

Universidad Del Valle De Guatemala

Departamento de Computación

Estructuras y algoritmos de datos



## **Proyecto 2: Sistema de Recomendaciones Entrega 1**

### **Recomendación de películas**

Grupo 8

Javier Valle 20159

Javier Mombiola 20067

María Fernanda Argueta 20458

## Documentación sobre cómo se emplea el programa

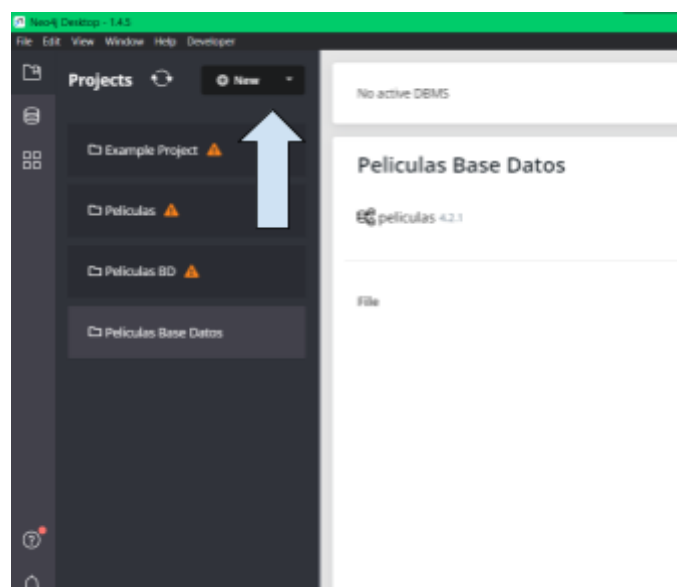
Para poder correr el programa de recomendación de películas se necesita:

1. Instalar Neo4J en la computadora en donde se correrá el programa.
2. Se necesita tener python instalado en la computadora.
3. Se necesitan tener los importados los módulos necesarios:
  - a. random
  - b. numpy
  - c. pandas
  - d. py2neo
  - e. matplotlib
  - f. neo4j
4. Se necesita un compilador, como Visual Studio por ejemplo.
5. Se necesitan las bases de datos .csv (proporcionadas por nosotros).
6. Se necesitan los programas .py (proporcionados por nosotros).

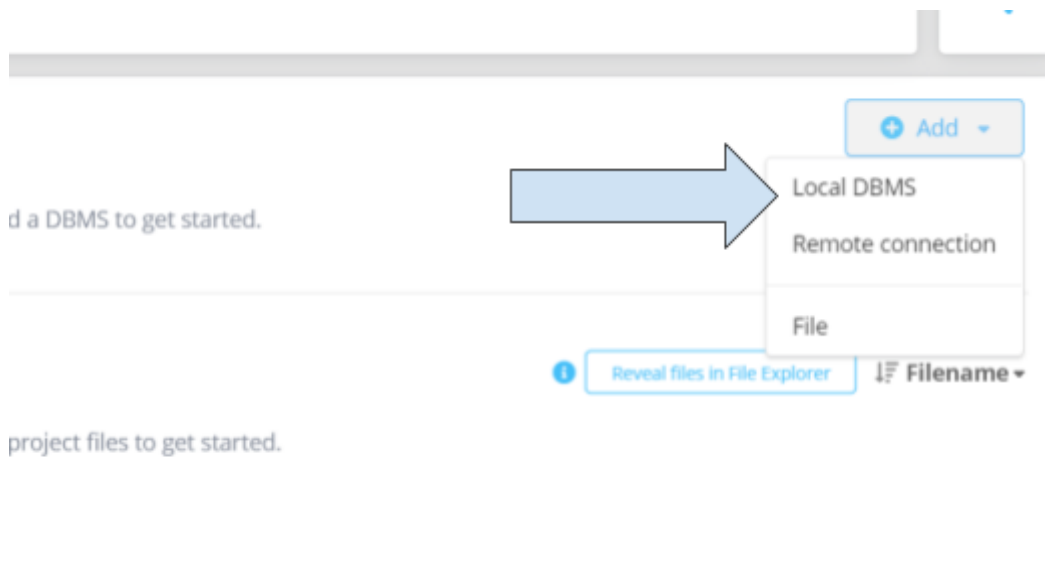
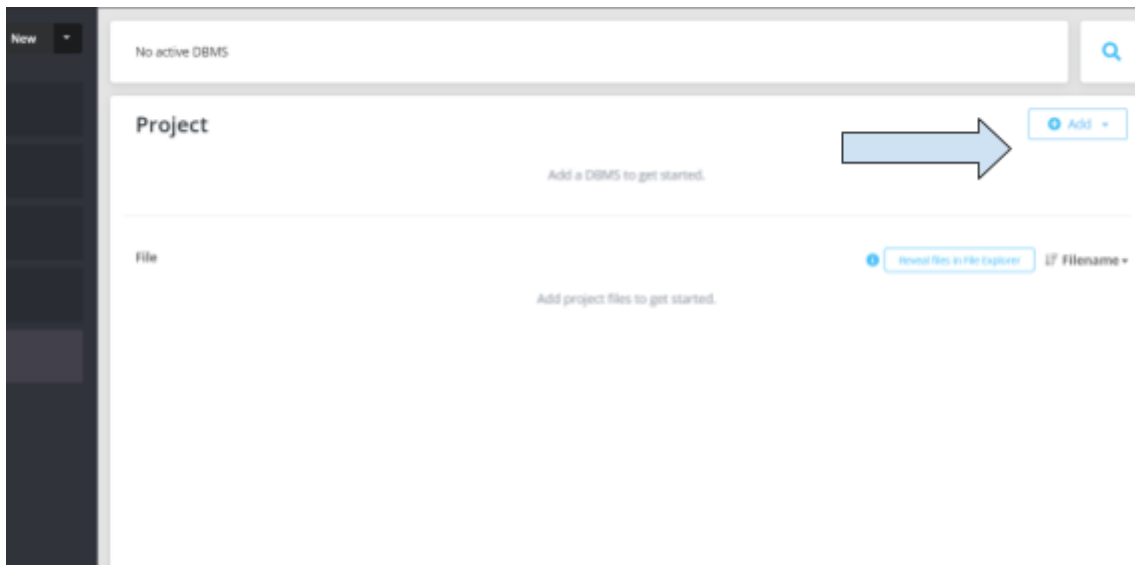
Los requisitos mínimos que necesita la computadora para poder correr Visual Studio y compilar el programa son los siguientes:

- Un procesador de 1.6 GHz o más rápido.
- 8 GB de memoria RAM.
- 60 GB de espacio disponible en el disco duro. El disco debe ser de 5400 RPM.
- Una tarjeta de video compatible con DirectX 9 (la cual tiene una resolución de 1024 x 768 o superior).
- Windows 7 SP1 en adelante.

Para poder correr el programa de forma óptima, primero se debe de crear una cuenta en Neo4J, ya instalado en su computadora, para luego poder crear un nuevo proyecto (no importa el nombre) agachado el botón “New”, ubicado arriba a la izquierda.



Después de tener el proyecto ya creado, debemos de agregarle una base de datos. Para hacer esto debemos de apachar el botón que dice “Add” a la derecha de la página. Esto desplegará un drop down menu y debemos de seleccionar la opción que dice “Local DBMS”.



Luego de agregar la base de datos, habrá dos espacios que podemos cambiar, el nombre y la contraseña. En el espacio del nombre se puede dejar como esta por default o se puede cambiar a lo que sea. Lo que en realidad importa es la contraseña. Para la contraseña debemos de poner la palabra “password”. De este modo aseguramos que los programas .py si lean esta base de datos. Cuando ya tengamos estos datos listos se apacha en la opción que dice “Create”.

### Project

**Name**  

 Graph DBMS 

**Password**  

 password 


**Version**  
4.2.1






 Cancel

 Create


Cuando ya hayamos creado la base de datos, lo que tenemos que hacer a continuación es apachar en “Start” para que dicha base de datos arranque.


 Start



 Open 




Cuando ya haya arrancada la base de datos, se verá algo así. A continuación debemos de darle click en “Open”.

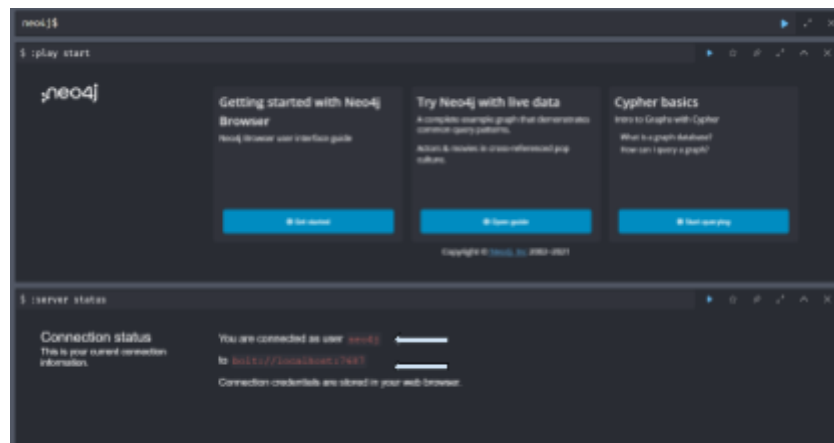
 Stop



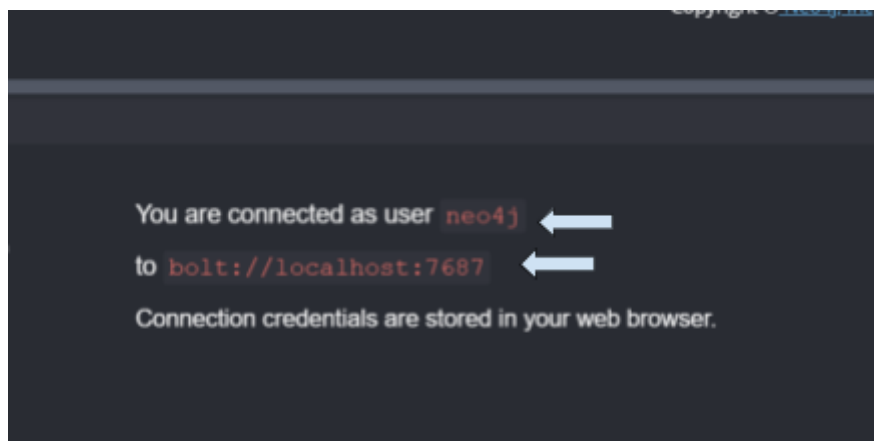
 Open 



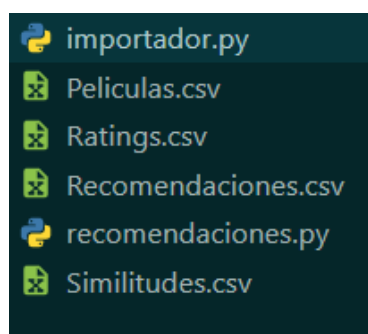
Esto nos desplegará una pantalla que se verá así.



Lo importante en esta pantalla es la parte inferior en donde aparecerá un texto igual o similar a este. El primer texto en rojo es el nombre de usuario de la base de datos, y el segundo texto es el uri de la base de datos. Ambos son importantes para que el programa sirva de manera óptima.



Después de tener nuestra base de datos lista. Debemos de descargar todos los archivos incluidos por nosotros (incluyen 4 archivos .csv y 2 archivos .py).



Lo primero que debemos de hacer con los archivos es abrir el archivo denominado como “importador.py” y asegurarnos que el uri, el nombre de usuario y la base de datos coincidan con los que están en el programa. Si no es así, hay que cambiar los datos en el programa, ya que los datos de Neo4J no se pueden cambiar. Si los datos están iguales entonces lo último que debemos de hacer es verificar que los archivos .csv tengan el mismo nombre que en el programa, de no ser así, se deben de cambiar para que estos coincidan.

```
#importando clases externas
import pandas as pd
import numpy as np
import matplotlib as plt
from py2neo import Graph
from neo4j import GraphDatabase

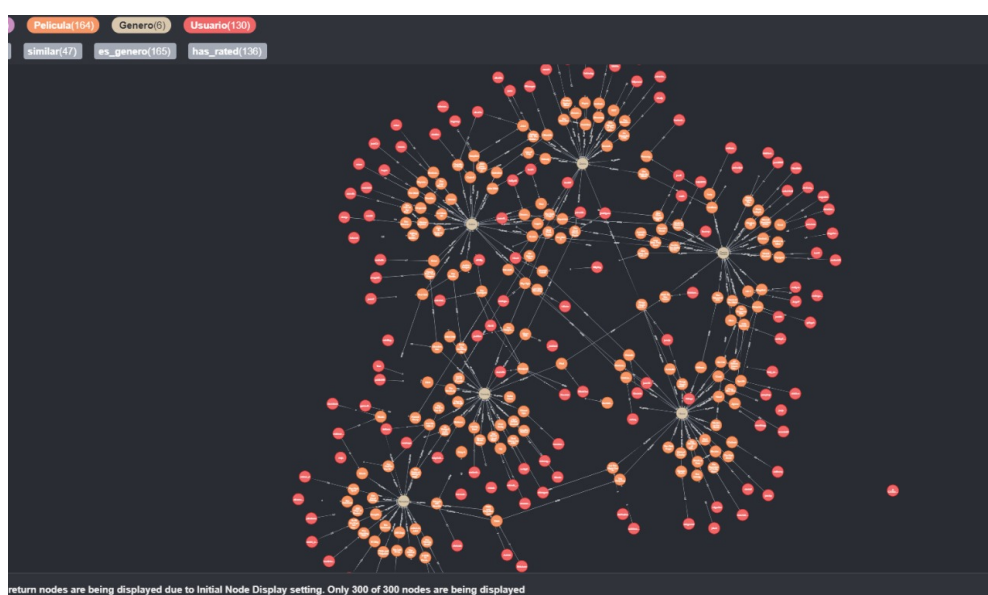
#establecer conexion con el uri, usuario y contraseña correctos
graphdp = GraphDatabase.driver(uri="bolt://localhost:7687", auth=("neo4j", "password"))

#iniciando la sesion de neo4j
session = graphdp.session()

#leer los archivos csv con pandas
catalogo = pd.read_csv("Peliculas.csv")
simis = pd.read_csv("Similitudes.csv")
rates = pd.read_csv("Ratings.csv")
```

A diagram with three blue arrows pointing downwards from the first three import statements to the database connection line, and one blue arrow pointing left from the CSV reading lines.

Cuando estemos seguros de que todos nuestros datos estén iguales, entonces solo queda correr el archivo “importador.py”, para que agregue todos los datos a la base de datos, lo cual se verá algo así.



Cuando ya tengamos nuestra base de datos lista, entonces ya podemos utilizar el archivo “recomendaciones.py”, para poder ver recomendaciones, dar rating a películas y modificar datos que ya estén dentro de la base de datos.

```
324         #preguntando por otro genero
325         print("\nAhora escoge tu segundo genero favorito")
326         for number, genero in enumerate(listagen2): #imp
327             print(number+1, genero)
328
329         op5 = 0
330         while op5 < 1:
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

Estas son las top 15 recomendaciones para usted:

	Título	Año	Duración	Rating	Genero
0	The Avengers	2012	224	8.0	Action
1	True Lies	1994	201	7.2	Action
2	Just Go With It	2011	157	6.4	Comedy
3	Things Heard and Seen	2021	201	5.3	Horror
4	47 Meters Down: Uncaged	2019	130	5.0	Horror
5	Karnan	2021	238	4.7	Action
6	How to Train Your Dragon	2010	138	4.1	Action
7	Up	2009	96	4.1	Comedy
8	Monsters Inc.	2001	132	4.1	Comedy
9	Sharknado	2013	126	3.3	Action
10	Godzilla vs Kong	2021	153	3.3	Action
11	Conan The Barbarian	2011	153	2.7	Action
12	Halloween	2018	146	2.6	Horror
13	Haunt	2019	132	2.5	Horror
14	Jigsaw	2017	132	2.3	Horror

Menu

-----

[1] Quiero Calificar |

[2] Quiero Recomendaciones |

[3] Quiero Editar Datos |

0 0 Live Share

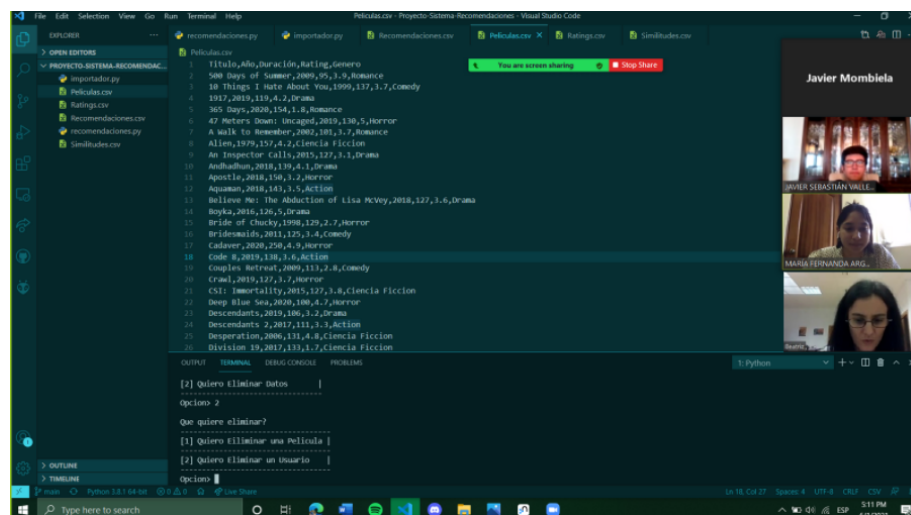
## Documentación de las pruebas con usuarios

Para poder tener una visión más clara sobre los comentarios recibidos por cada uno de los usuarios se realizó un cuadro con las preguntas realizadas a cada persona. Además se le solicitó a los usuarios indicar su nivel de satisfacción respecto al funcionamiento del programa, en una escala del 1 al 10. Siendo el 10 el mayor nivel de satisfacción y 0 el menor. La prueba de usuario se realizó virtualmente mediante zoom para evitar molestias de instalación.

### Prueba No.1

**Nombre:** Beatriz Valle

**Nivel de satisfacción:** 8



#### Qué le gustó

Se me hace un programa muy interesante y que utilizaría muy seguido.  
Es fácil de usar.

#### Que no le gustó

Siento que al momento de desplegar tantas recomendaciones me puede llegar a confundir más y poder escoger una película.

#### Recomendaciones

Se pueden poner estadísticas de los ratings en el programa.

#### Cambios

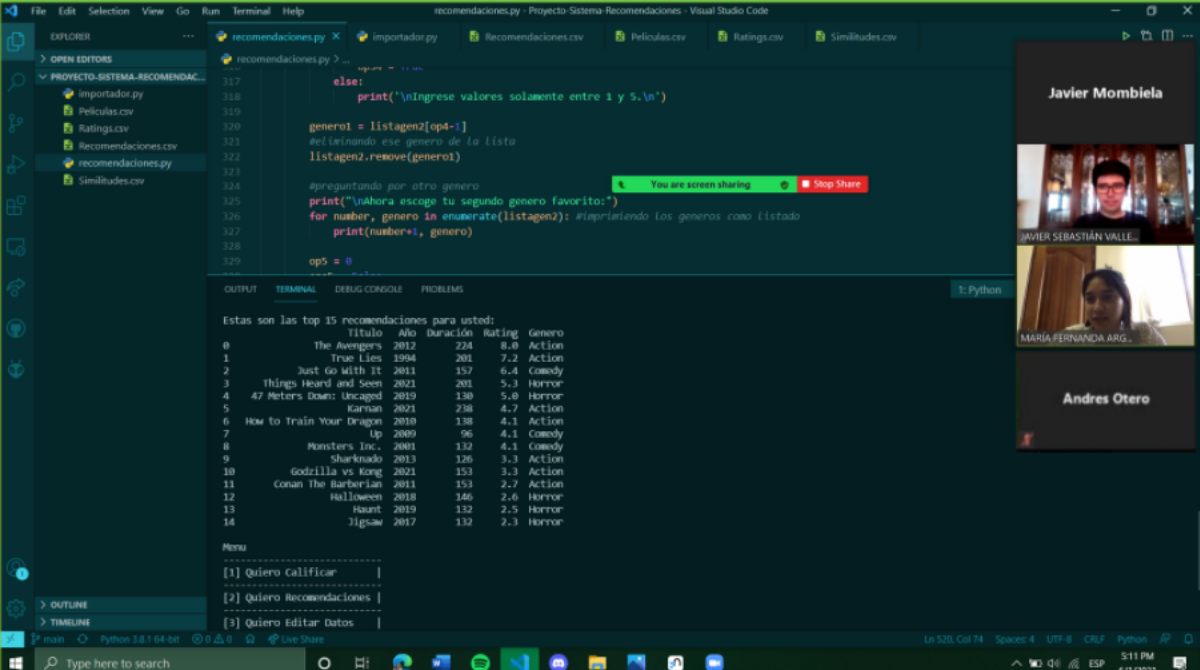
Me gustaría que me hiciera más preguntas para qué entonces las películas recomendadas fueran más específicas.



## Prueba No. 2

Nombre: Andres Otero

Nivel de satisfacción: 10



```
1177 else:
1178     print("\nIngrese valores solamente entre 1 y 5.\n")
1179
1180 genero1 = listagen2[op4-1]
1181 del(listagen2[op4-1])
1182 listagen2.remove(genero1)
1183
1184 #preguntando por otro genero
1185 print("\nAhora escoge tu segundo genero favorito:")
1186 for number, genero in enumerate(listagen2): #imprimiendo los generos como listado
1187     print(number+1, genero)
1188
1189 op5 = 0
```

Estas son las top 15 recomendaciones para usted:

	Titulo	Año	Duración	Rating	Genero
0	The Avengers	2012	224	8.0	Action
1	True Lies	1994	201	7.2	Action
2	Just Go with It	2011	157	6.4	Comedy
3	Things Heard and Seen	2021	201	5.3	Horror
4	47 Meters Down: Uncaged	2019	130	5.0	Horror
5	Karnan	2021	238	4.7	Action
6	How to Train Your Dragon	2010	138	4.1	Action
7	Up	2009	96	4.1	Comedy
8	Monsters Inc.	2001	132	4.1	Comedy
9	Sharknado	2013	126	3.3	Action
10	Godzilla vs Kong	2021	153	3.3	Action
11	Conan the Barbarian	2013	153	2.7	Action
12	Halloween	2018	146	2.6	Horror
13	Hunt	2019	132	2.5	Horror
14	Jigsaw	2017	132	2.3	Horror

Menu

- [1] Quiero Calificar
- [2] Quiero Recomendaciones
- [3] Quiero Editar Datos

### Qué le gustó

Me parece una idea muy innovadora y realmente funcional.  
Se me hizo una solución muy buena, ya que es rápida y accesible.

### Que no le gustó

No me gustó que hubieran tan pocos géneros de películas de los cuales escoger.

### Recomendaciones

Me hubiera gustado poder haber visto todo el repertorio de las películas que han sido recomendadas dentro del programa junto con su calificación.

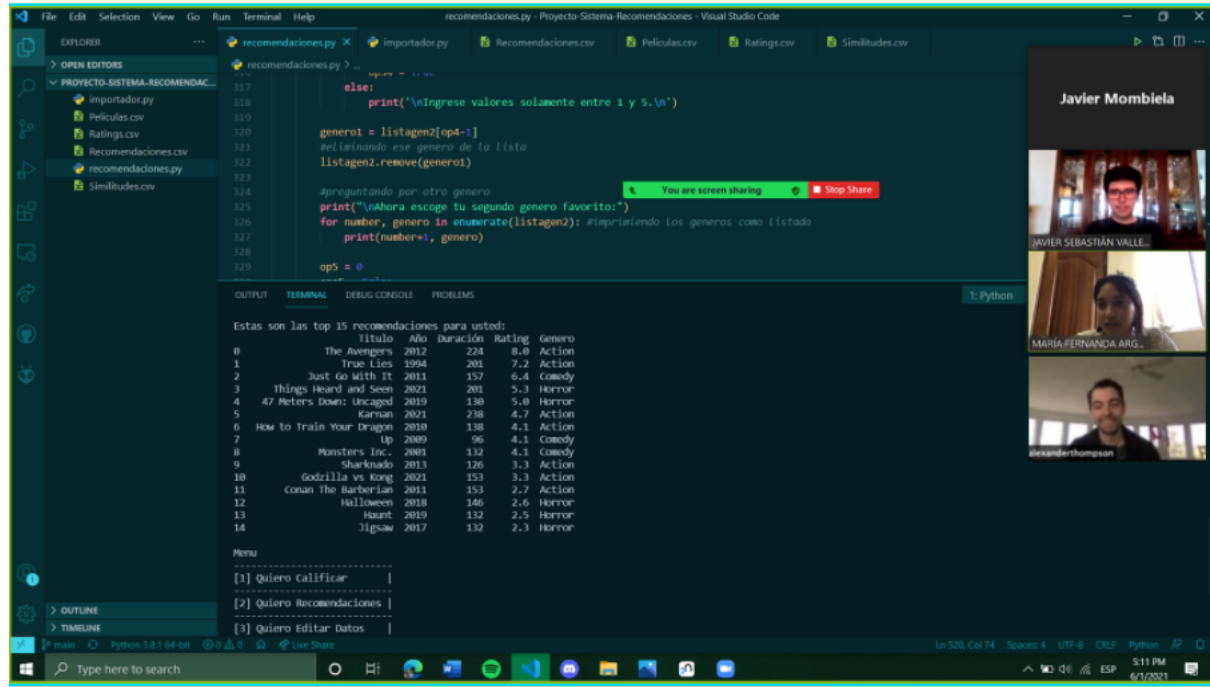
### Cambios

Me gustaría poder visualizar las películas ordenadas de otra manera, no por el rating.  
Me gustaría que estuvieran ordenadas por "la mejor opción para tí"

## Prueba No. 3

Nombre: Alexander Rivas

Nivel de satisfacción: 8



### Que le gustó

Le gustó la cantidad de recomendaciones que se le dieron y la variedad de títulos con sus géneros.

### Que no le gustó

La interfaz con el usuario.

### Recomendaciones

Agregar un sistema de inicio de sesión más elaborado, para que así se puedan guardar mis datos y recomendaciones anteriores.

### Cambios

Agregar una opción para poder darle al usuario la oportunidad de cambiar el nombre.

## **Conclusión**

Gracias a las recomendaciones recibidas por todos los usuarios se pudo concluir que en efecto se logró llegar a una buena solución ante la problemática que se da por la indecisión al momento de querer ver una película. Además todos los usuarios indicaron que consideran que esta es una problemática real y actual. Esta es una problemática que actualmente no cuenta con una solución que sea de fácil uso y funcional.

Por otro lado, debido a los comentarios recibidos en cuanto a la funcionalidad del programa se puede decir que se recibieron comentarios muy positivos. Sin embargo, se recibieron bastantes sugerencias en cuanto a posibles cambios para el funcionamiento. Entre estas se pueden mencionar dos que fueron muy buenas y muy funcionales. Primero, el incluir una opción en la que el usuario pueda tener acceso a todas las películas que se encuentran en la base de datos por si alguna le llama la atención. Y la segunda recomendación, sería el reducir el número de películas recomendadas para poder dar una solución directa a la búsqueda y que no genere más confusión.

Estos cambios no fueron realizados en el programa actual, sin embargo son buenas opciones que pueden llegar a ser tomadas en cuenta para futuros cambios en este. Lo más importante es que se pudo crear una solución rápida y amigable al usuario. Y encima de todo lograr que se solucionara la problemática de no saber qué película ver efectivamente.

Esta solución es la que mejor resuelve las necesidades destacadas en la fase de design thinking pues en esta fase se identificó que era necesario solicitar preguntas directas a los usuarios para poder realizar la recomendación. Además, se tomó muy en cuenta las sugerencias brindadas por los posibles usuarios durante la fase de idear. Estos aspectos, nos hicieron llegar entonces a la solución que resuelve de mejor manera la problemática detectada.

## **Base de datos**

La base de datos proporcionada por nosotros incluye cuatro archivos .csv “películas.csv”, el cual contiene la información sobre todas las películas que se utilizaron para poder crear esta base de datos. “Ratings.csv”, el cual contiene el rating que le dieron los usuarios a las películas que agregaron (de 1 a 5). “Similitudes.csv” el cual contiene películas similares, lo cual utilizamos para poder generar recomendaciones de géneros similares a los usuarios. “Recomendaciones.csv”, este es un archivo vacío que se llena a la hora de que el programa genera las recomendaciones y automáticamente se borra para que no se quede registro de las recomendaciones viejas y así aseguramos que cada recomendación sea única. Para poder guardar la privacidad de los usuarios, lo que hicimos fue pedirle que no ingresaran sus nombres, sino que ingresaran un nombre de usuario inventado por ellos. Tampoco pedimos ningún dato personal del usuario, lo único que le pedimos al usuario es que ingrese una película con toda su información y que le dé un rating.

## **Link repositorio de Github**

<https://github.com/javim7/Proyecto-Sistema-Recomendaciones.git>