

TAREA 1

APACHE CASSANDRA

Grupo 4

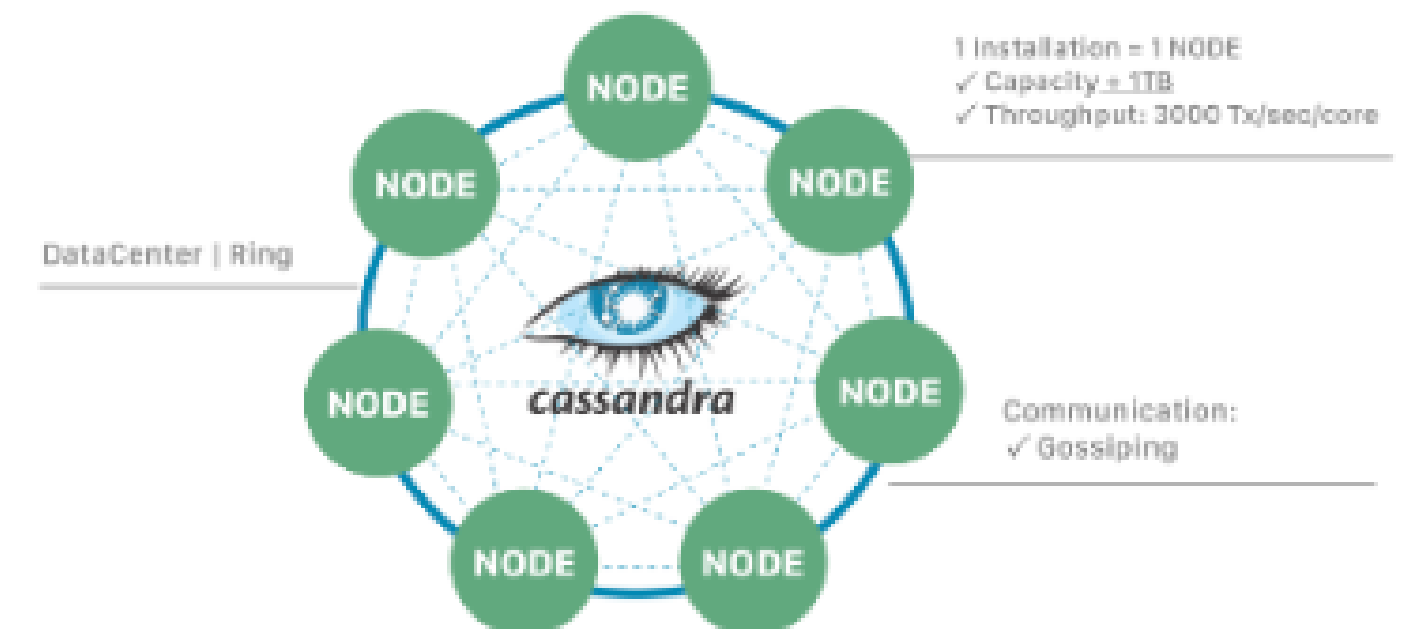


Ernesto Barría A.
Camilo Díaz G.
Sebastián Gutiérrez M.
Carlos Lagos C.
Luis Zegarra S.

APACHE CASSANDRA

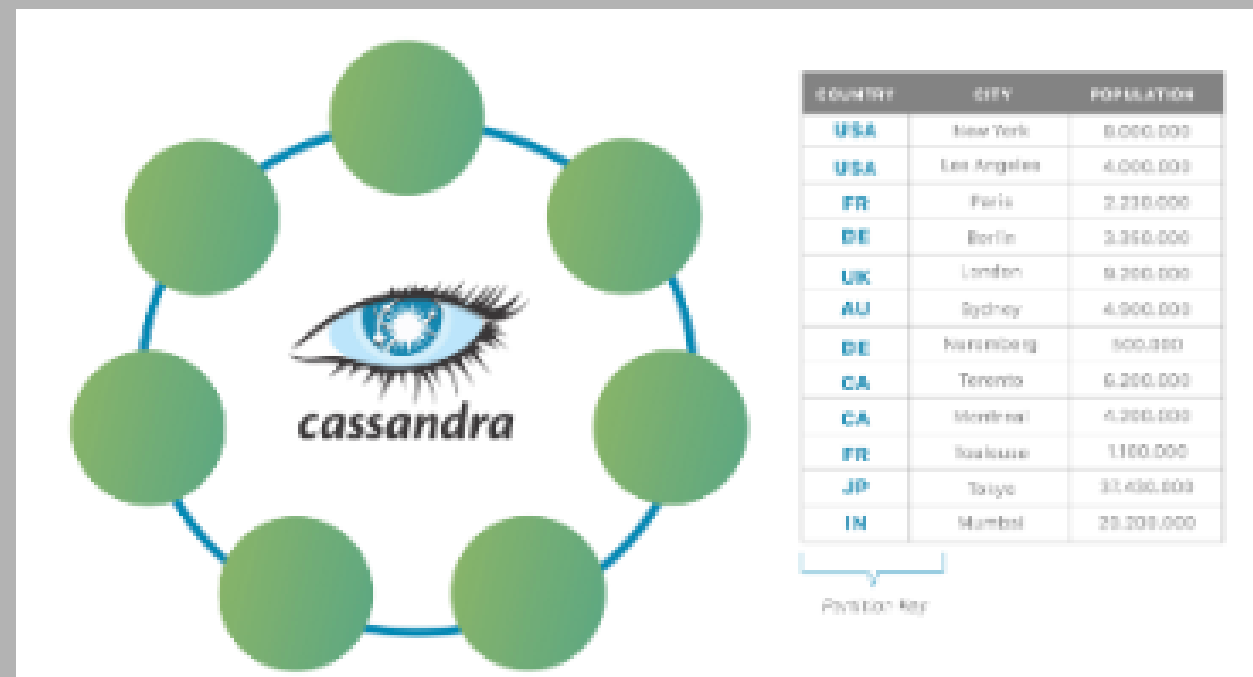
- DBMS NoSQL distribuido.
- Características:
 - No relacional.
 - Ampliamente distribuido.
 - Las bases de datos son livianas.
 - Familia de columnas.

ApacheCassandra™ = NoSQL Distributed Database

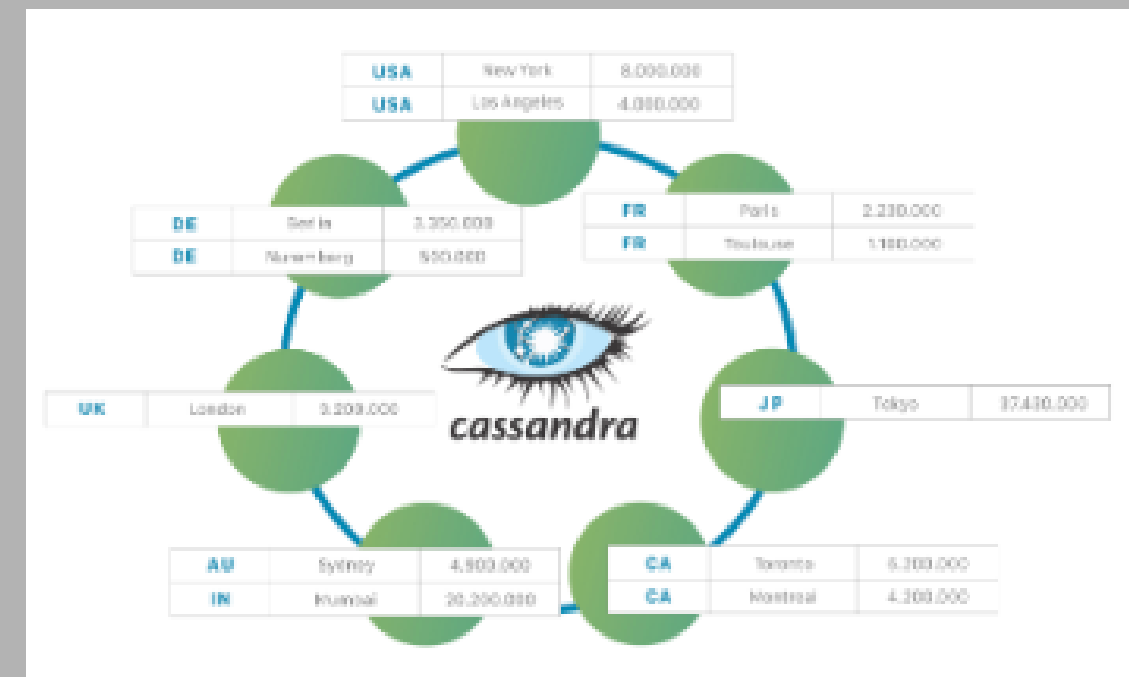


OBJETIVO

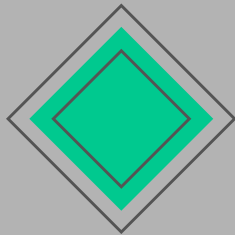
Implementar un clúster de Cassandra con 3 nodos para almacenar y gestionar datos de postulaciones.



lograr



CONFIGURACIÓN



Uso de Docker

- Trabajar con la imagen de Cassandra.
- Contruir el clúster y nodos.

Comandos utilizados:

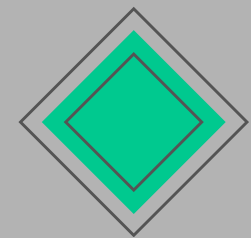
- `docker network create cassandra-cluster`
- `docker-compose.yml`

Características de `docker-compose.yml`

- Versión 3.8
- Imagen de Cassandra: latest
- Nodos creados: `cassandra1`, `cassandra2`, `cassandra3`.
- Configuración del entorno: nombre del clúster, semillas, memoria y límites.
- Uso de volúmenes



DISEÑO DEL KEYSPACE

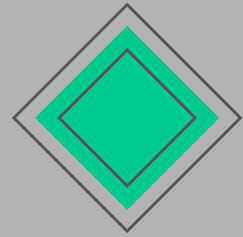


Creación del KEYSpace con factor de replicación 3:

```
``` CREATE KEYSpace mikeyspace WITH REPLICATION =  
{'class': 'SimpleStrategy', 'replication_factor': 3}; ```
```

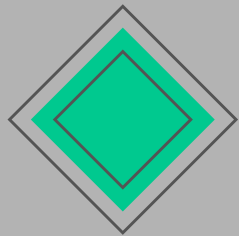


# ARCHIVO .PY

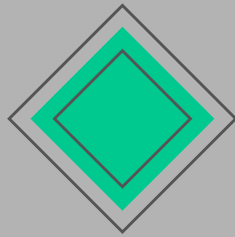


## Librerías:

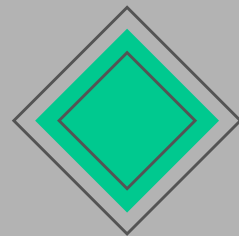
- Pandas
- cassandra-driver
- openpyxl



Valores por defecto.



Asignar valores desde tabla en pandas.



Diseño CQL tabla tablas.

- Claves de partición: carrera y matriculado.
- Claves de clustering: periodo y puntaje.
- Ordenamiento de clusteing:
  - a.puntaje descendente.
  - b.periodo ascendente.

# DISEÑO CQL

## Tabla estudiantes\_carrera

## Tabla estudiantes\_region

- 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.
- 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.

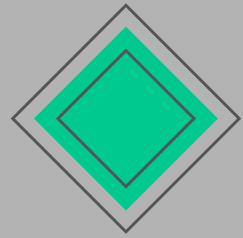
## Tabla estudiantes\_facultad

- 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.

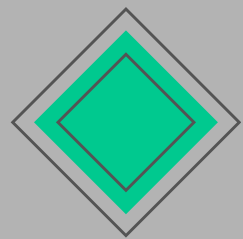
# PROBLEMAS



Problemas iniciales con las claves de partición y clustering:



Se intentó crear una tabla con la estructura  
`PRIMARY KEY((carrera, matriculado, facultad,  
región), puntaje, periodo).`

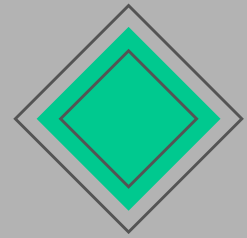


La falta de conocimiento en la sintaxis CQL  
impidió el éxito al consultar los datos, ya que las  
querys no funcionaban sin usar `ALLOW  
FILTERING.`

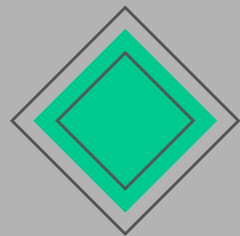




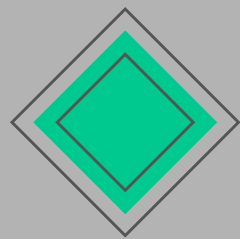
# PROBLEMAS



Solución al problema de las claves de partición y clustering:



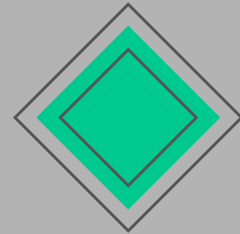
Una explicación de Carlos Bertuccini en StackOverflow fue clave para entender la diferencia entre partition key, composite key, y clustering key en Cassandra. [3]



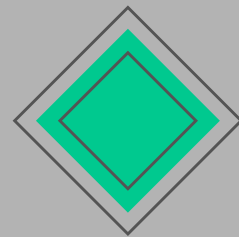
Se descubrió que la estructura inicial no era adecuada para el esquema de partición necesario.



# PROBLEMAS



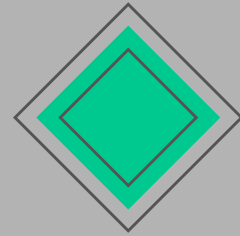
Problemas con las claves de clustering:



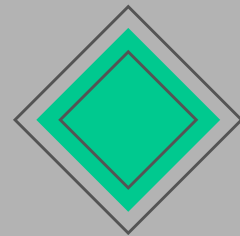
Se encontraron dificultades al intentar ordenar los datos por puntaje en algunas consultas.



# PROBLEMAS



Decisión final sobre el diseño de las tablas:



Creamos tres tablas distintas, cada una diseñada específicamente para satisfacer los requisitos de las consultas del negocio. Esto permitió un mejor manejo de los datos.



# REQUISITO 3

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

Consulta:

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

Retorno: tabla con 182 filas.

\*Nota: imágenes de tabla en informe



# REQUISITO 3

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

Consulta:

```
``cql
SELECT * FROM estudiantes_carrera WHERE matriculado = 'SI'
AND carrera = 'INGENIERIA CIVIL INFORMÁTICA' AND region =
'MAULE';``
```

Retorno: tabla con 92 filas.

\*Nota: imágenes de tabla en informe



# REQUISITO 3

C. Devolver todos los postulantes matriculados en la facultad de Ciencias de la Salud ordenado por puntaje PSU.

Consulta:

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

Retorno: Tabla con 824 filas.

\*Nota: imágenes de tabla en informe



# TABLA COMPARACIONES

Búsqueda en Exel por función	Resultado Exel	Resultado Cassandra
<code>`=COUNTIFS(F1:F16652; "SI";E1:E16652;"MEDICINA")`</code>	182	182
<code>`=COUNTIFS(F1:F16652; "SI";E1:E16652;"INGENIERÍA CIVIL INFORMÁTICA";J1:J16652;"MAULE")`</code>	92	92
<code>`=COUNTIFS(F1:F16652; "SI";G1:G16652;"CIENCIAS DE LA SALUD")`</code>	825	824

# POWER BI

CData ODBC Driver for Cassandra - DSN Configuration

**cdata**

Connection | Data Model | About

DSN Configuration

Data Source Name: CData Cassandra Source Test Connection Reset Connection

Connection Properties

**Basic** | Advanced

Server *	localhost
Port *	9042
Auth Scheme *	Basic
User *	
Password *	
Database	
Use SSL	False
Consistency Level	ONE

**Auth Scheme \***  
The scheme used for authentication. Accepted entries are Basic, DSE, Kerberos, and LDAP.

OK Cancel Help

Get Data

odbc

All

Database

Other

ODBC

Kyvos ODBC (Beta)

From ODBC

Data source name (DSN)  
CData Cassandra Source

Advanced options

Connection string (non-credential properties) (optional) ⓘ  
*Example: Driv...*

SQL statement (optional)  

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

Supported row reduction clauses (optional)  
(None) Detect

OK Cancel





THANK  
YOU





# REFERENCIAS

1. Apache Cassandra. (n.d.). Cassandra CQL documentation. Recuperado de <https://cassandra.apache.org/doc/stable/cassandra/cql/>
  2. Aache Cassandra on a Docker Container. Medium. <https://medium.com/swlh/building-a-python-data-pipeline-to-apache-cassandra-on-a-docker-container-fc757fbfafdd>
  3. Bertuccini, C. (2014, Julio 25). Difference between partition key, composite key and clustering key in Cassandra?. StackOverflow. <https://stackoverflow.com/a/24953331>
- 