

Máster en Bioinformática y Biología Computacional  
PROCESADO Y MANEJO DE DATOS MASIVOS

# Práctica de NEO4J

**Autores:**

*Lucas Mateos Pinilla*

*Tania Gonzalo Santana*

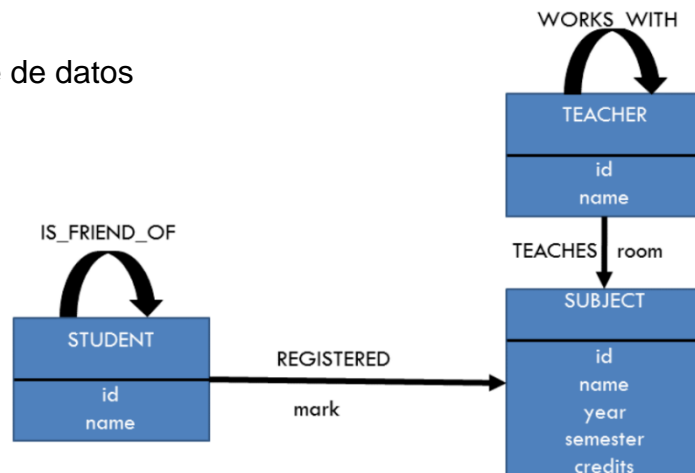
*Deyanira Borroto Albuquerque*

Noviembre 2024

Universidad Autónoma de Madrid

# Creación de la base de datos

## Diseño de la base de datos



### 1. Cargar los archivos .csv:

```
LOAD CSV WITH HEADERS FROM "file:/est.csv" AS csvLine
CREATE (e:Estudiante { id: toInteger(csvLine.id), nombre: csvLine.nombre});
```

```
LOAD CSV WITH HEADERS FROM "file:/profesores.csv" AS csvLine
CREATE (p:Profesor {id: toInteger(csvLine.id), nombre: csvLine.nombre});
```

```
LOAD CSV WITH HEADERS FROM "file:/esamigo.csv" AS csvLine
MATCH(e1:Estudiante), (e2:Estudiante)
WHERE e1.id= toInteger(csvLine.id1) AND e2.id= toInteger(csvLine.id2)
CREATE (e1)-[e:esAmigo]->(e2)
RETURN e;
```

```
LOAD CSV WITH HEADERS FROM "file:/asignaturas.csv" AS csvLine
CREATE (e:Asignatura { id: toInteger(csvLine.id), nombre: csvLine.nombre,
curso: toInteger(csvLine.curso), semestre: csvLine.semestre, credits:
toInteger(csvLine.credits)});
```

```
LOAD CSV WITH HEADERS FROM "file:/imparte.csv" AS csvLine
MATCH(p:Profesor), (a:Asignatura)
WHERE p.id= toInteger(csvLine.id1) AND a.id= toInteger(csvLine.id2)
CREATE (p)-[i:Imparte{aula:csvLine.aula}]->(a)
RETURN i;
```

```
LOAD CSV WITH HEADERS FROM "file:/matriculado.csv" AS csvLine
MATCH(e:Estudiante), (a:Asignatura)
WHERE e.id= toInteger(csvLine.id1) AND a.id= toInteger(csvLine.id2)
CREATE (e)-[m:Matriculado {nota:toInteger(csvLine.nota)}]->(a)
RETURN m;
```

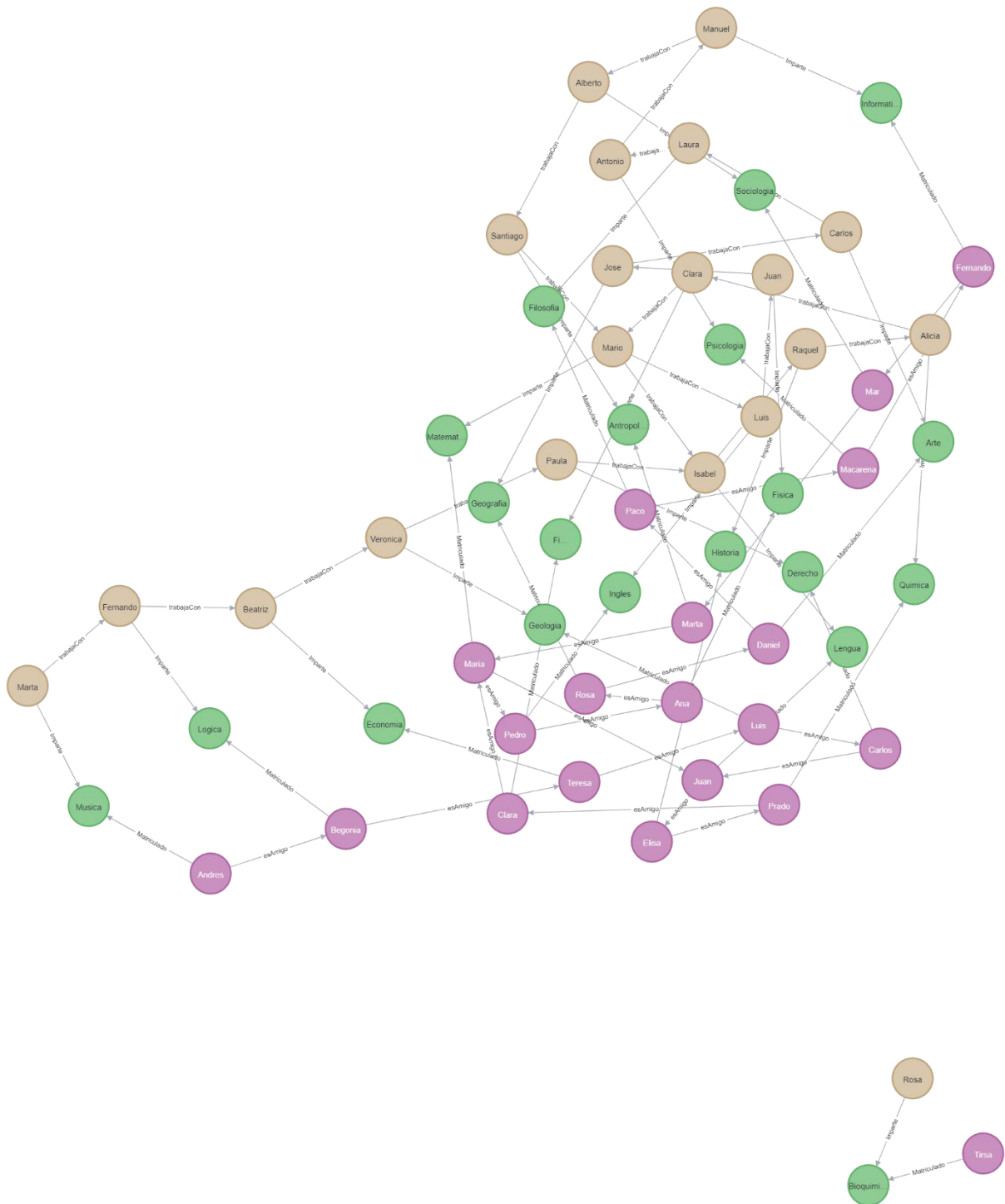
```
LOAD CSV WITH HEADERS FROM "file:/trabajacon.csv" AS csvLine
MATCH(p1:Profesor), (p2:Profesor)
WHERE p1.id= toInteger(csvLine.id1) AND p2.id= toInteger(csvLine.id2)
CREATE (p1)-[t:trabajaCon]->(p2)
RETURN t;
```

Borrar todo:

MATCH (n)

DETACH DELETE n

### Grafo de la base de datos de estudiantes



# Consultas

## 2. Ejecuta las siguientes consultas en Neo4j

### A. Nombre de las asignaturas que enseña la profesora Veronica:

```
MATCH (p:Profesor {nombre:"Veronica"}) - [:Imparte] -> (a:Asignatura)  
RETURN a.nombre AS Asignatura
```

Resultados:

Asignatura
"Geologia"

### B. Nombre de asignaturas del primer año:

```
MATCH (s:Asignatura {curso: 1})  
RETURN s.nombre AS asignaturas_primer_año
```

Resultados:

asignaturas_primer_año
"Matematicas"
"Ingles"
"Fisica"
"Geografia"
"Arte"
"Filosofia"
"Psicologia"
"Informatica"
"Sociologia"
"Antropologia"

C. Nombre de asignaturas por orden alfabético:

**MATCH** (a:Asignatura)  
**RETURN** a.nombre **AS** asignaturas\_alfabeticamente  
**ORDER BY** a.nombre

Resultados:

asignaturas_alfabeticamente
"Antropologia"
"Arte"
"Bioquimica"
"Derecho"
"Economia"
"Educacion Fisica"
"Filosofia"
"Fisica"
"Geografia"
"Geologia"
"Historia"
"Informatica"
"Ingles"
"Lengua"
"Logica"
"Matematicas"
"Musica"
"Psicologia"
"Quimica"
"Sociologia"

D. Nombre de asignaturas del tercer y el cuarto año:

```
MATCH (a:Asignatura)
WHERE a.curso = 3 OR a.curso = 4
RETURN a.nombre AS asignaturas_curso3y4
```

Resultados:

asignaturas_curso3y4
"Ingles"
"Historia"
"Geografia"
"Arte"
"Logica"
"Geologia"
"Derecho"
"Ingles"
"Historia"
"Geografia"
"Arte"
"Logica"
"Geologia"
"Derecho"

E. Elimina la propiedad semestre del nodo asignaturas:

```
MATCH (s:Asignatura)
REMOVE s.semestre
```

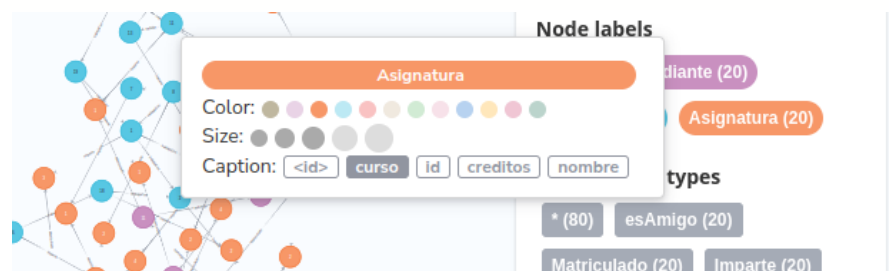


Imagen 1. Mostrando que la propiedad 'semestre' del nodo Asignatura ha sido eliminada.

#### F. Cambia el nombre de una asignatura:

```
MATCH (s:Asignatura)
WHERE s.nombre = 'Bioquimica' SET s.nombre = 'Genetica'
RETURN s.nombre AS nueva_asignatura
```

Resultados:

nueva_asignatura
"Genetica"

#### G. Número de asignaturas en las que el estudiante Daniel está matriculado:

```
MATCH (e:Estudiante {nombre:"Daniel"}) - [:Matriculado] -> (a:Asignatura)
RETURN COUNT(a) AS numero_asignaturas_Daniel
```

Resultados:

numero_asignaturas_Daniel
1

#### H. Nombres de los amigos de Pedro:

```
MATCH (e:Estudiante {nombre:"Pedro"}) - [:esAmigo] -> (amigo:Estudiante)
RETURN amigo.nombre AS amigos_Pedro
```

Resultados:

amigos_Pedro
"Ana"

#### I. Nombres de los amigos de los amigos de Pedro:

```
MATCH (e:Estudiante {nombre:"Pedro"}) - [:esAmigo] -> (amigo:Estudiante) - [:esAmigo] ->
(amigodeamigo:Estudiante)
RETURN amigodeamigo.nombre AS amigos_de_amigos_de_Pedro
```

Resultados:

amigos_de_amigos_de_Pedro
"Rosa"

J. Nombre de las asignaturas en las cuales uno de los amigos del estudiante Maria está matriculado:

```
MATCH (e:Estudiante {nombre:"Maria"}) - [:esAmigo] -> (amigo:Estudiante) - [:Matriculado] -  
(a:Asignatura)  
WITH amigo, a  
LIMIT 1  
RETURN amigo.nombre, a.nombre AS nombre_asignaturas
```

Resultados:

amigo.nombre	nombre_asignaturas
"Juan"	"Lengua"

K. Nombre de los estudiantes que están matriculados en cualquiera de las asignaturas que enseña el Profesor Fernando:

```
MATCH (p:Profesor {nombre:"Fernando"}) - [:Imparte] -> (a:Asignatura) <- [:Matriculado] -  
(e:Estudiante)  
RETURN e.nombre AS estudiante, a.nombre AS asignatura
```

Resultados:

estudiante	asignatura
"Begonia"	"Logica"

L. Nombres de los estudiantes que están matriculados en cualquiera de las asignaturas que enseña cualquiera de los profesores que trabaja con el Profesor Fernando:

```
MATCH (p:Profesor {nombre:"Fernando"}) - [:trabajaCon] -> (p2:Profesor) - [:Imparte] ->  
(a:Asignatura) <- [:Matriculado] - (e:Estudiante)  
RETURN e.nombre AS estudiante
```

Resultados:

estudiante
"Teresa"

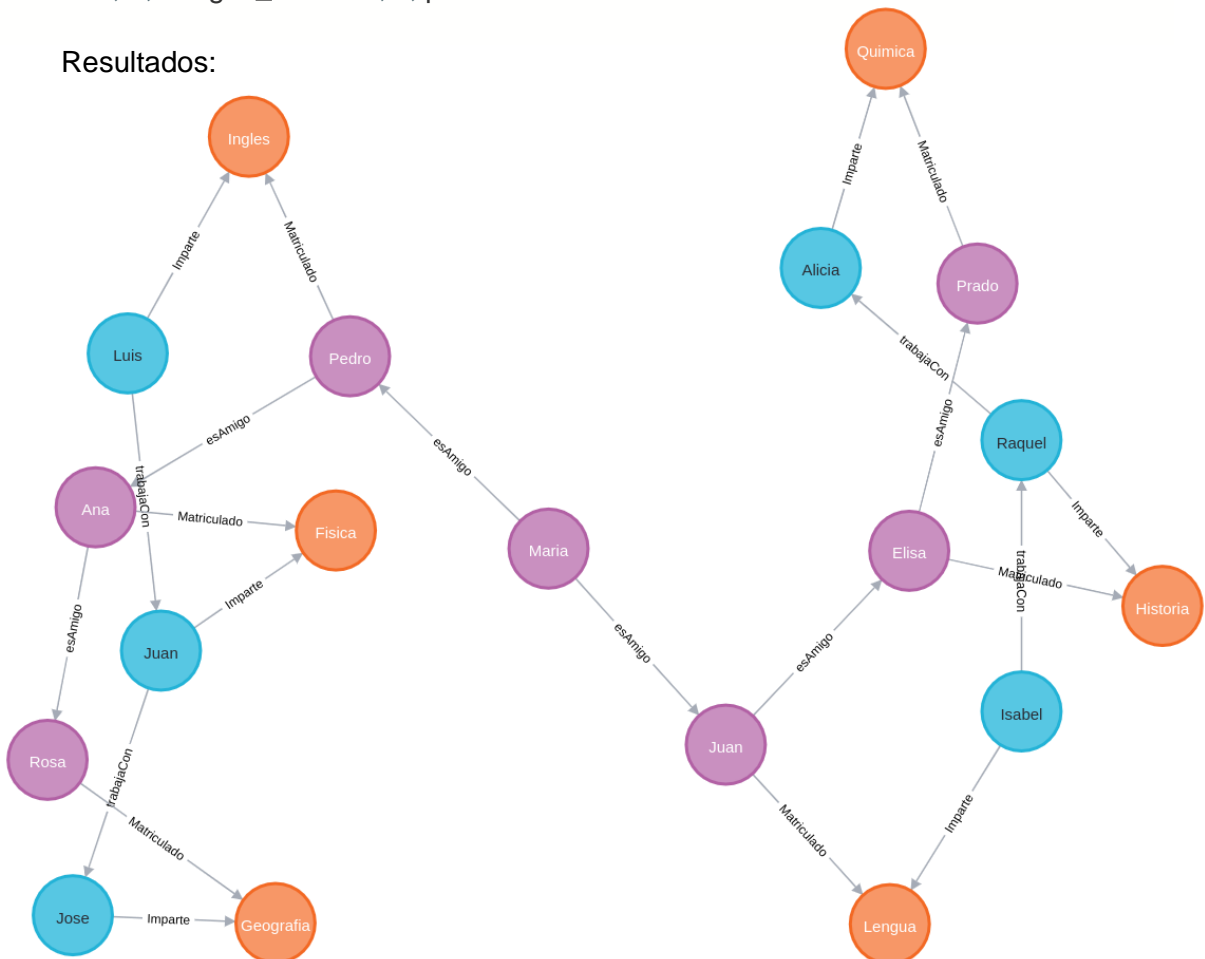


### 3. Proponed dos consultas de patrones para relaciones y resolverlos

a) Profesores que tienen y asignaturas a las que asisten los amigos y amigos remotos de María:

```
MATCH (e:Estudiante {nombre:"Maria"}) - [a:esAmigo*1..3] -> (amigos_remotos:Estudiante)
- [m:Matriculado] - (s:Asignatura) - [i:Imparte] - (p:Profesor)
RETURN e, a, amigos_remotos, s, p
```

Resultados:



b) Listar estudiantes con más de 1 amigo:

```
MATCH (e:Estudiante) - [r:esAmigo] -> (x:Estudiante)
WITH e, COUNT(r) AS rel_cnt
WHERE rel_cnt > 1
RETURN e.nombre AS estudiante_con_mas_amigos, rel_cnt AS num_amigos
```

Resultados:

estudiante_con_mas_amigos	num_amigos
"Maria"	2

c) *Nota promedio (redondeada) de estudiantes por curso:*

```
MATCH (s:Estudiante)-[r:Matriculado]->(a:Asignatura)
RETURN a.curso AS Curso, round(AVG(r.nota)) AS NotaPromedio
ORDER BY Curso
```

Curso	NotaPromedio
1	7.0
2	8.0
3	9.0
4	7.0