



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



ANÁLISIS DE DATOS

ASIGNATURA:

Análisis de Datos

PROFESOR:

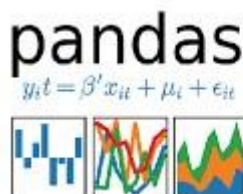
Ing. Lorena Chulde / Ing. Juan Pablo Zaldumbide

PERÍODO ACADÉMICO:

2023-B

EXAMEN BIMESTRAL

Marcelo Pinzón



	BandName	WaveLengthMin	WaveLengthMax
0	CustomAmplified	850	850
1	Blue	510	450
2	Green	560	530
3	Red	670	640
4	RedInfrared	880	850
5	ShortWaveInfrared_1	1100	1570
6	ShortWaveInfrared_2	2200	2100
7	Ceres	1300	1360

2023 - B

En los archivos proporcionados se encuentra datos de Netflix y la aceptación que cada película/serie tiene, 18337 registros se encuentran distribuidos en 4 archivos. Los campos del dataset son:

Title: Nombre de la película.

Available Globally?: Explica si se encuentra disponible a nivel mundial.

Release Date: Fecha de estreno en la plataforma

Hours Viewed: Horas vistas

Number of Ratings: Número de calificaciones "pulgar arriba"

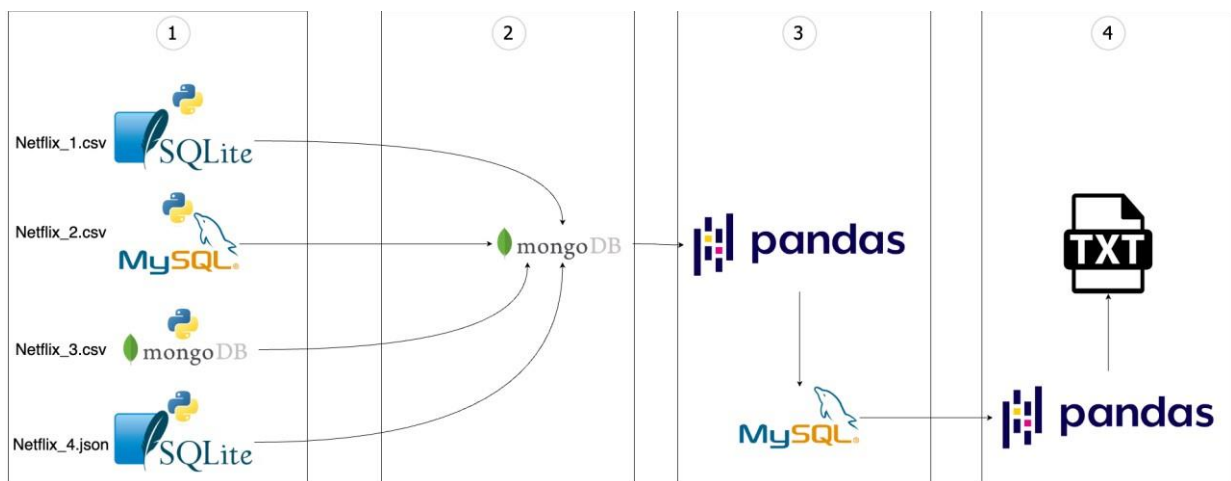
Rating: Clasificación general

Genre: Género

Key Words: Palabras clave Description:

Descripción

Dada la siguiente arquitectura:



Se solicita:

Etapas 1 - Importación de datos

Debe importar los archivos suministrados a cada una de las bases de datos mencionadas en el esquema. Puede usar **cualquier** método que usted crea conveniente. (25% del puntaje total)

Importar los datos a SQLite

```
[2]:
```

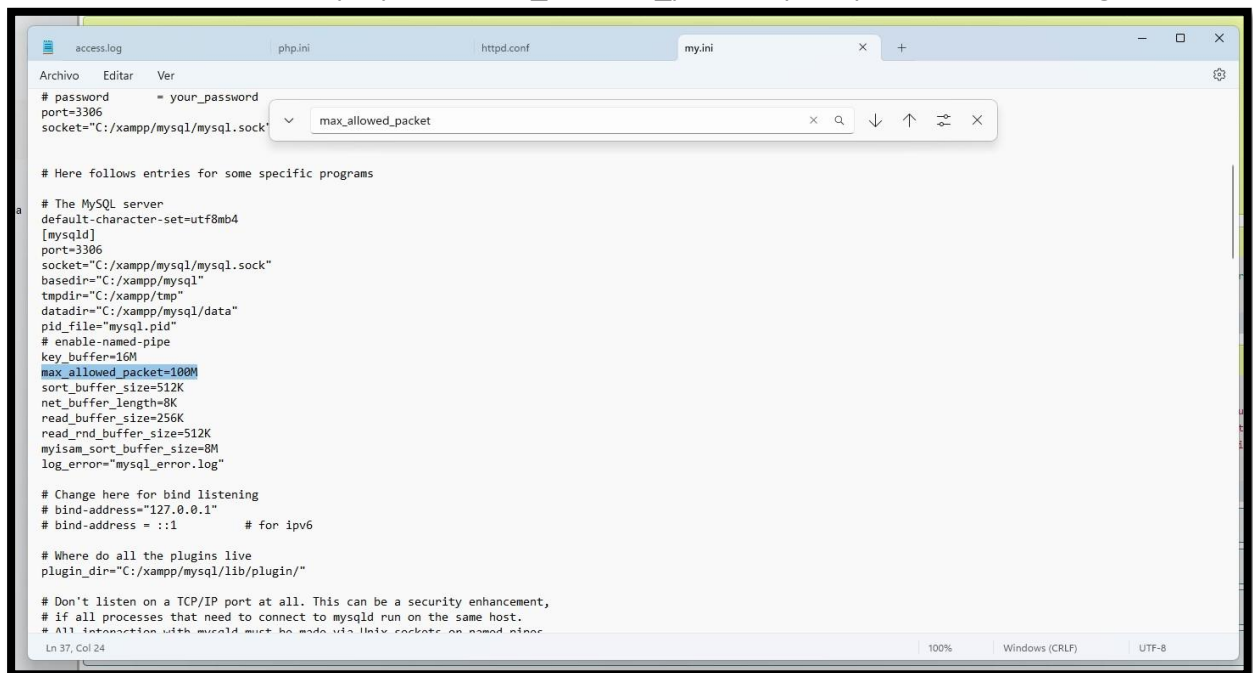
Title	Available Globally?	Release Date	Hours Viewed	Number of Ratings	Rating	Genre	Key Words	Description
The Worst Witch: Season 2	Yes	2018-07-27	10200000	97277.0	5.4	['Action', 'Adventure', 'Fantasy']	middle ages, 14th century, knight, monk, witch	14th-century knights transport a suspected wit...
All The Bright Places	Yes	2020-02-28	5500000	37176.0	6.6	['Drama', 'Romance']	based on novel, mental illness, young adult, inte...	The story of Violet and Theodore, who meet and...

```
[4]: #Crear la conexión y un cursor para ejecutar comandos SQL
conn=sqlite3.connect(ruta+"base1.db")
cursor=conn.cursor()
# En caso de generarse excepciones se maneja la excepción con el fin de que se guarde y se cierre la conexión sin importar que se genere o no la excepción
try:
    Diccionario=DataFrame1.to_sql(name="Datos_peliculas", con=conn)
except Exception as e:
    print(f"error {e}")
finally:
    conn.commit()
    conn.close()

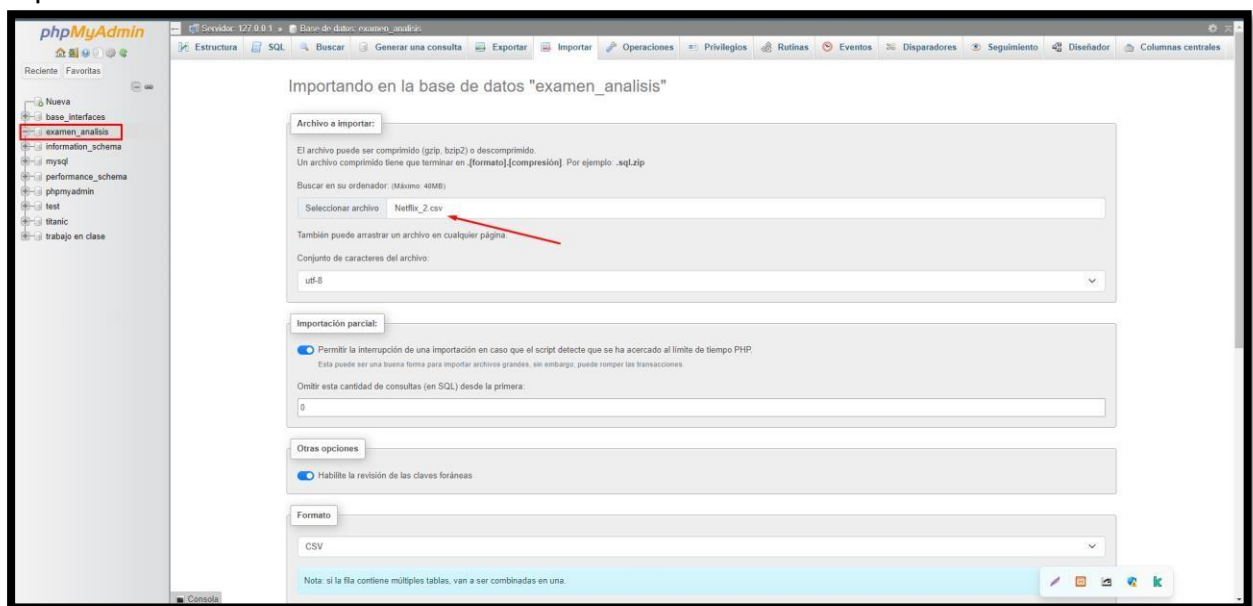
error Table 'Datos_peliculas' already exists.
```

Importación del archivo "Netflix_2.csv" a MySQL

Modificar el tamaño del paquete "max_allowed_packet" para permitir archivos grandes



Importar el Data Set



Opciones específicas al formato:

☐ Actualizar datos cuando las llaves importadas están duplicadas (agregar ON DUPLICATE KEY UPDATE)

Columnas separadas por:

Columnas encerradas entre:

Caracter de escape de columnas:

Líneas terminadas en:

Nombre de la tabla nueva (opcional):

Importe este gran número de filas (opcional):

☒ La primera línea del archivo contiene los nombres de columna de la tabla (si no está activado la primera línea será parte de los datos)

☐ No abortar si ocurre un error con INSERT

Importar

phpMyAdmin

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento Disparadores

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 0 - 24 (total de 4718. La consulta tardó 0.0002 segundos)

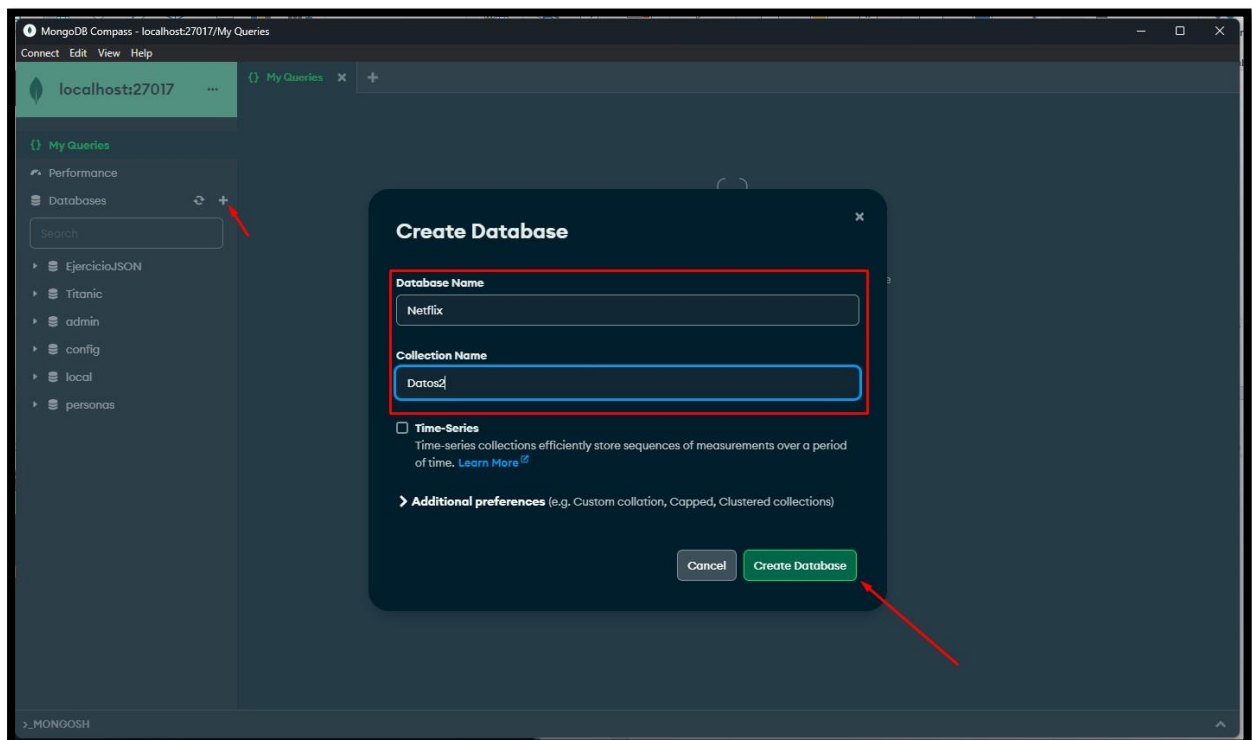
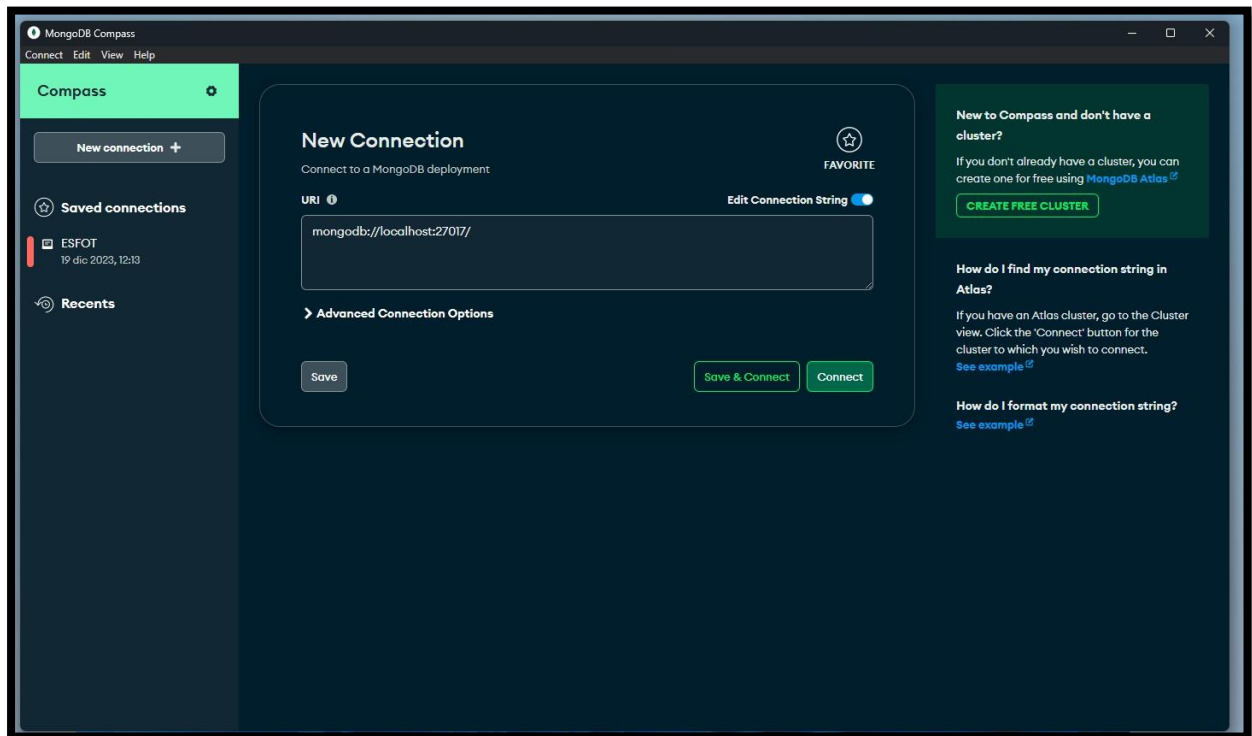
SELECT * FROM `netflix_2`

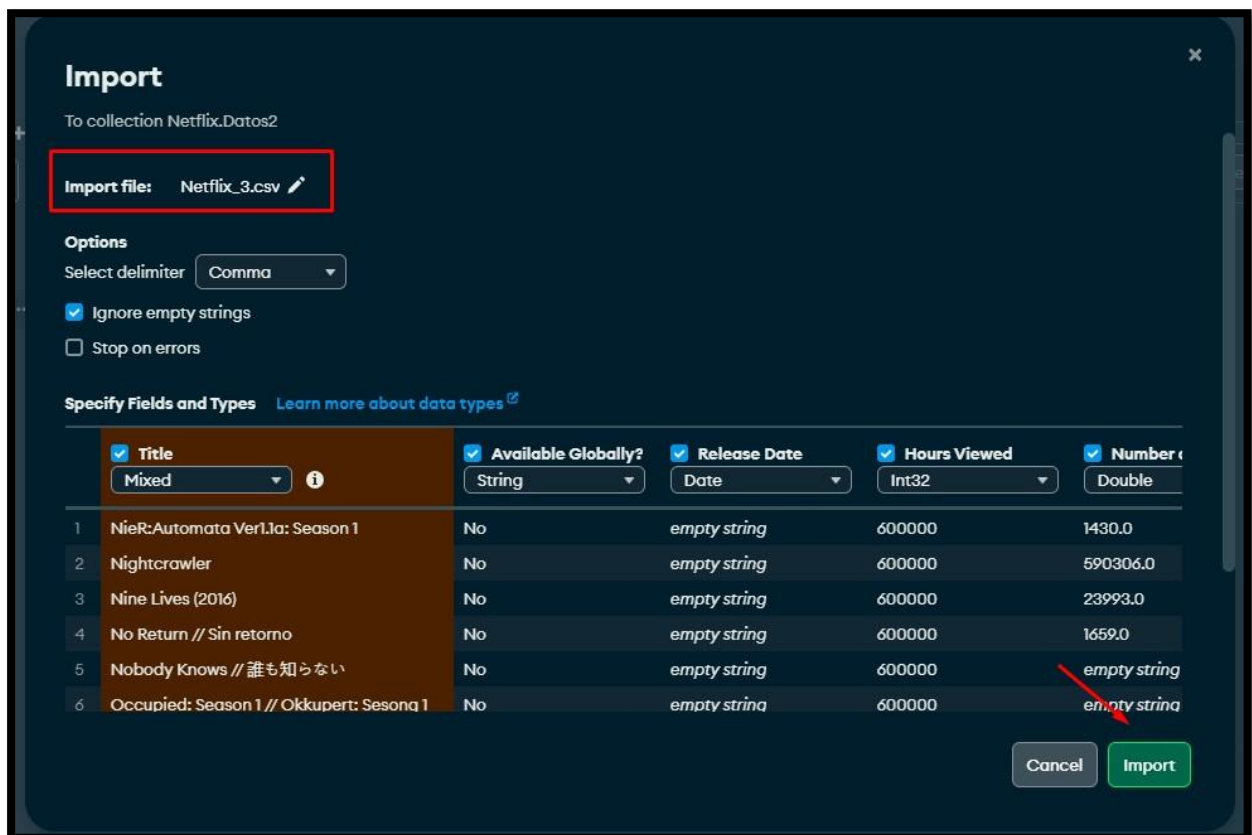
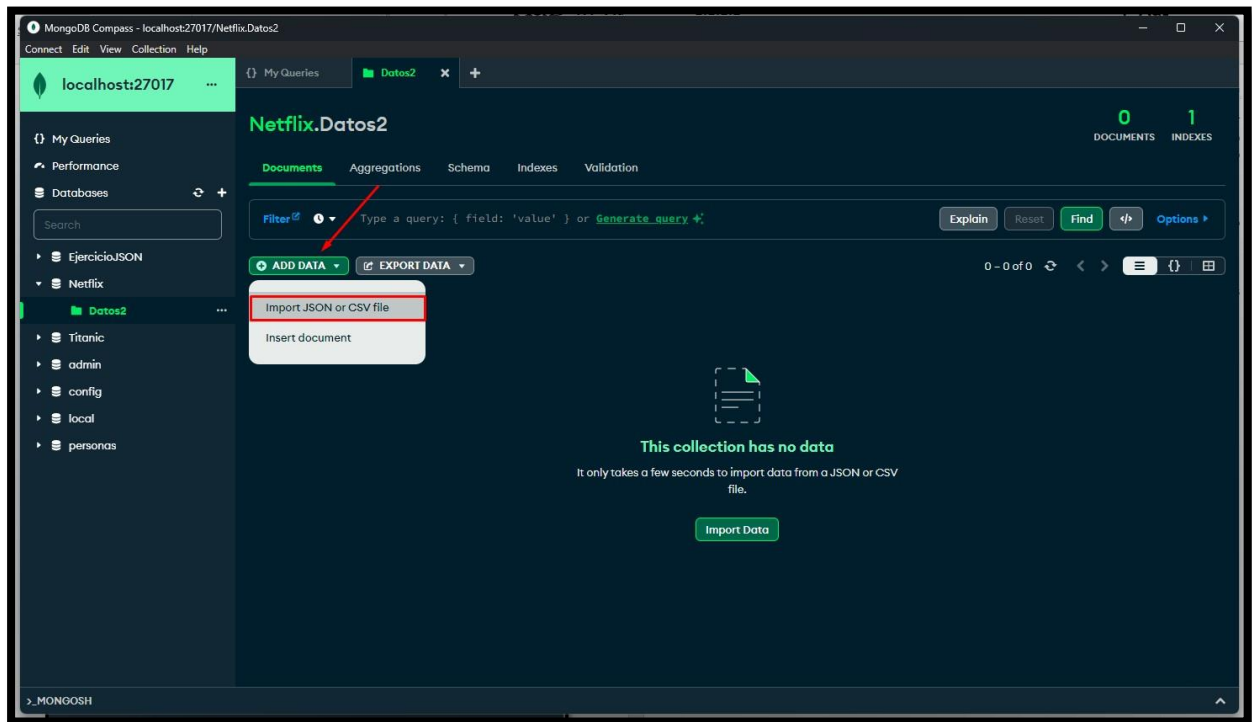
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

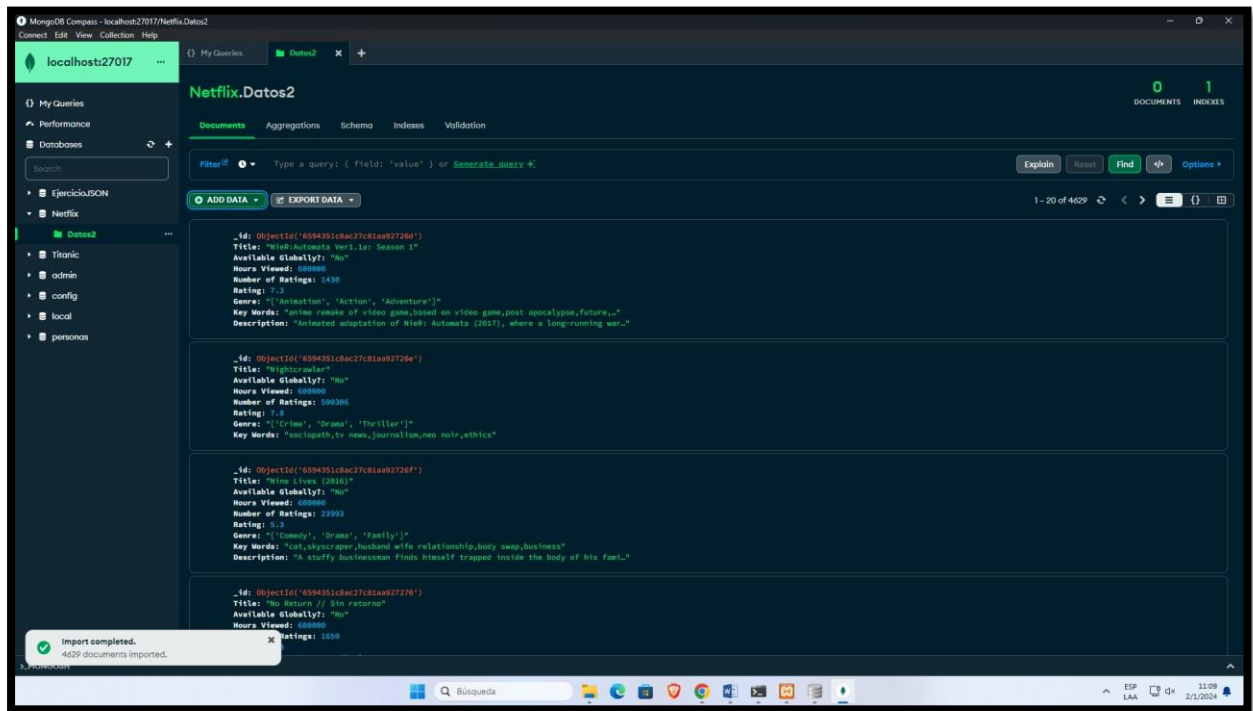
Número de filas: 25 Filtros: Buscar en esta tabla

Title	Available Globally?	Release Date	Hours Viewed	Number of Ratings	Rating	Genre	Key Words	Description
The Tale of Despereaux	No		2900000	39593.0	6.1	[Animation], [Adventure], [Comedy]	close up of eyes,close up of lips,close up of mout...	An unusually brave mouse helps to restore happiness.
The Vampire Diaries: Season 8	No		2900000	31.0	7.9	[Adventure], [Horror], [Mystery]	based on multiple sources.schoolteacher,teenager,1...	Snobbish teen snoops Elena Gilbert begins to suspect...
Tom and Jerry	No		2900000					
Trolls: The Beat Goes On! Season 7	Yes	2019-08-27	2900000	944.0	6.0	[Animation], [Short], [Adventure]	trill, singing, dreamworks, based on film	The party keeps on going for Poppy, Branch and the...
Welcome to Fishermen's Seafood Bar: Season 1 / 나만...	No		2900000					
Well-Intended Love: Season 2 / 후회 Boss 後悔我...	Yes	2020-03-31	2900000	87.0	5.7	[Short], [Comedy], [Musical]	copy boy, love, advice, autograph, waltz, ocean liner	Columnist Beatrice Blair dispenses love advice to...
Winx Club: Season 7	No		2900000	7.0	7.6	[Documentary], [Short]		
You Me Her: Season 1	No	2017-02-10	2900000	23.0	8.7	[Comedy], [Drama]	couple	Mags and Ash attempt to rekindle their marriage on...
3% Season 3	Yes	2019-06-07	2800000	12674.0	5.0	[Animation], [Adventure], [Comedy]	russian, circus, prince and pauper, look alike, role r...	BoogKapos's friends rally to bring him home from a...
A Cinderella Story: Christmas Wish	Yes		2800000	7584.0	5.3	[Comedy], [Family], [Fantasy]	christmas, holiday romance, teen fantasy, direct to v...	Despite her vain stepmother and mean stepsisters,...
An Astrological Guide for Broken Hearts: Season 2	Yes	2022-03-08	2800000	4058.0	7.1	[Comedy], [Romance]	love, life, astrology, horoscope, astrology enthusiast	Alice is heartbroken and hopelessly single. But af...
Ana Tramel: El juego: Limited Series	No		2800000	286.0	6.7	[Thriller]	female nudity, female full frontal nudity	Ana Tramel is a brilliant criminal lawyer in her l...
Ara You Human: Season 1 / 너도 인간이니: 시즌 1	No		2800000	26679.0	7.5	[Animation], [Action], [Drama]	mecha, pilot, flashback, fictional government agency...	The fate of the world is threatened by seemingly m...
Ashfall / 벅도산	No		2800000	5759.0	6.2	[Action], [Adventure], [Sci-Fi]	disaster, catclysm, destruction, race against time, j...	Stagnant since 1903, at an elevation of 2,744 m, a...
Console: netflix Public: Season 1	Yes	2019-06-14	2800000	876.0	6.8	[Drama], [Short]	back of clean, surface, film, soon fiction	74 hours. Team cost: 4 million dollars on the bus...

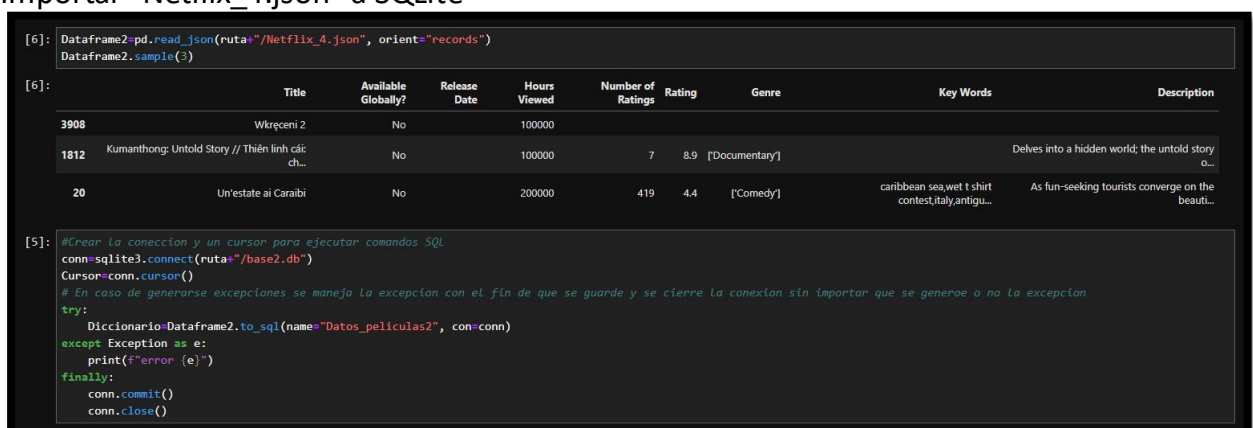
MongoDB Compass







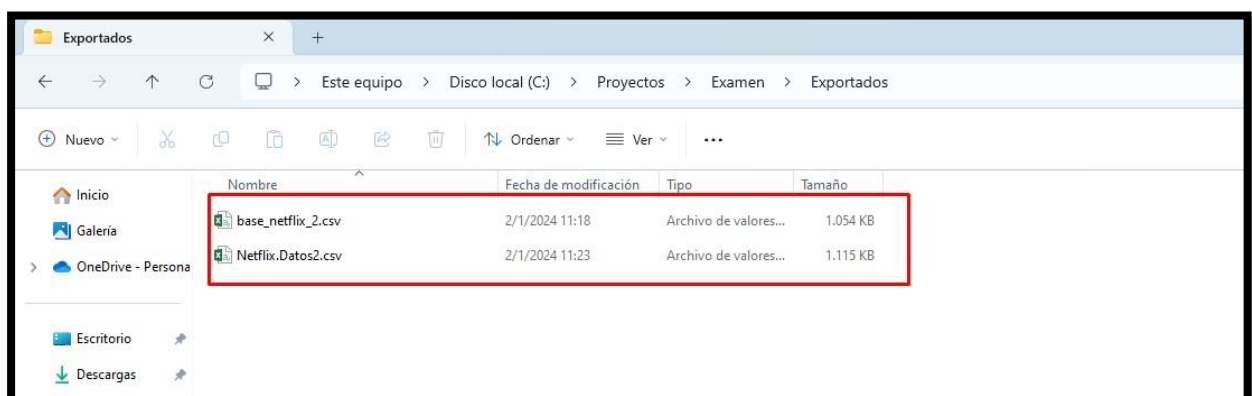
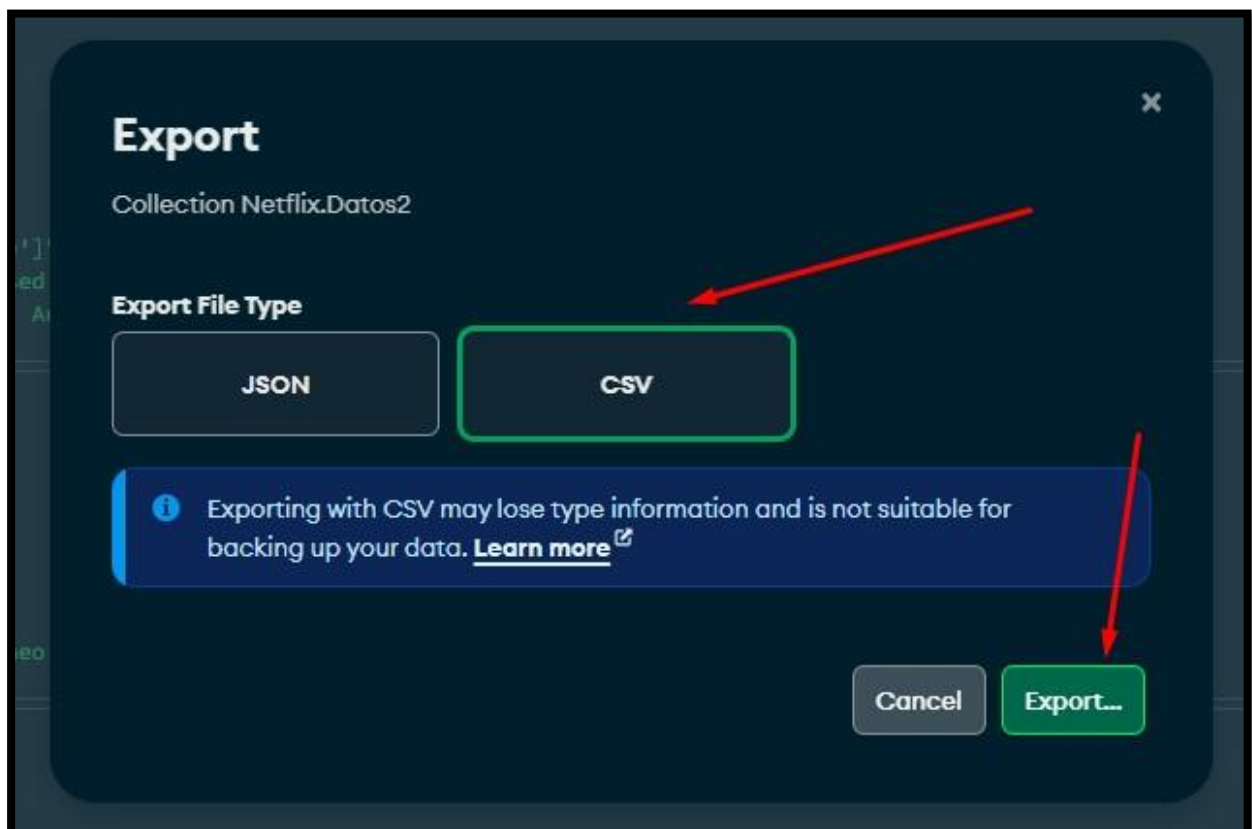
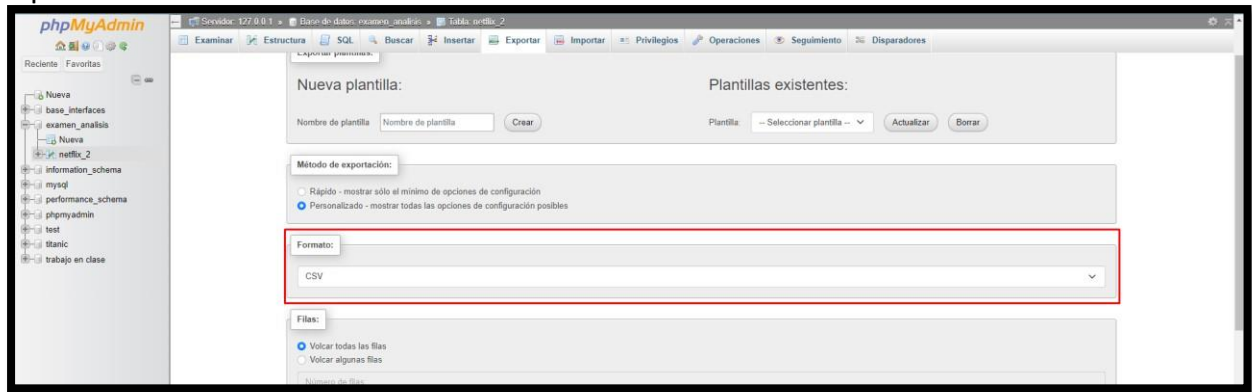
Importar "NetflixDatos2" a SQLite



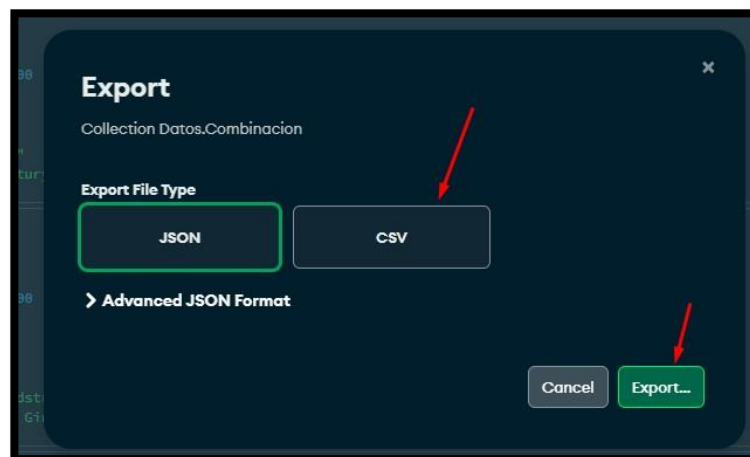
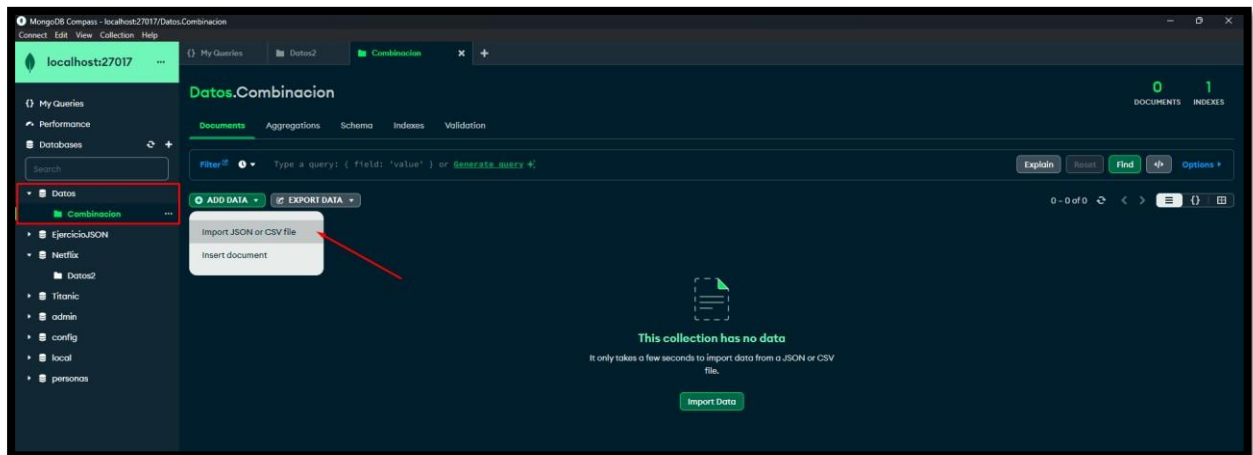
Etapas 2 - Consolidación de datos




Una vez que tiene las diferentes fuentes de datos debe exportar en cualquier formato y consolidar las bases de datos en mongoDB utilizando cualquier método (25% del puntaje total)

Exportar como CSV



Importar todos los Data Frames y exportar la combinación como archivo CSV



Nombre	Fecha de modificación	Tipo	Tamaño
 base_netflix_2.csv	2/1/2024 11:18	Archivo de valores...	1.054 KB
 Netflix.Datos2.csv	2/1/2024 11:23	Archivo de valores...	1.115 KB
 Datos.Combinacion.csv	2/1/2024 11:31	Archivo de valores...	4.369 KB

Etapas 3 - Limpieza de datos

En esta etapa debe importar los datos desde MongoDB y limpiarlos con pandas o cualquier herramienta que considere, deberá fijarse en campos vacíos y reemplazar por el promedio en el caso de que sea numérico, si es texto puede poner un texto estándar "info no disponible". Una vez limpia la data, se deberá exportar a MySQL (25% del puntaje total)

Etapas 3 - Limpieza de datos

```
[9]: Combination=pd.read_csv('archivos/Datos.Combinacion.csv', sep=',')
Combination
```

[9]:

	_id	Title	Available Globally?	Release Date	Hours Viewed	Number of Ratings	Rating	Genre	Key Words	Description
0	65943a2d8ac27c81aa928483	The Night Agent: Season 1	Yes	2023-03-23T00:00:00.000Z	812100000.0	7696.0	6.0	['Biography', 'Drama', 'History']	persian empire,empire,5th century b.c.,achae...	NaN
1	65943a2d8ac27c81aa928484	Ginny & Georgia: Season 2	Yes	2023-01-05T00:00:00.000Z	665100000.0	5216.0	5.7	['Comedy', 'Drama', 'Romance']	producer,three word title,headstrong,arranged ...	The film follows headstrong Ginny who meets Su...
2	65943a2d8ac27c81aa928485	The Glory: Season 1 // 더 글로리: 시즌 1	Yes	2022-12-30T00:00:00.000Z	622800000.0	11869.0	8.4	['Short']	NaN	NaN
3	65943a2d8ac27c81aa928486	Wednesday: Season 1	Yes	2022-11-23T00:00:00.000Z	507700000.0	NaN	NaN	['Talk-Show']	youtube video	MsMojo counts down the top 10 Wednesday (2022)...
4	65943a2d8ac27c81aa928487	Queen Charlotte: A Bridgerton Story	Yes	2023-05-04T00:00:00.000Z	503000000.0	50077.0	7.4	['Drama', 'History', 'Romance']	prequel,queen,historical,england,queen charlot...	Betrothed against her will to King George, you...
...
18328	65943a478ac27c81aa92cc1b	راس السنة	No	NaN	100000.0	383.0	4.8	['Drama']	live	A tale of different people whose lives interw...
18329	65943a478ac27c81aa92cc1c	心が騙びたがってるんだ。	No	NaN	100000.0	6209.0	7.3	['Animation', 'Drama', 'Family']	anime animation,anime	A young girl had her voice magically taken awa...
18330	65943a478ac27c81aa92cc1d	두근두근 내 인생	No	NaN	100000.0	NaN	NaN	NaN	NaN	NaN
18331	65943a478ac27c81aa92cc1e	라디오 스타	No	NaN	100000.0	NaN	NaN	NaN	NaN	NaN
18332	65943a478ac27c81aa92cc1f	선생 김봉두	No	NaN	100000.0	NaN	NaN	NaN	NaN	NaN

18333 rows x 10 columns

```
[10]: Combination.isnull().sum()
```

[10]:

	_id	Title	Available Globally?
	0	0	1


```
[10]: Combinacion.isnull().sum()

[10]: _id                0
      Title            0
      Available Globally?  1
      Release Date      13454
      Hours Viewed      2
      Number of Ratings  4112
      Rating            4112
      Genre             2573
      Key Words          5533
      Description        7714
      dtype: int64
```

```
..

•[16]: Columna="Number of Ratings"
      Disponibilidad=Combinacion[Columna]
      for clave, valor in Disponibilidad.items():
          if pd.isna(valor):
              Combinacion.at[clave, Columna]="Yes"
      Combinacion.isnull().sum()

[16]: _id                0
      Title            0
      Available Globally?  0
      Release Date      13454
      Hours Viewed      2
      Number of Ratings  4112
      Rating            4112
      Genre             2573
      Key Words          5533
      Description        7714
      dtype: int64

•[27]: Lanzamiento=Combinacion['Hours Viewed']
      Lanzamiento.()

[27]: 0    812100000.0
      1    665100000.0
      2    622800000.0
      3    507700000.0
      4    503000000.0
      Name: Hours Viewed, dtype: float64
```

```
[28]: Vistas=Combinacion["Hours Viewed"]
      promedio=Vistas.mean()
      for clave, valor in Vistas.items():
          if pd.isna(valor):
              Combinacion.at[clave, "Hours Viewed"]=promedio
      Combinacion.isnull().sum()

[28]: _id                0
      Title            0
      Available Globally?  0
      Release Date      0
      Hours Viewed      0
      Number of Ratings  0
      Rating            4112
      Genre             2573
      Key Words          5533
      Description        7714
      dtype: int64

[30]: Valoraciones=Combinacion["Number of Ratings"]
      for clave, valor in Valoraciones.items():
          if pd.isna(valor):
              Combinacion.at[clave, "Number of Ratings"]=float(random.randint(5, 100))
      Combinacion.isnull().sum()

[30]: _id                0
      Title            0
      Available Globally?  0
      Release Date      0
      Hours Viewed      0
      Number of Ratings  0
      Rating            4112
      Genre             2573
      Key Words          5533
      Description        7714
      dtype: int64
```

```
[40]: Genero=Combinacion["Genre"]
      Valores_no_nulos=Genero.dropna().to_list()

      for clave, valor in Genero.items():
          if pd.isna(valor):
              indice=random.randint(0, len(Valores_no_nulos))
              Combinacion.at[clave, "Genre"]=Valores_no_nulos[indice]
      Combinacion.isnull().sum()

[40]: _id                0
      Title            0
      Available Globally?  0
      Release Date      0
      Hours Viewed      0
      Number of Ratings  0
      Rating            4112
      Genre             0
      Key Words          5533
      Description        7714
      dtype: int64
```

```
[59]: Desc=Combinacion["Description"]
No_nulos=Desc.dropna().to_list()
for clave, valor in Desc.items():
    if pd.isna(valor):
        indice=random.randint(0, len(No_nulos))
        Combinacion.at[clave, "Description"]=No_nulos[indice]
Combinacion.isnull().sum()

[59]: _id          0
      Title     0
      Available Globally? 0
      Release Date 0
      Hours Viewed 0
      Number of Ratings 0
      Rating     0
      Genre      0
      Key Words  0
      Description 0
      dtype: int64
```

Etap4 - Presentación de resultados

Etap4 - Presentación de resultados

La respuesta a cada una de las siguientes preguntas se guardará en un archivo de texto diferente:

a.txt - ¿Cuál es porcentaje de películas/series que están disponibles a nivel mundial?

```
[69]: A=Combinacion[Combinacion["Available Globally?"]=="Yes"]
B=Combinacion["Available Globally?"]
Cf=len(A)
Ct=len(B)
Porcentaje=(Cf/Ct)*100
with open(archivos+"a.txt", "w", encoding="utf-8") as archivo1:
    archivo1.write(f"El porcentaje de películas/series que están disponibles a nivel es {round(Porcentaje, 4)}%\n")
```



b.txt - ¿Cuáles son las 10 películas/series más antiguas y las 10 más recientes?

```
Click here to ask Blackbox to help you code faster |
# Convertir la columna "Release Date" a objetos datetime en pandas (para el ordenamiento por fechas)
# Con "format='mixed'" se inferirá el formato de la fecha para cada uno de los valores de la columna en el DataFrame
# Asigna la zona horaria UTC a las fechas convertidas (se aplicó esto para prevenir un posible error)
Combinacion["Release date"]=pd.to_datetime(Combinacion["Release Date"], format="mixed", utc=True)
# Ordenar las fechas
Ordenamiento=Combinacion.sort_values(by='Release date')

[19] ✓ 0.3s Python

Click here to ask Blackbox to help you code faster |
with open(archivos + "b.txt", "wb") as archivo2:
    # Guardar las 10 películas más antiguas en el archivo
    archivo2.write("\t\t10 Películas antiguas\n".encode('utf-8'))
    archivo2.write(Ordenamiento["Title"].head(10).to_string().encode('utf-8') + b"\n")

    # Guardar las 10 películas más recientes en el archivo
    archivo2.write("\t\t10 Películas recientes\n".encode('utf-8'))
    archivo2.write(Ordenamiento["Title"].tail(10).to_string().encode('utf-8') + b"\n")

[20] ✓ 0.0s Python
```

```

Pegar Mover Copiar Eliminar Cambiar Nueva
b.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
10 Películas antiguas
2772 Arrested Development: Season 3
1216 Arrested Development: Season 1
1854 Arrested Development: Season 2
3221 Trailer Park Boys: Season 2
3408 Trailer Park Boys: Season 1
2849 Trailer Park Boys: Season 4
2848 Trailer Park Boys: Season 3
2406 Trailer Park Boys: Season 5
3741 Trailer Park Boys: Season 6
2644 Trailer Park Boys: Season 7
10 Películas recientes
17097 Rough Cut: Season 1 // Dal Y Mellt
6767 Lethal Weapon: Season 2
2488 Addicted (2014)
2479 Security (2017)
3824 Sicario: Day of the Soldado
3682 The Suicide Squad
1571 Captain Underpants: The First Epic Movie
15951 InuYasha the Movie 4: Fire on the Mystic Islan...
17905 The Witches (1990)
6626 Sherlock Holmes: A Game of Shadows

```

c.txt - Tomando en cuenta que el género es una lista de Python, ¿Cuál es el género que más se repite?

```

Click here to ask Blackbox to help you code faster
# Importamos la biblioteca "ast" para tratar cadenas de caracteres que contienen listas, tuplas, etc.,
# como estructuras validas en Python
import ast

# Extraer la columna "Genre" como lista (se extraera como lista de cadenas)
Generos=Combinacion["Genre"].to_list()
Elementos={} # Crear un diccionario para almacenar el conteo

# Crear una nueva lista con ast.literal_eval() para convertir las cadenas en listas, generara una lista de listas
Lista=[ast.literal_eval(valor) for valor in Generos]

# Acceder al valor de la lista y a su respectivo subvalor, realizar el conteo y guardarlo en el diccionario
for Sublista in Lista: # Accede a un elemento de la lista
    for valor in Sublista: # Accede al elemento de la sublista (en caso de ser lista de listas)
        if valor in Elementos: # Verifica si el elemento esta en el diccionario e incrementa el contador en uno
            Elementos[valor]+=1
        else: # Caso contrario lo añade y establece el contador en 1
            Elementos[valor]=1

# Buscar el elemento que mas se repite
Valor_maximo, elemento=0, str()
for i in Elementos.keys():
    if Elementos[i]>Valor_maximo:
        elemento=i
        Valor_maximo=Elementos[i]

# Guardar el resultado en un archivo
with open(archivos+"c.txt", "w") as archivo3:
    archivo3.write("El genero '{}' se repite {} veces en la columna \"Genre\"\\n".format(elemento, Valor_maximo))

```

```

c.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
El genero 'Drama' se repite 7085 veces en la columna "Genre"

```

d.txt - Según el rating ¿Las 10 mejores películas/series y las 10 peores películas/series?

```

Click here to ask Blackbox to help you code faster |
# Definición de la función para escribir en un archivo
Comment Code
def escribir_archivo(nombre, mensaje, titulos):
    with open(archivos + nombre, "a", encoding="utf-8") as archivo4:
        archivo4.write("\t\t\t\t\t"+mensaje+"\n")
        for i in range(len(titulos)):
            archivo4.write(titulos[i]+"\\n")

# Obtener la lista de valoraciones y ordenarla de forma descendente
Valoracion=Combinacion["Rating"].to_list()
Valoracion.sort(reverse=True)

# Obtener las 10 valoraciones mas altas y mas bajas de la columna
maximos=Valoracion[1:10]
minimos=Valoracion[-10:]

# Crear las listas donde se almacenaran los titulos de las películas
titulos=[]
menos_valoradas=[]
mas_valoradas=[]

# Obtener los títulos de las 10 películas menos valoradas
for valor in minimos:
    if len(titulos) < 10:
        indices = valor
        titulos.append(Combinacion["Title"][Combinacion["Rating"]==valor].to_list())

# Agregar los títulos a la lista de menos valorados y verificar que no se repitan los registros
for valor in titulos:
    for sub_valor in valor:
        if len(menos_valoradas) < 10 and sub_valor not in menos_valoradas:
            menos_valoradas.append(sub_valor)

# Escribir en el archivo la lista de las 10 películas menos valoradas
escribir_archivo("/d.txt", "Lista de 10 peores peliculas", menos_valoradas)

```

```

# Limpiar la lista de títulos
titulos.clear()

# Obtener títulos de las 10 películas más valoradas
for valor in maximos:
    if len(titulos) < 10:
        indices = valor
        titulos.append(Combinacion["Title"][Combinacion["Rating"] == valor].to_list())

# Agregar los títulos a la lista de mas valorados y verificar que no se repitan los registros
for valor in titulos:
    for sub_valor in valor:
        if len(mas_valoradas) < 10 and sub_valor not in mas_valoradas:
            mas_valoradas.append(sub_valor)

# Escribir en el archivo la lista de las 10 películas más valoradas
escribir_archivo("/d.txt", "Las 10 mejores peliculas", mas_valoradas)

```

d.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Lista de 10 peores peliculas

Squid Game: Season 1 // 오징어 게임: 시즌 1

Bake Squad: Season 2

Bunk'd: Season 2

Dark: Season 2

Odd Squad: Season 2

H (1998): Season 2

Sonic X: Season 2 // ソニックX: シーズン2

Still Game: Season 2

Baby: Season 2

Dogs: Season 2

Las 10 mejores peliculas

Big Time Rush: Season 2

Big Time Rush: Season 1

Big Time Rush: Season 4

Tom Papa: You're Doing Great!

Haroun

Love Is Blind: Brazil: Season 3 // Casamento às Cegas: Brasil: Temporada 3

The Country Cowboy // Il ragazzo di campagna

Zip & Zap and the Marble Gang // Zipi y Zape y el club de la canica

Safe House: Season 2

One Day at a Time: Season 1

e.txt - La keyword que más se repite.

```
Click here to ask Blackbox to help you code faster |
# Extraer la columna "Key Words" como una cadena sin incluir los indices
Palabras_clave = Combinacion["Key Words"].to_string(index=False)

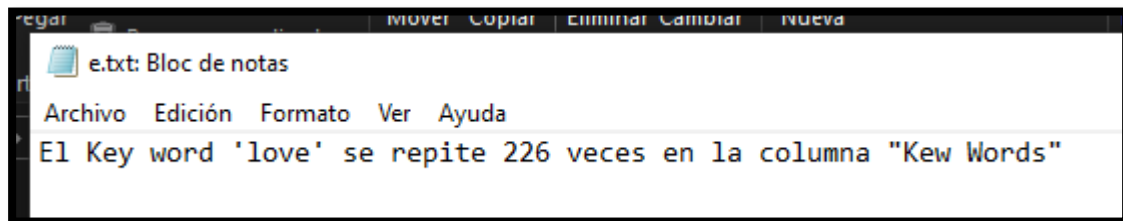
# Eliminar espacios en blanco y dividir la cadena en una lista de palabras clave
Lista_palabras = Palabras_clave.strip().split(",")

# Inicializar un diccionario para almacenar el conteo de palabras clave
Palabras = {}

# Contar las veces que se repite cada palabra clave en la lista
for P_Claves in Lista_palabras:
    if P_Claves in Palabras:
        Palabras[P_Claves] += 1
    else:
        Palabras[P_Claves] = 1

# Buscar el elemento que mas se repite
Valor_maximo, elemento = 0, str()
for i in Palabras.keys():
    if Palabras[i] > Valor_maximo:
        elemento = i
        Valor_maximo = Palabras[i]

# Guardar el resultado en un archivo
with open(archivos+"e.txt", "w") as archivo5:
    archivo5.write("El Key word '{}' se repite {} veces en la columna 'Kew Words'\n".format(elemento, Valor_maximo))
```



Cada consulta corresponde al 5% del puntaje total.

Entregables:

Subir en el aula virtual en un solo archivo comprimido:

- 1.- Código fuente <apellido.py> o <apellido.ipynb>
- 2.- pdf con capturas de pantalla de la ejecución de cada ejercicio.
- 3.- Archivos .txt generados