Trabajo Práctico
Modelado en Neo4j

Asignatura: Implementacion de Bases de Datos No SQL
Curso: K5571
Año: 2017

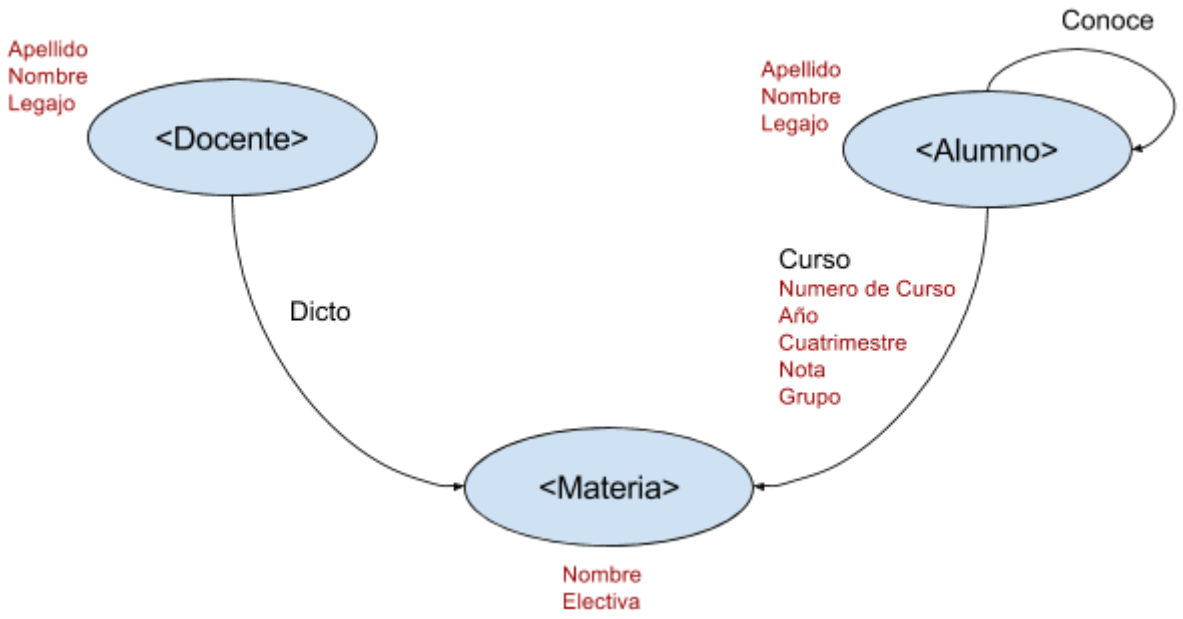
Integrantes:

- 149.169-6 - Crespi Gustavo
- 149.173-8 - Fernandes dos Santos Juan
- 149.334-6 - Rios Juan
- 149.324-3 - Santoalla Gastón
- 149.161-1 - Vazquez Ramiro

Índice

Modelado en Neo4j

Modelo



Se decidió modelar la problemática con 3 entidades principales (Alumno, Materia y Docente) guardando en ellas solo los datos necesarios para reconocerlas y guardar el resto de la información en las relaciones.

A continuación se explican algunas de las problemáticas o cuestiones que se tuvieron en cuenta. Solo vamos a explicar las más importantes ya que otras no tienen mucha complicación, por ejemplo la relación Dictó entre la materia y el Docente surge de manera bastante directa una vez que se decidió que existan esos dos nodos.

Relación Conoce

La relación *Conoce a* es conceptualmente “bidireccional”, es decir que si (a)-[:Conoce]->(b) entonces debería darse que (a)<-[:Conoce]-(b). Como Neo4j no provee un mecanismo para crear relaciones bidireccionales (las relaciones tienen un solo sentido) esto lo podemos resolver de dos formas, creando ambas relaciones o creando una sola relación y luego matcheandola sin sentido.

Para la relación Conoce entonces, decidimos la segunda opción. A la hora de crearla se lo hace con un sentido arbitrario por ejemplo (a)-[:Conoce]->(b) pero a la hora de matchear contra esa relación siempre se hace sin sentido particular, es decir MATCH (a)-[:Conoce]-(b). De esta forma se garantiza que la relación se considere para ambos sentidos siempre.

Alumno/Docente

Consideramos que tanto el alumno, como el docente tienen las mismas propiedades (legajo, nombre, apellido), de ahí que surgen varias formas de representar esta “dualidad”. En principio podría ser un nodo *Persona* que posea los atributos y que la separación de quien es docente y quien alumno surja de cómo se relaciona con las materias (esta es la opción que dejamos como variante del punto 6 pero no para el modelo).

Aunque esta opción es realizable y cumple con los requerimientos, nos pareció mejor separar la idea de quien es un Docente y quien un Alumno generando dos etiquetas distintas. De esta manera se logra no solo una separación conceptual más grande sino que a la hora de matchear no importa realmente las relaciones con las materias, sino que se matchea directamente por las etiquetas, dejando la query mucho más clara y limpia.

Cursos y Materias

A la hora de modelar al Alumno cursando una Materia surgieron diferentes dificultades y variantes con sus ventajas y desventajas. El Curso puede ser un nodo o una relación y los datos pueden estar guardados en la Materia, o en el Curso dependiendo lo que se elija. La decisión final se redujo a dos variantes principales.

La primera era dejar el Curso como un nodo intermedio entre el alumno y la Materia, relacionándose el Alumno con el Curso (con una relación aprobó o perteneció) en donde se guarda la nota el grupo, etc. Y el Curso con la Materia en sí. El problema con esta variante es que resultaba fácil matchear aquellos alumnos que habían “cursado juntos” pero complicaba el momento de matchear cuando simplemente cursaron la misma materia (pero no necesariamente el mismo curso).

La segunda opción que se planteó es dejar al curso como una relación entre el alumno y la materia, dejando que los nodos guarden la información exclusiva de las entidades y la relación Cursó como el lugar donde guardar los datos de esa “cursada” (el año, cuatrimestre, número de curso, etc). La ventaja de esta variante es que facilitaba ambos métodos de acceso tanto por curso (aunque requiere la comparación de muchos campos) como de materia. El problema es que generaba mucha más redundancia de datos (los datos del curso se repiten por cada alumno que forme parte y mezcla los datos del curso con los del alumno en ese curso).

Aunque ninguna de las dos variantes restringe del todo las queries que se pueden hacer, la segunda opción nos pareció más clara y práctica para realizar las queries. A pesar del costo extra de memoria creemos que vale la pena pagarlo en pos de tener queries más simples.

Grupos

Aunque en un principio parecía una decisión sencilla, resultó ser un poco más compleja. Como primera aproximación habíamos planteado una relación similar a Conoce entre los alumnos, el problema con esta es que se dificulta reconocer a todos los que forman parte del mismo grupo en una materia (y por ende también a alguno que no forme parte).

Debido a esto planteamos dos alternativas, modelar el grupo como un nodo o simplemente poner el nombre del grupo en la relacion Curso. De estas dos preferimos la segunda, ya que aunque sobrecargar un poco la relación Curso, no agrega complejidad al modelo y dejaba las queries muy sencillas (si ya vas a buscar que sean parte del mismo curso, decidir que sea del mismo grupo o de otro era una condición más en el where).

Queries de Creación

Creación de Nodos

```
CREATE
(Gu:Alumno:Docente { apellido: "Crespi", nombre: "Gustavo", legajo: 1491696 }),
(H:Alumno:Docente { apellido: "Fernandes dos Santos", nombre: "Juan", legajo: 1491738  }),
(J:Alumno:Docente { apellido: "Rios", nombre: "Juan", legajo: 1493346 }),
(G:Alumno:Docente { apellido: "Santoalla", nombre: "Gaston", legajo: 1493243 }),
(HURLEY:Alumno { apellido: "Reyes", nombre: "Hugo", legajo: 1324567 }),
(R:Alumno { apellido: "Vazquez", nombre: "Ramiro", legajo: 1491611 }),
(DZN:Alumno { apellido: "Daniel", nombre: "Zinghini", legajo: 1492800 }),
(SRLMP:Alumno { apellido: "Sarlomp", nombre: "Juan", legajo:1330298 }),
(AED:Materia { nombre: "Algoritmos y Estructura de Datos", electiva: false }),
(DDS:Materia { nombre: "Diseño de Sistemas", electiva: false }),
(GDD:Materia { nombre: "Gestion de Datos", electiva: false }),
(CRIP:Materia { nombre: "Criptografia", electiva: true }),
(PDEP:Materia { nombre: "Paradigmas de Programacion", electiva: false }),
(AG:Materia { nombre: "Administracion Gerencial", electiva: false }),
(SO:Materia { nombre: "Sistemas Operativos", electiva: false }),
(ADP:Materia { nombre: "Administracion de Personal", electiva: true }),
(TADP:Materia { nombre: "Tecnicas Avanzadas de Programacion", electiva: true });
```

Creación de Relaciones

```
MATCH (Gus:Alumno { legajo: 1491696 }),  (Juan:Alumno { legajo: 1493346 }),
(Gas:Alumno { legajo: 1493243 }),  (R:Alumno { legajo: 1491611 }),
(Jfds:Alumno { legajo: 1491738 }), (hurley:Alumno { legajo: 1324567 }),
(dzn:Alumno { legajo: 1492800 }),  (srlmp:Alumno { legajo:1330298 }),
(pdep:Materia { nombre: "Paradigmas de Programacion"}),
(tadp:Materia { nombre: "Tecnicas Avanzadas de Programacion"}),
(crip:Materia { nombre: "Criptografia" }),
(so:Materia { nombre: "Sistemas Operativos"}),
(gdd:Materia { nombre: "Gestion de Datos" }),
(aed:Materia { nombre: "Algoritmos y Estructura de Datos"}),
(dds:Materia { nombre: "Diseño de Sistemas"}),
(adp:Materia { nombre: "Administracion de Personal"}),
(ag:Materia { nombre: "Administracion Gerencial"})
CREATE (Gus)-[:Conoce]->(Juan), (Gus)-[:Conoce]->(Gas), (Gus)-[:Conoce]->(R),
(Gus)-[:Conoce]->(Jfds),
(Juan)-[:Conoce]->(Gas), (Juan)-[:Conoce]->(R), (Juan)-[:Conoce]->(Jfds),
(Gas)-[:Conoce]->(R), (Gas)-[:Conoce]->(Jfds),
(R)-[:Conoce]->(Jfds),
(Gas)-[:Conoce]->(dzn), (dzn)-[:Conoce]->(srlmp),
(Juan)-[:Dicto]->(pdep), (Juan)-[:Dicto]->(dds),
(Gus)-[:Dicto]->(pdep), (Gus)-[:Dicto]->(dds),
(Jfds)-[:Dicto]->(pdep), (Jfds)-[:Dicto]->(tadp),
(Gas)-[:Dicto]->(pdep),
(Gas)-[:Curso { nroCurso: "K3001", ano: 2015, cuatrimestre: "A", nota: 10, grupo: "Grupo 4" }]->(dds),
(Gas)-[:Curso { nroCurso: "K3021", ano: 2015, cuatrimestre: "2C", nota: 9, grupo: "Grupo 2" }]->(tadp),
(Gas)-[:Curso { nroCurso: "K2004", ano: 2014, cuatrimestre: "A", nota: 4, grupo: "Grupo 4" }]->(pdep),
(Gas)-[:Curso { nroCurso: "K5054", ano: 2017, cuatrimestre: "1C", nota: 10  }]->(ag),
(Gas)-[:Curso { nroCurso: "K3053", ano: 2015, cuatrimestre: "2C", nota: 4, grupo: "SACHIDA"}]->(so),
(Gas)-[:Curso { nroCurso: "K1025", ano: 2013, cuatrimestre: "A", nota: 10 }]->(aed),
(Gas)-[:Curso { nroCurso: "K3022", ano: 2015, cuatrimestre: "2C", nota: 8, grupo: "Ñuflo" }]->(gdd),
(Gas)-[:Curso { nroCurso: "K4071", ano: 2016, cuatrimestre: "2C", nota: 10, grupo: "Clefia" }]->(crip),
(Juan)-[:Curso { nroCurso: "K3001", ano: 2015, cuatrimestre: "A", nota: 4, grupo: "Grupo 4" }]->(dds),
(Juan)-[:Curso { nroCurso: "K3021", ano: 2015, cuatrimestre: "2C", nota: 8, grupo: "Grupo
8"}]->(tadp),
(Juan)-[:Curso { nroCurso: "K2004", ano: 2014, cuatrimestre: "A", nota: 6, grupo: "Grupo 3" }]->(pdep),
(Juan)-[:Curso { nroCurso: "K3013", ano: 2015, cuatrimestre: "2C", nota: 4, grupo: "socketes
planificaDOS" }]->(so),
```

(Juan)-[:Curso { nroCurso: "K1025", ano: 2013, cuatrimestre: "A", nota: 5 }]->(aed),
(Juan)-[:Curso { nroCurso: "K3022", ano: 2015, cuatrimestre: "2C", nota: 7, grupo: "Ñuflo" }]->(gdd),
(Juan)-[:Curso { nroCurso: "K4071", ano: 2016, cuatrimestre: "2C", nota: 10, grupo: "Clefia" }]->(crip),
(Jfds)-[:Curso { nroCurso: "K3001", ano: 2015, cuatrimestre: "A", nota: 4, grupo: "Grupo 5" }]->(dds),
(Jfds)-[:Curso { nroCurso: "K3021", ano: 2015, cuatrimestre: "2C", nota: 8, grupo: "Grupo 2" }]->(tadp),
(Jfds)-[:Curso { nroCurso: "K2004", ano: 2014, cuatrimestre: "A", nota: 8, grupo: "Grupo 4" }]->(pdep),
(Jfds)-[:Curso { nroCurso: "K3013", ano: 2015, cuatrimestre: "2C", nota: 6, grupo: "socketes planificaDOS" }]->(so),
(Jfds)-[:Curso { nroCurso: "K1025", ano: 2013, cuatrimestre: "A", nota: 9 }]->(aed),
(Jfds)-[:Curso { nroCurso: "K3022", ano: 2015, cuatrimestre: "2C", nota: 10, grupo: "Ñuflo" }]->(gdd),
(Jfds)-[:Curso { nroCurso: "K4071", ano: 2016, cuatrimestre: "2C", nota: 10, grupo: "Clefia" }]->(crip),
(hurley)-[:Curso { nroCurso: "K4071", ano: 2016, cuatrimestre: "2C", nota: 9, grupo: "Mickey"}]->(crip),
(Gus)-[:Curso { nroCurso: "K3001", ano: 2015, cuatrimestre: "A", nota: 8, grupo: "Grupo 4" }]->(dds),
(Gus)-[:Curso {nroCurso: "K3021", ano: 2015, cuatrimestre: "2C", nota: 10, grupo: "Grupo 2"}]->(tadp),
(Gus)-[:Curso { nroCurso: "K2005", ano: 2014, cuatrimestre: "A", nota: 10, grupo: "Grupo 3"}]->(pdep),
(Gus)-[:Curso { nroCurso: "XXX", ano: 2017, cuatrimestre: "1C", nota: 6 }]->(ag),
(Gus)-[:Curso { nroCurso: "K3053", ano: 2015, cuatrimestre: "2C", nota: 4, grupo: "F" }]->(so),
(Gus)-[:Curso { nroCurso: "K1025", ano: 2013, cuatrimestre: "A", nota: 8 }]->(aed),
(Gus)-[:Curso { nroCurso: "K3022", ano: 2015, cuatrimestre: "2C", nota: 7, grupo: "B" }]->(gdd),
(Gus)-[:Curso { nroCurso: "K0000", ano: 2016, cuatrimestre: "2C", nota: 10 }]->(adp),
(R)-[:Curso { nroCurso: "K3001", ano: 2015, cuatrimestre: "A", nota: 8, grupo: "Grupo 4" }]->(dds),
(R)-[:Curso { nroCurso: "K3021", ano: 2015, cuatrimestre: "2C", nota: 6, grupo: "Grupo 8" }]->(tadp),
(R)-[:Curso { nroCurso: "K2001", ano: 2014, cuatrimestre: "A", nota: 8, grupo: "Grupo 3" }]->(pdep),
(R)-[:Curso { nroCurso: "XXX", ano: 2017, cuatrimestre: "1C", nota: 10 }]->(ag),
(R)-[:Curso { nroCurso: "K3013", ano: 2015, cuatrimestre: "2C", nota: 7, grupo: "socketes planificaDOS" }]->(so),
(R)-[:Curso { nroCurso: "K1025", ano: 2013, cuatrimestre: "A", nota: 10 }]->(aed),
(R)-[:Curso { nroCurso: "K3022", ano: 2015, cuatrimestre: "2C", nota: 7, grupo: "Ñuflo" }]->(gdd)

Queries

A continuación se muestran las queries junto con las screen de respuesta de cada una, en algunos casos se expresó el RETURN de forma que sea más claro para leer y que los datos no fueran confusos (en cada una está explicado).

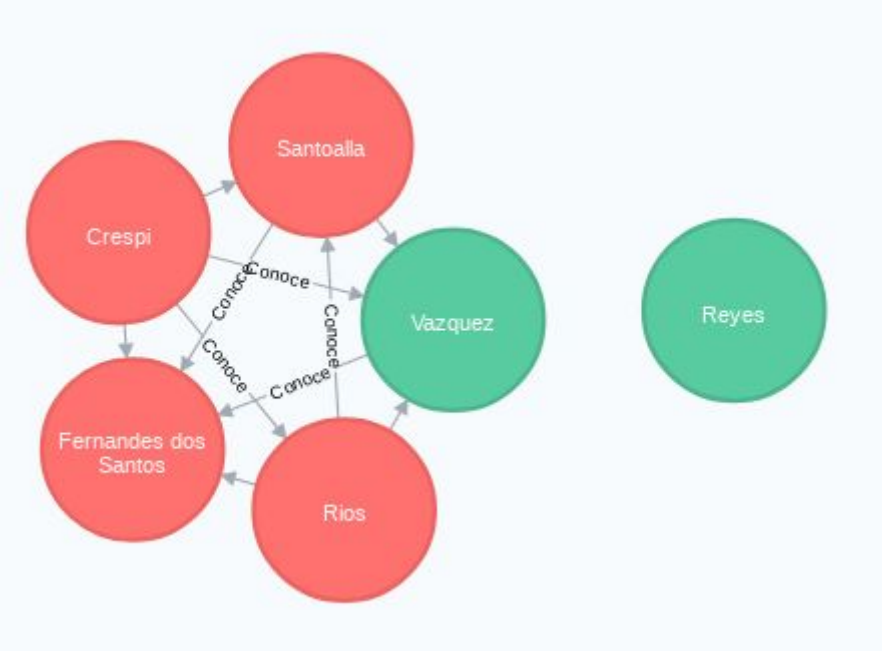
1

Listado de alumnos que cursaron materias juntos, pero en esta materia son de distintos grupos.

En esta query si se devuelve tal cual lo que se pide se obtendría una lista con todos los alumnos (por los datos cargados en la base). Para la query original se debería poner al final RETURN DISTINCT a1 as Alumno sin embargo para que se obtenga una respuesta más clara se cambió el return para que devuelva por fila Materia+Alumno con los alumnos que cursaron esa misma materia pero en otro grupo

Como de esta forma la query en formato fila devuelve muchos valores, se ejemplifica con una o dos materias en las screenshots. (La screen de grafo es exactamente igual con ambos RETURNs)

```
MATCH (a1:Alumno)-[c1:Curso]->(m: Materia)<-[c2:Curso]-(a2:Alumno)
WHERE c1.nroCurso = c2.nroCurso
AND c1.ano = c2.ano
AND c1.cuatrimestre = c2.cuatrimestre
AND c1.grupo <> c2.grupo
RETURN m.nombre as Materia, a1 as Alumno, collect(a2) as `Alumnos de otros Grupos`
```



"Tecnicas Avanzadas de Programacion"

```
{
  "legajo": 1491611,
  "apellido": "Vazquez",
  "nombre": "Ramiro"
}
```

```
[
{
  "legajo": 1491696,
  "apellido": "Crespi",
  "nombre": "Gustavo"
}
,
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
,
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
]
```

"Diseño de Sistemas"

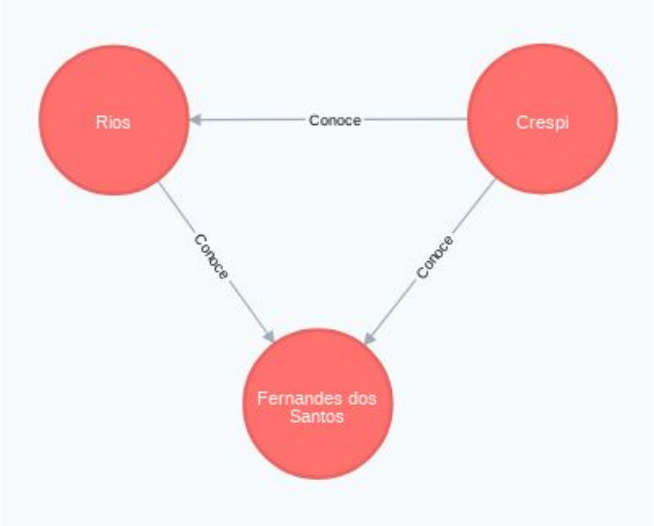
```
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
```

```
[
{
  "legajo": 1491696,
  "apellido": "Crespi",
  "nombre": "Gustavo"
}
,
{
  "legajo": 1493346,
  "apellido": "Rios",
  "nombre": "Juan"
}
,
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
,
{
  "legajo": 1491611,
  "apellido": "Vazquez",
  "nombre": "Ramiro"
}
]
```

2

Listado de docentes que dictaron más de una materia.

```
MATCH (d:Docente)-[di:Dicto]->(m:Materia)
WITH d as Docente, count(di) as materias
WHERE materias > 1
RETURN Docente
```



Docente

```
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
```

```
{
  "legajo": 1491696,
  "apellido": "Crespi",
  "nombre": "Gustavo"
}
```

```
{
  "legajo": 1493346,
  "apellido": "Rios",
  "nombre": "Juan"
}
```

3

Tu propio promedio de calificaciones. (No devuelve grafo)
De un alumno específico

```
MATCH (a:Alumno { legajo: 1493243 })-[c:Curso]->()
RETURN toFloat(sum(c.nota))/count(c) as Promedio
```

Promedio
8.125

Variante, de todos los alumnos

```
MATCH (a:Alumno)-[c:Curso]->()
RETURN a.apellido as Alumno, toFloat(sum(c.nota))/count(c) as Promedio
```

Alumno	Promedio
"Santoalla"	8.125
"Crespi"	7.875
"Fernandes dos Santos"	7.857142857142857
"Vazquez"	8
"Rios"	6.285714285714286
"Reyes"	9

4

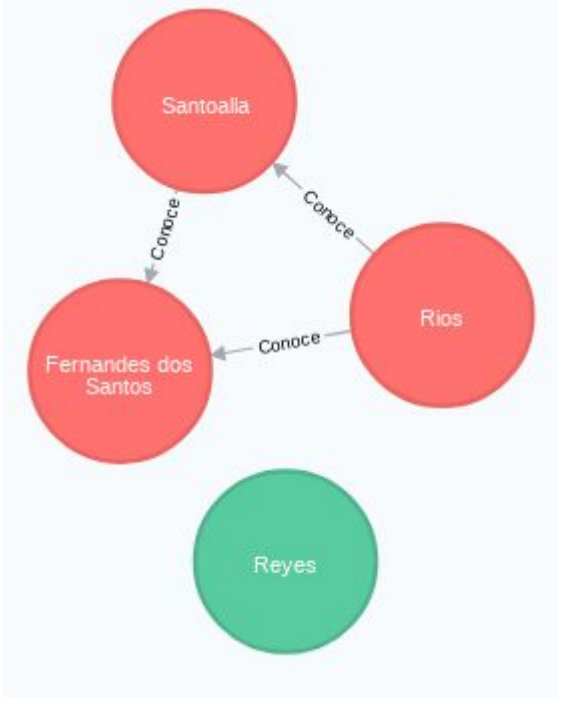
Listado para recomendación de alumnos que cursaron en el mismo curso y cuatrimestre pero no se conocen entre sí.

En este caso se muestra agrupado por alumno, todos los que cursaron con él pero son desconocidos.

```
MATCH (a1:Alumno)-[c1:Curso]->(m:Materia)<-[c2:Curso]-(a2:Alumno)
WHERE c1.nroCurso = c2.nroCurso
AND c1.ano = c2.ano
AND c1.cuatrimestre = c2.cuatrimestre
AND NOT (a1)-[:Conoce]-(a2)
RETURN a1 as Alumno, collect(a2) as Desconocidos
```

Alumno


```
{
  "legajo": 1324567,
  "apellido": "Reyes",
  "nombre": "Hugo"
}
```

Desconocidos


```
[
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
,
{
  "legajo": 1493346,
  "apellido": "Rios",
  "nombre": "Juan"
}
,
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
]
```

```
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
```

```
[
  {
    "legajo": 1324567,
    "apellido": "Reyes",
    "nombre": "Hugo"
  }
]
```

```
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
```

```
[
  {
    "legajo": 1324567,
    "apellido": "Reyes",
    "nombre": "Hugo"
  }
]
```

```
{
  "legajo": 1493346,
  "apellido": "Rios",
  "nombre": "Juan"
}
```

```
[
  {
    "legajo": 1324567,
    "apellido": "Reyes",
    "nombre": "Hugo"
  }
```

5

Listado de los conocidos de tus conocidos, hasta longitud 2, e indefinida.

Longitud 2

```
MATCH (a:Alumno { legajo: 1493346 })-[:Conoce*..2]-(b)
WHERE a <> b
RETURN DISTINCT b as Conocido
```

Conocido

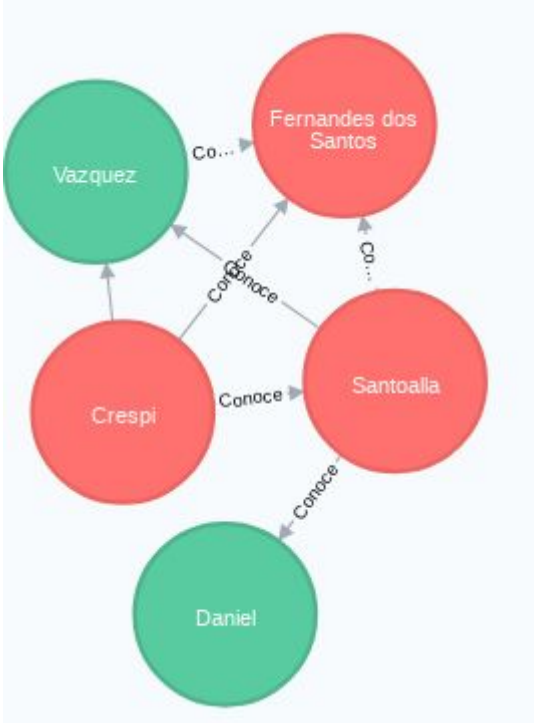

```
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
```

```
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
```

```
{
  "legajo": 1491611,
  "apellido": "Vazquez",
  "nombre": "Ramiro"
}
```

```
{
  "legajo": 1491696,
  "apellido": "Crespi",
  "nombre": "Gustavo"
}
```

```
{
  "legajo": 1492800,
  "apellido": "Daniel",
  "nombre": "Zinghini"
}
```

Started streaming 5 records after 3 ms and completed after 4 ms.

Longitud Indefinida

```
MATCH (a:Alumno  { legajo: 1493346 })-[:Conoce*]-(b)
WHERE a <> b
RETURN DISTINCT b as Conocido
```

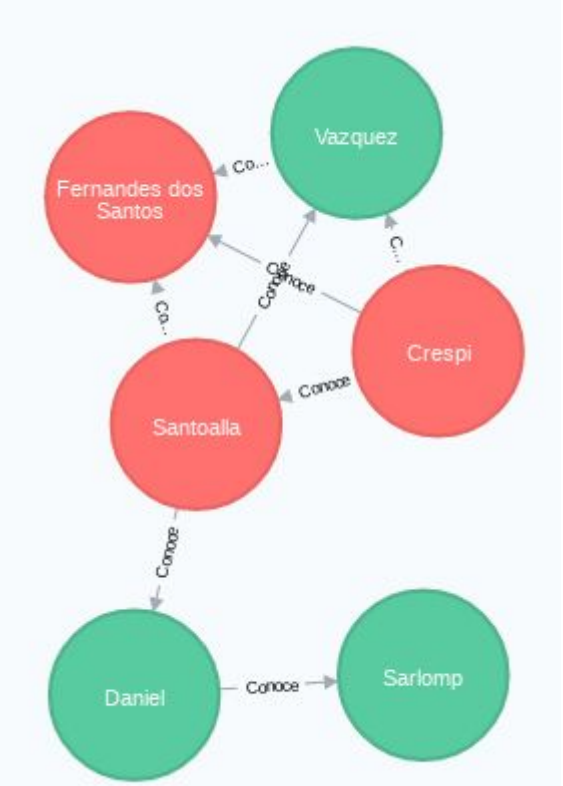
Conocido


```
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
```

```
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
```

```
{
  "legajo": 1492800,
  "apellido": "Daniel",
  "nombre": "Zinghini"
}
```

```
{
  "legajo": 1330298,
  "apellido": "Sarlomp",
  "nombre": "Juan"
}
```


```
{
  "legajo": 1491611,
  "apellido": "Vazquez",
  "nombre": "Ramiro"
}
```

```
{
  "legajo": 1491696,
  "apellido": "Crespi",
  "nombre": "Gustavo"
```


arted streaming 6 records after 3 ms and completed after 58 ms.

6

Apellido y nombre de alumnos que también son docentes (ver pedido adicional para esta consulta).

```
MATCH (a: Alumno: Docente)
RETURN DISTINCT a.nombre, a.apellido
```

a.nombre	a.apellido
"Gustavo"	"Crespi"
"Juan"	"Fernandes dos Santos"
"Juan"	"Rios"
"Gaston"	"Santoalla"

Variante

En vez de tener un nodo Alumno y uno Docente, tener un Nodo Persona que tenga ambas relaciones (curso y dicto)

```
MATCH (p: Persona)-[:Curso]-(), (p)-[:Dicto]-()
return DISTINCT p.nombre, p.apellido
```

7

Dado un alumno en particular, recomendación de materias electivas que no haya cursado, en base al criterio de haber sido cursadas por otros alumnos que cursaron por lo menos una en común con él.

```
MATCH (a1:Alumno { legajo: 1493243 })-[:Curso]->(m:Materia)<-[:Curso]-(a2:Alumno),
(a2)-[:Curso]->(e:Materia { electiva: true })
WHERE NOT (a1)-[:Curso]-(e)
RETURN DISTINCT e as `Electiva Recomendada`
```

Electiva Recomendada


```
{
  "nombre": "Administracion de
Personal",
  "electiva": true
}
```

8

Hacer una variante del ítem anterior, recomendando sólo si el otro alumno es un contacto (directo o indirecto).

```
MATCH (a1:Alumno { legajo: 1493243 })-[:Conoce*]-(a2:Alumno),
(a1)-[:Curso]->(m:Materia)<-[:Curso]-(a2), (a2)-[:Curso]->(e:Materia { electiva: true })
WHERE NOT (a1)-[:Curso]-(e)
RETURN DISTINCT e as `Electiva Recomendada`
```

Si se hace con el legajo 1324567 por ejemplo, no se obtienen resultados ya que no es contacto de nadie, en cambio con la query 7 se obtiene *Tecnicas Avanzadas de Programcion* como sugerencia.

9

Consulta adicional, decidida por el grupo.

Listado para recomendación de compañeros de grupo teniendo como criterio que hayan cursado más de 5 materias juntos.

```
MATCH (a1:Alumno)-[c1:Curso]->(m:Materia)<-[c2:Curso]-(a2:Alumno)
WHERE c1.ano = c2.ano
AND c1.cuatrimestre = c2.cuatrimestre
AND c1.nroCurso = c2.nroCurso
WITH a1, a2, count(c1) as cantMateriasJuntos
WHERE cantMateriasJuntos > 5
RETURN a1 as Alumno, collect(DISTINCT a2) as Recomendados
```

Alumno


```
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
```


Recomendados

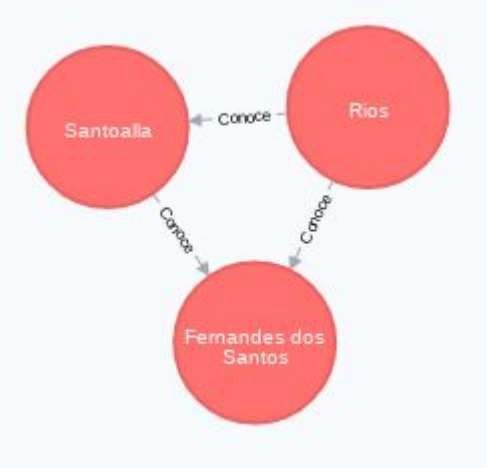

```
[
  {
    "legajo": 1493346,
    "apellido": "Rios",
    "nombre": "Juan"
  }
,
  {
    "legajo": 1493243,
    "apellido": "Santoalla",
    "nombre": "Gaston"
  }
]
```



```
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
```



```
[
  {
    "legajo": 1493346,
    "apellido": "Rios",
    "nombre": "Juan"
  }
,
  {
    "legajo": 1491738,
    "apellido": "Fernandes dos Santos",
    "nombre": "Juan"
  }
]
```

```
{
  "legajo": 1493346,
  "apellido": "Rios",
  "nombre": "Juan"
}
```



```
[
{
  "legajo": 1491738,
  "apellido": "Fernandes dos Santos",
  "nombre": "Juan"
}
,
{
  "legajo": 1493243,
  "apellido": "Santoalla",
  "nombre": "Gaston"
}
```

Grafo Completo