

Universidad del Valle de Guatemala
Algoritmos y Estructura de Datos
Catedrático:
Moises Alonso
Sección: 40



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

Manual de Usuario
Proyecto Final

Carné 22674, Emilio Reyes
Carné 22097, Sara Guzman

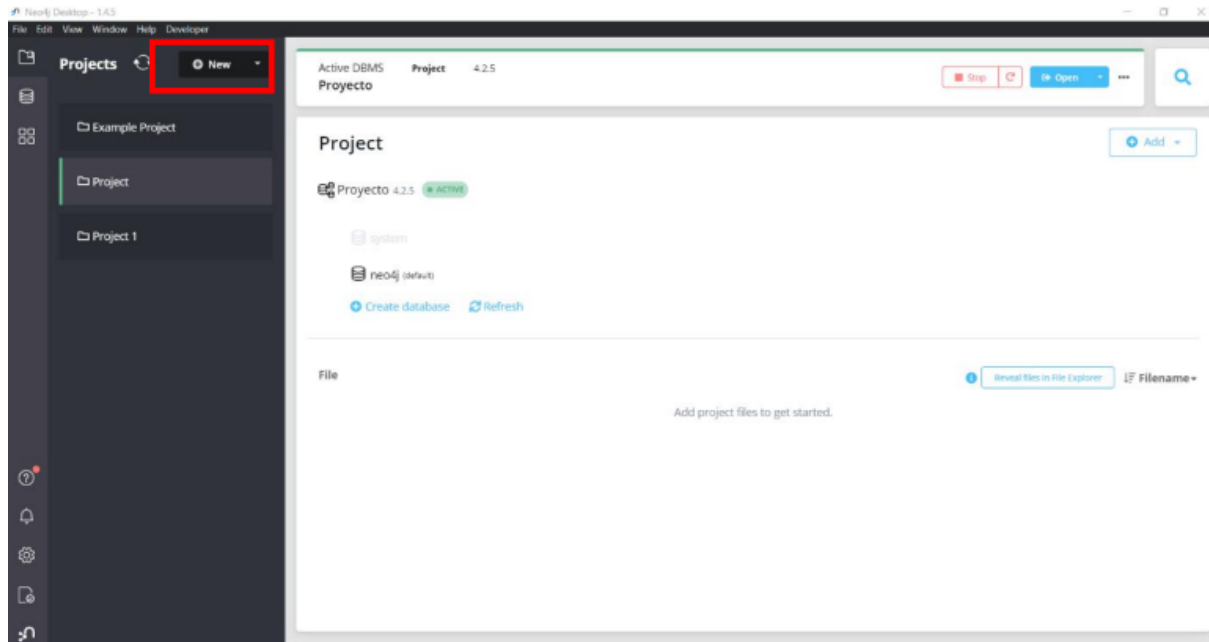
Paso 1.

Se debe instalar Neo4j desktop, si aún no se tiene, usando el siguiente enlace:

<https://neo4j.com/download/>

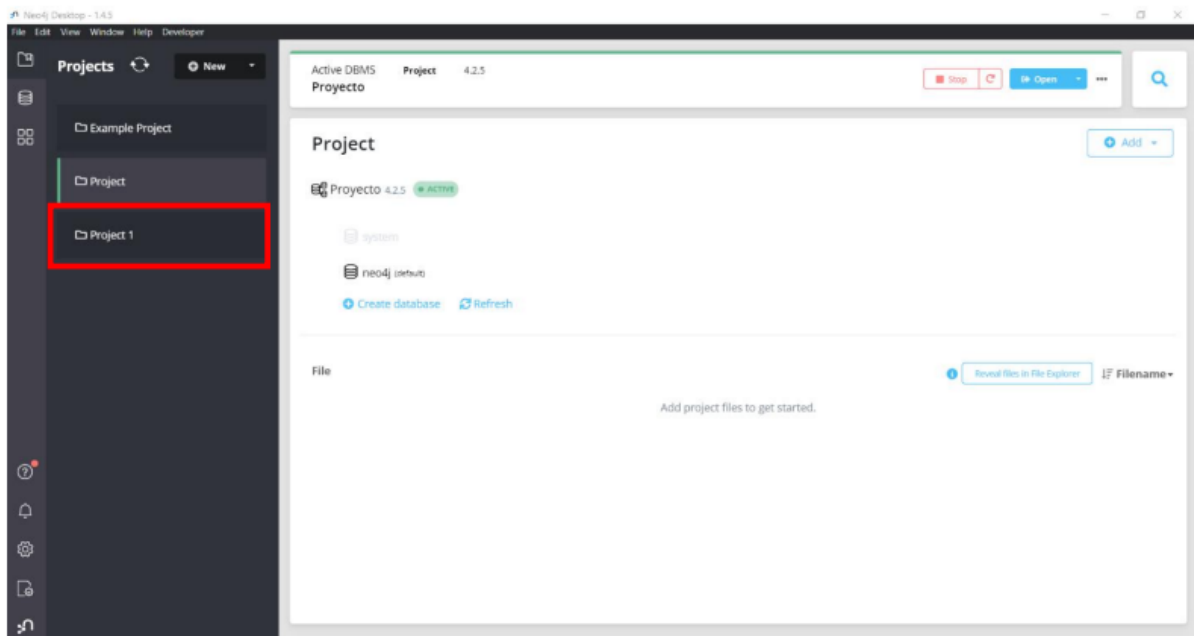
Paso 2.

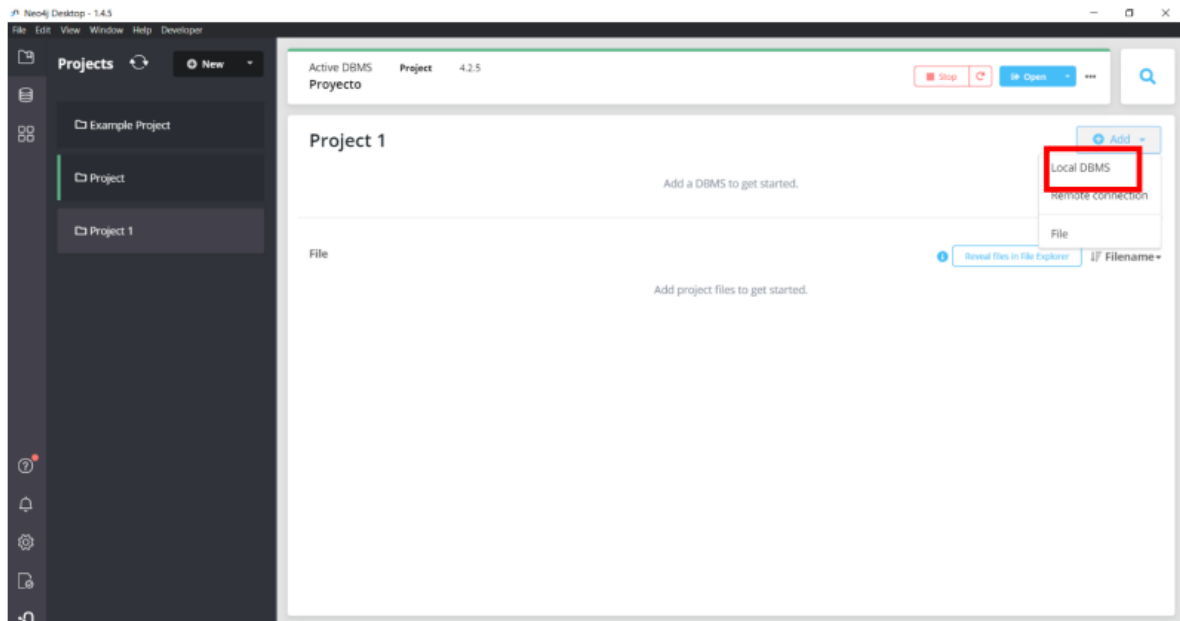
Se debe crear un nuevo proyecto, como mostrado a continuación.



Paso 3.

El proyecto ha sido creado, se debe crear una nueva base de datos en add -> local DBMS.



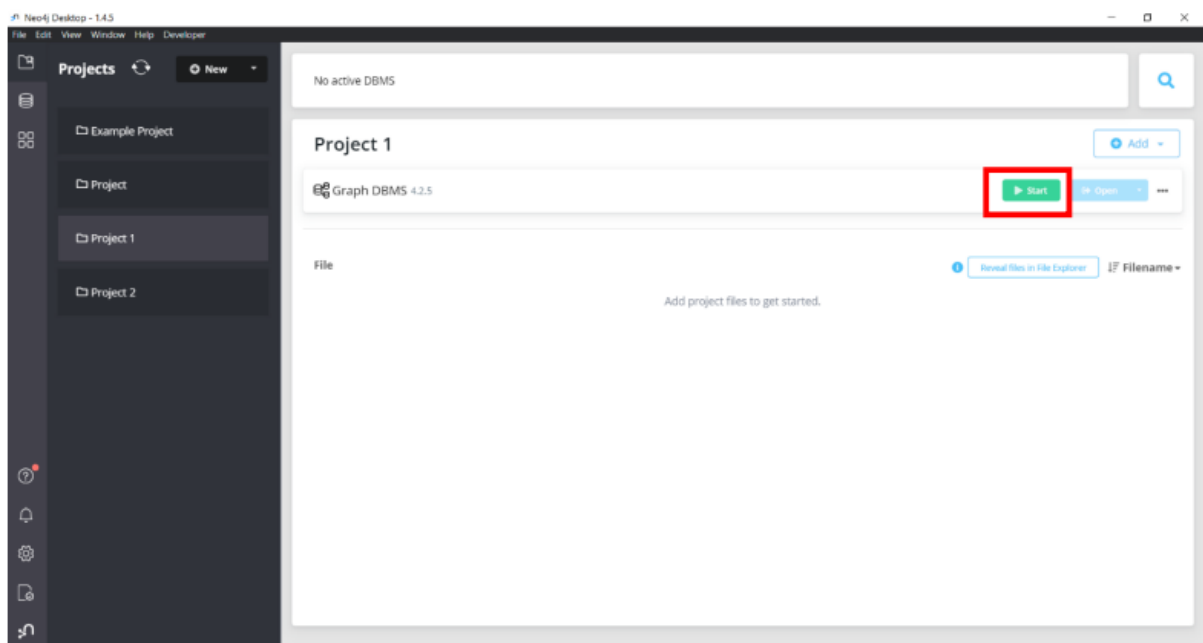


Paso 4.

La contraseña que pida ingresar para estos propósitos será “12345678”.

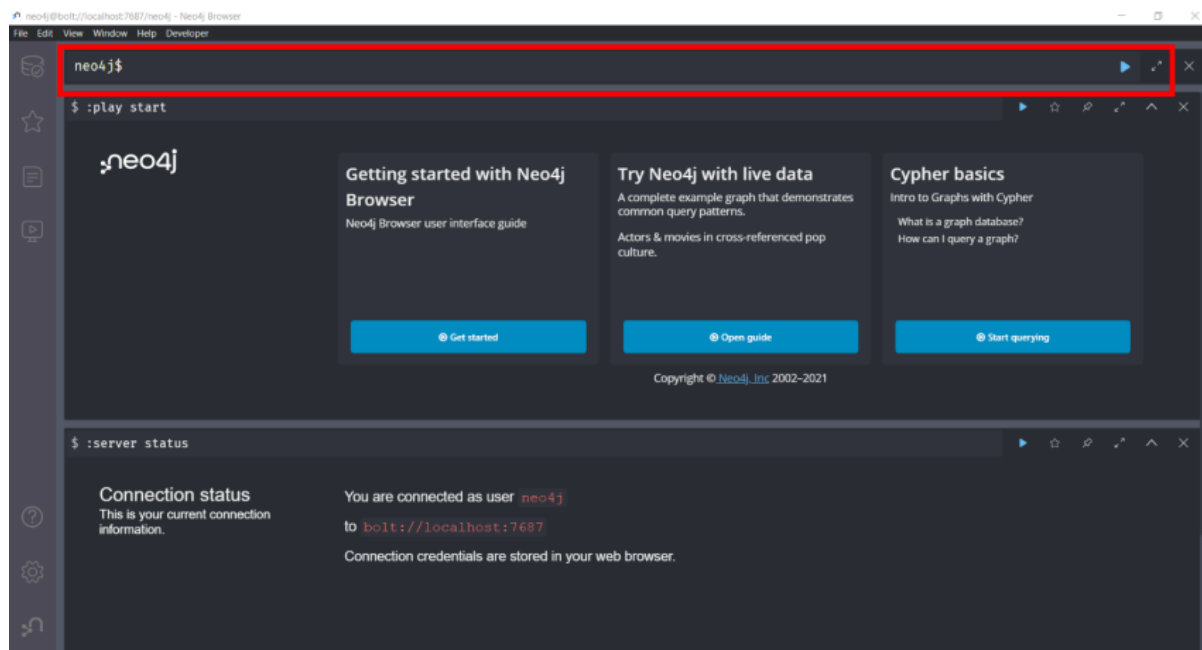
Paso 5.

Se inicia la base de datos y se abre.



Paso 6.

En la terminal, colocar el siguiente código.



```
MERGE (p:Cancion{cancion:row.Cancion, ranking:row.Ranking})
MERGE (c:Artista{artista:row.Artista})
MERGE (c2:Artista{artista:row.Artista2})
MERGE (c3:Artista{artista:row.Artista3})
MERGE (e:Genero{genero:row.Genero})
MERGE (tp:EstadoDeAnimo{estadoDeAnimo:row.EstadoDeAnimo})
MERGE (t:Estrellas{estrellas:row.Estrellas})
CREATE (p)-[:EstadoDeAnimo]->(tp)
CREATE (p)-[:Artista]->(c)
CREATE (p)-[:Artista]->(c2)
CREATE (p)-[:Artista]->(c3)
CREATE (p)-[:Estrellas]->(t)
CREATE (p)-[:Genero]->(e)
RETURN p
```

Paso 7.

Correr el archivo Main, sin apagar la base de datos.

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  main.py - PROYECTO FINAL - Visual Studio Code

main.py  Bienvenido  connection.py  main.py 1 X

main.py > ConsultaUsuario
1  from neo4j import GraphDatabase
2  from connection import *
3  import random
4
5
6  contra="Computologo"
7
8  """
9  agregar:
10 una opcion para agregar mas componentes al grafo. En esta opcion
11 el usuario ingresa la informacion de la cancion que esta agregando
12 """
13
14 def agregar(connection, db):
15     #se le solicita al usuario que ingrese una contraseña
16     palabra=input("Por favor ingrese la palabra clave \n")
17     if palabra == contra:
18         # Se le solicita al usuario que ingrese los datos
19         cancion=input("Ingrese el nombre de la cancion a agregar a la base de datos \n >")
20         datos = {"Artista":"","Genero":"","EstadoDeAnimo":"","Estrellas":""}
21         # El ciclo FOR recorre el diccionario datos y guarda la informacion que el usuario ingrese.
22         for keys in datos:
23             print("Seleccione la opcion con la que se siente más identificado según "+keys )
24             dictionary = elementos(connection, db, keys)
25             elec = eleccion(dictionary)
26             datos[keys] = elec
27         # Se le asigna un ranking de 0 porque al ser un dato nuevo, no se le ha recomendado a nadie
28         datos["Ranking"]=0
29         # Primer query para crear la cancion
30         query = '''
31             CREATE (p:Cancion{cancion:"%s", ranking:'0'})
32             '''%(cancion)
33         connection.query(query, db)
34         # Se envia otro query uniendo a la cancion con sus características. Se envia en estructura del código de cypher para que la plataforma entienda el query.
35         query="""MATCH (p:Cancion) WHERE p.cancion = '%s'
36                 MATCH (p1:Artista) WHERE p1.artista = '%s'
37                 MERGE (p) -[:Artista]-> (p1) WITH p
38                 MATCH (p2:Genero) WHERE p2.genero = '%s'
39                 MERGE (p) -[:Genero]-> (p2) WITH p
40                 """
```