

Descripción de la Práctica 1.-

El/la profesional en formación deberá desarrollar un algoritmo basado en el proceso ETL, para Extraer-Transformar-Cargar un conjunto de datos (DataSet) relativo a Películas de TV, el cual posee diversos errores o datos incorrectos que provocarán errores en las etapas de transformación y análisis. Para ello, se utilizará un DataSet con datos abiertos, publicado en:

[https://raw.githubusercontent.com/sundeeblue/movie_rating_prediction/master/
movie_metadata.csv](https://raw.githubusercontent.com/sundeeblue/movie_rating_prediction/master/movie_metadata.csv)

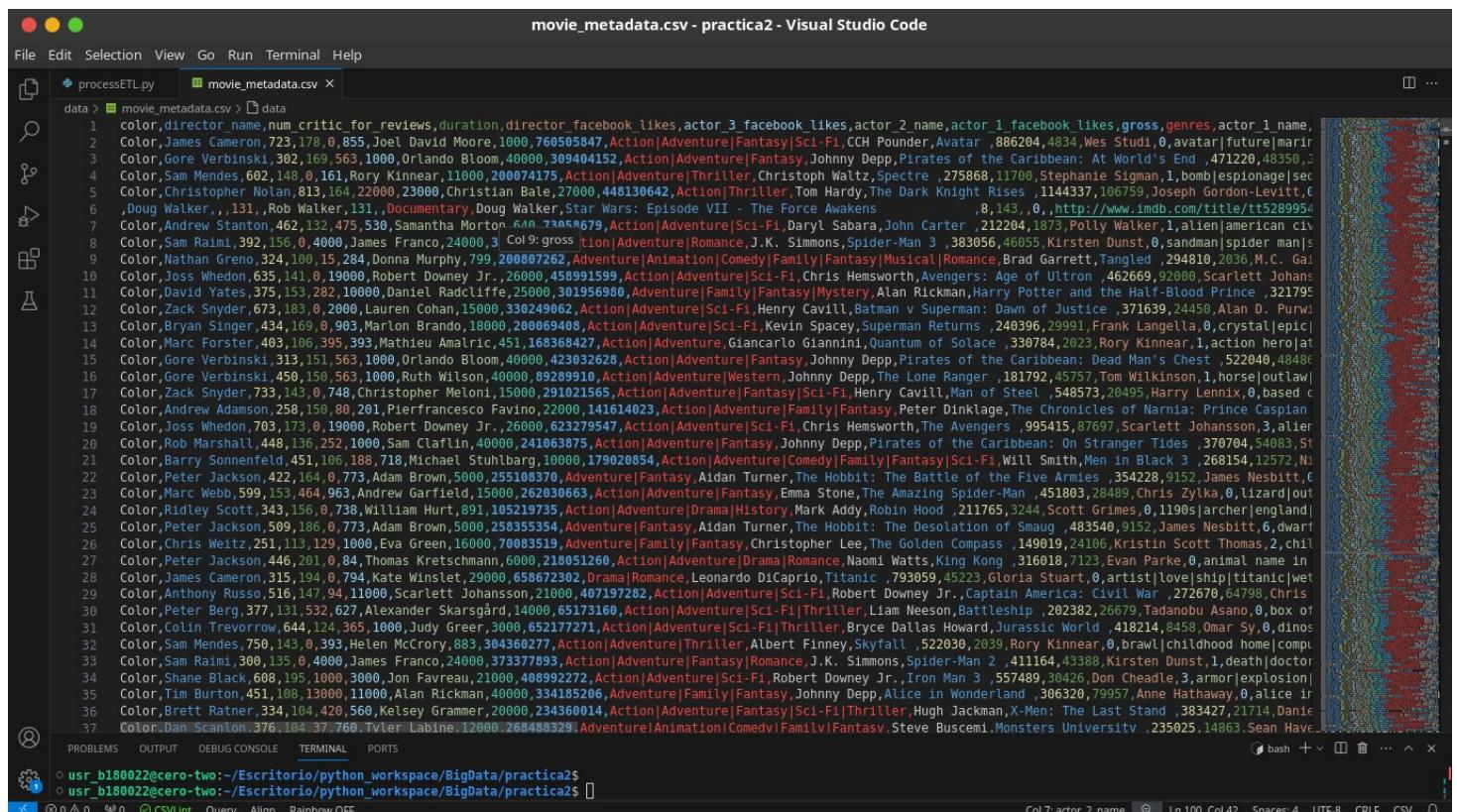
El archivo requerido para esta práctica se denomina **movies_dataset.csv**

Nota: Es necesario elaborar un documento en formato (*.pdf) que posee las impresiones de pantalla que validen la resolución de cada pregunta de la práctica 1, visualizando código fuente en python y resultado de la ejecución.

El proceso de ETL debe considerar los siguientes criterios:

En esta actividad se crea un archivo datos_limpios.csv se acualizara el sus datos conforme vayamos realizando los puntos requeridos en esta practica2.

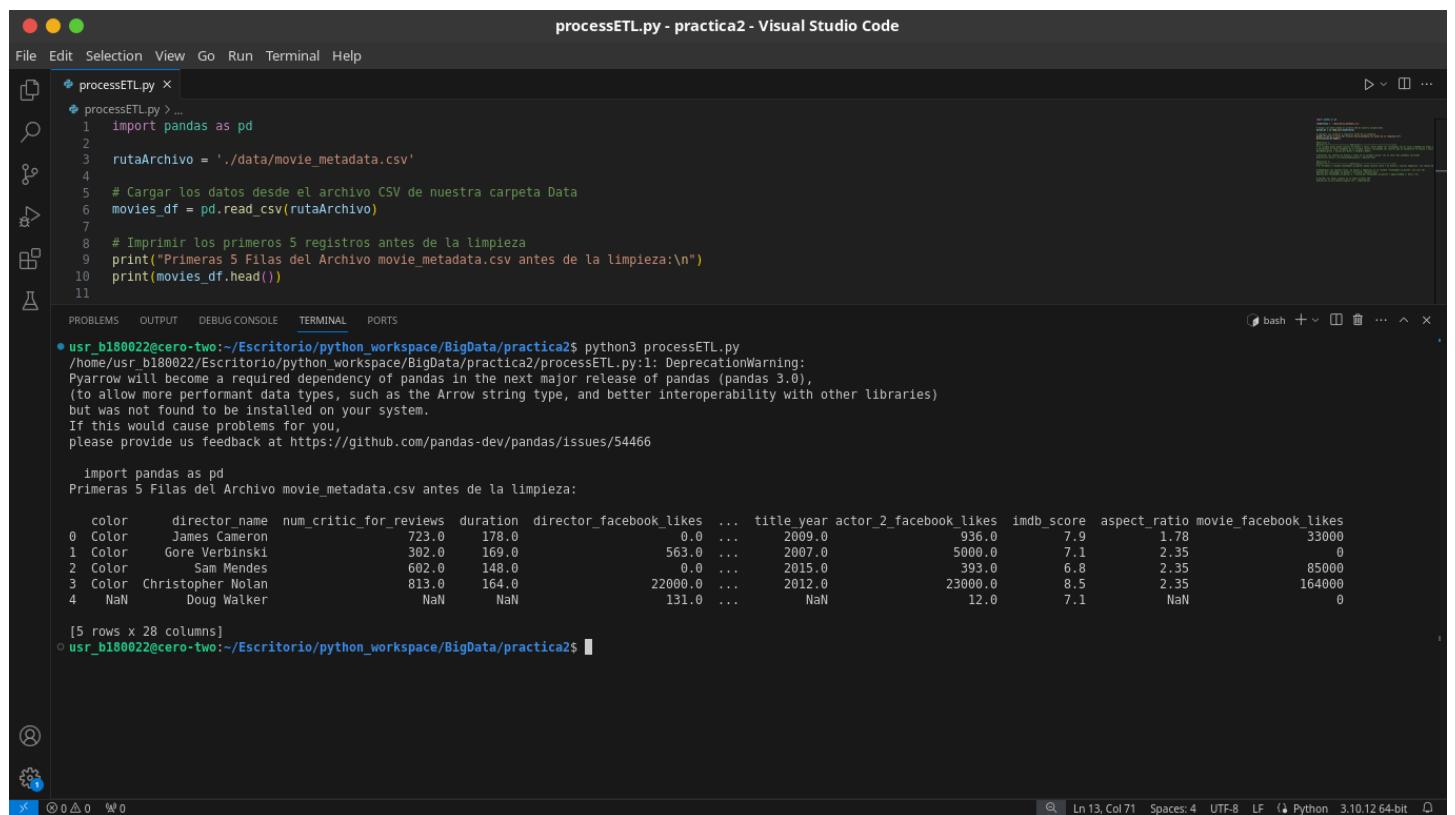
Consultamos el contenido del archivo movies.csv:



```
movie_metadata.csv - practica2 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
processETL.py movie_metadata.csv
data > movie_metadata.csv > □ data
1 color,director_name,num_critic_for_reviews,duration,director_facebook_likes,actor_3_facebook_likes,actor_2_name,actor_1_facebook_likes,gross,genres,actor_1_name,
2 Color,James Cameron,723,178,0,855,Joel David Moore,1000,760505847,Action|Adventure|Fantasy|Sci-Fi,CCH Pounder,Avatar ,886204,4834,Wes Studi,0,avatar|future|marin
3 Color,Gore Verbinski,302,169,563,1000,Orlando Bloom,40000,309404152,Action|Adventure|Fantasy,Johnny Depp,Pirates of the Caribbean: At World's End ,471220,48350,J
4 Color,Sam Mendes,602,148,0,161,Rory Kinnear,11000,200074175,Action|Adventure|Thriller,Christoph Waltz,Spectre ,275868,11700,Stephanie Sigman,1,bomb|espionage|sec
5 Color,Christopher Nolan,813,164,22000,Christian Bale,27000,448130642,Action|Thriller,Tom Hardy,The Dark Knight Rises ,1144337,106759,Joseph Gordon-Levitt,c
6 ,Doug Walker,,,131,,Rob Walker,Star Wars: Episode VII - The Force Awakens ,8,143,,0,,http://www.imdb.com/title/tt5289954
7 Color,Andrew Stanton,462,132,475,530,Samantha Morton ,640,738458679,Action|Adventure|Sci-Fi,Daryl Sabara,John Carter ,212204,1873,Polly Walker,l,alien|american civ
8 Color,Sam Raimi,392,156,0,4000,James Franco,24000,3 Col:9:gross tion|Adventure|Romance,J.K. Simmons,Spider-Man 3 ,383056,46055,Kirsten Dunst,0,sandman|spider man|s
9 Color,Nathan Greno,375,153,282,10000,Donna Murphy,799,20080762,Adventure|Animation|Comedy|Family|Fantasy|Musical|Romance,Brad Garrett,Tangled ,294810,2036,M.C. Gai
10 Color,Joss Whedon,635,141,0,19000,Robert Downey Jr.,26000,458991599,Action|Adventure|Sci-Fi,Chris Hemsworth,Avengers: Age of Ultron ,462669,92000,Scarlett Johans
11 Color,David Yates,375,153,282,10000,Daniel Radcliffe,25000,301956980,Adventure|Family|Fantasy|Mystery,Alan Rickman,Harry Potter and the Half-Blood Prince ,321795
12 Color,Zack Snyder ,673,183,0,2000,Lauren Cohan,15000,330249062,Action|Adventure|Sci-Fi,Henry Cavill,Batman v Superman: Dawn of Justice ,371639,24450,Alan D. Poul
13 Color,Bryan Singer,434,169,0,903,Marlon Brando,18000,200069408,Action|Adventure|Sci-Fi,Kevin Spacey,Superman Returns ,240396,29991,Frank Langella,0,crystal|epic|
14 Color,Marc Forster,463,106,395,393,Mathieu Amalric,451,168368427,Action|Adventure,Giancarlo Giannini,Quantum of Solace ,330784,2023,Rory Kinnear,1,action hero|at
15 Color,Gore Verbinski,313,151,563,1000,Orlando Bloom,40000,423032628,Action|Adventure|Fantasy,Johnny Depp,Pirates of the Caribbean: Dead Man's Chest ,522040,48486
16 Color,Gore Verbinski,450,150,563,1000,Ruth Wilson,40000,89289910,Action|Adventure|Western,Johnny Depp,The Lone Ranger ,181794,45757,Tom Wilkinson,1,horse|outlaw|
17 Color,Zack Snyder ,733,143,0,748,Christopher Meloni,15000,291021565,Action|Adventure|Fantasy|Sci-Fi,Henry Cavill,Man of Steel ,548573,20495,Harry Lennix,0,based on
18 Color,Andrew Adamson,258,150,80,201,Pierfrancesco Favino,22000,141614023,Action|Adventure|Family|Fantasy,Peter Dinklage,The Chronicles of Narnia: Prince Caspian
19 Color,Joss Whedon ,703,173,0,19000,Robert Downey Jr.,26000,623279547,Action|Adventure|Sci-Fi,Chris Hemsworth,The Avengers ,995415,87697,Scarlett Johansson,3,alien
20 Color,Rob Marshall,448,136,252,1000,Sam Clafin,40000,241063875,Action|Adventure|Fantasy,Johnny Depp,Pirates of the Caribbean: On Stranger Tides ,370704,54083,St
21 Color,Barry Sonnenfeld,451,106,188,718,Michael Stuhlbarg,10000,719020854,Action|Adventure|Comedy|Family|Fantasy|Sci-Fi,Will Smith,Men in Black 3 ,268154,12572,Ni
22 Color,Peter Jackson,422,164,0,773,Adam Brown,5000,255108370,Adventure|Fantasy,Aidan Turner,The Hobbit: The Desolation of Smaug ,354228,9152,James Nesbitt,0
23 Color,Marc Webb,599,153,464,963,Andrew Garfield,15000,262030663,Action|Adventure|Fantasy,Emma Stone,The Amazing Spider-Man ,451803,28489,Chris Zylka,0,lizard|out
24 Color,Ridley Scott,343,156,0,738,William Fichtner,891,105219735,Action|Adventure|Drama|History,Mark Addy,Robin Hood ,211765,3244,Scott Grimes,0,1190s|archer|england|
25 Color,Peter Jackson,509,186,0,773,Adam Brown,5000,258335534,Adventure|Fantasy,Aidan Turner,The Hobbit: The Desolation of Smaug ,483540,9152,James Nesbitt,6,dwarf
26 Color,Chris Weitz,251,113,129,1000,Eva Green,16000,70083519,Adventure|Family|Fantasy,Christopher Lee,The Golden Compass ,149019,24106,Kristin Scott Thomas,2,child
27 Color,Peter Jackson,446,201,0,84,Thomas Kretschmann,6000,218051260,Action|Adventure|Drama|Romance,Naomi Watts,King Kong ,316018,7123,Evan Parke,0,animal name in
28 Color,James Cameron,315,194,0,794,Kate Winslet,29000,658672302,Drama|Romance,Leonardo DiCaprio,Titanic ,793059,45223,Gloria Stuart,0,artist|love|ship|titanic|wet
29 Color,Anthony Russo,516,147,94,11000,Scarlett Johansson,21000,407197282,Action|Adventure|Sci-Fi,Robert Downey Jr.,Captain America: Civil War ,272670,64798,Chris
30 Color,Peter Berg,377,131,532,627,Alexander Skarsgård,14000,65173160,Action|Adventure|Sci-Fi|Thriller,Liam Neeson,Battleship ,202382,26679,Tadanobu Asano,0,box of
31 Color,Colin Trevorrow,644,124,365,10000,Judy Greer,883,304360277,Action|Adventure|Sci-Fi|Thriller,Bryce Dallas Howard,Jurassic World ,418214,8458,Omar Sy,0,dinos
32 Color,Sam Mendes,750,143,0,393,Helen Mirren,883,304360277,Action|Adventure|Thriller,Albert Finney,Skyfall ,522030,2639,Rory Kinnear,0,brawl|childhood home|compu
33 Color,Sam Raimi,388,135,0,4000,James Franco,24000,373377893,Action|Adventure|Fantasy|Romance,J.K. Simmons,Spider-Man 2 ,411164,43388,Kirsten Dunst,1,death|doctor
34 Color,Shane Black,608,195,1000,3000,Jon Favreau,21000,408992272,Action|Adventure|Sci-Fi,Robert Downey Jr.,Iron Man 3 ,557489,36426,Don Cheadle,3,armor|explosion|
35 Color,Tim Burton,451,108,13000,11000,Alan Rickman,40000,334185206,Adventure|Family|Fantasy,Johnny Depp,Alice in Wonderland ,306320,79957,Anne Hathaway,0,alice in
36 Color,Brett Ratner,334,104,420,560,Kelsey Grammer,20000,234360014,Action|Adventure|Fantasy|Sci-Fi|Thriller,Hugh Jackman,X-Men: The Last Stand ,383427,21714,Danie
37 Color,Dan Scanlon,376,104,37,760,Tyler Labine,12000,268488329,Adventure|Animation|Comedy|Family|Fantasy,Steve Buscemi,Monsters University ,235025,14863,Sean Have
```

Ahora mostramos las primeras filas antes de la limpieza:

Pesenta: B18022, Jose Colombio Gonzalez Perez



```
processETL.py - practica2 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
processETL.py x
processETL.py > ...
1 import pandas as pd
2
3 rutaArchivo = './data/movie_metadata.csv'
4
5 # Cargar los datos desde el archivo CSV de nuestra carpeta Data
6 movies_df = pd.read_csv(rutaArchivo)
7
8 # Imprimir los primeros 5 registros antes de la limpieza
9 print("Primeras 5 Filas del Archivo movie_metadata.csv antes de la limpieza:")
10 print(movies_df.head())
11

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash + x ... ^ x

● usr_b180022@cero-two:~/Escritorio/python_workspace/BigData/practica2$ python3 processETL.py
/home/usr_b180022/Escritorio/python_workspace/BigData/practica2/processETL.py:1: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

import pandas as pd
Primeras 5 Filas del Archivo movie_metadata.csv antes de la limpieza:

   color    director_name  num_critic_for_reviews  duration  director_facebook_likes  ...  title_year  actor_2_facebook_likes  imdb_score  aspect_ratio  movie_facebook_likes
0  Color      James Cameron        723.0          178.0                  0.0  ...       2009.0            936.0     7.9      1.78           33000
1  Color       Gore Verbinski        302.0          169.0                 563.0  ...       2007.0            5000.0     7.1      2.35             0
2  Color        Sam Mendes        602.0          148.0                  0.0  ...       2015.0            393.0     6.8      2.35          85000
3  Color  Christopher Nolan        813.0          164.0                22000.0  ...       2012.0            23000.0     8.5      2.35         164000
4    NaN        Doug Walker           NaN              NaN                  131.0  ...             NaN            12.0     7.1      NaN             0

[5 rows x 28 columns]
● usr_b180022@cero-two:~/Escritorio/python_workspace/BigData/practica2$
```

1.-La columna gross posee valores en blanco o nulos (NaN), estos deben ser rellenados con el valor promedio de todos los valores de esa columna.

Se imprimira solo los valores de la columna Gross antes de realizar la limpieza y despues.

Nota: Nan Indica que es valor carece de algun valor dentro de ese campo.

Pesenta: B18022, Jose Colombio Gonzalez Perez

```

◆ processETL.py X
◆ processETL.py > ...
14 # La columna gross posee valores en blanco o nulos, estos deben ser rellenos con el valor promedio de todos los valores de esa columna.
15 #Imprimir la columna Gross pra ver el contenido antes de realizar la limpieza
16
17 print("\nValores de la columna 'gross' antes de la limpieza:")
18 print(movies_df['gross'].head(10))
19 print("\n\t----> Realizando Cambios en < Gross > .....\\n")
20
21 # Calculamos el valor promedio de la columna "gross" excluyendo los valores que se encuentren en blanco o nulos
22 promedio_gross = movies_df['gross'].dropna().mean()
23
24 # Reemplazar/rellenar los valores nulos en la columna "gross" con el valor del promedio calculado
25 movies_df['gross'] = movies_df['gross'].fillna(promedio_gross)
26 # Imprimir la columna "gross" despues de la limpieza
27 print("\nValores de la columna 'gross' despues de la limpieza:")
28 print(movies_df['gross'].head(10))
29
30
31
32 #Ejercicio 2:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[5 rows x 28 columns]
----- Ejercicio 1: -----
Valores de la columna 'gross' antes de la limpieza:
0    766505847.0
1    309404152.0
2    200074175.0
3    448130642.0
4     NaN
5    73058679.0
6    336530303.0
7    200807262.0
8    458991599.0
9    301956980.0
Name: gross, dtype: float64
-----> Realizando Cambios en < Gross > ......

Valores de la columna 'gross' despues de la limpieza:
0    6.605959e+08
1    3.094042e+08
2    2.000742e+08
3    4.481306e+08
4    4.846841e+07
5    7.305868e+07
6    3.365303e+08
7    2.008073e+08
8    4.589916e+08
9    3.019570e+08
Name: gross, dtype: float64
o usr_b18002@ecero-two:~/Escritorio/python_workspace/BigData/practica2s []

```

2.- El atributo o columna **facenumber_in_poster** posee valores nulos o en blanco y valores negativos, los cuales deber rellenados o reemplazados con el valor de cero **0**.

```

29
30
31
32 #Ejercicio 2:
33 # El atributo o columna facenumber_in_poster posee valores nulos o en blanco y valores negativos, los cuales deber rellenados o reemplazados con el valor de cero 0.
34 print("\n----- Ejercicio 2: -----")
35 print("Columna facenumber_in_poster posee valores nulos o en blanco y valores negativos, los cuales fueron rellenos/reemplazados con el valor de cero 0.")
36 # Reemplazar los valores nulos o negativos en la columna 'facenumber_in_poster' con ceros
37 movies_df['facenumber_in_poster'] = movies_df['facenumber_in_poster'].fillna(0)
38 movies_df['facenumber_in_poster'] = movies_df['facenumber_in_poster'].apply(lambda x: 0 if x < 0 else x)
39
40 # Mostrar los campos de la columna 'facenumber_in_poster' despues de la limpieza
41 print("\nSe esta aplicando la limpieza de 'facenumber_in_poster' : \t...TERminado...\n")
42
43
44

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

----- Ejercicio 2: -----

Columna facenumber_in_poster posee valores nulos o en blanco y valores negativos, los cuales fueron rellenos/reemplazados con el valor de cero 0.

Se esta aplicando la limpieza de 'facenumber_in_poster' : ...TERminado...

usr_b18022@cerro-two:~/Escritorio/python_workspace/BigData/practica2\$

3.- Crear una nueva columna denominada **TitleCode** y los valores que serán asignados resultar de realizar una extracción o subcadena de la columna **movie_imdb_link**.

Ejemplo: http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1 se extrae el dato: **tt0499549**

Nota: Para utilizar expresiones regulares para la búsqueda de lo solicitado se requierenos/necesitamos importar la libreria **<re>** de python. Por ende se crea una función para realizar la búsqueda de la expresión regular para poder crear la nueva columna y ingresar los datos extraídos apartir del link.

```

File Edit Selection View Go Run Terminal Help
processETL.py x
processETL.py > ...
52 print("\nSe esta aplicando la limpieza de 'facenumber_in_poster' : \t...TERminado...\n")
53
54 #Ejercicio 3:
55 #3.- Crear una nueva columna denominada TitleCode y los valores que serán asignados resultar de realizar una extracción o subcadena de la columna movie_imdb_link.
56 # Ejemplo: http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1 se extrae el dato: tt0499549
57 print("\n----- Ejercicio 3: -----")
58
59 print()
60     "3.- Crear una nueva columna denominada TitleCode y los valores que serán asignados resultar de realizar una extracción o subcadena de la columna movie_imdb_link."
61     "\n\tEjemplo: http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1 se extrae el dato: <tt0499549>"
62 []
63 #hacemos uso de la función declarada en el comienzo del programa:
64 # Aplicar la función a la columna 'movie_imdb_link' para crear la nueva columna 'TitleCode'
65 movies_df['TitleCode'] = movies_df['movie_imdb_link'].apply(extract_title_code)
66 print("Creando nueva columna.....\n")
67 # Mostrar los primeros registros de la columna 'TitleCode'
68 print("\nPrimeros registros de la columna 'TitleCode':")
69 print(movies_df['TitleCode'].head())
70

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

----- Ejercicio 3: -----

3.- Crear una nueva columna denominada TitleCode y los valores que serán asignados resultar de realizar una extracción o subcadena de la columna movie_imdb_link.

Ejemplo: http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1 se extrae el dato: <tt0499549>

Creando nueva columna.....

Primeros registros de la columna 'TitleCode':

0	tt0499549
1	tt0449088
2	tt2379713
3	tt1345836
4	tt5289954

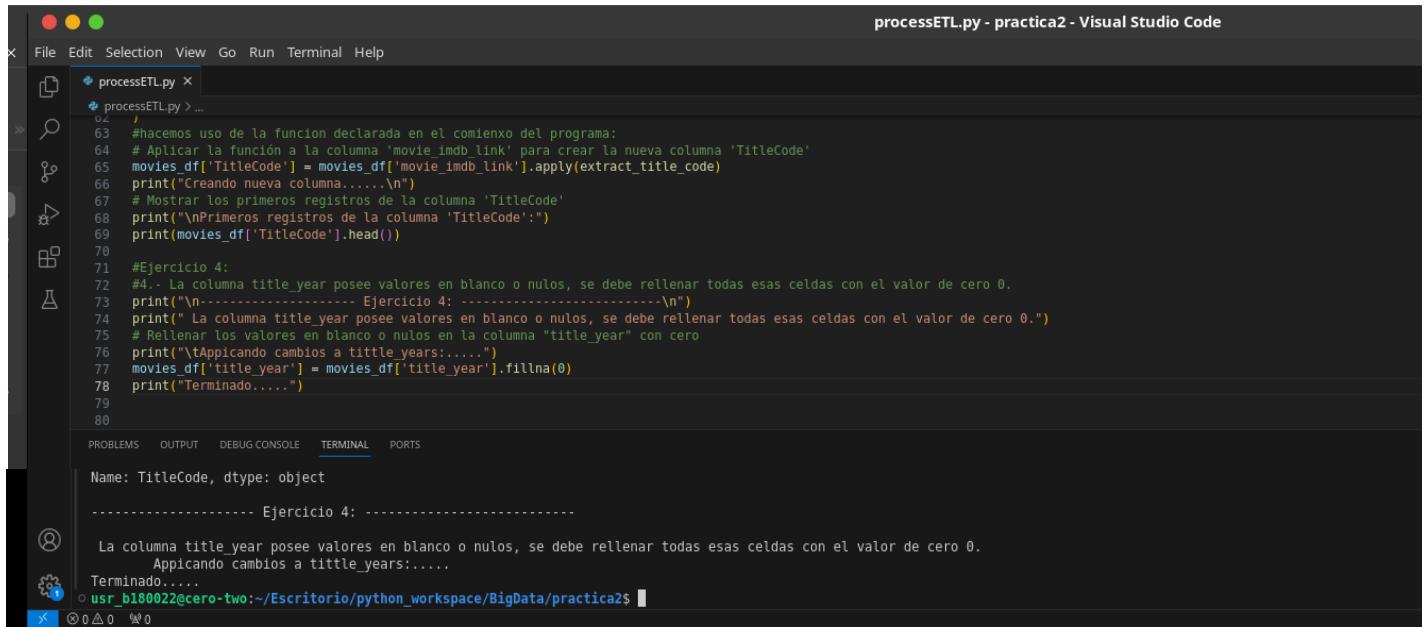
Name: TitleCode, dtype: object

usr_b18022@cerro-two:~/Escritorio/python_workspace/BigData/practica2\$

4.- La columna **title_year** posee valores en blanco o nulos, se debe llenar todas esas celdas con el valor de cero **0**.

Presenta: B18022, Jose Colomio Gonzalez Perez

Nota: para llenar los campos nulos que posee title_year usaremos el metodo Fillna().



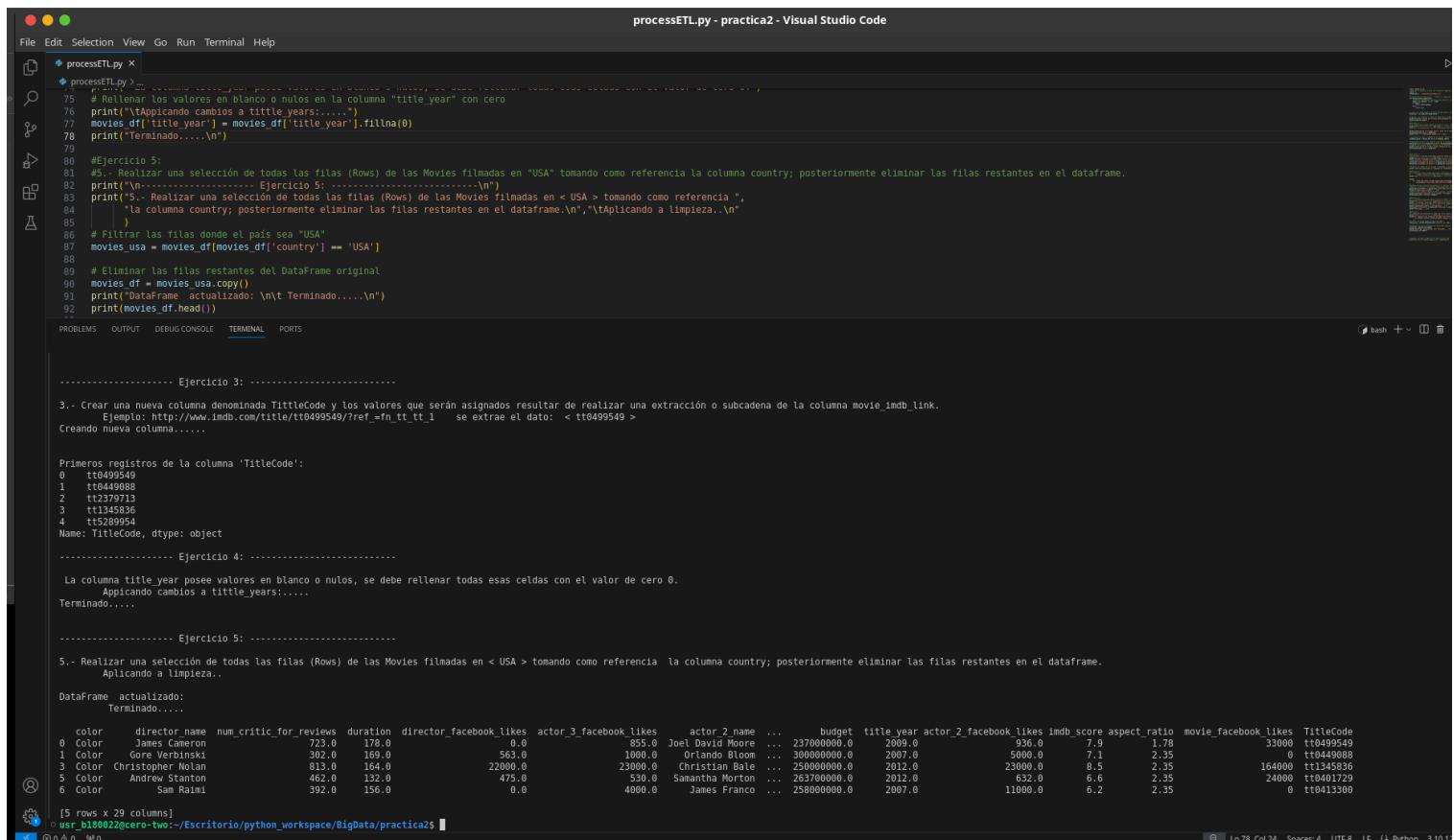
```
processETL.py - practica2 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
processETL.py > ...
02
63 #hacemos uso de la funcion declarada en el comienzo del programa:
64 # Aplicar la función a la columna 'movie_imdb_link' para crear la nueva columna 'TitleCode'
65 movies_df['TitleCode'] = movies_df['movie_imdb_link'].apply(extract_title_code)
66 print("Creando nueva columna.....\n")
67 # Mostrar los primeros registros de la columna 'TitleCode':
68 print("\nPrimeros registros de la columna 'TitleCode':")
69 print(movies_df['TitleCode'].head())
70
71 #Ejercicio 4:
72 #.- La columna title_year posee valores en blanco o nulos, se debe llenar todas esas celdas con el valor de cero 0.
73 print("\n----- Ejercicio 4: -----")
74 print(" La columna title_year posee valores en blanco o nulos, se debe llenar todas esas celdas con el valor de cero 0 .")
75 # Rellenar los valores en blanco o nulos en la columna "title_year" con cero
76 print("\tAplicando cambios a tittle_years:....")
77 movies_df['title_year'] = movies_df['title_year'].fillna(0)
78 print("Terminado.....")
79
80

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Name: TitleCode, dtype: object
----- Ejercicio 4:
La columna title_year posee valores en blanco o nulos, se debe llenar todas esas celdas con el valor de cero 0 .
    Aplicando cambios a tittle_years:.....
Terminado.....
usr_b180022@cer0-two:~/Escritorio/python_workspace/BigData/practica2$
```

5.- Realizar una selección de todas las filas (Rows) de las Movies filmadas en "USA" tomando como referencia la columna **country; posteriormente eliminar las filas restantes en el dataframe.**



```

File Edit Selection View Go Run Terminal Help
processETL.py - practica2 - Visual Studio Code
processETL.py
75 # Rellenar los valores en blanco o nulos en la columna "title_year" con cero
76 print("\tAplicando cambios a title_years.....")
77 movies_df['title_year'] = movies_df['title_year'].fillna(0)
78 print("Terminado....\n")
79
80 #Ejercicio 5:
81 #5.. Realizar una selección de todas las filas (Rows) de las Movies filmadas en "USA" tomando como referencia la columna country; posteriormente eliminar las filas restantes en el dataframe.
82 print("\n----- Ejercicio 5: ----- \n")
83 print("5.. Realizar una selección de todas las filas (Rows) de las Movies filmadas en < USA > tomando como referencia ", 
84 "la columna country; posteriormente eliminar las filas restantes en el DataFrame.\n", "\tAplicando a limpieza..\n"
85 )
86 # Filtrar las filas donde el país sea "USA"
87 movies_usa = movies_df[movies_df['country'] == 'USA']
88
89 # Eliminar las filas restantes del DataFrame original
movies_df = movies_usa.copy()
90 print("DataFrame actualizado: \n\t Terminado....\n")
91 print(movies_df.head())
92
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash + □
----- Ejercicio 3: -----
3.. Crear una nueva columna denominada TitleCode y los valores que serán asignados resultar de realizar una extracción o subcadena de la columna movie_imdb_link.
    Ejemplo: http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1 se extrae el dato: <tt0499549>
Creando nueva columna.....
Primeros registros de la columna 'TitleCode':
0 tt0499549
1 tt0180880
2 tt2379713
3 tt1345836
4 tt5289954
Name: TitleCode, dtype: object
----- Ejercicio 4: -----
La columna title_year posee valores en blanco o nulos, se debe llenar todas esas celdas con el valor de cero 0.
    Aplicando cambios a title_years....
Terminado.....
----- Ejercicio 5: -----
5.. Realizar una selección de todas las filas (Rows) de las Movies filmadas en < USA > tomando como referencia la columna country; posteriormente eliminar las filas restantes en el dataframe.
    Aplicando a limpieza..
DataFrame actualizado:
Terminado.....
color director_name num_critic_for_reviews duration director_facebook_likes actor_3_facebook_likes actor_2_name ... budget title_year actor_2_facebook_likes imdb_score aspect_ratio movie_facebook_likes TitleCode
0 Color James Mangold 723.0 170.0 850.0 185.0 Joel Edgerton ... 237000000.0 2007.0 1350.0 7.9 1.78 330000 tt0499549
1 Color Gore Verbinski 307.0 160.0 563.0 1800.0 Orlando Bloom ... 300000000.0 2007.0 5000.0 7.1 2.35 0 tt0499088
3 Color Christopher Nolan 813.0 164.0 23000.0 23000.0 Christian Bale ... 250000000.0 2012.0 23000.0 8.5 2.35 164000 tt1345936
5 Color Andrew Stanton 462.0 132.0 475.0 530.0 Samantha Morton ... 263700000.0 2012.0 632.0 6.6 2.35 24000 tt0401729
6 Color Sam Raimi 392.0 156.0 0.0 4000.0 James Franco ... 258000000.0 2007.0 11000.0 6.2 2.35 0 tt0413300
[5 rows x 29 columns]

```

6.- Generar un nuevo archivo llamado "FilmTV_USAMoviesClean.csv"

```

processETL.py - practica2 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER PRACTICA2
  data FilmTV_USAMoviesClean.csv
  movie_metadata.csv
  S2_AIPractica2.odt#
processETL.py
S2_AIPractica2.odt

processETL.py
  processETL.py
    processETL.py...
      print("Dataframe actualizado: \n\t terminado....\n")
      print(movies_df.head())
      # Ejercicio 6:
      # Generar un nuevo archivo CSV con las peliculas filmadas en "USA"
      print("\n----- Ejercicio 6: -----")
      print("Generar un nuevo archivo CSV con las peliculas filmadas en < USA > \n")
      # Guardar los datos limpios en un nuevo archivo CSV en nuestra carpeta data
      movies_usa.to_csv("./data/FilmTV_USAMoviesClean.csv", index=False)
      print("Generando archivo... \n")
      print("El archivo se guardo con exito en practica2/data/FilmTV_USAMoviesClean.csv\n")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ejercicio 5:
5.- Realizar una seleccion de todas las filas (Rows) de las Movies filmadas en < USA > tomando como referencia la columna country; posteriormente eliminar las filas restantes en el dataframe.
Aplicando a limpieza.

DataFrame actualizado:
Terminado.....
color director name num_critic_for_reviews duration director_facebook_likes actor_3_facebook_likes ... title_year actor_2_facebook_likes imdb_score aspect_ratio movie_facebook_likes TitleCode
0 color James Cameron 723.0 178.0 0.0 855.0 ... 2009.0 936.0 7.9 1.78 33000 tt0499549
1 Color Gore Verbinski 302.0 169.0 563.0 1000.0 ... 2007.0 5000.0 7.1 2.35 0 tt0449088
3 Color Christopher Nolan 813.0 164.0 22000.0 23000.0 ... 2012.0 23000.0 8.5 2.35 164000 tt1345836
5 Color Andrew Stanton 462.0 152.0 475.0 530.0 ... 2012.0 652.0 6.6 2.35 24000 tt0401729
6 Color Sam Raimi 392.0 156.0 0.0 4000.0 ... 2007.0 11000.0 6.2 2.35 0 tt0413200
[5 rows x 29 columns]
Ejercicio 6:
Generar un nuevo archivo CSV con las peliculas filmadas en < USA >
Generando archivo...
TERminado
El archivo se guardo con exito en practica2/data/FilmTV_USAMoviesClean.csv
* user_b180022@cer0-two:~/Escritorio/python_workspace/BigData/practica2$ ls
* data processETL.py S2_AIPractica2.odt
* user_b180022@cer0-two:~/Escritorio/python_workspace/BigData/practica2$ tree
.
+- data
  +- FilmTV_USAMoviesClean.csv
  +- movie_metadata.csv
  +- processETL.py
  +- S2_AIPractica2.odt

1 directory, 4 files
* user_b180022@cer0-two:~/Escritorio/python_workspace/BigData/practica2$ tree -la
.
+- data
  +- FilmTV_USAMoviesClean.csv
  +- movie_metadata.csv
  +- processETL.py
  +- S2_AIPractica2.odt

1 directory, 4 files
* user_b180022@cer0-two:~/Escritorio/python_workspace/BigData/practica2$ ln 101, Col 48 Spaces: 4 UTF-8 Python 3.10.12 64-bit

```

7.- Efectuar la carga (**Load**) del archivo limpio y transformado en un Gestor de Base de Datos Relacional

Para llevar cabo este ultimo ejercicio es necesario realizar los siguiente spasos:

1.- Instalar libreria `psycopg2` de python para conectarnos con el SGBD PostgreSQL, como nos encontramos en el entorno linux Ubuntu Budgy en su version 24.1.0 , ejecutamos el comando:

`pip install pandas psycopg2`

```

done
user_b180022@cer0-two:~$ pip install psycopg2
Defaulting to user installation because normal site-packages is not writeable
Collecting psycopg2
Using cached psycopg2-2.9.9.tar.gz (384 kB)
Preparing metadata (setup.py) ... done
Building wheels for collected packages: psycopg2
Building wheel for psycopg2 (setup.py) ... error
  error: subprocess-exited-with-error

  x python setup.py bdist_wheel did not run successfully.
  | exit code: 1
  + [38 lines of output]
    running bdist_wheel
    running build
    running build_py
    creating build
    creating build/lib.linux-x86_64-3.10
    creating build/lib.linux-x86_64-3.10 psycopg2
    copying lib/pool.py --> build/lib.linux-x86_64-3.10 psycopg2
    
```

Si te salio el mismo error que yo, no te preocunes, aqui lo solucionaremos.,

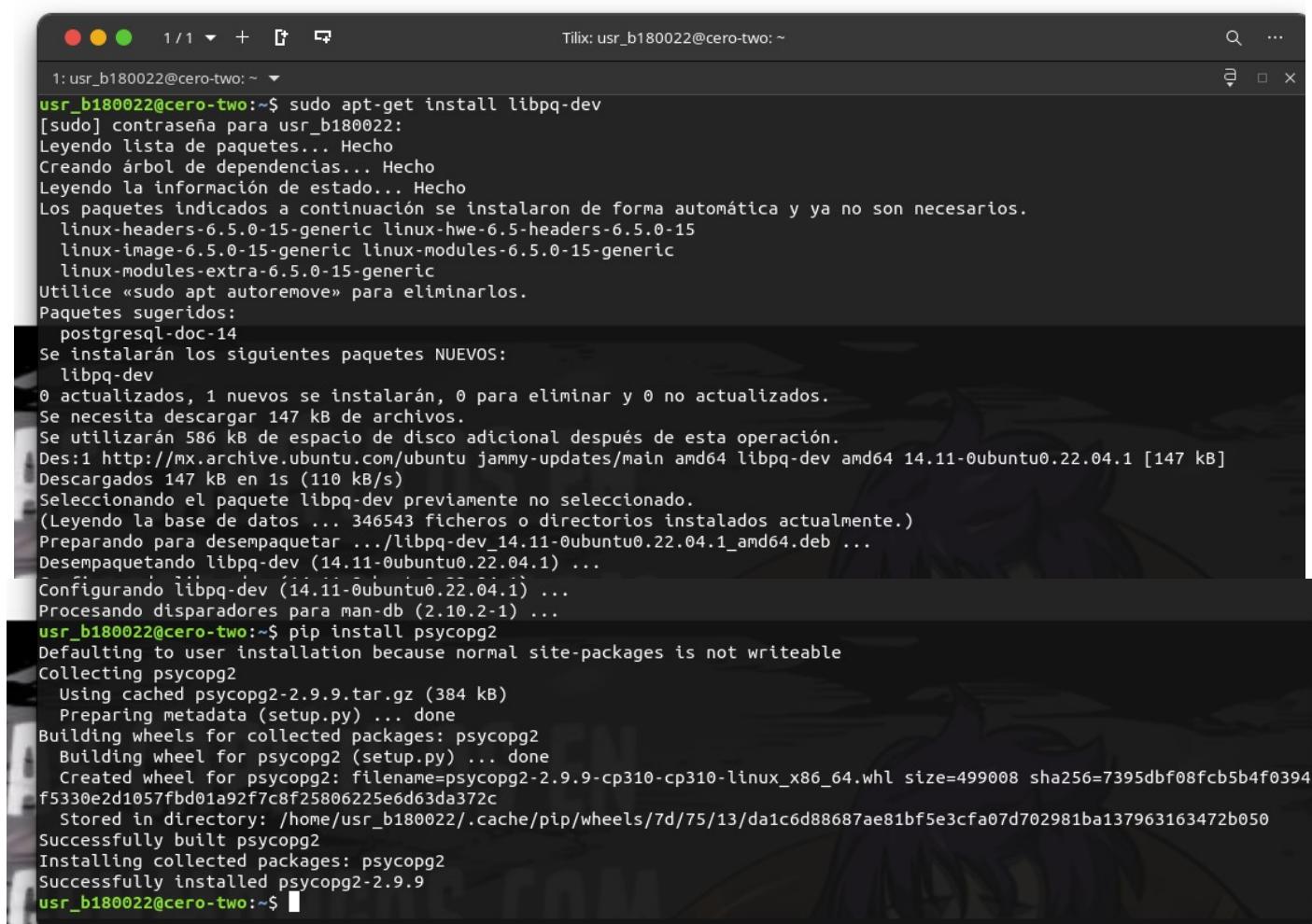
El error radica en que nos hace falta algunos complementos que necesita psycopg2 para realizar la conexión con postgres. Es decir nos hace falta complementos o librerías necesarias para que postgresql trabaje bien., en otras palabras necesitamos instalar las bibliotecas de desarrollo de PostgreSQL en nuestro sistema.

Biblioteca de postgres: libpq-dev lo instalamos con el comando:

```
sudo apt-get install libpq-dev
```

Una vez instaladas las bibliotecas de desarrollo de PostgreSQL, intenta instalar psycopg2 nuevamente usando pip:

```
pip install psycopg2
```



```
1:usr_b180022@cero-two:~$ sudo apt-get install libpq-dev
[sudo] contraseña para usr_b180022:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  linux-headers-6.5.0-15-generic linux-hwe-6.5-headers-6.5.0-15
  linux-image-6.5.0-15-generic linux-modules-6.5.0-15-generic
  linux-modules-extra-6.5.0-15-generic
Utilice «sudo apt autoremove» para eliminarlos.
Paquetes sugeridos:
  postgresql-doc-14
Se instalarán los siguientes paquetes NUEVOS:
  libpq-dev
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 147 kB de archivos.
Se utilizarán 586 kB de espacio de disco adicional después de esta operación.
Des:1 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpq-dev amd64 14.11-0ubuntu0.22.04.1 [147 kB]
Descargados 147 kB en 1s (110 kB/s)
Seleccionando el paquete libpq-dev previamente no seleccionado.
(Leyendo la base de datos ... 346543 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libpq-dev_14.11-0ubuntu0.22.04.1_amd64.deb ...
Desempaquetando libpq-dev (14.11-0ubuntu0.22.04.1) ...
Configurando libpq-dev (14.11-0ubuntu0.22.04.1) ...
Procesando disparadores para man-db (2.10.2-1) ...
usr_b180022@cero-two:~$ pip install psycopg2
Defaulting to user installation because normal site-packages is not writeable
Collecting psycopg2
  Using cached psycopg2-2.9.9.tar.gz (384 kB)
    Preparing metadata (setup.py) ... done
Building wheels for collected packages: psycopg2
  Building wheel for psycopg2 (setup.py) ... done
    Created wheel for psycopg2: filename=psycopg2-2.9.9-cp310-cp310-linux_x86_64.whl size=499008 sha256=7395dbf08fc5b4f0394
f5330e2d1057fdb01a92f7c8f25806225e6d63da372c
    Stored in directory: /home/usr_b180022/.cache/pip/wheels/7d/75/13/da1c6d88687ae81bf5e3cfa07d702981ba137963163472b050
Successfully built psycopg2
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.9
usr_b180022@cero-two:~$
```

Nota: Si sigues teniendo problemas, también puedes intentar instalar psycopg2-binary, que es una versión binaria precompilada de psycopg2 y te funcionara de igual manera sin problemas:

```
pip install psycopg2-binary
```

2.- Creamos una base de datos usando nuestro Cliente postgresql , en Ubuntu PostgreSQL ya viene por defecto,

```
usr_b180022@cero-two:~$ psql --version
psql (PostgreSQL) 14.11 (Ubuntu 14.11-0ubuntu0.22.04.1)
usr_b180022@cero-two:~$
```

asi que procedemos a conectar a postgres en nuestra terminal y ejecutar el comando:

CREATE DATABASE movies_database;

```
usr_b180022@cero-two:~$ psql --version
psql (PostgreSQL) 14.11 (Ubuntu 14.11-0ubuntu0.22.04.1)
usr_b180022@cero-two:~$ psql -U postgres
psql: error: connection to server on socket "/var/run/postgresql/.s.PGSQL.5432" failed: FATAL:  Peer authentication failed for user "postgres"
usr_b180022@cero-two:~$ sudo su postgres
postgres@cero-two:/home/usr_b180022$ psql
could not change directory to "/home/usr_b180022": Permisio denegado
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1))
Type "help" for help.

asi que procedemos a conectar a postgres en nuestra terminal y ejecutar el comando:
postgres=# CREATE DATABASE movies_database;
CREATE DATABASE movies_database;
```

3.- Crear la Base de Datos y la Tabla: Utilizando un cliente de base de datos o un script en Python, puedes crear una base de datos y una tabla que coincida con la estructura del DataFrame limpio. En este caso usaremos un script Python para ello:

```

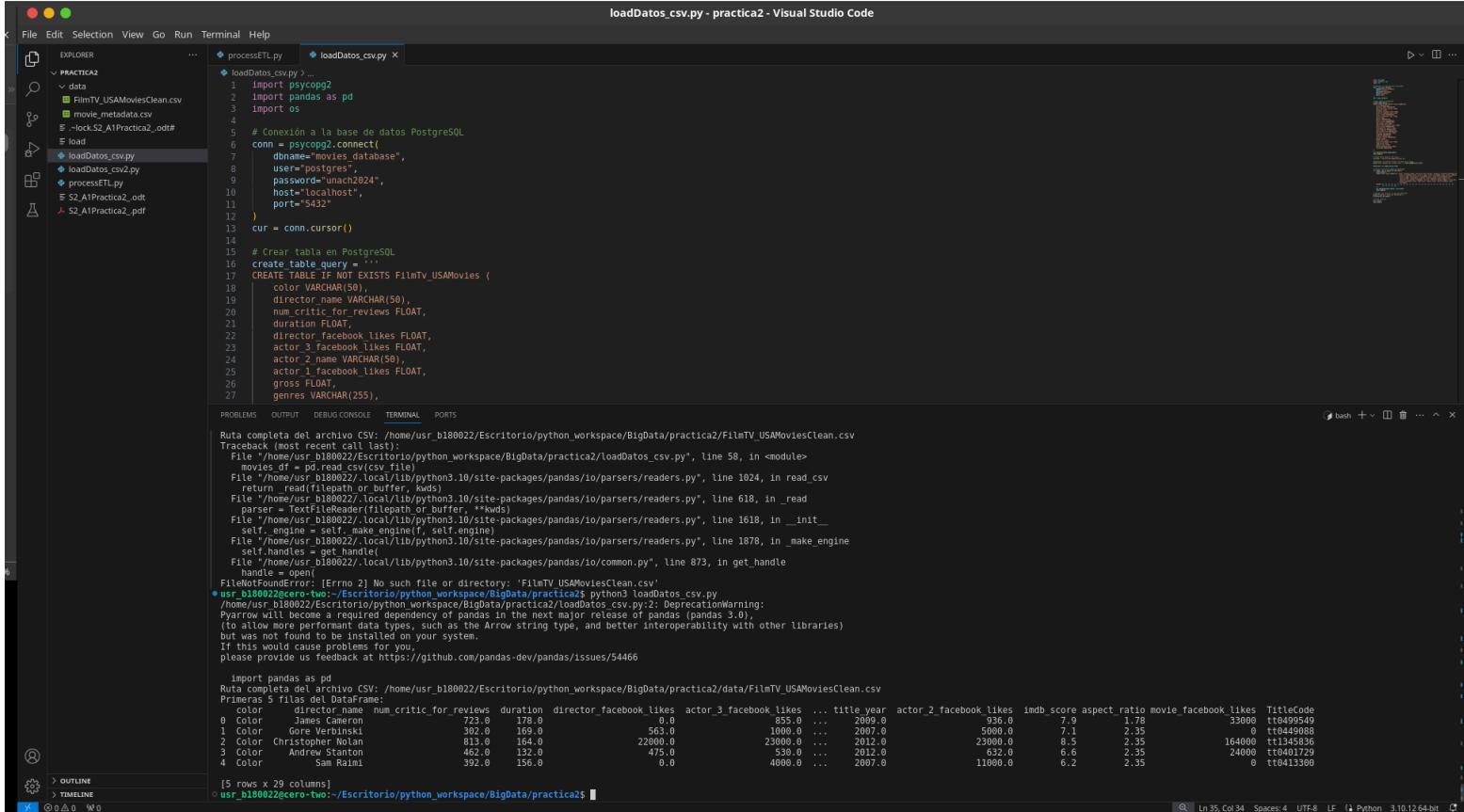
Terminal Help
  processETL.py      loadDatos_csv.py x

loadDatos_csv.py > ...
1  import psycopg2
2  import pandas as pd
3  import os
4
5  # Conexión a la base de datos PostgreSQL
6  conn = psycopg2.connect(
7      dbname="movies_database",
8      user="postgres",
9      password="unach2024",
10     host="localhost",
11     port="5432"
12 )
13 cur = conn.cursor()
14
15 # Crear tabla en PostgreSQL
16 create_table_query = """
17 CREATE TABLE IF NOT EXISTS FilmTv_USAMovies (
18     color VARCHAR(50),
19     director_name VARCHAR(50),
20     num_critic_for_reviews FLOAT,
21     duration FLOAT,
22     director_facebook_likes FLOAT,
23     actor_3_facebook_likes FLOAT,
24     actor_2_name VARCHAR(50),
25     actor_1_facebook_likes FLOAT,
26     gross FLOAT,
27     genres VARCHAR(255),
28     actor_1_name VARCHAR(50),
29     movie_title VARCHAR(255),
30     num_voted_users FLOAT,
31     cast_total_facebook_likes FLOAT,
32     actor_3_name VARCHAR(50),
33     facenumber_in_poster FLOAT,
34     plot_keywords VARCHAR(255),
35     movie_imdb_link VARCHAR(255),
36     num_user_for_reviews FLOAT,
37     language VARCHAR(50),
38     country VARCHAR(50),
39     content_rating VARCHAR(50),
40     budget FLOAT,
41     title_year FLOAT,
42     actor_2_facebook_likes FLOAT,
43     imdb_score FLOAT,
44     aspect_ratio FLOAT,
45     movie_facebook_likes FLOAT,
46     TitleCode VARCHAR(50)
47 );
48 ...
49 cur.execute(create_table_query)
50 conn.commit()
51
52 # Cargar datos desde el CSV limpio
53 csv_file = 'data/FilmTV_USAMoviesClean.csv'
54
55 #imprimimos la ruta del archivo csv antes de la carga
56 print("Ruta completa del archivo CSV:", os.path.abspath(csv_file))
57
58 movies_df = pd.read_csv(csv_file)
59
60 # Insertar datos en la tabla de PostgreSQL
61 for index, row in movies_df.iterrows():
62     insert_query = """
63     INSERT INTO FilmTv_USAMovies (color, director_name, num_critic_for_reviews, duration, director_facebook_likes,
64                                     actor_3_facebook_likes, actor_2_name, actor_1_facebook_likes, gross, genres,
65                                     actor_1_name, movie_title, num_voted_users, cast_total_facebook_likes,
66                                     actor_3_name, facenumber_in_poster, plot_keywords, movie_imdb_link,
67                                     num_user_for_reviews, language, country, content_rating, budget, title_year,
68                                     actor_2_facebook_likes, imdb_score, aspect_ratio, movie_facebook_likes,
69                                     TitleCode)
70     VALUES (%s, %s, %s,
71             %s, %s, %s, %s, %s)
72     """
73     cur.execute(insert_query, tuple(row))
74     conn.commit()
75
76 # Imprimir las primeras 5 filas del DataFrame
77 print("Primeras 5 filas del DataFrame:")
78 print(movies_df.head())
79
80 # Cerrar conexión
81 cur.close()
82 conn.close()
83

```

Con estos pasos, habremos cargado exitosamente los datos del archivo CSV limpio *FilmTV_USAMoviesClean.csv* en una tabla *FilmTv_USAMovies*. De nuestra base de datos creada anteriormente.

En nuestra terminal nos debio de arrojar un resultado parecido a este:



The screenshot shows a Visual Studio Code interface with a Python script named `loadData_csv.py` open in the editor. The script connects to a PostgreSQL database named "movies" using psycopg2 and creates a table if it doesn't exist. It then reads a CSV file named `FilmTV_USAMoviesClean.csv` and inserts its data into the table. The code includes imports for `psycopg2`, `pandas`, and `os`. The terminal tab shows the execution of the script, which completes successfully, creating the table and inserting data. The output shows the first five rows of the inserted data.

```
Ruta completa del archivo CSV: '/home/usr_b180022/Escritorio/python_workspace/BigData/practica2/FilmTV_USAMoviesClean.csv'
Traceback (most recent call last):
  File "/home/usr_b180022/Escritorio/python_workspace/BigData/practica2/loadData_csv.py", line 58, in <module>
    movies_df = pd.read_csv(csv_file)
  File "/home/usr_b180022/.local/lib/python3.10/site-packages/pandas/io/parsers/readers.py", line 1024, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "/home/usr_b180022/.local/lib/python3.10/site-packages/pandas/io/parsers/readers.py", line 618, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "/home/usr_b180022/.local/lib/python3.10/site-packages/pandas/io/parsers/readers.py", line 1618, in __init__
    self._engine = self._get_engine(f, self.engine)
  File "/home/usr_b180022/.local/lib/python3.10/site-packages/pandas/io/parsers/readers.py", line 1878, in _make_engine
    self._handle = get_handle(f, engine)
  File "/home/usr_b180022/.local/lib/python3.10/site-packages/pandas/io/common.py", line 873, in get_handle
    handle = open(f, 'r', encoding='utf-8')
FileNotFoundError: [Errno 2] No such file or directory: 'FilmTV_USAMoviesClean.csv'
*usr_b180022@cero-two:~/Escritorio/python_workspace/BigData/practica2$ python3 loadData_csv.py:2: DeprecationWarning:
Pyyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

import pandas as pd
Ruta completa del archivo CSV: '/home/usr_b180022/Escritorio/python_workspace/BigData/practica2/data/FilmTV_USAMoviesClean.csv'
Primero 5 filas del DataFrame:
   color      director_name num_critic_for_reviews duration director_facebook_likes actor_3_facebook_likes ... title_year actor_2_facebook_likes  ...   imbd_score aspect_ratio movie_facebook_likes  TitleCode
0  Color      James Cameron          723.0        178.0             0.0                855.0     ...    2009.0           936.0       7.9      1.78            33000  tt0499549
1  Color      Gore Verbinski         302.0        169.0             563.0               1000.0     ...    2007.0           5000.0       7.1      2.35            0  tt0449088
2  Color Christopher Nolan        181.0        160.0             22000.0              23000.0     ...    2012.0           23000.0       8.5      2.35            164000  tt0401730
3  Color      Andrew Stanton        462.0        132.0             475.0                530.0     ...    2012.0           632.0       6.0      2.35            24000  tt0401729
4  Color      Sam Raimi            392.0        156.0             0.0                4000.0     ...    2007.0           11000.0      6.2      2.35            0  tt04113300
```

Nota: en nuestra creacion de tabla dentro del script de python debemos revisar que los datos y los nombres de columnas del archivo.csv sean las mismas que en la tabla creada en la base dedatos para no tener problemas de conversion a momento de inserccion de datos de nuestro DataFrame a nuestra tabla en base de datos.

Mas adelante comarto una imagen del resultado de la inserccion de datos del script hacia la tabla de nuestra base de datos:

Una disculpa por la imagen.

Pesenta: B18022, Jose Colombio Gonzalez Perez

Esta es la estructura de la practica

```
usr_b180022@cero-two:~/Escritorio/python_workspace/BigData/practica2$ tree
```

```
.practica2
├── data
│   ├── FilmTV_USAMoviesClean.csv (nuevo csv)
│   └── movie_metadata.csv (csv usado para la practica)
└── load
    ├── loadDatos_csv.py (script para exportar csv a tabla PostgreSQL)
    ├── processETL.py (script para limpiar el csv siguiendo los requerimientos de la practica)
    └── S2_A1Practica2_.odt (esté Archivo <reporte de la practica>)
```

1 directory, 6 files

Anexo

Descargar Archivo o proyecto de la practica 2:

Guardare la practica en una carpeta llamada **Practica2**. Dependiendo del numero de prácticas.

En este caso la guardare en practica2. En el repositorio de github el cual contiene todas las practicas que realizare en todo el semestre:

<https://github.com/pepe1603/repo-BigData.git>

Referencias

Tutorial de prácticas:

- ✓ <https://realpython.com/python-data-cleaning-numpy-pandas/>
- ✓ <https://www.analyticsvidhya.com/blog/2021/06/data-cleaning-using-pandas/>
- ✓ <https://datagy.io/pandas-data-cleaning/>
- ✓ <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html>
- ✓ <https://python.plainenglish.io/importing-csv-data-into-postgresql-using-python-aee6b5b11816>