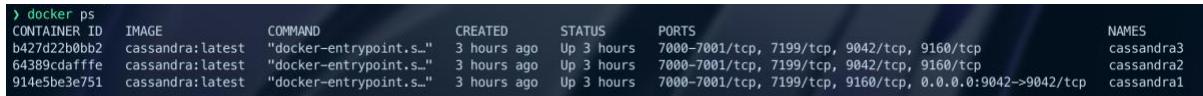
Se solicita implementar un clúster utilizando la estrategia simple, lo que implica contar con tres nodos operativos en un entorno Docker y un factor de replicación de 3. Los pasos para llevar a cabo esta tarea son los siguientes:

1. Iniciar Cassandra en el entorno de Docker ejecutando el comando ``docker network create cassandra-cluster``.
2. Crear el archivo ``docker-compose.yml`` para la definición de los nodos. Las características de este archivo son las siguientes:
 - Versión: 3.8.
 - Imagen de Cassandra utilizada: ``latest``.
 - Nombre del contenedor para cada nodo: [cassandra1, cassandra2, cassandra3].
 - Cada contenedor tiene su propio entorno configurado con el nombre del clúster, las semillas, el parámetro ``start_rpc``, el tamaño máximo de la memoria heap y el tamaño inicial de la memoria heap.
 - El puerto en el que está disponible cada nodo.
 - La red utilizada: ``cassandra-net``.
 - Límite de memoria establecido en 2 GB (esto se debe a que, al no establecer un límite, Cassandra utilizaba toda la memoria disponible).

- Volúmenes para el almacenamiento de los datos.
- Por último, se configura un puente en la red para conectar los nodos.

3. Levantar el clúster utilizando el comando `docker-compose` en la consola.

Se accede al clúster mediante comandos para crear un KEYSPACE destinado a la carga de datos. Al crear este KEYSPACE, se utilizó el siguiente comando para definir un factor de replicación de 3: "CREATE KEYSPACE mikeyspace WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 3};". [8]



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b427d22b0bb2	cassandra:latest	"docker-entrypoint.s..."	3 hours ago	Up 3 hours	7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp	cassandra3
64389cdaaffe	cassandra:latest	"docker-entrypoint.s..."	3 hours ago	Up 3 hours	7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp	cassandra2
914e5be3e751	cassandra:latest	"docker-entrypoint.s..."	3 hours ago	Up 3 hours	7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0.0:9042->9042/tcp	cassandra1

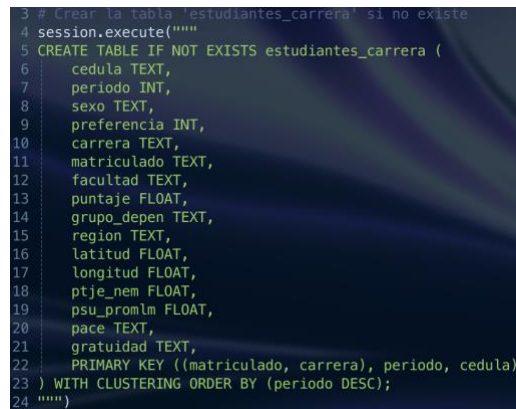
Figura 1. Contenedores en Docker de cada nodo UP.

2.2 Desarrollo Requisito 2

Se utilizaron 2 diseños de tablas diferentes que serán explicados a continuación.

Diseño estudiantes_carrera:

Este diseño fue implementado para satisfacer las consultas del requisito 3.A, su estructura es la siguiente:



```
3 # Crear la tabla 'estudiantes_carrera' si no existe
4 session.execute("""
5 CREATE TABLE IF NOT EXISTS estudiantes_carrera (
6     cedula TEXT,
7     periodo INT,
8     sexo TEXT,
9     preferencia INT,
10    carrera TEXT,
11    matriculado TEXT,
12    facultad TEXT,
13    puntaje FLOAT,
14    grupo_depen TEXT,
15    region TEXT,
16    latitud FLOAT,
17    longitud FLOAT,
18    ptje_nem FLOAT,
19    psu_promlm FLOAT,
20    pace TEXT,
21    gratuidad TEXT,
22    PRIMARY KEY ((matriculado, carrera), periodo, cedula)
23 ) WITH CLUSTERING ORDER BY (periodo DESC);
24 """)
```

Figura 2. Diseño CQL de estudiantes_carrera

- Las claves de partición son: matricula y carrera.
- Las claves de clustering son: periodo y cedula.
 - Cedula se agregó para que sea el diferenciador de cada valor, ya que es único.
- El ordenamiento del clustering se especifica mediante el comando `WITH CLUSTERING`, que indica que, dentro de cada partición, los registros deben ordenarse por periodo en orden descendente.

Diseño estudiantes_region:

Este diseño fue implementado para satisfacer las consultas del requisito 3.B, su estructura es la siguiente:

```
3 # Crear la tabla 'estudiantes_region' si no existe
4 session.execute("""
5 CREATE TABLE IF NOT EXISTS estudiantes_region (
6     cedula TEXT,
7     periodo INT,
8     sexo TEXT,
9     preferencia INT,
10    carrera TEXT,
11    matriculado TEXT,
12    facultad TEXT,
13    puntaje FLOAT,
14    grupo_depen TEXT,
15    region TEXT,
16    latitud FLOAT,
17    longitud FLOAT,
18    ptje_nem FLOAT,
19    psu_promlm FLOAT,
20    pace TEXT,
21    gratuidad TEXT,
22    PRIMARY KEY ((matriculado, carrera), region, periodo, cedula)
23 ) WITH CLUSTERING ORDER BY (region DESC, periodo DESC);
24 """)
25
```

Figura 3. Diseño CQL de estudiantes_region

- Las claves de partición son: matricula y carrera.
- Las claves de clustering son: region, periodo y cedula.
 - Cedula se agregó para que sea el diferenciador de cada valor, ya que es único.
- El ordenamiento del clustering se especifica mediante el comando `WITH CLUSTERING`, que indica que, dentro de cada partición, los registros deben ordenarse primero por region en orden descendente y luego por periodo en orden descendente.

Diseño estudiantes_facultad:

Este diseño fue implementado para satisfacer las consultas del requisito 3.C, su estructura es la siguiente:

```
3 # Crear la tabla 'estudiantes_facultad' para la consulta 3 si no existe
4 session.execute("""
5 CREATE TABLE IF NOT EXISTS estudiantes_facultad (
6     cedula TEXT,
7     periodo INT,
8     sexo TEXT,
9     preferencia INT,
10    carrera TEXT,
11    matriculado TEXT,
12    facultad TEXT,
13    puntaje FLOAT,
14    grupo_depen TEXT,
15    region TEXT,
16    latitud FLOAT,
17    longitud FLOAT,
18    ptje_nem FLOAT,
19    psu_promlm FLOAT,
20    pace TEXT,
21    gratuidad TEXT,
22    PRIMARY KEY ((matriculado, facultad), puntaje, cedula)
23 ) WITH CLUSTERING ORDER BY (puntaje DESC);
24 """)
25
```

Figura 4. Diseño CQL de estudiantes_facultad

- Las claves de partición son: matricula y facultad.
- Las claves de clustering son: puntaje y cedula.
 - Cedula se agregó para que sea el diferenciador de cada valor, ya que es único.
- El ordenamiento del clustering se especifica mediante el comando `WITH CLUSTERING`, que indica que, dentro de cada partición, los registros deben ordenarse por puntaje en orden descendente.

Estos diseños se aplican dentro de un script en Python, en este se lee el archivo `postulaciones.xlsx`, recuperar los datos y agregarlos a los nodos, mediante el uso de las librerías de python: [pandas](#) (manejo de datos), [cassandra-driver](#) y [openpyxl](#) (librería de escritura/lectura de archivos excel).

Se puede observar que las tablas se llaman `estudiantes_carrera`, `estudiantes_region` y `estudiantes_facultad`, contienen todas las columnas solicitadas en las instrucciones de la

tarea. En la figura 5, 6, y7, se puede ver que se han cargado 16,651 líneas cargada en todas las tablas.

Para los campos que no contengan datos, es decir, aquellos que estén vacíos, se definió un valor por defecto: `` para los textos y `0` para los números. En las figura 8 y 9 se puede evidenciar que la carga dentro de los nodos es similar.

```
root@93016ec078be:~# nodetool tablestats mikeyspace
Total number of tables: 3

-----
Keyspace: mikeyspace
  Read Count: 10
  Read Latency: 1.4350999999999998 ms
  Write Count: 49953
  Write Latency: 0.037848717794727045 ms
  Pending Flushes: 0
    Table: estudiantes_carrera
      SSTable count: 0
      Old SSTable count: 0
      Max SSTable size: 0B
      Space used (live): 0
      Space used (total): 0
      Space used by snapshots (total): 0
      Off heap memory used (total): 0
      SSTable Compression Ratio: -1.00000
      Number of partitions (estimate): 58
      Memtable cell count: 16651
      Memtable data size: 5947457
      Memtable off heap memory used: 0
      Memtable switch count: 0
      Speculative retries: 0
      Local read count: 0
      Local read latency: NaN ms
      Local write count: 16651
      Local write latency: NaN ms
      Local read/write ratio: 0.00000
      Pending flushes: 0
      Percent repaired: 100.0
      Bytes repaired: 0B
      Bytes unrepaired: 0B
      Bytes pending repair: 0B
      Bloom filter false positives: 0
      Bloom filter false ratio: 0.00000
      Bloom filter space used: 0
      Bloom filter off heap memory used: 0
      Index summary off heap memory used: 0
      Compression metadata off heap memory used: 0
      Compacted partition minimum bytes: 0
      Compacted partition maximum bytes: 0
      Compacted partition mean bytes: 0
      Average live cells per slice (last five minutes): NaN
      Maximum live cells per slice (last five minutes): 0
      Average tombstones per slice (last five minutes): NaN
      Maximum tombstones per slice (last five minutes): 0
      Droppable tombstone ratio: 0.00000
```

Figura 5. Estadísticas tabla
estudiantes_carrera

```
Table: estudiantes_region
SSTable count: 0
Old SSTable count: 0
Max SSTable size: 0B
Space used (live): 0
Space used (total): 0
Space used by snapshots (total): 0
Off heap memory used (total): 0
SSTable Compression Ratio: -1.00000
Number of partitions (estimate): 58
Memtable cell count: 16651
Memtable data size: 5614437
Memtable off heap memory used: 0
Memtable switch count: 0
Speculative retries: 0
Local read count: 1
Local read latency: NaN ms
Local write count: 16651
Local write latency: NaN ms
Local read/write ratio: 0.00006
Pending flushes: 0
Percent repaired: 100.0
Bytes repaired: 0B
Bytes unrepaired: 0B
Bytes pending repair: 0B
Bloom filter false positives: 0
Bloom filter false ratio: 0.00000
Bloom filter space used: 0
Bloom filter off heap memory used: 0
Index summary off heap memory used: 0
Compression metadata off heap memory used: 0
Compacted partition minimum bytes: 0
Compacted partition maximum bytes: 0
Compacted partition mean bytes: 0
Average live cells per slice (last five minutes): NaN
Maximum live cells per slice (last five minutes): 0
Average tombstones per slice (last five minutes): NaN
Maximum tombstones per slice (last five minutes): 0
Droppable tombstone ratio: 0.00000
```

Figura 6. Estadísticas tabla
estudiantes_facultad

```
Table: estudiantes_facultad
SSTable count: 0
Old SSTable count: 0
Max SSTable size: 0B
Space used (live): 0
Space used (total): 0
Space used by snapshots (total): 0
Off heap memory used (total): 0
SSTable Compression Ratio: -1.00000
Number of partitions (estimate): 15
Memtable cell count: 16651
Memtable data size: 5407001
Memtable off heap memory used: 0
Memtable switch count: 0
Speculative retries: 0
Local read count: 9
Local read latency: NaN ms
Local write count: 16651
Local write latency: NaN ms
Local read/write ratio: 0.00054
Pending flushes: 0
Percent repaired: 100.0
Bytes repaired: 0B
Bytes unrepaired: 0B
Bytes pending repair: 0B
Bloom filter false positives: 0
Bloom filter false ratio: 0.00000
Bloom filter space used: 0
Bloom filter off heap memory used: 0
Index summary off heap memory used: 0
Compression metadata off heap memory used: 0
Compacted partition minimum bytes: 0
Compacted partition maximum bytes: 0
Compacted partition mean bytes: 0
Average live cells per slice (last five minutes): NaN
Maximum live cells per slice (last five minutes): 0
Average tombstones per slice (last five minutes): NaN
Maximum tombstones per slice (last five minutes): 0
Droppable tombstone ratio: 0.00000
```

Figura 7. Estadísticas tabla
estudiantes_facultad

```
root@93016ec078be:~# nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens     Owns (effective)  Host ID                               Rack
UN  172.20.0.3    806.1 KiB     16         100.0%            0b6147ec-2af8-4cd0-9724-6a69bff2cdc2  rack1
UN  172.20.0.2    816.79 KiB   16         100.0%            1f95d0d3-1eed-4479-b9f1-56e3218385a4  rack1
UN  172.20.0.4    805.28 KiB   16         100.0%            4d7a3641-2f2f-4715-9c3d-d72154c65e85  rack1
```

Figura 8. Estado de los nodos

```

root@93016ec078be:/# nodetool rings
nodetool: Found unexpected parameters: [rings]
See 'nodetool help' or 'nodetool help <command>'.
root@93016ec078be:/# nodetool ring

```

Datcenter: datacenter1

Address	Rack	Status	State	Load	Owns	Token
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	8717333109442339795
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-9084675580240443939
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-8633369064032273531
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-8066824175498816923
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-7542071953340861271
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-72491322428566129
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-6697695305848899514
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	-6173396361660886589
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-575632583824467327
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-5437480668949102948
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-5257427075946316274
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	-4701886237643476324
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-4358107861501591291
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-418592645606283462
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-3589355815992132213
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-3168075034863856446
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-311458785820999317
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	-2645852832305317313
172.20.0.2	rack1	Up	Normal	805.28 KIB	100.00%	-2350403122251505652
172.20.0.4	rack1	Up	Normal	816.79 KIB	100.00%	-2147223612742463819
172.20.0.2	rack1	Up	Normal	806.1 KIB	100.00%	-1666127580966435597
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	-1274647241825926112
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	-831510186780549579
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	-557219060228565181
172.20.0.2	rack1	Up	Normal	805.28 KIB	100.00%	-538731548315032472
172.20.0.4	rack1	Up	Normal	816.79 KIB	100.00%	-14211334852684132
172.20.0.2	rack1	Up	Normal	806.1 KIB	100.00%	111842263622871780
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	540641259132569185
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	1061804235468478115
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	1467872780220920271
172.20.0.2	rack1	Up	Normal	805.28 KIB	100.00%	1704161264813161157
172.20.0.4	rack1	Up	Normal	816.79 KIB	100.00%	2639494687984592191
172.20.0.2	rack1	Up	Normal	806.1 KIB	100.00%	2459294445357062162
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	2751525507891696656
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	3154594328691533523
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	34123617856688980658
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	381850398086179319
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	434153806405595294
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	4914625880843117608
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	5538564791548553683
172.20.0.2	rack1	Up	Normal	805.28 KIB	100.00%	596655356182638591
172.20.0.4	rack1	Up	Normal	816.79 KIB	100.00%	608922990725329867
172.20.0.2	rack1	Up	Normal	806.1 KIB	100.00%	6507235135574911425
172.20.0.4	rack1	Up	Normal	816.79 KIB	100.00%	6850299647226491625
172.20.0.2	rack1	Up	Normal	805.28 KIB	100.00%	693869888618752667
172.20.0.4	rack1	Up	Normal	806.1 KIB	100.00%	739682398388273376
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	7759815960718998100
172.20.0.4	rack1	Up	Normal	805.28 KIB	100.00%	8290567877129074863
172.20.0.2	rack1	Up	Normal	816.79 KIB	100.00%	8717333109442339795

Figura 9. Estado del anillo en el Cluster

Enfrentamos diferentes problemas para lograr entender como funcionaban las claves de particiones y clustering. Primero se trató de crear una única tabla con la estructura `PRIMARY KEY((carrera, matriculado, facultad, región) puntaje, periodo)`, debido al bajo conocimiento dentro de la sintaxis CQL, no llegamos muy lejos después de cargar los datos, ya que la primera llamada para el requisito 3.A no funcionaba y necesitaba del `ALLOW FILTERING` [9] para funcionar. Sin embargo, luego descubrimos la impresionante explicación de Carlos Bertuccini en StackOverflow sobre “la diferencia entre partition key, composite key y clustering key en cassandra” [10], donde logramos comprender como funcionaba el tema de las particiones y descubrimos que lo que estábamos creando no era correcto en lógica. Por esto tratamos de crear 2 tablas que tuvieran las claves necesarias para cumplir con cada requisito, utilizando una tabla para las consultas de 3.A y 3.B y otra para 3.C, logramos resultados exitosos en el retorno de los datos al hacer las queries, pero decidimos cambiarlos porque no cumplían explícitamente con las instrucciones (cumplía, pero al momento de ordenar, se ordenaban por 2 claves a la vez), esta solución era una buena idea ya que evitaba replicar tres veces la información dentro de los nodos. También tuvimos que reconsiderar el uso de las claves de clustering, ya que en ocasiones se presentaban problemas al intentar ordenar por puntaje, lo que dificultaba la realización de consultas. Por último definimos crear tres tablas, una para cada requisito, esto para cumplir con lo solicitado explícitamente dentro de las instrucciones.

2.3 Desarrollo Requisito 3

- Devolver todos los postulantes matriculados en la carrera de medicina ordenados por periodo.

En este caso, la consulta diseñada es:



```
```sql
```

```
SELECT * FROM estudiantes_carrera WHERE matriculado = 'SI' AND carrera = 'MEDICINA';```
```

Esto retorna la siguiente tabla con 182 filas:

Figura 10. Parte 2 de la tabla para carrera medicina

Para ver la tabla completa ir a [repositorio GitHub](#).

- b. Devolver todos los postulantes matriculados provenientes de la región del Maule en la carrera Ingeniería Civil Informática ordenados por periodo.

En este caso la consulta realizada es:

```
```sql
```

```
SELECT * FROM estudiantes_region WHERE matriculado = 'SI' AND carrera = 'INGENIERÍA CIVIL INFORMÁTICA' AND region = 'MAULE';```
```

La tabla retornada es la siguiente con 92 filas:

Figura 11. Tabla con filtro matriculados en facultad de ciencias

- En este caso la consulta realizada es:

```
SELECT * FROM estudiantes_facultad WHERE matriculado = 'SI' AND facultad = 'CIENCIAS DE LA SALUD';``
```

La tabla obtenida tiene 824 filas:

(824 rows)

Figura 12. Tabla parte final de la consulta 3.C

Para ver la tabla completa ir a [repositorio GitHub](#).

Para comprobar la efectividad de las consultas, en la siguiente tabla, se presentan las consultas directamente realizadas en el archivo Exel, donde se obtuvieron los mismos resultados para la primeras 2 consultas y en la tercera hay una diferencia de 1 dato.

Búsqueda en Excel por función	Resultado
<code>=COUNTIFS(F1:F16652; "SI";E1:E16652;"MEDICINA")</code>	182
<code>=COUNTIFS(F1:F16652; "SI";E1:E16652;"INGENIERÍA CIVIL INFORMÁTICA";J1:J16652;"MAULE")</code>	92
<code>=COUNTIFS(F1:F16652; "SI";G1:G16652;"CIENCIAS DE LA SALUD")</code>	825

2.4 Desarrollo Requisito 4

Para realizar la conexión a Cassandra desde Power BI se instaló el ODBC “CData” y se configuró en localhost con el puerto 9042.

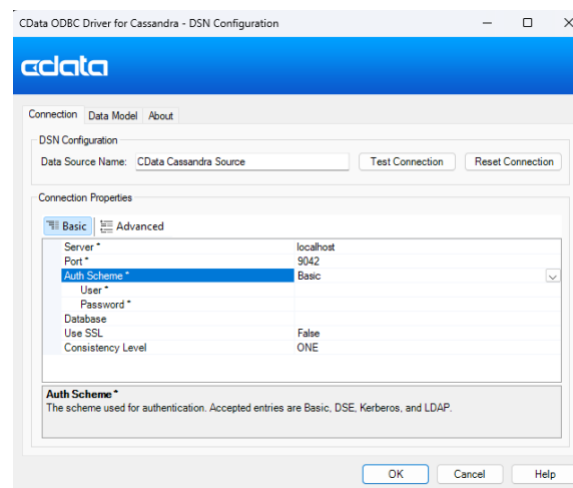
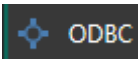


Figura 13. Configuración ODBC “CData”

Una vez configurado el ODBC, desde Power BI se realiza la conexión siguiendo los siguientes pasos:

1. Presionar “Get Data”.
2. Seleccionar “ODBC”.
3. Escribir la query correspondiente en opciones avanzadas.





From ODBC

Data source name (DSN)
CData Cassandra Source

Advanced options
Connection string (non-credential properties) (optional) ☐
Example: ODBC;

SQL statement (optional)
SELECT * FROM mikespace.estudiantes_facultad WHERE matriculado = 'SI'
AND facultad = 'CIENCIAS DE LA SALUD';

Supported row reduction clauses (optional)
(None) ☐ Detect

OK Cancel

OK

Load

4. Presionar “OK”.

5. Presionar “Load”.

Al realizar las 3 consultas, se obtienen los siguientes resultados:

carretera	cedula	matriculado	periodo	facultad	gratuidad	grupo dapem	latitud	longitud	pase	preferencia	pm-presencia	plje-som	postaje	region	sexo
MEDICINA	1943108	SI	2017	MEDICINA	SI	MUNICIPAL	-35.600000155273	-71.7419967801387	2	6885	764	75130	MALE	FEMENINO	
MEDICINA	1951638	SI	2017	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-34.97801778002	-71.2239960234375	2	7140	775	74600	MALE	FEMENINO	
MEDICINA	1961973	SI	2017	MEDICINA	NO	PARTICULAR INGRESADO	-34.97801778002	-71.2239960234375	2	7140	775	74600	MALE	FEMENINO	
MEDICINA	1969627	SI	2017	MEDICINA	NO	MUNICIPAL	-35.420000126834	-71.656997686641	1	7045	740	74600	MALE	FEMENINO	
MEDICINA	1969716	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-34.97801778002	-71.2239960234375	2	7145	762	75000	MALE	FEMENINO	
MEDICINA	1971837	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-34.97801778002	-71.2239960234375	2	7080	750	75675	MALE	FEMENINO	
MEDICINA	1974381	SI	2017	MEDICINA	NO	MUNICIPAL	-35.420000126834	-71.656997686641	2	7155	734	74200	MALE	FEMENINO	
MEDICINA	1975682	SI	2017	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-35.846000195484	-72.3119817407754	1	6955	760	74600	MALE	FEMENINO	
MEDICINA	1976485	SI	2017	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.2239960234375	2	6790	760	75000	MALE	FEMENINO	
MEDICINA	1984990	SI	2017	MEDICINA	NO	MUNICIPAL	-34.97801778002	-71.2239960234375	1	6845	805	73900	MALE	FEMENINO	
MEDICINA	1985911	SI	2017	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	6790	781	74765	MALE	FEMENINO	
MEDICINA	1987379	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.420000126834	-71.656997686641	2	7250	781	74740	MALE	FEMENINO	
MEDICINA	1990380	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.4499991591809	-71.8217980819194	2	7145	770	74600	MALE	FEMENINO	
MEDICINA	2007170	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	2	6890	764	74400	MALE	FEMENINO	
MEDICINA	2009638	SI	2017	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	7045	791	72770	MALE	MASCULINO	
MEDICINA	2009521	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.420000126834	-71.656997686641	1	6885	754	74200	MALE	MASCULINO	
MEDICINA	2013971	SI	2017	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	7050	725	72720	MALE	FEMENINO	
MEDICINA	2037633	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.51598848332	-71.571980861914	3	7275	760	77780	MALE	MASCULINO	
MEDICINA	2036679	SI	2017	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-35.420000126834	-71.656997686641	2	7015	764	74600	MALE	FEMENINO	
MEDICINA	2037170	SI	2017	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	2	6890	764	74400	MALE	FEMENINO	
MEDICINA	2010885	SI	2017	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-35.9646007019043	-72.3170013407754	1	6840	785	75075	MALE	FEMENINO	
MEDICINA	1947406	SI	2016	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-35.4499991591809	-71.8217980819194	2	7265	764	74600	MALE	FEMENINO	
MEDICINA	1947520	SI	2016	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	6740	795	74885	MALE	FEMENINO	
MEDICINA	1947513	SI	2016	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	7105	760	74695	MALE	MASCULINO	
MEDICINA	1947188	SI	2016	MEDICINA	SI	MUNICIPAL	-35.423000126834	-71.656997686641	1	6775	781	74490	MALE	MASCULINO	
MEDICINA	1951670	SI	2016	MEDICINA	NO	PARTICULAR INGRESADO	-34.97801778002	-71.2239960234375	2	7175	754	74730	MALE	FEMENINO	
MEDICINA	1951655	SI	2016	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	7105	760	74695	MALE	MASCULINO	
MEDICINA	1951638	SI	2016	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-35.600000155273	-71.7419967801387	1	6595	764	73885	MALE	FEMENINO	
MEDICINA	1960978	SI	2016	MEDICINA	SI	MUNICIPAL	-34.92000054802	-71.819997686641	2	7175	754	74730	MALE	MASCULINO	
MEDICINA	1961034	SI	2016	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-34.97801778002	-71.2239960234375	2	6870	764	74690	MALE	FEMENINO	
MEDICINA	1964472	SI	2016	MEDICINA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	2	7125	750	74615	MALE	FEMENINO	
MEDICINA	1965189	SI	2016	MEDICINA	SI	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	2	6840	740	74760	MALE	MASCULINO	
MEDICINA	1967227	SI	2016	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	2	6725	754	74730	MALE	MASCULINO	
MEDICINA	1969624	SI	2016	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	7075	785	76255	MALE	FEMENINO	
MEDICINA	1969427	SI	2016	MEDICINA	NO	MUNICIPAL	-35.423000126834	-71.656997686641	1	7240	740	73885	MALE	FEMENINO	
MEDICINA	1967224	SI	2016	MEDICINA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	2	7455	720	77465	MALE	MASCULINO	
MEDICINA	1971089	SI	2016	MEDICINA	SI	PARTICULAR SUBVENCIONADO	-35.3360007019043	-72.414014464638	2	6855	775	74640	MALE	FEMENINO	

Figura 14. Consulta A

cedula	carretera	periodo	matriculado	region	facultad	gratuidad	grupo dapem	latitud	longitud	pase	preferencia	pm-presencia	plje-som	postaje	sexo
1946695	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.963999622773	-71.68199808343	1	5180	631	58830	FEMENINO	
1951469	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.51598848332	-71.571980861914	3	6210	610	62610	MASCULINO	
1951670	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.963999622773	-71.68199808343	1	5790	686	63400	MASCULINO	
1951151	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.506999622773	-72.733602643177	1	6230	682	61070	MASCULINO	
1960883	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	5980	538	57470	MASCULINO	
1951410	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.51598848332	-71.571980861914	3	5510	750	66880	MASCULINO	
1971913	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.4499991591809	-71.8217980819194	2	5825	528	56530	MASCULINO	
1960890	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.423000126834	-71.656997686641	1	6250	607	60880	MASCULINO	
1960817	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	6170	647	62130	MASCULINO	
1960815	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.51598848332	-71.571980861914	3	6185	661	61830	MASCULINO	
1960450	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.4499991591809	-71.8217980819194	2	6775	486	56660	MASCULINO	
1961751	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.84999841211	-71.544999644737	1	6015	637	62010	MASCULINO	
1967784	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.963999622773	-71.68199808343	1	5480	476	52200	MASCULINO	
1961911	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.963999622773	-71.68199808343	1	5810	539	58110	MASCULINO	
1961887	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.709000918207	-72.526998170698	1	5130	589	56710	MASCULINO	
1951690	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.716999530851	-71.25	1	5570	617	61230	FEMENINO	
1960972	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.4499991591809	-71.8217980819194	2	5470	389	58230	MASCULINO	
2005403	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.84999841211	-71.544999644737	1	5880	523	59110	MASCULINO	
2007078	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	5805	544	53670	MASCULINO	
2005885	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	6000	595	59510	MASCULINO	
2023989	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-34.97801778002	-71.2239960234375	1	6420	647	65410	MASCULINO	
2006544	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.423000126834	-71.656997686641	1	5380	585	55330	MASCULINO	
2009844	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR INGRESADO	-35.423000126834	-71.656997686641	1	6000	595	59510	MASCULINO	
2017964	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	5250	617	57720	MASCULINO	
2011690	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	6180	548	57740	MASCULINO	
2114686	INGENIERIA CIVIL INFORMÁTICA	2017	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	6785	607	66880	MASCULINO	
1980905	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	6400	610	61420	MASCULINO	
1961660	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	5830	641	60335	MASCULINO	
1951470	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	PARTICULAR SUBVENCIONADO	-35.77980320308	-71.644999644737	1	5765	539	57680	MASCULINO	
1951671	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.423000126834	-71.656997686641	1	5810	642	60335	MASCULINO	
1960979	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	NO	MUNICIPAL	-35.90000120879	-71.789998308133	1	5485	488	50385	MASCULINO	
1960983	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.51598848332	-71.571980861914	3	6425	637	63825	FEMENINO	
1961794	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.423000126834	-71.656997686641	1	6000	607	62130	MASCULINO	
1947444	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	NO	PARTICULAR SUBVENCIONADO	-35.84999841211	-71.544999644737	1	5480	476	52200	MASCULINO	
1951478	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.963999622773	-71.68199808343	1	5810	539	58110	MASCULINO	
1951473	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.84999841211	-71.544999644737	1	5880	523	59110	MASCULINO	
1951475	INGENIERIA CIVIL INFORMÁTICA	2016	SI	MALE	CENCAS DE LA INGENIERIA	SI	MUNICIPAL	-35.716999530851	-71.25	1	5570	617	61230	FEMENINO	



cedula	facultad	matriculador	programa	carrera	matricula	grado	departamento	latitud	longitud	peso	promedio	profesional	peso_promedio	psicologo	sexo
000000	Ciencias de la Salud	SI	7700	Enfermería	SI	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	6017	714	MADE	FIANANNO	M
000001	Ciencias de la Salud	SI	6800	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	6100	713	MADE	FIANANNO	M
000002	Ciencias de la Salud	SI	6700	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	5800	701	MADE	FIANANNO	M
000003	Ciencias de la Salud	SI	6600	Enfermería	SI	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	5821	719	MADE	FIANANNO	M
000004	Ciencias de la Salud	SI	6500	Enfermería	SI	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	6100	692	MADE	FIANANNO	M
000005	Ciencias de la Salud	SI	6400	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2017	1	6000	690	MADE	MASCUPADO	M
000006	Ciencias de la Salud	SI	6300	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	6100	689	MADE	FIANANNO	M
000007	Ciencias de la Salud	SI	6200	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	6100	708	MADE	MASCUPADO	M
000008	Ciencias de la Salud	SI	6100	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2015	1	6100	695	MADE	MASCUPADO	M
000009	Ciencias de la Salud	SI	6000	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	5811	722	MADE	FIANANNO	M
000010	Ciencias de la Salud	SI	5900	Enfermería	SI	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	6045	706	MADE	FIANANNO	M
000011	Ciencias de la Salud	SI	5800	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2017	1	6020	701	MADE	FIANANNO	M
000012	Ciencias de la Salud	SI	5700	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	5821	692	MADE	FIANANNO	M
000013	Ciencias de la Salud	SI	5600	Enfermería	SI	MUNICIPAL	15.433000155634	-71.65000708647	2017	1	6100	690	MADE	FIANANNO	M
000014	Ciencias de la Salud	SI	5500	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	5830	682	MADE	FIANANNO	M
000015	Ciencias de la Salud	SI	5400	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2015	1	6100	690	MADE	MASCUPADO	M
000016	Ciencias de la Salud	SI	5300	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	6100	687	MADE	FIANANNO	M
000017	Ciencias de la Salud	SI	5200	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	6100	691	MADE	FIANANNO	M
000018	Ciencias de la Salud	SI	5100	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	6100	688	MADE	FIANANNO	M
000019	Ciencias de la Salud	SI	5000	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	5700	690	MADE	FIANANNO	M
000020	Ciencias de la Salud	SI	4900	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	5800	719	MADE	FIANANNO	M
000021	Ciencias de la Salud	SI	4800	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	5800	692	MADE	FIANANNO	M
000022	Ciencias de la Salud	SI	4700	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2016	1	6100	691	MADE	FIANANNO	M
000023	Ciencias de la Salud	SI	4600	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	6100	677	MADE	FIANANNO	M
000024	Ciencias de la Salud	SI	4500	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	5800	678	MADE	FIANANNO	M
000025	Ciencias de la Salud	SI	4400	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	5800	678	MADE	FIANANNO	M
000026	Ciencias de la Salud	SI	4300	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	6045	702	MADE	FIANANNO	M
000027	Ciencias de la Salud	SI	4200	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	6017	692	MADE	FIANANNO	M
000028	Ciencias de la Salud	SI	4100	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2015	1	6020	581	MADE	MASCUPADO	M
000029	Ciencias de la Salud	SI	4000	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	5821	581	MADE	FIANANNO	M
000030	Ciencias de la Salud	SI	3900	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	5815	657	MADE	MASCUPADO	M
000031	Ciencias de la Salud	SI	3800	Enfermería	NO	MUNICIPAL	15.433000155634	-71.65000708647	2017	1	6000	581	MADE	MASCUPADO	M
000032	Ciencias de la Salud	SI	3700	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	5800	647	MADE	MASCUPADO	M
000033	Ciencias de la Salud	SI	3600	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2017	1	6100	657	MADE	FIANANNO	M
000034	Ciencias de la Salud	SI	3500	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	5810	652	MADE	FIANANNO	M
000035	Ciencias de la Salud	SI	3400	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2015	1	5800	651	MADE	MASCUPADO	M
000036	Ciencias de la Salud	SI	3300	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	5800	657	MADE	MASCUPADO	M
000037	Ciencias de la Salud	SI	3200	Enfermería	NO	INTEGRAR SUBVENCIONADO	15.433000155634	-71.65000708647	2016	1	5800	647	MADE	FIANANNO	M

Figura 16 Consulta C

2.5 Desarrollo Requisito 5

2.5.1 Consistencia

Código `consistency.py` que agregar un nuevo valor a la tabla y luego hace la consulta ([link](#) [GitHub](#)). Se aplica `ConsistencyLevel.QUORUM` para establecer que el valor escrito/leído, sea aceptado por quorum entre los nodos. [11,12]

```
12 from cassandra.cluster import Cluster
13 from cassandra import ConsistencyLevel
14
15 # Configuración de la conexión a la base de datos
16 cluster = Cluster(['127.0.0.1']) # Cambia la IP por la de tu nodo Cassandra
17 session = cluster.connect('mikeyspace') # Reemplaza con el keyspace que usas
18
19 # Establecer el nivel de consistencia a QUORUM
20 session.default_consistency_level = ConsistencyLevel.QUORUM
21
22 # Consulta de inserción: Insertar datos en la tabla
23 insert_query = """
24 INSERT INTO estudiantes (cedula, periodo, sexo, preferencia, carrera, matriculador, facultad, puntaje, grado, departamento, latitud, longitud, peso_prom, psicologo, pac)
25 VALUES ('12345678-9', 2024, 'MASCULINO', 1, 'MEDICINA', 'SI', 'CIENCIAS DE LA SALUD', 650.5, 'MUNICIPAL', 'NM', -33.4378, -78.5505, 558, 886, 'SI', 'SI');
26 """
27
28 # Ejecutar la consulta de inserción
29 session.execute(insert_query)
30
31 # Confirmar que los datos fueron insertados (opcional)
32 print("Datos insertados correctamente.")
33
34 # Consulta de lectura: Seleccionar datos de la tabla
35 select_query = """
36 SELECT cedula, facultad, matriculador, puntaje, carrera, sexo
37 FROM estudiantes_facultad
38 WHERE matriculador = 'SI' AND facultad = 'CIENCIAS DE LA SALUD'
39 AND puntaje >= 500;
40 """
41
42 # Ejecutar la consulta de lectura
43 rows = session.execute(select_query)
44
45 # Mostrar los resultados
46 for row in rows:
47     print(row)
```

Figura 17. Código de consistency.py

Se ejecuta el código consistency.py

```
Row(cedula='19696333', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=54140.0, carrera='NUTRICION Y DIETETICA', sexo='MASCULINO')
Row(cedula='19696643', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=53860.0, carrera='ENFERMERIA', sexo='FEMENINO')
Row(cedula='19390264', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=53000.0, carrera='NUTRICION Y DIETETICA', sexo='FEMENINO')
Row(cedula='19658446', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=52580.0, carrera='KINESIOLOGIA', sexo='FEMENINO')
Row(cedula='19695433', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=52530.0, carrera='PSICOLOGIA', sexo='FEMENINO')
Row(cedula='19695497', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=52240.0, carrera='KINESIOLOGIA', sexo='MASCULINO')
Row(cedula='19308014', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=52100.0, carrera='NUTRICION Y DIETETICA', sexo='FEMENINO')
Row(cedula='12345678-9', facultad='CIENCIAS DE LA SALUD', matriculador='SI', puntaje=650.5, carrera='MEDICINA', sexo='MASCULINO')
```

Figura 18. Resultado al ejecutar el archivo consistency.py

2.5.2 Alta Disponibilidad

Al apagar el nodo 3 como se observa en la figura 12, el cliente de Power BI mantiene su conexión con el servidor.



Figura 19. Nodo 3 apagado

Al apagar el nodo 2 como se observa en la figura 13, el cliente de Power BI mantiene su conexión con el servidor.

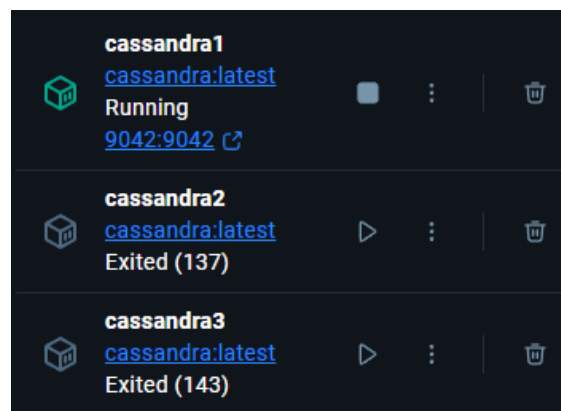


Figura 20. Nodos 2 y 3 apagados

Al apagar el nodo 1 como se observa en la figura 14, el cliente de Power BI pierde su conexión con el servidor, como se puede apreciar en la figura 15.

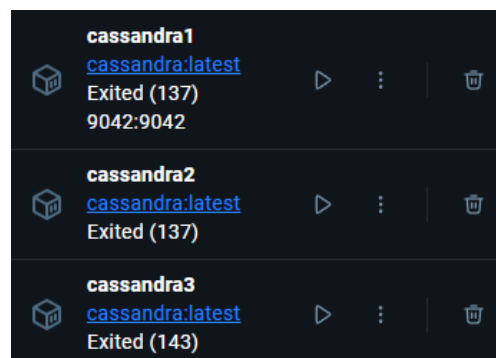


Figura 21. Nodos 2, 3 y 4 apagados

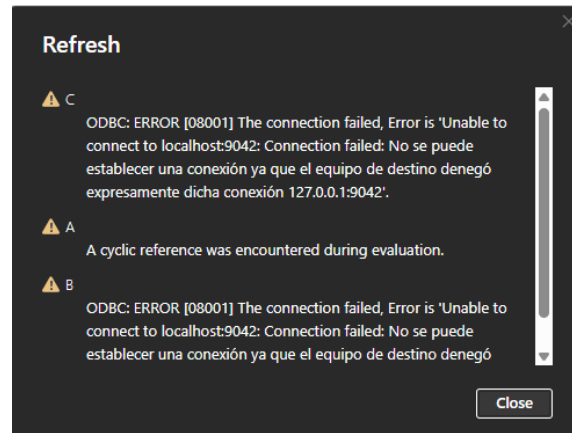


Figura 23. Conexión perdida de Power BI al servidor de Cassandra.

3 Conclusiones

Implementar el clúster Cassandra con tres nodos operativos en Docker ha sido exitoso. Configurar un Factor de Replicación de 3 garantiza que los datos se repliquen entre los nodos de forma eficiente, brindando alta disponibilidad y tolerancia a fallos. El diseño de las tablas, basado en las claves de partición y clustering, se ajusta a las consultas requeridas, optimizando el acceso a la información. Además, realizar la carga de datos mediante un script en Python, utilizando las bibliotecas adecuadas, permitió cargar 16,651 registros en las tablas sin errores. Se ha demostrado la consistencia de los datos entre los nodos mediante el nivel de consistencia QUORUM, y también la alta disponibilidad al simular la caída de un nodo, observando que el clúster sigue respondiendo sin interrupciones para el usuario. Conectar Power BI a Cassandra ha permitido visualizar los datos de forma coherente y eficiente, validando la efectividad de la arquitectura y el diseño de las tablas para realizar consultas en tiempo real. En conjunto, optimizar el rendimiento del clúster con tres nodos ha demostrado que el sistema es capaz de escalar sin comprometer la integridad ni el tiempo de respuesta. En resumen, cumplir con los requisitos de consistencia y alta disponibilidad, así como la integración con Power BI, ha permitido una visualización fluida y precisa de los datos.



4 Referencias Bibliográficas

1. Figueroa Colarte, M. (n.d.). *Bases de datos Avanzadas*. INF-325 Bases de Datos Avanzadas. Universidad Técnica Federico Santa María.
2. Meier, A., Kaufmann, M., Meier, A., & Kaufmann, M. (2019). Nosql databases. *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management*, 201-218.
3. Chen, J. K., & Lee, W. Z. (2018, December). A study of NoSQL Database for enterprises. In *2018 International Symposium on Computer, Consumer and Control (IS3C)* (pp. 436-440). IEEE.
4. Praschl, C., Pritz, S., Krauss, O., & Harrer, M. (2022, November). A comparison of relational, NoSQL and NewSQL database management systems for the persistence of time series data. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)* (pp. 1-6). IEEE.
5. Pavlo, A., & Aslett, M. (2016). *What's really new with NewSQL?*. *ACM Sigmod Record*, 45(2), 45-55.
6. Figueroa Colarte, M. (n.d.). *Tarea de laboratorio 1: Cassandra*. INF-325 Bases de Datos Avanzadas. Universidad Técnica Federico Santa María.
7. Apache Software Foundation. (n.d.). *Cassandra basics*. Apache Cassandra. https://cassandra.apache.org/_/cassandra-basics.html (Foundation, n.d.).
8. Apache Software Foundation. (n.d.). *Apache Cassandra CQL documentation*. <https://cassandra.apache.org/doc/stable/cassandra/cql/>
9. Apache Software Foundation. (n.d.). *Cassandra Query Language (CQL) documentation: SELECT with allow filtering*. Apache Cassandra. https://cassandra.apache.org/doc/4.1/cassandra/cql/cql_singlefile.html#selectAllowFiltering
10. Bertuccini, C. (2014, Julio 25). *Difference between partition key, composite key and clustering key in Cassandra?*. StackOverflow. <https://stackoverflow.com/a/24953331>
11. Gorbenko, A., Romanovsky, A., & Tarasyuk, O. (2020). Interplaying Cassandra NoSQL consistency and performance: A benchmarking approach. In *Dependable Computing-EDCC 2020 Workshops: AI4RAILS, DREAMS, DSOGR, SERENE 2020*, Munich, Germany, September 7, 2020, Proceedings 16 (pp. 168-184). Springer International Publishing.
12. DataStax. (n.d.). *How is the consistency level configured?* DataStax Documentation. <https://docs.datastax.com/en/cassandra-oss/3.0/cassandra/dml/dmlConfigConsistency.html>