



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Bases de Datos No-SQL



Integrantes:

Brandon Martinez

Erika Rosales

Proyecto Fase II - Neo4J

En esta fase, se migró la base de datos desde MongoDB a Neo4j para aprovechar el modelo de grafos en la gestión de datos sobre videojuegos. Esta transición permite explorar las relaciones entre videojuegos, géneros, empresas, plataformas y etiquetas, facilitando consultas complejas basadas en conexiones entre nodos.

A continuación, detallamos la estructura y las tareas realizadas para llevar a cabo el proyecto.

Proceso de Migración y Modelado

1. **Extracción de Datos:** Los datos fueron extraídos de MongoDB y preparados para ser insertados en Neo4j.
2. **Transformación de Datos:** Se realizaron transformaciones para ajustar los datos al modelo de grafos de Neo4j, creando nodos y relaciones según el esquema definido.
3. **Carga de Datos:** Los datos transformados se cargaron en la base de datos Neo4j, estableciendo las relaciones entre los nodos.
4. **Implementación de Consultas:** Se desarrollaron y probaron consultas Cypher para explorar y analizar los datos en el nuevo modelo de grafos.

Estructura de la base de datos

El modelo de datos en Neo4j se definió con los siguientes nodos y relaciones:

★ **Nodos:**

- **Juego:** Representa un videojuego con propiedades como name, rating (anteriormente original_game_rating), y release_date (anteriormente original_release_date).
- **Género:** Representa un género de videojuego.

- Empresa: Representa una empresa desarrolladora.
- Plataforma: Representa una plataforma en la que los juegos están disponibles.
- Tag: Representa una etiqueta asociada a un videojuego.

★ **Relaciones:**

- DESARROLLADO_POR: Conecta un juego con la empresa desarrolladora.
- PERTENECE_A: Conecta un juego con uno o más géneros.
- SU_PLATAFORMA: Conecta un juego con una o más plataformas.
- SU_ETIQUETA: Conecta un juego con una o más etiquetas.

Queries

Todas las consultas se desarrollaron siguiendo el esquema sugerido en el enunciado, utilizando el lenguaje de consultas cypher de neo4j.

Listado de consultas:

Consulta 1: Lista los juegos que estén relacionados con todos los géneros especificados.

```
:param generos => ['Action', 'Adventure']

MATCH (j:Juego)-[:SU_GENERO]->(g:Genero)
WHERE g.name IN $generos
RETURN DISTINCT j
```

Consulta 2: Busca juegos desarrollado por una empresa específica

```
:param nombreEmpresa => ""

MATCH (j:Juego)-[:DESARROLLADO_POR]->(e:Empresa)
WHERE e.name = $nombreEmpresa
RETURN j.name AS Juego, e.name AS Empresa
```

Consulta 3: Encuentra los juegos que estén disponibles en más de un número específico de plataformas.

```
:param n => 2

MATCH (j:Juego)-[:SU_PLATAFORMA]->(p:Plataforma)
WITH j, COUNT(p) AS numPlataformas
WHERE numPlataformas > $n
```

```
RETURN j.name AS Juego, numPlataformas AS TotalPlataformas
```

Consulta 4: Contar los nodos juegos que están en más de n plataformas

```
:param n => 4  
MATCH (j:Juego)-[:SU_PLATAFORMA]->(p:Plataforma)  
WITH j, COUNT(p) AS numPlataformas  
WHERE numPlataformas > $n  
RETURN COUNT(j) AS TotalJuegos
```

Consulta 5: Lista los juegos que tienen una calificación mayor a la especificada

Consulta 6: Juegos que están relacionadas a las etiquetas y géneros indicados

```
:param etiquetas => ['Multiplayer', 'Singleplayer']  
:param generos => ['Action', 'Adventure']  
  
WITH $etiquetas AS etiquetas, $generos AS generos  
MATCH (j:Juego)-[:SU_ETIQUETA]->(t:Tag)  
WHERE t.name IN etiquetas  
WITH j, COUNT(t) AS etiquetaCount  
WHERE etiquetaCount = SIZE(etiquetas)  
MATCH (j)-[:SU_GENERO]->(g:Género)  
WHERE g.name IN generos  
WITH j, etiquetaCount, COUNT(g) AS generoCount  
WHERE generoCount = SIZE(generos)  
RETURN j.name AS juego, etiquetaCount, generoCount
```

Consulta 7: Busca los juegos que pertenecen a los géneros especificados y tienen una calificación mayor al promedio calculado de los juegos pertenecientes a esos géneros

```
:param generos => ['Shooter', 'Action']  
  
MATCH (j:Videojuego)-[:SU_GENERO]->(g:Genero)
```

```
WHERE g.name IN $generos
WITH AVG(j.rate) AS promedio
MATCH (j:Videojuego)-[:SU_GENERO]->(g:Genero)
WHERE g.name IN $generos AND j.rate > promedio
RETURN DISTINCT(j.name) AS Videojuego, j.rate AS rating
```