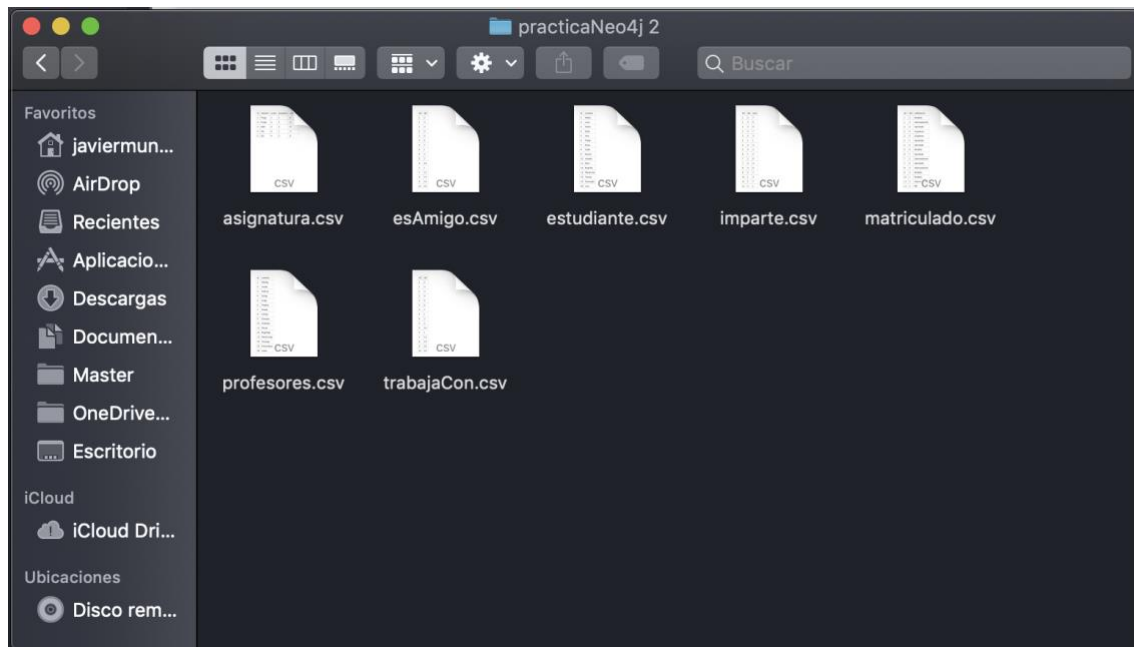


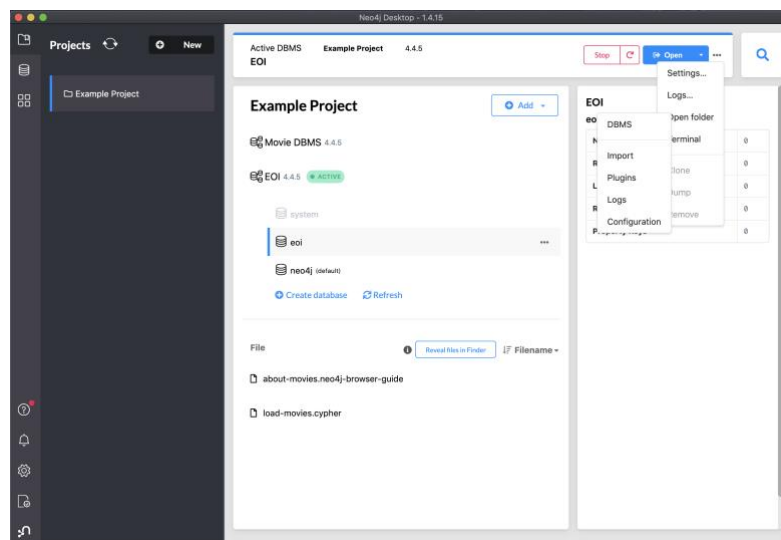
Ejercicio 1

Descargamos el archivo .zip facilitado en la página y lo descomprimos. Se descomprimirá en una carpeta que tendrá un total de 7 ficheros:

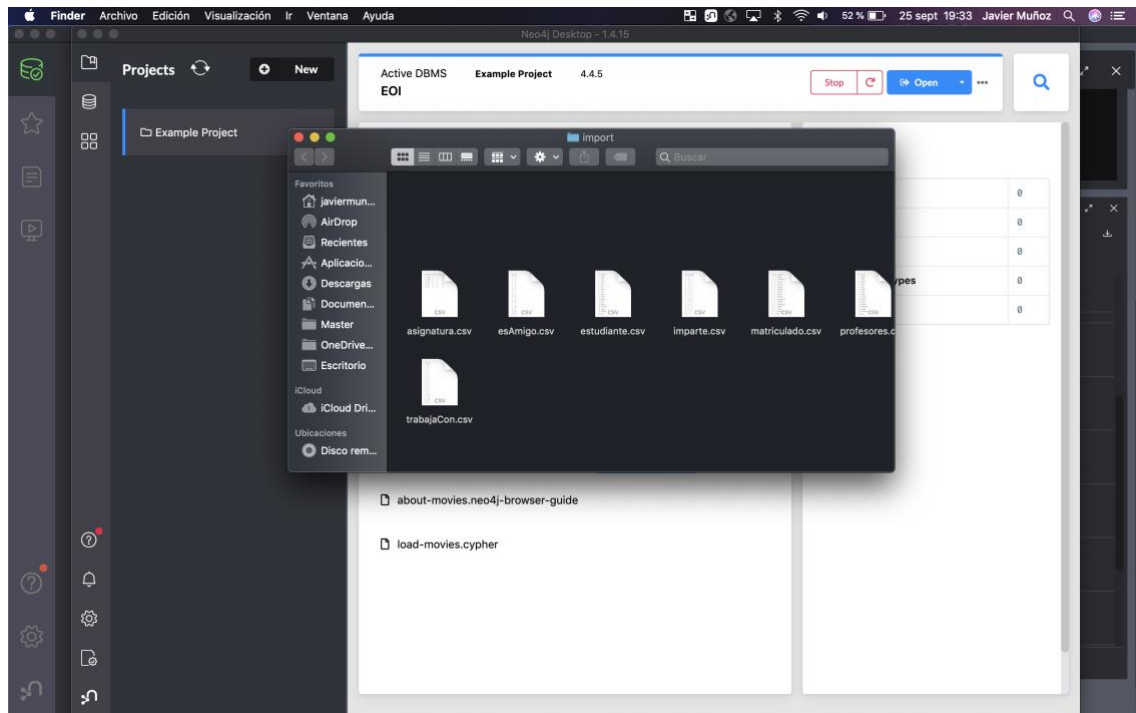


Cada uno de los ficheros representa un conjunto de datos relacionado con el ejercicio que se va a realizar, para las consultas en Cypher sobre la base de datos de Neo4j.

Para que Neo4j “reconozca” estos ficheros, se ha de importar al entorno cada uno de ellos.



Una vez seleccionamos la opción “import”, se nos abrirá una carpeta propia de Neo4j donde podremos dejar los ficheros que utilizaremos para cargar nuestros datos:



Ya con la información importada, podemos usar los ficheros para cargar los datos en Neo4j.

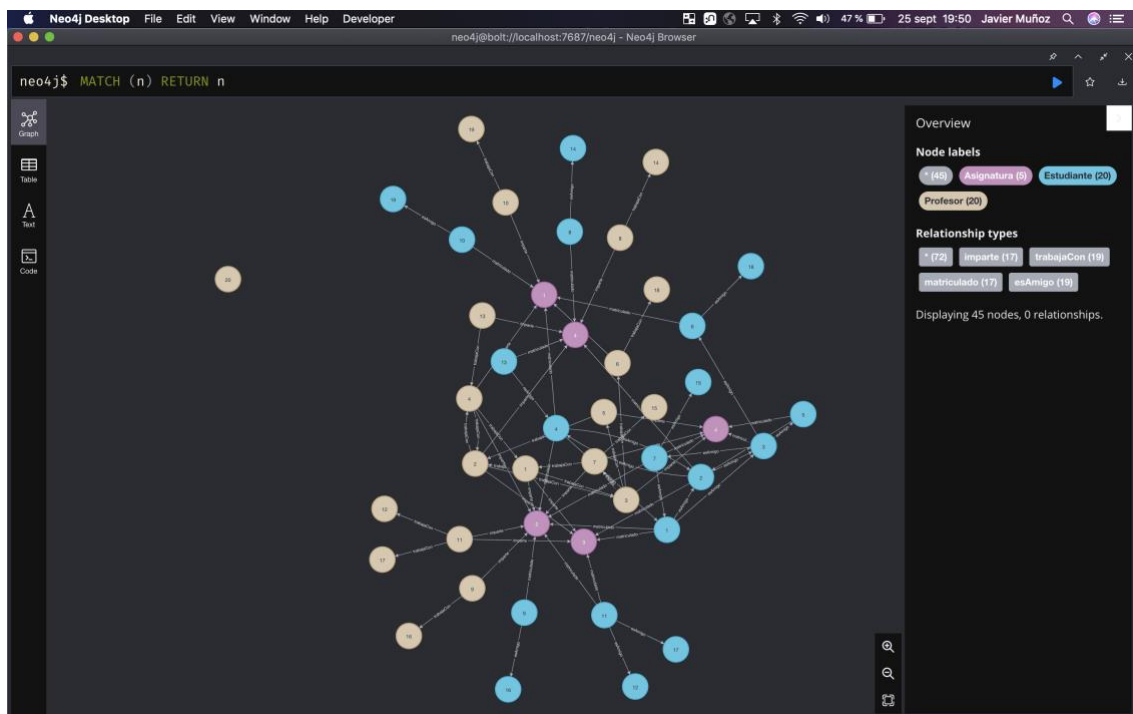
Ejercicio 2

(Las sentencias que se han realizado para realizar la carga de datos en el grafo de la base de datos están en el fichero **Ej2.txt**)

El criterio que se ha seguido para hacer la carga de datos ha sido la siguiente:

- Se cargan primero los datos que representan las entidades. En este caso, los ficheros serían: **asignatura.csv**, **estudiante.csv** y **profesores.csv**.
- Una vez verificados que las entidades (en forma de nodos en el grafo) están cargados, procedemos a cargar los ficheros que representan las relaciones entre las entidades. Estos ficheros serían: **esAmigo.csv**, **imparte.csv**, **matriculado.csv** y **trabajaCon.csv**. Hay que destacar que, en este caso, la creación de las relaciones ha sido mas compleja que la de las entidades.

Una vez la carga de los datos está completada, nos queda un grafo como el siguiente:



Donde se pueden apreciar las distintas relaciones que se han ido construyendo en las conexiones entre las entidades.

Ya con el grafo construido, procedemos a realizar las consultas del ejercicio 3.

Ejercicio 3

Consulta 1: Nombres de asignaturas impartidas por la profesora 'Clarap' en el Aula 5

```
1 MATCH (p:Profesor)-[i:imparte]→(a:Asignatura)
2 WHERE p.nombre = 'Clarap' AND i.aula = 5
3 RETURN a.nombre
```

(no changes, no records)

Como se puede observar en la imagen superior, la consulta no tiene ningún resultado, por lo que nos tomamos la libertad de modificar los campos sobre los que filtrar, para obtener una consulta con resultado. En este caso, comprobando que clases imparte el profesor 'Pedrop' en el aula 3.

```
1 MATCH (p:Profesor)-[i:imparte]→(a:Asignatura)
2 WHERE p.nombre = 'Pedrop' AND i.aula = 3
3 RETURN a.nombre
```

	a.nombre
1	"SI1"

Consulta 2: Calificación obtenida por el estudiante Pérez en la asignatura de Física

```
1 MATCH (e:Estudiante)-[m:matriculado]→(a:Asignatura)
2 WHERE e.nombre = 'Pérez' AND a.nombre = 'Física'
3 RETURN m.calificacion
```

(no changes, no records)

De nuevo, nos vemos ante la situación de que no hay ningún resultado ante la consulta. Revisando los datos, vemos que no existe ningún estudiante llamado Pérez y ninguna asignatura 'Física'. De nuevo, cambiamos los criterios de filtro a unos que puedan dar resultado sobre la consulta, obteniendo el siguiente resultado.

```
1 MATCH (e:Estudiante)-[m:matriculado]→(a:Asignatura)
2 WHERE e.nombre = 'Juan' AND a.nombre = 'SI1'
3 RETURN m.calificacion
```

	m.calificacion
1	"Suspenso"

Consulta 3: Número de créditos matriculados por la estudiante 'IsabelJ'

```
1 MATCH (e:Estudiante)-[m:matriculado]→(a:Asignatura)
2 WHERE e.nombre = 'IsabelJ'
3 RETURN SUM(a.creditos)
```

SUM(a.creditos)
0

Ningún resultado aparece de nuevo para esta consulta, buscamos por la estudiante cuyo nombre es 'Elisa' y obtenemos el siguiente resultado:

```
1 MATCH (e:Estudiante)-[m:matriculado]→(a:Asignatura)
2 WHERE e.nombre = 'Elisa'
3 RETURN SUM(a.creditos)
```

SUM(a.creditos)
9

Consulta 4: Nombres de las asignaturas de 1º curso.

```
1 MATCH (a:Asignatura{curso:1})
2 RETURN a.nombre
```

a.nombre
"Prog1"

Consulta 5: Nombres de las asignaturas ordenadas alfabéticamente

```
1 MATCH (a:Asignatura)
2 RETURN a.nombre ORDER BY a.nombre
```

a.nombre
1 "EDA"
2 "Prog1"
3 "Prog2"
4 "SI1"
5 "SI1"

Anotar en este caso, que por defecto, Neo4j ordena cadenas de caracteres alfabéticamente si se pone la cláusula ORDER BY de manera automática. Si se quisiese hacer el orden inverso, se usaría la cláusula **DESC**.

Consulta 6: Nombres de asignaturas de tercer o cuarto curso

```
1 MATCH (a:Asignatura)
2 WHERE a.curso = 2 OR a.curso = 3
3 RETURN a.nombre ORDER BY a.nombre
```

	a.nombre
1	"EDA"
2	"Prog2"

Consulta 7: Eliminar la propiedad semestre de una asignatura

```
1 MATCH (a:Asignatura{nombre:'SI1'})
2 REMOVE a.semestre
3 RETURN a.nombre, a.semestre
```

	a.nombre	a.semestre
1	"SI1"	null
2	"SI1"	null

Consulta 8: Cambiar el nombre de una de las asignaturas

```
1 MATCH (a:Asignatura{nombre:'SI1'})
2 SET a.nombre = 'SI2'
3 RETURN a.nombre, a.semestre
```

	a.nombre	a.semestre
1	"SI2"	null
2	"SI2"	null

Como se puede observar, una vez lanzada la consulta, si tratamos de filtrar sobre esa propiedad con el valor antiguo, la consulta no retorna nada.

```
1 MATCH (a:Asignatura{nombre:'SI1'})
2 RETURN a.nombre, a.semestre
```

	(no changes, no records)
--	--------------------------

Consulta 9: Número de asignaturas en las que está matriculado la estudiante Elisa

```
1 MATCH (:Estudiante{nombre:'Elisa'})-[:matriculado]->(a:Asignatura)
2 RETURN COUNT(a.nombre)
```

	COUNT(a.nombre)
1	2

Consulta 10: Nombres de los amigos del estudiante Pedro

```
1 MATCH (:Estudiante{nombre:'Paco'})-[:esAmigo]->(e:Estudiante)
2 RETURN e.nombre
```

	e.nombre
1	"Mar"
2	"Begoña"

Consulta 11: Nombres de los amigos y de los amigos de los amigos del estudiante Pedro

```
1 MATCH (:Estudiante{nombre:'Pedro'})->(amigosde:Estudiante)->(amigosdeamigos:Estudiante)
2 RETURN DISTINCT(amigosde.nombre), (amigosdeamigos.nombre)
```

	(amigos1.nombre)	(amigos2.nombre)
1	"Rosa"	"Fernando"
2	"Rosa"	"María"
3	"Prado"	"Carlos"
4	"Ana"	"Juan"

Consulta 12: Nombres de asignaturas en las que está matriculado alguno de los amigos de la estudiante Elisa

```
1 MATCH (e:Estudiante{nombre:'Elisa'})->(:Estudiante)-[:matriculado]->(a:Asignatura)
2 RETURN a.nombre
```

	a.nombre
1	"SI2"
2	"EDA"
3	"EDA"
4	"Prog2"

Consulta 13: Nombres de estudiantes que están matriculados en alguna de las asignaturas que imparte la profesora Elisap

1 MATCH (p:Profesor{nombre:'Elisap'})-[:imparte]→(a:Asignatura)←[:matriculado]-(e:Estudiante)

2 RETURN e.nombre

3

Table

e.nombre

1 "Andrés"

2 "Elisa"

3 "Prado"

4 "Elisa"

5 "Rosa"

6 "Paco"

7

Consulta 14: Nombres de estudiantes que están matriculados en alguna de las asignaturas que imparte alguno de los profesores que trabaja con la profesora Elisap

1 MATCH (p:Profesor{nombre:'Elisap'})-[:trabajaCon]→(:Profesor)-[:imparte]→(a:Asignatura)←[:matriculado]-(e:Estudiante)

2 RETURN e.nombre

Table

e.nombre

1 "Juan"

2 "Macarena"

3 "Clara"

4 "Paco"

5 "María"

6 "Juan"

7