

Introducción – parte 2

LOAD CSV

Para pequeñas cantidades de datos; análisis y POC (proof of concept).

Carpeta para importar datos

Por default es la carpeta “**import**”

\neo4jDatabases\database-324d5caa-131b-41b5-aeda-21991ebd01cd\installation-4.0.3\import

//Si ejecutan esta consulta

```
LOAD CSV FROM 'file:///C:/Users/julio/Documents/t.txt' AS line
```

```
CREATE (c:Course {name:line[1]})
```

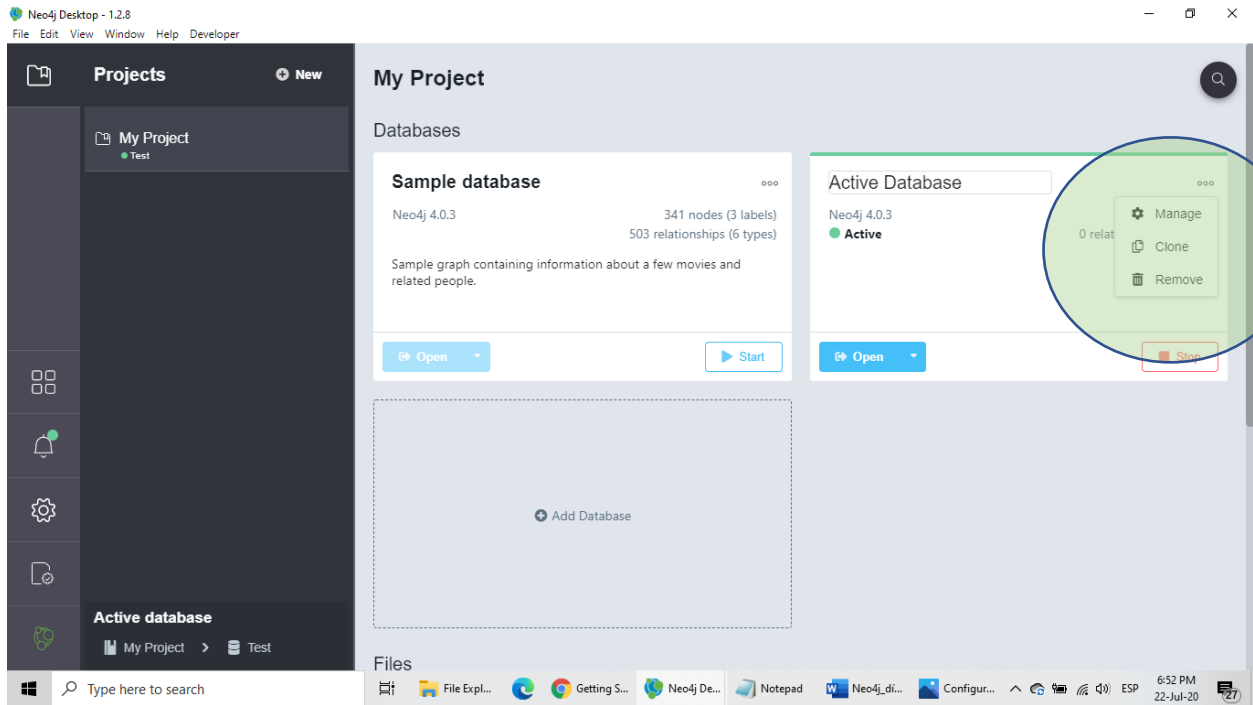
//El error indica que busca cargar el archivo pero no lo localiza.

ERROR Neo.ClientError.Statement.ExternalResourceFailed

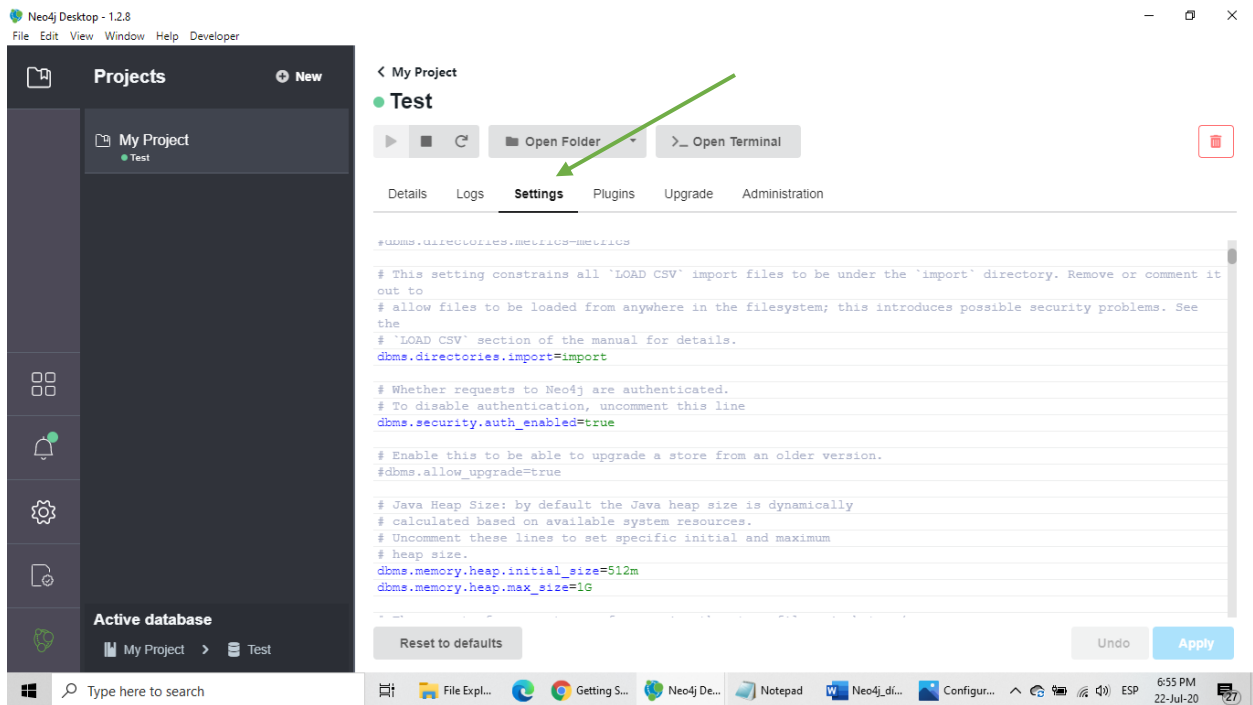
```
Couldn't load the external resource at:  
file:/C:/Users/julio/OneDrive/Escritorio/SA/neo4jDatabases/database-1d7f0075-a654-4d3b-b127-  
6b1884527ecb/installation-4.0.3/import/Users/julio/Documents/t.txt
```

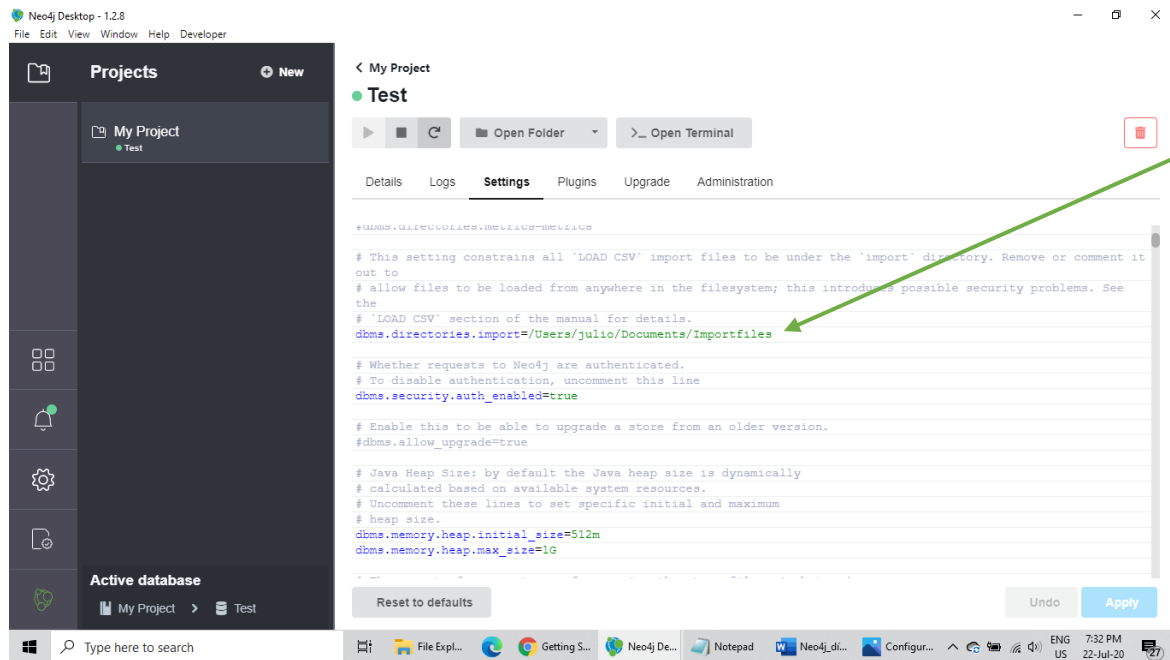
Para elegir la carpeta de carga:

Ir a Neo4j Desktop



Pestaña de configuración





Aplicar y reiniciar el servidor

//Ahora esto funciona

```
LOAD CSV FROM 'file:///file.csv' AS line
```

```
CREATE (c:Course {name:line[1]})
```

Indica el número de columna (la primera es cero)

Ejemplo de Estudiantes y Grupos

//estudiantes de MCD

```
LOAD CSV WITH HEADERS FROM 'file:///Estudiantes.csv' AS line
```

```
MERGE (titulo:Titulo {name: line.Titulo})
```

```
CREATE (estudiante:Estudiante {id: toInteger(line.id), name:line.APELLIDO})
```

```
CREATE (estudiante)-[:DEGREE]->(titulo)
```

Crea un nodo y lo etiqueta como "Estudiante"

//grupos de BD

```
LOAD CSV WITH HEADERS FROM 'file:///grupos.csv' AS line
```

```
CREATE (g:Grupo {id: toInteger(line.id), name:line.nombre, DER:line.DER,  
implementation:line.Implementation })
```

//relación entre estudiantes y grupos

```
LOAD CSV WITH HEADERS FROM "file:///joins.csv" AS line
```

```
MATCH (estudiante:Estudiante {id: toInteger(line.estudianteld)}),(grupo:Grupo {id:  
toInteger(line.grupold)})
```

```
CREATE (estudiante)-[:WORKED]->(grupo)
```

//(ver que se arma un grafo conexo)

queries

// devuelve el Estudiante 'Vigo' y su etiqueta Grupo

```
MATCH (:Estudiante { name: 'Vigo' })--(grupo:Grupo)
```

```
RETURN grupo.name
```

//Ídem anterior pero con etiqueta Título

```
MATCH (:Estudiante { name: 'Yoon' })--(tit:Título)
```

```
RETURN tit.name
```

//Todos los estudiantes del grupo M

```
MATCH (g:Grupo { name: 'M' })--(est:Estudiante)
```

```
RETURN est.name
```

//los nodos conectados con Vigo en una relación saliente

```
MATCH (:Estudiante { name: 'Vigo' })-->(nodo)
```

```
RETURN nodo.name
```

//los nodos conectados con Ingeniero Industrial en una relación entrante

```
MATCH (:Titulo { name: 'Ingeniero Industrial' })<--(nodo)
```

```
RETURN nodo.name
```

//a qué nodos Título llega Yoon en hasta 7 saltos

```
MATCH (br { name: 'Yoon' })-[:DEGREE|WORKED*1..7]-(titulo:Titulo)
```

```
RETURN titulo.name
```

//el grupo MN como grafo

```
MATCH (mn:Grupo { name: 'MN' })-[*0..1]-(x)
```

```
RETURN x
```

//el grafo de “Ingeniero Industrial”

```
MATCH (ingind:Titulo { name: 'Ingeniero Industrial' })-[*0..1]-(x) // distancia del grafo
```

```
RETURN x
```

//otra forma

```
MATCH p =(licecon { name: 'Lic. en Economía' })<--()
```

```
RETURN p
```

//el grafo de Matys flecha saliente

```
MATCH p =(Matys { name: 'Matys' })-->()
```

```
RETURN p
```

//nodo entrante y saliente de una relación dada por id

```
MATCH (a)-[r]-(b)
```

```
WHERE id(r)= 82
```

```
RETURN a,b
```

// Encontrar la ruta más corta entre dos nodos, siempre que la ruta tenga un máximo de 10 relaciones.

//Dentro de los paréntesis, define un enlace único de una ruta: el nodo inicial, la relación de conexión y el nodo final.

MATCH

(ingind:Titulo { name: 'Ingeniero Industrial' })

,(contpub:Titulo { name: 'Contador Público' })

, p = shortestPath((ingind)-[*..10]-(contpub))

RETURN p

(Hacer un explain de esta consulta)

EXPLAIN: estima la cantidad de filas, mientras PROFILE las calcula. EXPLAIN es el plan de ejecución, no la actual ejecución del query.

//¿Algún ingeniero industrial se vincula con la implementación del Relevamiento en salud? Si es así, dar

//el camino más corto

MATCH

(ingind:Titulo { name: 'Ingeniero Industrial' })

,(relsal:Grupo { implementation: 'Relevamiento en salud' })

, p = shortestPath((ingind)-[*..10]-(relsal))

RETURN p

//¿Qué no sea Peirano?

MATCH

(ingind:Titulo { name: 'Ingeniero Industrial' })

,(relsal:Grupo { implementation: 'Relevamiento en salud' })

, p = shortestPath((ingind)-[*..30]-(relsal))

WHERE NONE (n IN nodes(p) WHERE n.name= 'Peirano')

RETURN p

//¿Qué tampoco sea Assereto Farroni?

MATCH

(ingind:Titulo { name: 'Ingeniero Industrial' })

,(relsal:Grupo { implementation: 'Relevamiento en salud' })

, p = shortestPath((ingind)-[*..30]-(relsal))

WHERE NONE (n IN nodes(p) WHERE n.name= 'Peirano' or n.name= 'Assereto Farroni')

RETURN p

//Todos los caminos entre el DER y la implementación del problema del Religiones

MATCH

(derrel:Grupo { DER: 'Religiones' })

,(implrel:Grupo { implementation: 'Religiones' })

, p = ((derrel)-[*..20]-(implrel))

RETURN p