

DESCRIPCIÓN

Esta labor consiste en la elaboración de un informe de Power BI, aprovechando las capacidades analíticas de Python. Se utilizarán los scripts de Python creados previamente en la Tarea I para generar visualizaciones personalizadas con las bibliotecas Seaborn y Matplotlib. Estas visualizaciones estarán integradas en el informe de Power BI para ofrecer una comprensión más profunda de la capacidad del lenguaje de programación en la herramienta Power BI.

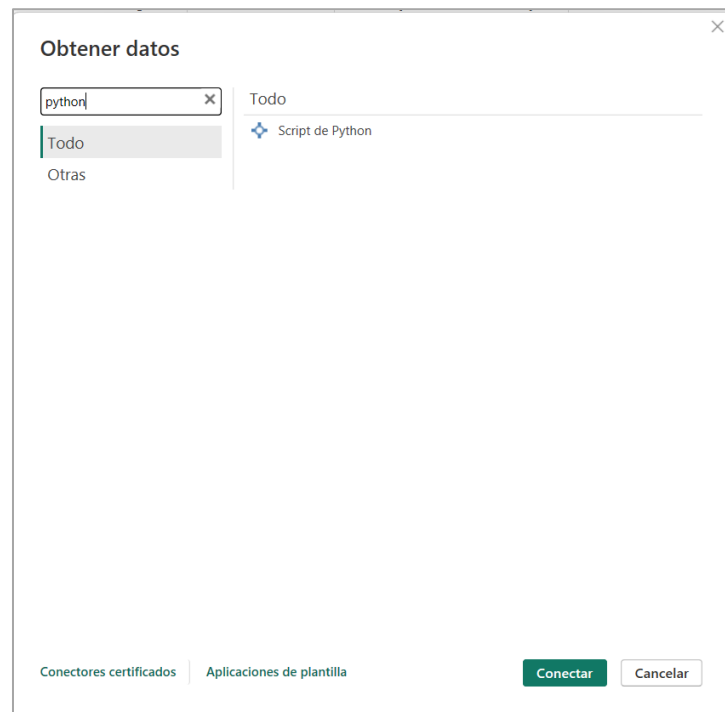
NIVEL I

LOS 7 EJERCICIOS DEL NIVEL I DE LA TAREA 01

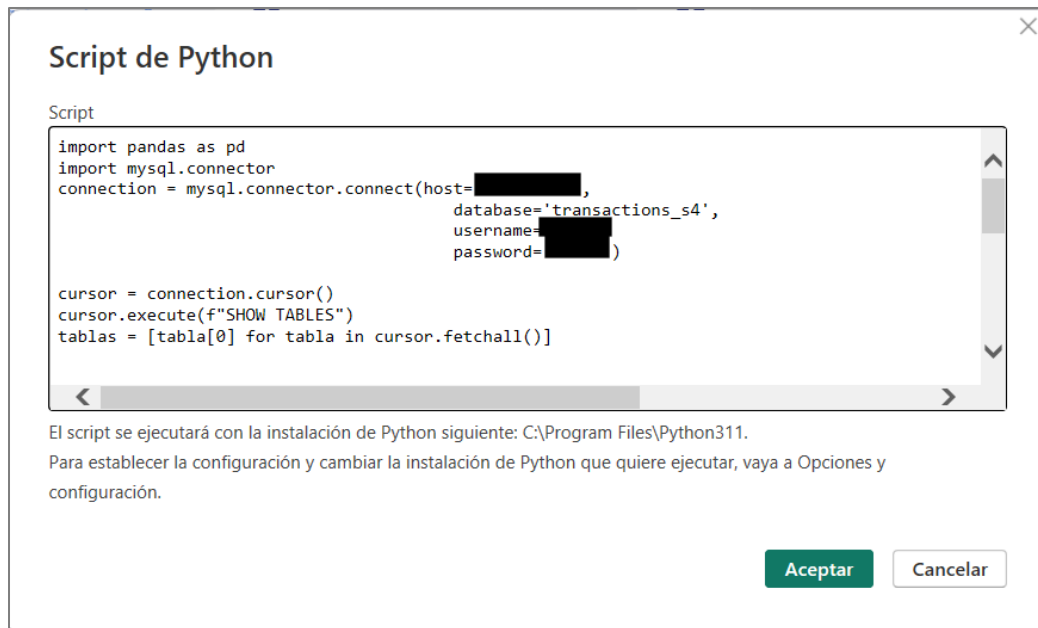
Luego de haber realizado el ejercicio 8.1, donde utilicé Python para generar visualizaciones con las librerías Seaborn y Matplotlib, ahora corresponde trasladar esos scripts a Power BI. Estas visualizaciones me permitieron analizar y representar los datos de manera efectiva dentro del entorno de Jupyter Notebook.

Para ello, primero tuve que configurar Power BI para que fuera compatible con Visual Studio Code. En mi caso, debido a problemas en la configuración inicial, tuve que desinstalar tanto Visual Studio Code como Power BI y luego reinstalarlos en mi equipo. Posteriormente, también fue necesario instalar nuevamente Python, asegurándome de descargar una versión compatible con Power BI, en este caso, la versión 3.11.9.

Una vez completada la configuración, pasé a la fase de carga de los DataFrames. Para ello, en el módulo "Inicio", seleccioné la opción "Obtener datos", busqué "Python" y elegí la opción "Script de Python".



Luego se abrió una nueva ventana que me permitió cargar el script de python:



En este punto cabe aclarar que tomé como base el script trabajado en el ejercicio 8.1, pero realicé modificaciones para garantizar su correcta ejecución sin errores dentro del entorno de Power BI. Estas modificaciones incluyeron ajustes en la conexión a MySQL, en la transformación de los datos y en el manejo de tipos de datos, asegurando compatibilidad con Power BI.

Por lo tanto, el script final cargado en Power BI es el siguiente:

```
import pandas as pd
import mysql.connector
connection = mysql.connector.connect(host=[redacted],
                                     database='transactions_s4',
                                     username=[redacted],
                                     password=[redacted])

cursor = connection.cursor()
cursor.execute(f'SHOW TABLES')
tablas = [tabla[0] for tabla in cursor.fetchall()]

for tabla in tablas:
    cursor.execute(f'SELECT * FROM {tabla}')
    columnas = [columna[0] for columna in cursor.description]
    globals()[f'df_{tabla}'] = pd.DataFrame(cursor.fetchall(), columns=columnas)

cursor.close()
connection.close()

df_products = df_products.astype({'price': object, 'weight': object})
df_transactions = df_transactions.astype({'amount': object, 'declined': bool})

df_transactions['amount'] = df_transactions['amount'].astype(str).str.replace('.', ',')
df_products['price'] = df_products['price'].astype(str).str.replace('.', ',')
df_products['weight'] = df_products['weight'].astype(str).str.replace('.', ',')

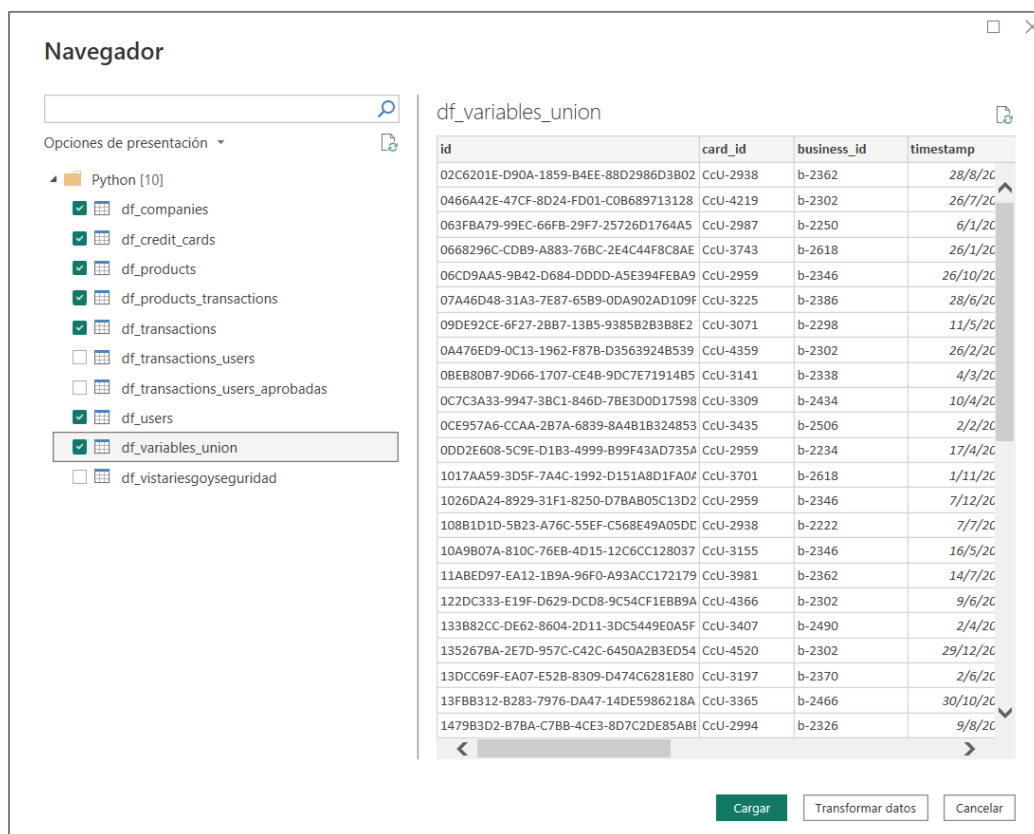
df_users.rename(columns={'id': 'user_id'}, inplace=True)
df_transactions_users = df_transactions.merge(df_users, on='user_id', how='left')
df_transactions_users_aprobadas = df_transactions_users[df_transactions_users['declined'] == False]
df_transactions_users_aprobadas = df_transactions_users_aprobadas.groupby("user_id")["user_id"].count().reset_index(
    name='cant_transactions')
df_variables_union = df_transactions.merge(df_transactions_users_aprobadas, left_on='user_id', right_on='user_id', how='left')
```

Este código permite la carga y transformación de datos desde la base de datos MySQL en Power BI mediante Python.

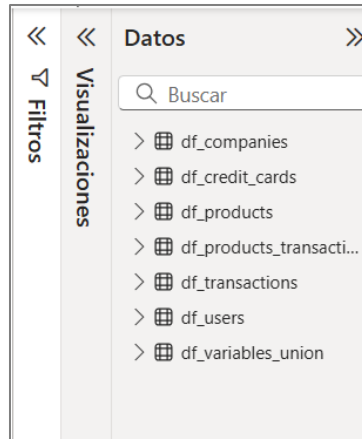
En primer lugar, establece una conexión con la base de datos transactions_s4, extrae todas las tablas disponibles y las convierte en DataFrames de Pandas para su manipulación. Posteriormente, realiza ajustes en los tipos de datos de df_products y df_transactions, convirtiendo ciertos valores numéricos a formato de texto y reemplazando el punto decimal por una coma para garantizar la compatibilidad con Power BI.

Adicionalmente, dado que en el ejercicio 8.1 generé nuevos DataFrames para el análisis, seleccioné aquellos que serían relevantes para las visualizaciones en Power BI. Además de los DataFrames base obtenidos desde MySQL, incorporé df_variables_union, asegurando que sus transformaciones y cálculos previos estuvieran correctamente implementados.

Al ejecutar este código en Power BI, se despliega una ventana de selección de datos, donde elegí los DataFrames necesarios para la creación de las visualizaciones para los siguientes tres niveles:



A continuación, se muestran los DataFrames seleccionados, correctamente cargados en Power BI y pongo a modo de ejemplo el df_companies (de la misma manera se encuentran cargados los demás DataFrame):

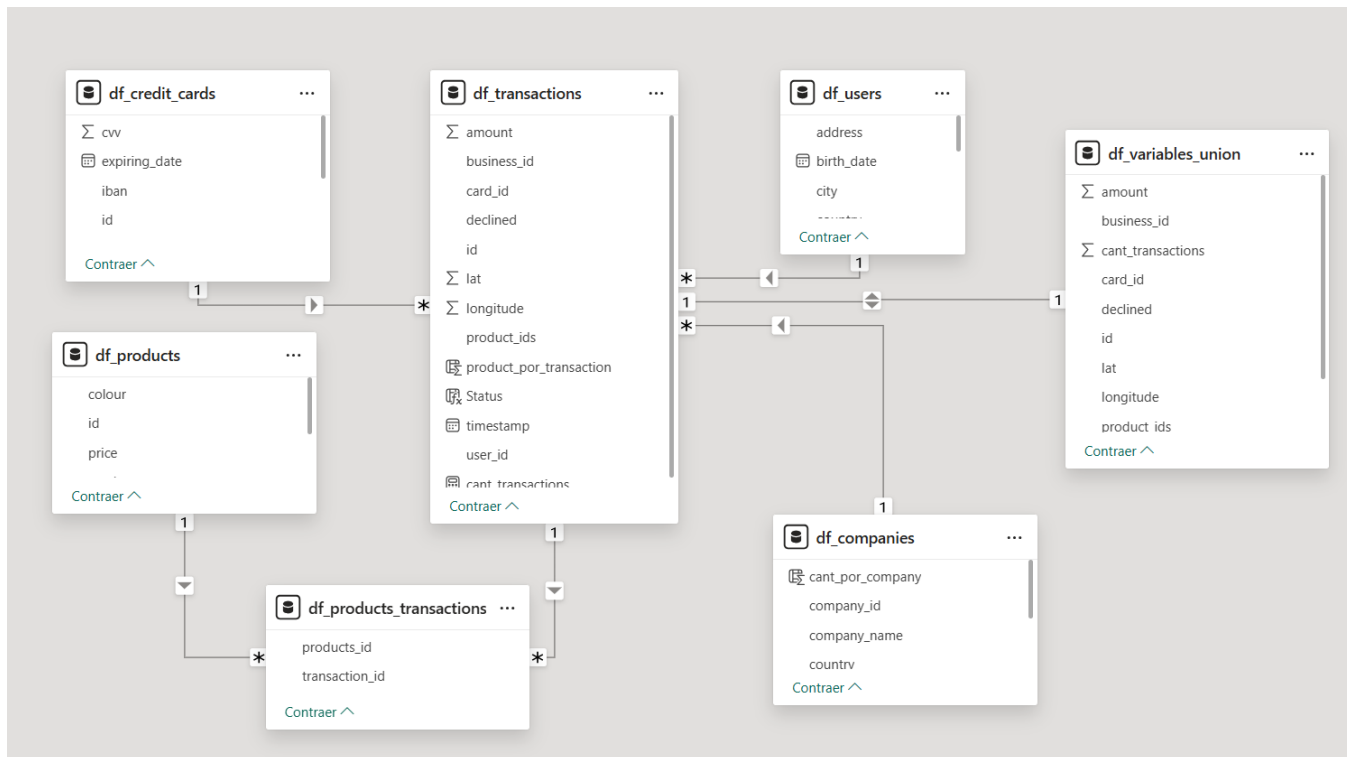


company_id	company_name	phone	email	country	website	cant_por_company
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site	8
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9	6
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars	9
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110	8
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings	6
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110	7
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one	9
b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars	9
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States	https://ebay.com/sub	9
b-2258	Vestibulum Lorem PC	02 02 87 33 40	aenean.massai.integer@aol.net	Belgium	https://pinterest.com/sub/cars	8
b-2262	Gravida Sagittis LLP	03 81 28 33 97	turpis.vitae@google.ca	Sweden	https://naver.com/site	11
b-2266	Mus Aenean Eget Foundation	06 25 15 52 43	mi.duis@hotmail.net	Sweden	https://instagram.com/group/9	11
b-2270	Dis Parturient Institute	05 36 29 78 74	purus@protonmail.org	Ireland	https://google.com/one	6
b-2274	Sed LLC	01 63 16 26 52	at@outlook.com	Belgium	https://reddit.com/fr	8
b-2278	Arcu LLP	06 46 04 41 45	dui@aol.ca	Norway	https://yahoo.com/sub	7
b-2282	Pretium Neque Corp.	07 77 48 55 28	eleifend.nec.malesuada@protonmail.couk	Australia	https://netflix.com/sub	6
b-2286	Fringilla LLC	08 29 15 93 57	gravida@protonmail.couk	New Zealand	https://reddit.com/user/110	6
b-2290	Quisque Libero LLC	01 45 48 71 11	sapien.molestie.orci@hotmail.couk	China	https://baidu.com/group/9	2
b-2294	Auctor Mauris Vel LLP	08 09 28 74 14	nec.tempus@icloud.couk	United States	https://instagram.com/fr	9
b-2298	Elit Etiam Laoreet Associates	07 69 74 17 45	ultrices@google.couk	Canada	https://yahoo.com/fr	5
b-2302	Nunc Interdum Incorporated	05 18 15 48 13	non@outlook.com	Germany	https://wikipedia.org/en-us	8
b-2306	Augue Foundation	06 88 43 15 63	mauris@yahoo.com	Germany	https://baidu.com/sub/cars	8
b-2310	Non Magna LLC	06 71 73 13 17	nisl.quisque.fringilla@hotmail.ca	United Kingdom	https://whatsapp.com/site	9
b-2314	A Institute	03 34 91 68 65	metus.aliquam@google.edu	Belgium	https://reddit.com/fr	8
b-2318	Quam A Felis Industries	04 87 10 27 16	proin.velit@icloud.edu	Italy	https://ebay.com/settings	9
b-2322	Integer Mollis Corp.	03 12 20 45 24	eu.eros@protonmail.ca	Italy	https://netflix.com/group/9	9
b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://cnn.com/group/9	9
b-2330	Donec Fringilla PC	01 51 58 14 44	ut.tincidunt@hotmail.ca	France	https://google.com/fr	3
b-2334	Amet Institute	06 33 40 21 33	nullam.lobortis.quam@outlook.net	Australia	https://nytimes.com/one	6
b-2338	Magna Incorporated	07 73 69 44 18	non@google.net	New Zealand	https://twitter.com/en-us	6
b-2342	Fringilla Porttitor Incorporated	06 30 30 97 81	ultrices.a.auctor@protonmail.couk	Sweden	https://wikipedia.org/settings	11
b-2346	Ut Semper Foundation	01 60 36 33 06	consectetuer@google.ca	Sweden	https://ebay.com/settings	11

Como fase final antes de comenzar con los ejercicios, realicé ajustes en Power BI para modificar el tipo de dato de algunas columnas de los DataFrames cargados. Por ejemplo:

- **df_transactions** → convertí user_id de texto a número.
- **df_transactions** → cambié amount a formato de moneda y le añadí el símbolo de euros (€)

También tuve que configurar las relaciones entre los DataFrames al importarlos desde Python a Power BI, ya que no estaban definidas. Como resultado, el diagrama quedó estructurado de la siguiente manera:



Una vez completados estos pasos, comencé con la realización de los ejercicios correspondientes a los tres niveles.

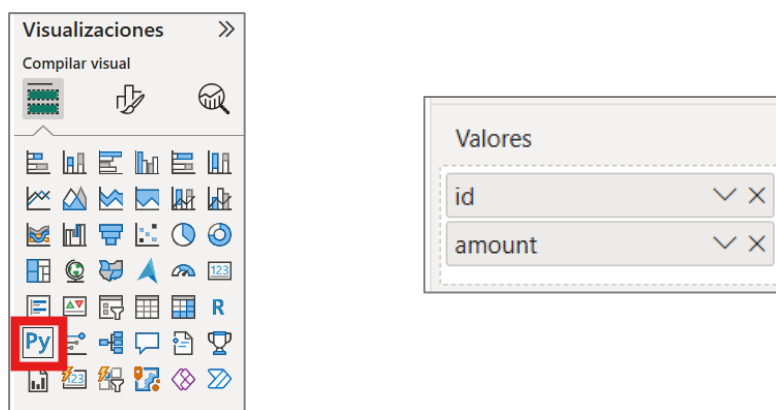
EJERCICIO I

Una variable numérica.

Para realizar este ejercicio, seleccioné en el panel "Visualización" el ícono de PY, que corresponde al Objeto Visual de Python. Luego, en la sección de valores, agregué los datos necesarios para generar el gráfico, en este caso, los campos `id` y `amount` del DataFrame `df_transactions`.

A continuación, en el Editor de scripts de Python, ingresé el código utilizado en el ejercicio 8.1 para este análisis. Es importante destacar que, en Power BI, es necesario reemplazar el nombre del DataFrame por "dataset" en todas sus referencias dentro del código.

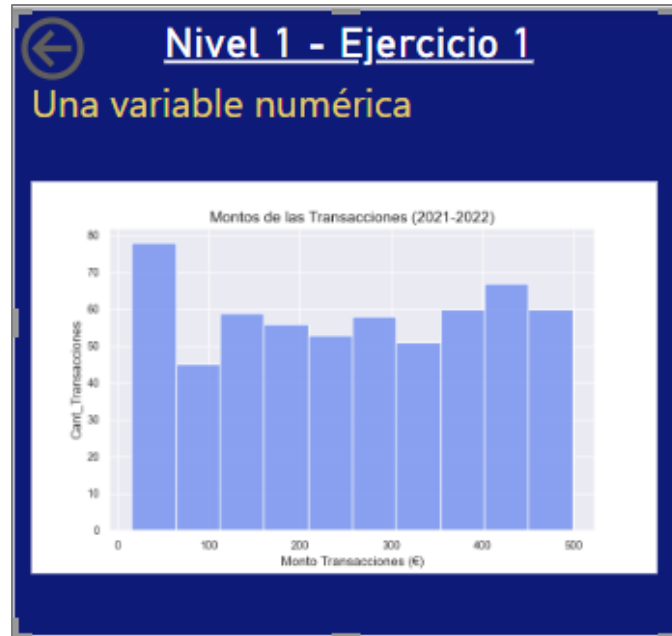
Finalmente, ejecuté el script y obtuve la misma gráfica generada previamente en Visual Studio Code con Python.



```
Editor de scripts de Python

⚠ Las filas duplicadas se quitarán de los datos.

1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 plt.figure(figsize=(8, 5))
5 sns.set_theme(style='darkgrid', palette='coolwarm')
6 ax = sns.histplot(data=dataset, x="amount", bins=10)
7
8 plt.title('Montos de las Transacciones (2021-2022)', size=14)
9 plt.xlabel('Monto Transacciones (€)', size=12)
10 plt.ylabel('Cant_Transacciones', size=12)
11 plt.tick_params(axis='both', labelsize=10)
12
13 plt.show()
14 |
```



Aplicé este mismo procedimiento en los demás ejercicios de los tres niveles. En algunos casos, debido a que no importé todos los DataFrames creados en el ejercicio 8.1, fue necesario implementar medidas adicionales o agregar columnas calculadas en Power BI para replicar correctamente las visualizaciones. Asimismo, en ciertos escenarios, tuve que incluir ajustes específicos en el “Editor de scripts de Python” dentro de Power BI para garantizar que los gráficos se generaran de manera precisa. Sin embargo, estos casos fueron excepcionales y la mayoría de las visualizaciones pudieron reproducirse sin inconvenientes.

VISUALIZACIONES DE LOS EJERCICIOS 1 AL 7

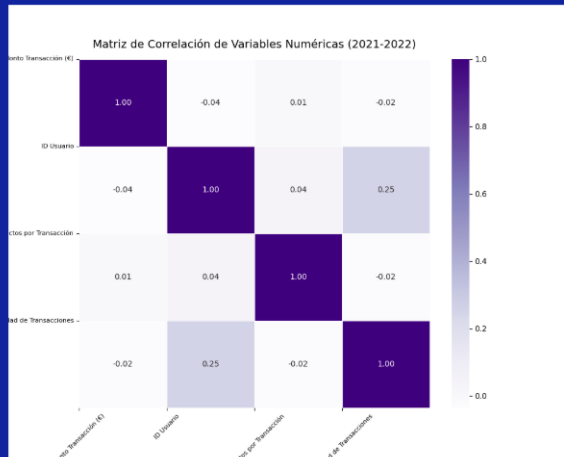


NIVEL 2

VISUALIZACIONES DE LOS EJERCICIOS 1 Y 2

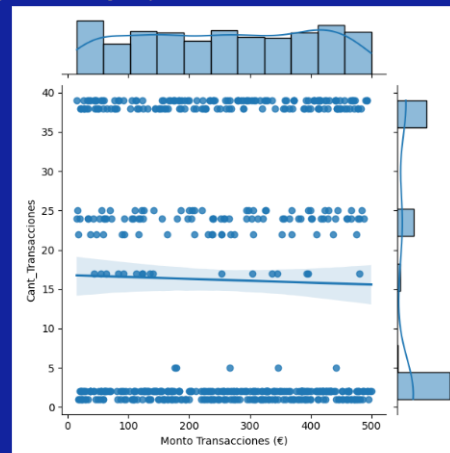
Nivel 2 - Ejercicio 1

Correlación de todas las variables numéricas.



Nivel 2 - Ejercicio 2

Implementa un jointplot.



NIVEL 3

VISUALIZACIONES DE LOS EJERCICIOS 1 Y 2

Nivel 3 - Ejercicio 1

Implementa un violinplot combinado con otro tipo de gráfico.



Nivel 3 - Ejercicio 2

Genera un FacetGrid para visualizar múltiples aspectos de datos simultáneamente.

