

Tarea: recrea las visualizaciones (gráficos / diagramas) del nivel 1 a Power BI usando scripts de Python.

RESUMEN TAREA:

1. Carga de datos directamente a Power BI con script Python.
2. Transformación a Power Query: INT→FLOAT, configuración regional OS por decimales.
3. Modelado de datos: crear relaciones
4. Por cada tipo de visualización:
 - Arrastrar columnas → Power BI crea dataset automático
 - ``df = dataset.copy()``
 - Eliminar merges (Power BI ya une tablas)
 - Si hay alguna visualización con Plotly, transformar el código a matplotlib/seaborn

Asimilado en Nivel 3 (tarea paso a paso):

Debes preparar el código Python para adaptarse a los requerimientos para ejecutar el Script en Power BI.

-Fuente de datos:

OBTENER DATOS – SCRIPT DE PYTHON – Copiar código para importar datos, por ejemplo, para establecer una conexión con MYSQL:

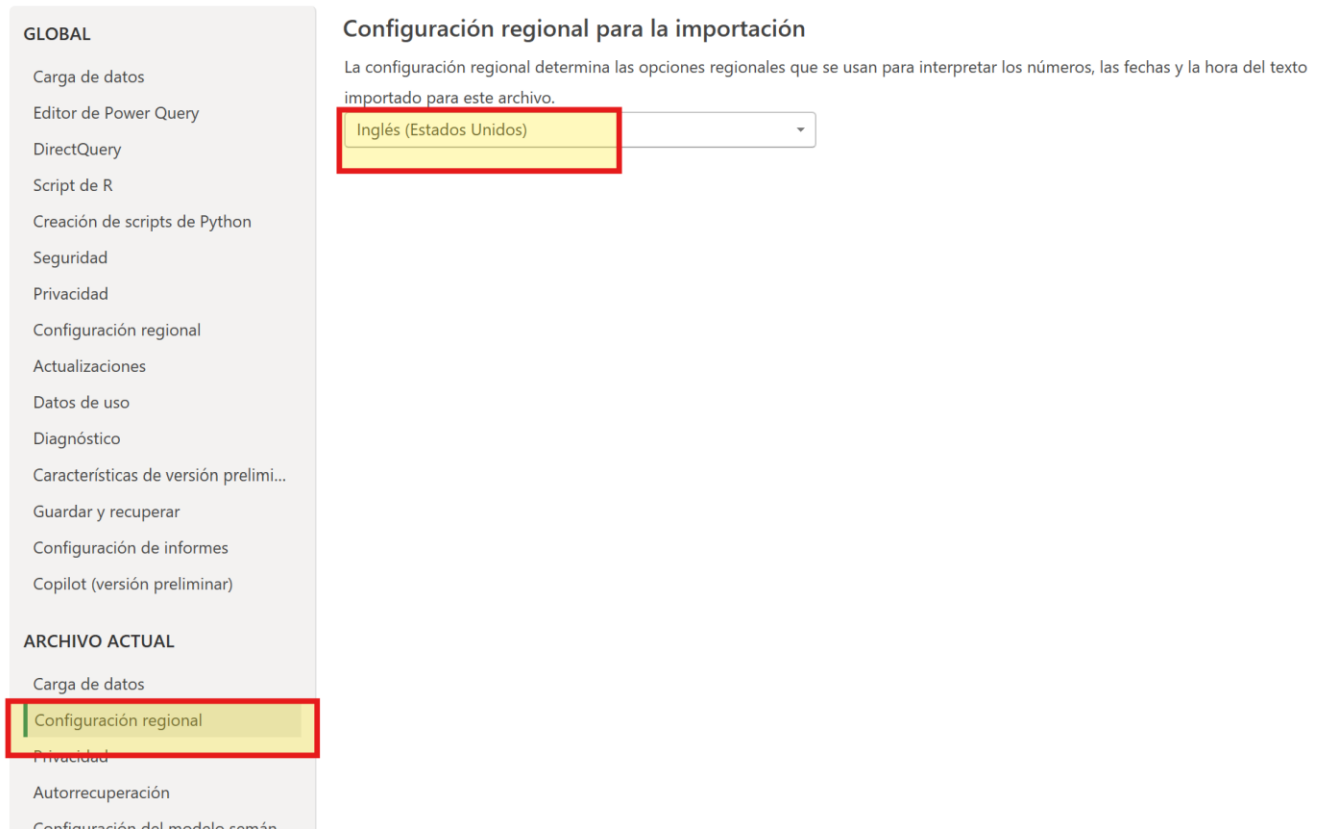
```
conn = mysql.connector.connect(  
    host="localhost",  
    user="XXXX",  
    password="XXXXXX",  
    database="transactions_s4")  
df_transactions = pd.read_sql("SELECT * FROM `transaction`", conn)  
df_users = pd.read_sql("SELECT * FROM users", conn)  
df_cards = pd.read_sql("SELECT * FROM credit_cards", conn)  
df_companies = pd.read_sql("SELECT * FROM companies", conn)  
df_transaction_product = pd.read_sql("SELECT * FROM transaction_product", conn)  
df_products = pd.read_sql("SELECT * FROM products", conn)  
df_targetes_actives = pd.read_sql("SELECT * FROM targetes_actives", conn)  
conn.close()
```

Al cargar los datos, antes transformar datos en POWER QUERY para asegurarse que los data type corresponden con los originales.

En el ejercicio de este repositorio, todas las columnas numéricas fueron transformadas a INTEGER, por lo que hubo que cambiarlas a FLOAT.

Además, para **mantener el uso del PUNTO como separador decimal** en los datos originales, hubo que cambiar la configuración en **OPCIONES - ARCHIVO ACTUAL – CONFIGURACIÓN REGIONAL** para tener inglés (ESTADOS UNIDOS) y que así mantuviera los floats según los originales al pasar de INT a FLOAT en POWER QUERY.

Opciones



GLOBAL

- Carga de datos
- Editor de Power Query
- DirectQuery
- Script de R
- Creación de scripts de Python
- Seguridad
- Privacidad
- Configuración regional
- Actualizaciones
- Datos de uso
- Diagnóstico
- Características de versión prelimi...
- Guardar y recuperar
- Configuración de informes
- Copilot (versión preliminar)

ARCHIVO ACTUAL

- Carga de datos
- Configuración regional
- Privacidad
- Autorrecuperación
- Configuración del modelo semán

Configuración regional para la importación

La configuración regional determina las opciones regionales que se usan para interpretar los números, las fechas y la hora del texto importado para este archivo.

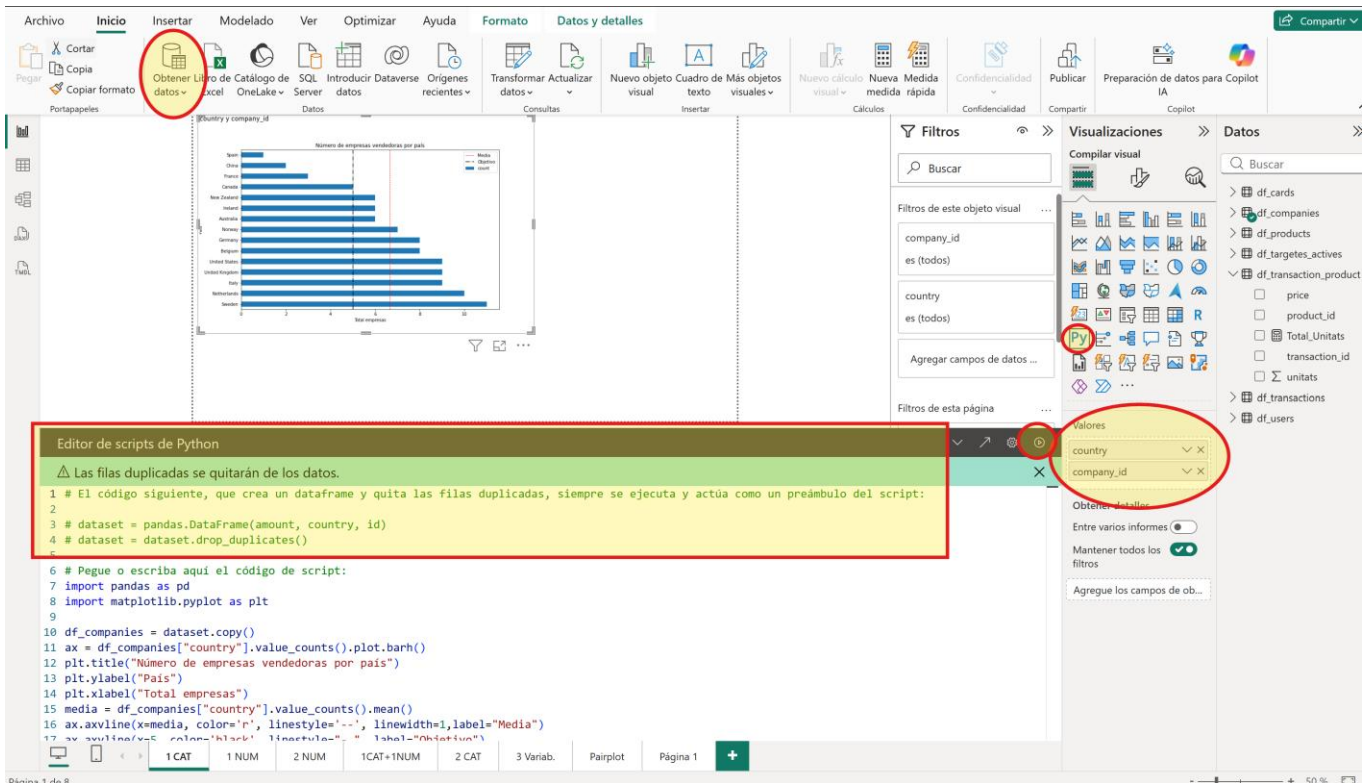
Inglés (Estados Unidos)

A continuación, hay que **revisar las relaciones entre las tablas** – dataframes importados. Además, modificar en cada relación el DIRECCIÓN DE FILTRO CRUZADO para que sea en AMBAS direcciones.

Ahora, en “visualizaciones” **elegir “PY”**.

Arrastrar al apartado VALORES las columnas o métricas de datos que nos interesan para que POWER BI cree un DATASET propio de esa visualización y haga automáticamente los merge/join correspondientes.

Copiar el código de Python adaptado y ejecutar run en el script.



Editor de scripts de Python

⚠ Las filas duplicadas se quitarán de los datos.

1 # El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta y actúa como un preámbulo del script:

2

3 # dataset = pandas.DataFrame(amount, country, id)

4 # dataset = dataset.drop_duplicates()

5

6 # Pegue o escriba aquí el código de script:

7 import pandas as pd

8 import matplotlib.pyplot as plt

9

10 df_companies = dataset.copy()

11 ax = df_companies["country"].value_counts().plot.barh()

12 plt.title("Número de empresas vendedoras por país")

13 plt.ylabel("País")

14 plt.xlabel("Total empresas")

15 media = df_companies["country"].value_counts().mean()

16 ax.axvline(x=media, color="r", linestyle="--", linewidth=1, label="Media")

17 ax.axvline(x=5, color="black", linestyle="solid", label="Objetivo")

Adaptaciones de código que hay que tener en cuenta para usarlo en POWER BI.

dataset es la variable automática que **Power BI** crea con todos los datos cargados (unión de tablas).

El nombre/s de dataframe del código no existe a priori en la ejecución de POWER BI. Para poder aprovecharlos sin cambiarlos, por comodidad, hay que añadir:

nom_df_original = dataset.copy()

- El .copy() es opcional, pero asegura que al ejecutar cambios, no modifiquen los datos originales.

- Los merge de df ("joins") hay que eliminarlos, ya que Power Bi crea un dataset automáticamente uniendo las tablas de origen de los distintos valores arrastrados.

POWER BI soporta .plot de Pandas, matplotlib y Seaborn, pero no Plotly. Hay que pasar los gráficos de Plotly a uno de los soportados o ejecutar y se abrirá una ventana externa.