



Algoritmos y Estructuras de Datos

Sección 20

Proyecto 2 Fase 2

Derek Arreaga- 22537

César López- 22535

Isabella Miralles- 22293

Mónica Salvatierra- 22249

Giovanni Santos- 22523

Link del repositorio:

<https://github.com/alee2602/Proyecto2ED>

Pruebas con Usuarios

A continuación se muestra la retroalimentación de los usuarios a los que se les entrevistó durante la primera fase del proyecto para que expusieran su punto de vista con respecto a las recomendaciones realizadas.

Nombre:	Esteban Meza
Opinión:	Me gustó mucho el enfoque que le dieron al sistema para realizar las recomendaciones. De cierta manera, acierta a mis gustos, además de mostrar algunas canciones que no pensaría que el sistema me sugiriera. Pero en general, me parece que está muy bien hecho.

Nombre:	Sebastián Huertas
Opinión:	Me pareció interesante que el sistema tuviera un amplio repertorio de géneros que escoger, porque suelo escuchar música muy variada. Además, la opción para agregar una canción podría ser útil para cualquiera que quisiera recomendar una nueva canción en base a los gustos de los demás. Sin embargo, me hubiese encantado que en las recomendaciones salieran el nombre de los artistas que cantan las canciones. No solo porque hay títulos que suelen repetirse con frecuencia, sino también hay canciones que no había escuchado del todo, u otras que no pude reconocer en el momento.

Nombre:	Bianca Calderón
Opinión:	Me gustó mucho el hecho de que implementaran una variedad de géneros y que al mismo tiempo, nos dejara colocar varios. Siento que ese aspecto ayudó con la precisión de la recomendación. También

	me gustó que mostrara canciones de mi gusto y algunas que no conocía del todo.
--	--

Nombre:	José Marchena
Opinión:	Considero interesante el hecho de que había muchos géneros para escoger, a pesar de que solo escucho a unos específicos. Me gustó que la recomendación se enfocara en cómo me quería sentir, porque soy consciente de lo que escucho con respecto a mi estado de ánimo. Las recomendaciones fueron precisas y las canciones que obtuve son cercanas o iguales a las que suelo escuchar día a día.

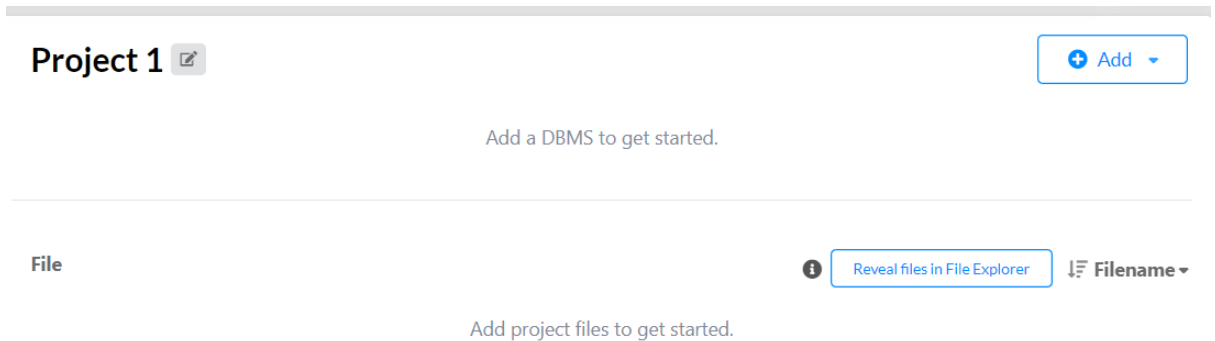
Nombre:	Adriana Santos
Opinión:	La verdad, me gustó la variedad de opciones que obtuve, precisamente porque mi género preferido es el pop. Quería probar varias opciones y ver las diferentes recomendaciones utilizando los diferentes sentimientos. Al final obtuve muchas canciones recomendadas; unas clásicas que suelo escuchar con cierta frecuencia y otras que nunca había escuchado. Me gustó la practicidad del programa y lo completa que es.

Documentación para utilizar el sistema de recomendaciones

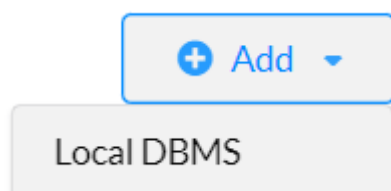
Requerimientos previos:

- Contar con Python IDLE instalado
- En la consola de cmd, instalar neo4j utilizando “`pip install neo4j`”
- Instalar Neo4j Desktop
- Descargar la carpeta del repositorio y abrir la de Fase 2

1. Crear un nuevo proyecto en Neo4j




2. Agregar una Base Local




3. Agregar contraseña para utilizarla luego en el archivo .py

Name



 Graph DBMS

Password

|


Version

5.3.0


 Cancel  Create

4. Una vez creada, ir al directorio de los archivos del proyecto

Project 1

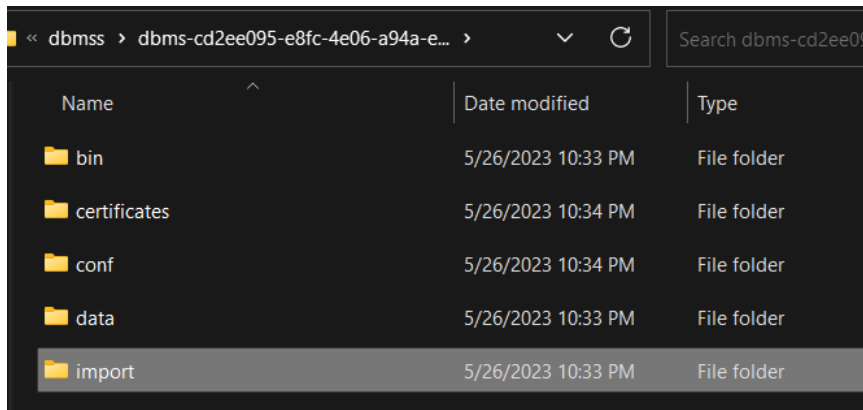
 Graph DBMS 5.3.0

File

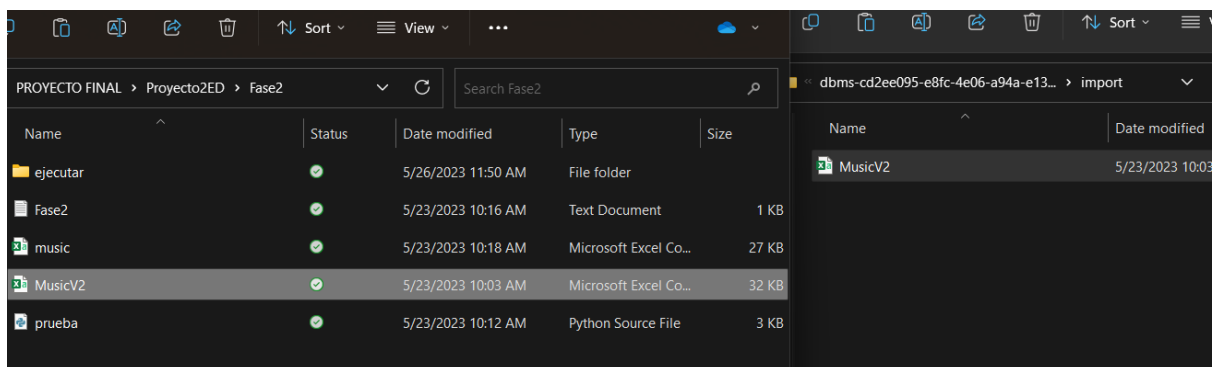
 [Reveal files in File Explorer](#)

Add project files to get started.

- El directorio es: C:\Users\SU USUARIO\.Neo4j\Desktop\relate-data\dbmss
- En el caso de tener varios proyectos creados, seleccionar el creado más recientemente, y luego seleccionar la carpeta “import”



c. Copiar el archivo “MusicV2” a la carpeta import



5. Iniciar la base de datos e ingresar a la consola

a. Copiar el siguiente código en CYPHER para crear la base de datos:

LOAD CSV WITH HEADERS FROM 'file:///musicVFinal.csv' AS row

MERGE (s:SONG {Title: row.title})

WITH s, row

UNWIND split(row.gen, ':') AS Genre

MERGE (g:GENRE {type: Genre})

MERGE (a:ARTIST {name: row.artist})

MERGE (f:FEELING {Feeling: row.feeling})

MERGE (s)-[p:ES_DE_ARTISTA]->(a)

MERGE (a)-[q:CANTA]->(s)

MERGE (s)-[r:ES_DE_GENERO]->(g)

MERGE (f)-[z:LO_TRANSMITE]->(s)

MERGE (s)-[x:TRANSMITE]->(f)

```
1 LOAD CSV WITH HEADERS FROM 'file:///musicV2.csv' AS row
2 MERGE (s:SONG {Title: row.title})
3
4
5 WITH s, row
6 UNWIND split(row.gen, ':') AS Genre
7 MERGE (g:GENRE {type: Genre})
8
9 MERGE (a:ARTIST {name: row.artist})
10 MERGE (f:FEELING {Feeling: row.feeling})
11
12 MERGE (s)-[p:ES_DE_ARTISTA]->(a)
```

Added 904 labels, created 904 nodes, set 904 properties, created 3086 relationships, completed after 643 ms.

6. Una vez creada la base de datos, vamos al código

- a. En el archivo programa.py, la contraseña está inicializada en dos métodos, el método modificar y el método buscar.
- b. Hay dos opciones para ingresar la contraseña
 - i. La primera es modificar el ejemplo de contraseña en ambos métodos, la cuál es **12345678**, modificar por la contraseña creada anteriormente

```

def modificar(query):

    #password = input("Ingrese la contraseña de la Base de Datos")
    password = "12345678"

    db = Neo4jDatabase("bolt://localhost:7687", "neo4j", password)

    #CONSULTA
    db.run_query(query)

    db.close()

def buscar(query,nodeType):

    devuelta = []

    #password = input("Ingrese la contraseña de la Base de Datos")
    password = "12345678"

```

ii. La segunda opción es descomentar las expresiones password =

input("Ingrese la contraseña de la Base de Datos")

Lo malo de este método es que pide la contraseña cada vez que se hace una búsqueda o modificación en el programa

```

261 password = input("Ingrese la contraseña de la Base de Datos")
262 #password = "12345678"
263
264 db = Neo4jDatabase("bolt://localhost:7687", "neo4j", password)
265
266 #CONSULTA
267 db.run_query(query)
268
269
270 db.close()
271
272 def buscar(query,nodeType):
273
274     devuelta = []
275
276     password = input("Ingrese la contraseña de la Base de Datos")
277     #password = "12345678"

```

Ahora ya puede probar ejecutando el archivo **Programa.py** el programa de recomendación (entre más géneros musicales elija, será mejor la recomendación)