

1.CARGA DE NODOS Y RELACIONES

Carga de estudiantes

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

Carga de profesores

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

Carga de relación “esAmigo” entre estudiantes

```
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)-[m:esAmigo]->(e2) return m
```

Carga de asignaturas

```
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:
toInteger(csvLine.semestre),creditos: toInteger(csvLine.creditos)})
```

Carga de relación “imparte” entre profesor y asignatura

```
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)-[m:Imparte]->(a) return m
```

Carga de relación “matriculado” entre estudiante y asignatura

```
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]->(a) return m
```

Carga de relación “trabajaCon” entre profesores

```
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)-[m:trabajaCon]->(p2) return m
```

2.QUERIES DE CONSULTA

A.Names of subjects taught by Professor Clarap

match (p:Profesor)-[m:Imparte]->(a:Asignatura) where p.nombre = 'Clarap' return a.nombre

B.Names of first year subjects

match (a:Asignatura) where a.curso = 1 return a.nombre

C.Names of subjects arranged alphabetically

match (a:Asignatura) return a.nombre order by a.nombre

D.Names of third or fourth year subjects

match (a:Asignatura) where a.curso = 3 or a.curso = 4 return a.nombre

E.Eliminate the semester property of the nodes Subject

match (a:Asignatura) remove a.semestre return a

F.Change the name of one of the subjects

match (a:Asignatura{nombre:'Estadística'}) set a.nombre = 'Bioestadística' return a

G.Number of subjects in which the student Maria is enrolled

match (e:Estudiante)-[m:Matriculado]->(a:Asignatura) where e.nombre = 'María' return count(a)

En caso de que haya varias Marías matriculadas, se tendría que usar e.id = el id de la María específica.

El uso del id es importante ya que puede haber valores repetidos en los nombres u otras propiedades en una clase de nodos, al ser una propiedad con valor único para cada nodo. (ej. Puede haber dos estudiantes con nombre María, pero cada una tendría un id único). En este caso utilizamos e.names porque tenemos una lista de estudiantes muy pequeña y sabemos que no hay nombres repetidos a priori, pero por lo general lo conveniente es utilizar la propiedad id.

H.Names of Pedro's friends

match (e1:Estudiante)-[m:esAmigo]->(e2:Estudiante) where e1.nombre = 'Pedro' return e2.nombre

Names of friends and friends of Pedro's friends

```
match (e1:Estudiante)-[m:esAmigo]->(e2:Estudiante)-[m2:esAmigo]->(e3:Estudiante) where  
e1.nombre = 'Pedro' return e2.nombre,e3.nombre
```

J. Names of subjects in which one of the friends of the student Mario is enrolled

```
match (e1:Estudiante)-[m:esAmigo]->(e2:Estudiante)-[m2:Matriculado]->(a:Asignatura) where  
e1.nombre = 'Luis' return a.nombre
```

K. Names of students who are enrolled in any of the subjects taught by Professor López

```
match (p:Profesor)-[m:Imparte]->(a:Asignatura)<-[m2:Matriculado]-(e:Estudiante) where  
p.nombre='Lopez' return e.nombre
```

L. Names of students who are enrolled in any of the subjects taught by any of the teachers who work with Professor Sergio

```
match (p1:Profesor)-[m1:trabajaCon]->(p2:Profesor)-[m2:Imparte]->(a:Asignatura)<-  
[m3:Matriculado]-(e:Estudiante) where p1.nombre='Sergio' return e.nombre
```

3. QUERIES DE CONSULTA PLANTEADAS

Nombre los profesores que imparten clase a los amigos de los amigos de 'Rosa', y que asignaturas son impartidas por estos profesores a dichos alumnos.

```
match (e1:Estudiante)-[m1:esAmigo]->(e2:Estudiante)-[m2:Matriculado]->(a:Asignatura)<-  
[m3:Imparte]-(p1:Profesor) where e1.nombre = 'Rosa' return p1.nombre,a.nombre
```

Conocer el número de asignaturas que imparten los profesores que trabajan con Clarap

```
match (p1:Profesor)-[m1:trabajaCon]->(p2:Profesor)-[m2:Imparte]->(a:Asignatura) where  
p1.nombre = 'Clarap' return count(a)
```

Contar el número de años que tiene la carrera

```
match (a:Asignatura) return max(a.curso)
```