



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Máster **Big Data** Analytics

2023 -2024



NOSQL

Profesor Juan Carlos Casamayor Ródenas

Proyecto Base de Datos NOSQL
para la gestión de una biblioteca

*Tarea 4: Implementación en **Neo4j***

Alumnos **TRABAJO1**:
Juan José Romero León
José Carlos Ávila Palazón
M.Teresa Granero Ruiz

I DISEÑO

Diseño Conceptual

Para el modelo no relacional que se presenta en este proyecto, el diseño conceptual para el modelo de datos en Neo4j se deben tener en cuenta que contamos con un modelo que de forma generica se basa en nodos y relaciones.

Esto nos quiere decir que se deberán generar nodos respecto a las tablas de origen y sus relaciones respectivas entre nodos. Para ello, es importante tener en cuenta las claves ajenas dentro de las tablas para poder realizar las relaciones entre los nodos.

Los cambios dentro del diseño conceptual respecto con Cassandra o Mongo DB son los siguientes:

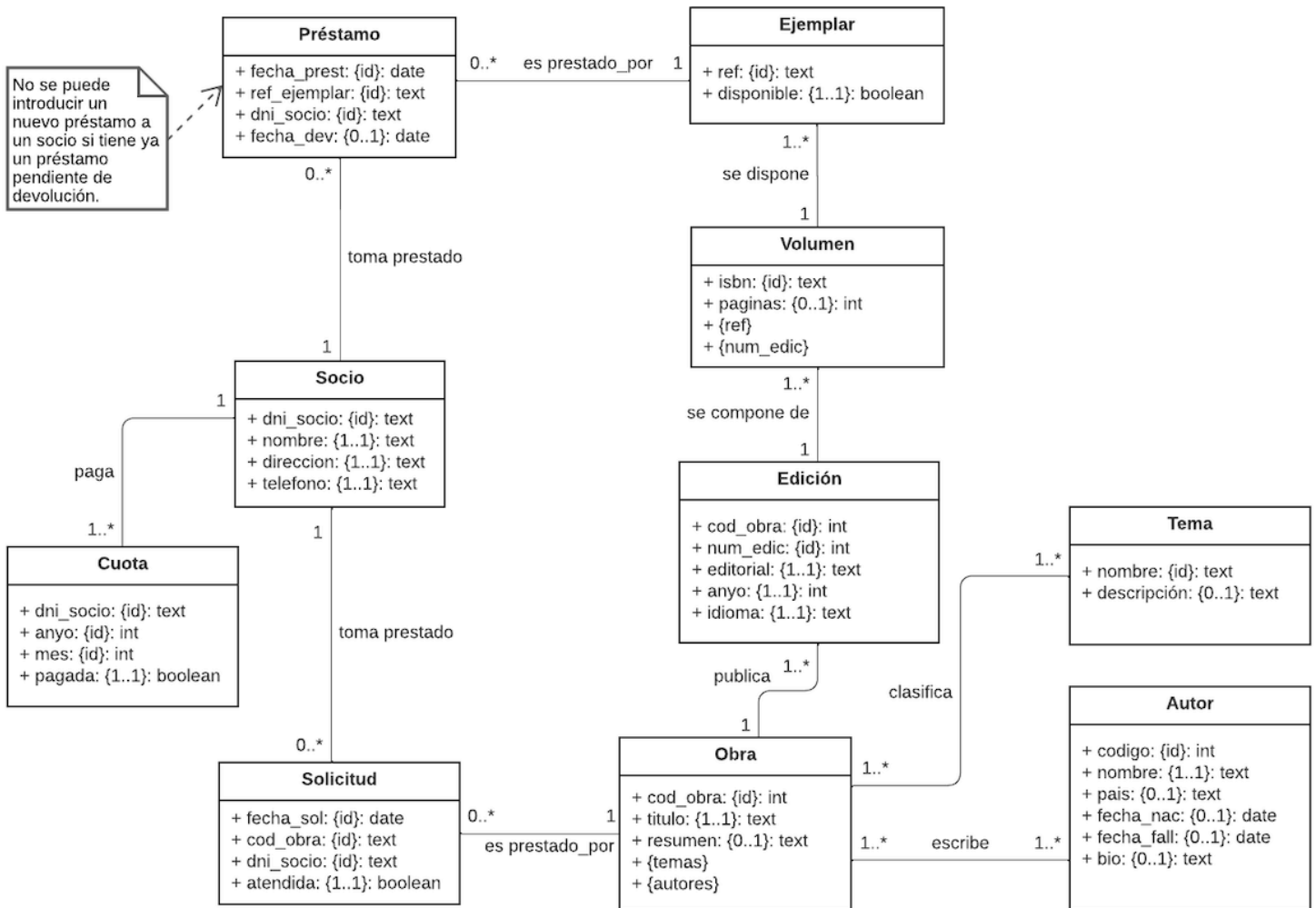
-Para la tabla “Volumen” es importante tener en cuenta para poder incorporar las relaciones de “se_dispone” y “se_compone” debemos añadir las claves ajenas propias de “Ejemplar” y “Edición” quedando la tabla con los siguientes componentes:

- isbn:{id} int
- paginas:{1..1} int
- {ref_ejemplar} int
- {num_editorial} int

-Cambios de nombre de relaciones por posibles confusiones respecto al diseño conceptual de MongoDB y Cassandra. “Es_prestado” y “Toma_prestado” para diferenciar entre la Solicitud y el Préstamo del Socio se incluirá un "por" y el Préstamo del Ejemplar con la Solicitud de la Obra se incluire de la misma forma un “por”.

Proyecto NOSQL - Diseño conceptual Neo4j

Juan José Romero León - José Carlos Ávila Palazón - MªTeresa Granero Ruiz | February 25, 2024

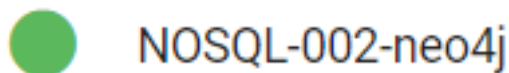


II IMPLEMENTACIÓN

Configuración de 1MV con Neo4j

Para implementar nuestro modelo en Neo4j se debe configurar el programa dentro de nuestra máquina virtual para acceder desde nuestro navegador.

En primer lugar, entramos como root a la máquina virtual:



Dentro de un terminal ejecutaremos en el directorio respectivo /root/Software/neo4j los siguientes comandos respectivamente:

```
# rpm -ivf cypher-shell-5.15.0-1.noarch.rpm
# rpm -ivf neo4j-community-5.15.0-1.noarch.rpm
```

Después, se procede a desinstalar cortafuegos con los siguientes comandos:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

En el fichero que se encuentra dentro de /etc/neo4j/neo4j.conf

Se descomentará el siguiente comando para poder tener acceso de forma local a Neo4j.

```
server.default_listen_address=0.0.0.0
```

Y en `initial.dbms.default_database="nombre_database"` se colocará el nombre del database donde se encuentran los ficheros de las tablas a cargar.

```
# The name of the default database
#initial.dbms.default_database=neo4j
#initial.dbms.default_database=northwind
#initial.dbms.default_database=ciclismo
initial.dbms.default_database=library_db|
```

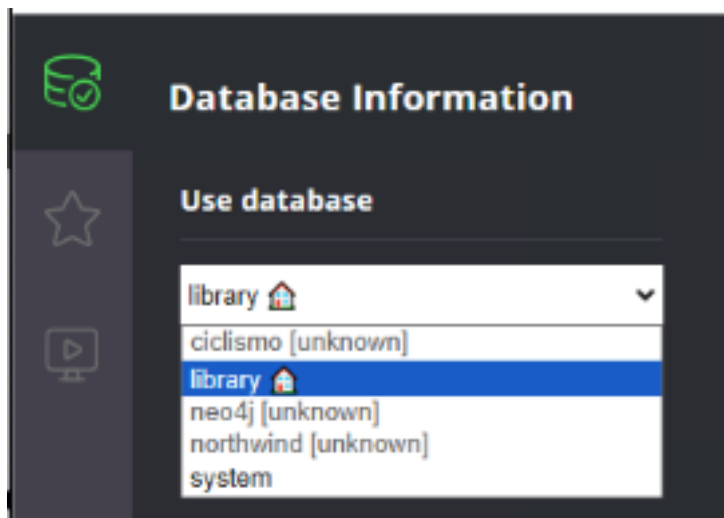
Además, en el directorio /var/lib/neo4j/import/ añadimos los ficheros csv en un subdirectorio desde donde se realiza la importación.

Y con todo ello podremos acceder por medio de nuestro navegador de forma local con la siguiente url: <http://nosql-nnn-neo4j.dsicv.upv.es:7474>

Donde “nnn” será nuestro número de cada alumno de las máquinas virtuales.

Implementación del esquema de la DB con Neo4j

Para implementar el modelo DB dentro del entorno de Neo4j debemos situar nuestra conexión a la DB library



Ya seleccionado el DB correctamente ejecutaremos el script de creación de los nodos y relaciones de nuestro modelo.

Para la solución del diseño cabe recalcar una serie de aspectos claves para la generación del modelo en Neo4j es importante a la hora de generar relaciones entre nodos, incluir en la tabla .csv las claves ajenas propias de la relación.

El script de ejecución es el siguiente con una estructura de en primer lugar creación de nodos y posteriormente las respectivas relaciones y restricciones.

label
"Tema"
"Autor"
"Obra"
"Solicitud"
"Volumen"
"Cuota"
"Prestamo"
"Socio"
"Ejemplar"
"Edicion"

//Creación de cada uno de los nodos sin incluir las claves ajenas de las tablas

```
LOAD CSV WITH HEADERS FROM "file:///library/Tema.csv" as line
FieldTerminator ','
CREATE (n:Tema)
SET n=line;
```

```
LOAD CSV WITH HEADERS FROM "file:///library/Autor.csv" as line
FieldTerminator ','
CREATE (n:Autor)
SET n.cod_autor=ToInteger(line.cod_autor),
n.nombre_tema=line.nombre_tema,
n.pais=line.pais,
n.fecha_nac=line.fecha_nac,
n.fecha_fall=line.fecha_fall,
n.bio=line.bio;
```

```
LOAD CSV WITH HEADERS FROM "file:///library/Obra.csv" as line
FieldTerminator ','
CREATE (n:Obra)
SET n.cod_obra=ToInteger(line.cod_obra),
n.titulo=line.titulo,
n.nombre_tema=line.nombre_tema,
n.cod_autor=line.cod_autor
n.resumen=line.resumen;
```

```
LOAD CSV WITH HEADERS FROM "file:///library/Solicitud.csv" as line
FieldTerminator ','
CREATE (n:Solicitud)
SET n.fecha_sol=line.fecha_sol,
n.cod_obra=ToInteger(line.cod_obra),
n.dni_socio=line.dni_socio,
n.atendida=ToBoolean(line.atendida);
```

```

LOAD CSV WITH HEADERS FROM "file:///library/Volumen.csv" as line
FieldTerminator ','
CREATE (n:Volumen)
SET n.isbn=ToInteger(line.isbn),
n.ref=line.ref,
n.num_edic=line.num_edic,
n.paginas=ToInteger(line.paginas);

LOAD CSV WITH HEADERS FROM "file:///library/Cuota.csv" as line
FieldTerminator ','
CREATE (n:Cuota)
SET n.anyo=ToInteger(line.anyo),
n.mes=ToInteger(line.mes),
n.dni_socio=line.dni_socio,
n.pagada=ToBoolean(line.pagada);

LOAD CSV WITH HEADERS FROM "file:///library/Prestamo.csv" as line
FieldTerminator ','
CREATE (n:Prestamo)
SET n=line;

LOAD CSV WITH HEADERS FROM "file:///library/Socio.csv" as line
FieldTerminator ','
CREATE (n:Socio)
SET n=line;

LOAD CSV WITH HEADERS FROM "file:///library/Ejemplar.csv" as line
FieldTerminator ','
CREATE (n:Ejemplar)
SET n=line;

LOAD CSV WITH HEADERS FROM "file:///library/Edicion.csv" as line
FieldTerminator ','
CREATE (n:Edicion)
SET n.ref=line.ref,
n.cod_obra=line.cod_obra,
n.disponible=ToBoolean(line.disponible);

//Creación de cada relación entre los nodos
//Utilizamos las tablas en las que se encuentran
//las claves ajenas para poner entablar las relaciones

//Tema-clasifica-Obra
MATCH (o_ob:Obra),(t_tema:Tema)
WHERE o_ob.nombre_tema=t_tema.nombre_tema
MERGE (o_ob)-[r:CLASIFICA]->(t_tema);

//Autor-escribe-Obra
MATCH (o:Obra),(a:Autor)
WHERE o.cod_autor=a.cod_autor
MERGE (o)-[r:ESCRIBE]->(a);

//Obra-es_prestado-Solicitud
MATCH (s:Solicitud),(o:Obra)
WHERE s.cod_obra=o.cod_obra
CREATE (s)-[r:ES_PRESTADO]-(o);

//Obra-publica-Edicion

```

```

MATCH (e:Edicion)(o:Obra)
WHERE e.cod_obra=o.cod_obra
CREATE (e)-[r:PUBLICA]->(o);

//Edicion-se_compone_de-Volumen
MATCH (v:Volumen),(e:Edicion)
WHERE v.num_edic=e.num_edic
CREATE (v)-[r:SE_COMPONE_DE]->(e);

//Ejemplar-se_dispone-Volumen
MATCH (v:Volumen),(e:Ejemplar)
WHERE v.ref=e.ref
CREATE (v)-[r:SE_DISPONE]->(e);

//Ejemplar-es_prestado_por-Prestamo
MATCH (p:Prestamo),(e:Ejemplar)
WHERE p.ref=e.ref
CREATE (p)-[r:ES_PRESTADO_POR]->(e);

//Socio-toma_prestado-Prestamo
MATCH (p:Prestamo),(s:Socio)
WHERE p.dni_socio=s.dni_socio
CREATE (p)-[r:TOMA_PRESTADO]-(s);

//Socio-toma_prestado_por-Prestamo
MATCH (p:Solicitud),(s:Socio)
WHERE p.dni_socio=s.dni_socio
CREATE (p)-[r:TOMA_PRESTADO_POR]->(s);

//Cuota-paga-Socio
MATCH (c:Cuota),(s:Socio)
WHERE c.dni_socio=s.dni_socio
CREATE (c)-[r:PAGA]-(s);

```


Node labels

*(51)

Autor

Cuota

Edicion

Ejemplar

Obra

Prestamo

Socio

Solicitud

Tema

Volumen

Relationship types

*(56)

CLASIFICA

ESCRIBE

ES_PRESTADO

ES_PRESTADO_POR

PAGA

PUBLICA

SE_COMPONE_DE

SE_DISPONE

TOMA_PRESTADO

TOMA_PRESTADO_POR

