

Transformación de Datos y Preparación para Análisis en Power BI y Python - Isabel Mañero

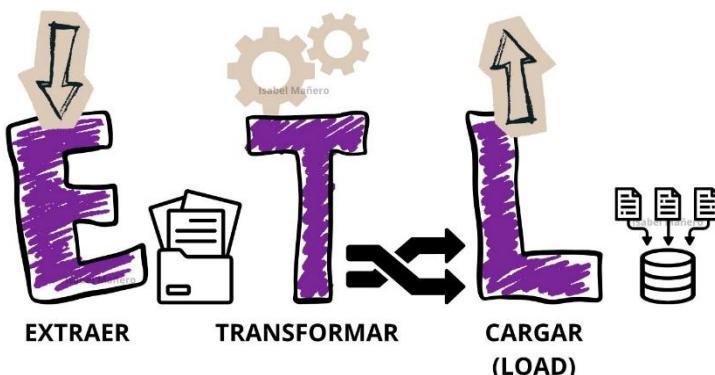
Proceso de transformación con PowerBI

El objetivo de este informe es mostrar cómo, mediante un **flujo de trabajo automatizado** en Power BI, es posible convertir datos en bruto en una estructura ordenada y preparada para el análisis, facilitando la creación de visualizaciones, dashboards y métricas avanzadas que apoyen una toma de decisiones más eficiente y fundamentada.

Partimos de un **archivo Excel (.xlsx)** con información ficticia sobre deportistas y eventos deportivos por año (cada hoja es un año diferente), que requiere transformaciones para ser analizable. Mediante **Power Query y/o Python**, se limpian y reestructuran los datos, unificando la información histórica y automatizando procesos para incorporar futuros archivos con el mismo formato. Gracias a esto, se obtiene una tabla organizada y lista para generar análisis temporales, visualizaciones y medidas DAX, optimizando el tiempo y mejorando la calidad de los resultados obtenidos.

TIPO DE DEPORTE													
DEPORTES		Fútbol		Tenis		Baloncesto		Natación		Hockey		Submarinismo	
Por grandes zonas	Por pequeñas zonas	Nº de Deportistas	Eventos										
ESPAÑA	ANDALUCÍA	240	7	120	61	205	3	200	21	50	2	70	4
	MADRID	400	12	260	34	400	56	150	14	500	12	15	21
	EXTREMADURA	205	14	56	68	167	22	32	15			6	4
	ASTURIAS	78	11	30	64			12	5			80	7
	CANTABRIA	190	2	30	12	23	204	7	25			50	2
Total		923	44	466	227	772	81	394	55	550	14	171	36
INTERNACIONAL	PORTUGAL	200	12	306	4	306	23	306	5	306	4	223	85
	FRANCIA	45	2	277	19	277	14	277	4			38	11
	ITALIA	260	8	101	11	200	7	101	5	101	5	113	21
	REINO UNIDO	157	9	13	23	13	9	13	9	13	7		
	Total	662	31	697	57	796	53	697	23	420	16	374	117
Total general		1.585	75	1.163	284	1.568	134	1.091	78	970	30	545	153

Este tipo de tablas presentan varias limitaciones para el análisis directo, ya que los datos no están normalizados ni estructurados de forma óptima para obtener **insights significativos**. Para poder analizarlos de manera eficiente, es necesario aplicar transformaciones que permitan organizar la información, un proceso central en las **ETL** (**Extract, Transform, Load**).



Vista general PowerBI - Editor/Transformaciones

The screenshot shows the Power BI Editor interface with the following elements:

- Left pane:** Shows two datasets: "2012" and "2013". A callout points to the "2012" dataset with the text "Hojas cargadas del archivo .xlsx".
- Main area:** Displays a table titled "Origen[[Item='2012',Kind='Sheet']]Data". The table has columns: Column1, Column2, Column3, Column4, Column5, Column6, and Column7. The data includes rows for various categories like DEPORTES, ESPAÑA, and CANTABRIA, along with a "Total general" row.
- Right pane:** Contains the "Configuración de la consulta" (Query Configuration) window. It shows the "Nombre" (Name) as "2012" and the "PASOS APLICADOS" (Applied Steps) section. The "Navegación" step is highlighted with a callout. Other steps listed include "Encabezados promovidos", "Encabezados promovidos1", "Encabezados promovidos2", "Tipo cambiado1", "Columnas con nombre cambiado", "Rellenar hacia abajo", "Filas filtradas", "Otras columnas con anulación", "Dividir columna por delimitador", "Tipo cambiado2", "Columna dinamizada", "Personalizada agregada", and "Consulta anexada".
- Bottom right:** A callout points to the "Otras columnas con anulación" step in the applied steps list with the text "Estructura inicial del dataset".
- Bottom right:** Another callout points to the "Otras columnas con anulación" step in the applied steps list with the text "Transformaciones que se aplican al dataset".

En este contexto, Power BI ofrece herramientas muy poderosas:

- **Power Query (lenguaje M):** permite realizar transformaciones en los datos antes de cargarlos en el modelo. Con Power Query podemos:
 - Limpiar datos (eliminar duplicados, corregir errores, homogeneizar formatos, elegir el tipo de dato (str,fecha,int...)).
 - Reestructurar tablas (unir hojas de distintos años, pivotar o despivotar columnas).
 - **Automatizar** procesos mediante scripts en el **Editor Avanzado**, de manera que cualquier nuevo archivo que cumpla con el mismo formato pueda incorporarse automáticamente al análisis.

Vista general PowerBI - Editor avanzado

The screenshot shows the Power BI Advanced Editor interface with the following elements:

- Left pane:** Shows the same two datasets: "2012" and "2013".
- Main area:** Displays the "2012" dataset table with the same structure as before.
- Right pane:** Contains the "Configuración de la consulta" (Query Configuration) window. The "Nombre" (Name) is set to "2012". The "PASOS APLICADOS" (Applied Steps) section is expanded, showing the "Navegación" step highlighted with a callout. Other steps listed are identical to the previous screenshot.
- Bottom left:** A callout points to the "Otras columnas con anulación" step in the applied steps list with the text "En el Editor avanzado, encontramos la parte interna de esas transformaciones. Podemos comentar el código para reutilizarlo y hacerlo más legible." A large callout arrow points from this text towards the bottom of the Power Query script.
- Bottom center:** The Power Query script itself is shown in the "Script" pane:


```

let
    // 1. Cargar archivo Excel
    Origen = Excel.Workbook(File.Contents("C:\Users\isau9\OneDrive\Escritorio\Deportes.xlsx")),
    // 2. Seleccionar la hoja 2012
    "#2012_Sheet" = Origen[[Item="2012",Kind="Sheet"]][Data],
    // 3. Promover encabezados
    #>[#"Encabezados promovidos" = Table.PromoteHeaders("#2012_Sheet", [PromoteAllScalars=True])],
    #>[#"Encabezados promovidos1" = Table.PromoteHeaders("#Encabezados promovidos", [PromoteAllScalars=True])],
    #>[#"Encabezados promovidos2" = Table.PromoteHeaders("#Encabezados promovidos1", [PromoteAllScalars=True])],
    // 4. Cambiar tipos de columnas iniciales (ajustar según la tabla real)
    #>[#"Tipo cambiado1" = Table.TransformColumnTypes(#"Encabezados promovidos2",{
        {"Por grandes zonas", type text}, {"("Eventos",",", type text),
        {"("Nº de Deportistas",",", Int64.Type), {"Eventos", Int64.Type},
        {"("Nº de Deportistas_1",",", Int64.Type), {"Eventos_1", Int64.Type},
        {"("Nº de Deportistas_2",",", Int64.Type), {"Eventos_2", Int64.Type},
        {"("Nº de Deportistas_3",",", Int64.Type), {"Eventos_3", Int64.Type},
        {"("Nº de Deportistas_4",",", Int64.Type), {"Eventos_4", Int64.Type},
        {"("Nº de Deportistas_5",",", Int64.Type), {"Eventos_5", Int64.Type},
        {"("Nº de Deportistas_6",",", Int64.Type), {"Eventos_6", Int64.Type},
        {"("Nº de Deportistas_7",",", Int64.Type), {"Eventos_7", Int64.Type},
        {"("Nº de Deportistas_8",",", Int64.Type), {"Eventos_8", Int64.Type},
        {"("Nº de Deportistas_9",",", Int64.Type), {"Eventos_9", Int64.Type},
        {"("Nº de Deportistas_10",",", Int64.Type), {"Eventos_10", Int64.Type}
    })],
    // 5. Renombrar columnas de deportes
    #>[#"Columnas con nombre cambiado" = Table.RenameColumns(#"Tipo cambiado1", {
        {"Eventos", "Nº de Deportistas"}, {"Eventos_1", "Nº de Deportistas_1"}, {"Eventos_2", "Nº de Deportistas_2"}, {"Eventos_3", "Nº de Deportistas_3"}, {"Eventos_4", "Nº de Deportistas_4"}, {"Eventos_5", "Nº de Deportistas_5"}, {"Eventos_6", "Nº de Deportistas_6"}, {"Eventos_7", "Nº de Deportistas_7"}, {"Eventos_8", "Nº de Deportistas_8"}, {"Eventos_9", "Nº de Deportistas_9"}, {"Eventos_10", "Nº de Deportistas_10"}]
in
    #>[#"Columnas con nombre cambiado"]

```

Por ejemplo, en este script:

```
// 8. Despivotar columnas de deportes
#"Otras columnas con anulación de dinamización" = Table.UnpivotOtherColumns(
    {"Filas filtradas", {"Por grandes zonas", "Por pequeñas zonas"}, "Atributo", "Valor"},
```

Despivotar columnas se refiere, a que transforma una tabla organizada de forma horizontal en una tabla organizada de forma vertical. En este proceso, las columnas que contienen valores de datos se convierten en filas. Es algo así como “girar” la tabla. Lo que antes era una sola fila con mucha información repartida hacia la derecha, ahora se convierte en **varias filas** (una por cada combinación de deporte/métrica) organizadas hacia abajo.

Gracias a estas herramientas, es posible generar un flujo de trabajo completamente **automatizado** que:

1. Limpia y transforma los datos desde el Excel.
2. Unifica la información histórica por años.
3. Permite realizar **visualizaciones y análisis temporales** de manera eficiente.

Finalmente obtenemos una tabla de este formato (con los datos de 2013 anexados también). Cómo se puede comprobar nuestra tabla tiene un formato ordenado y “analizable”.

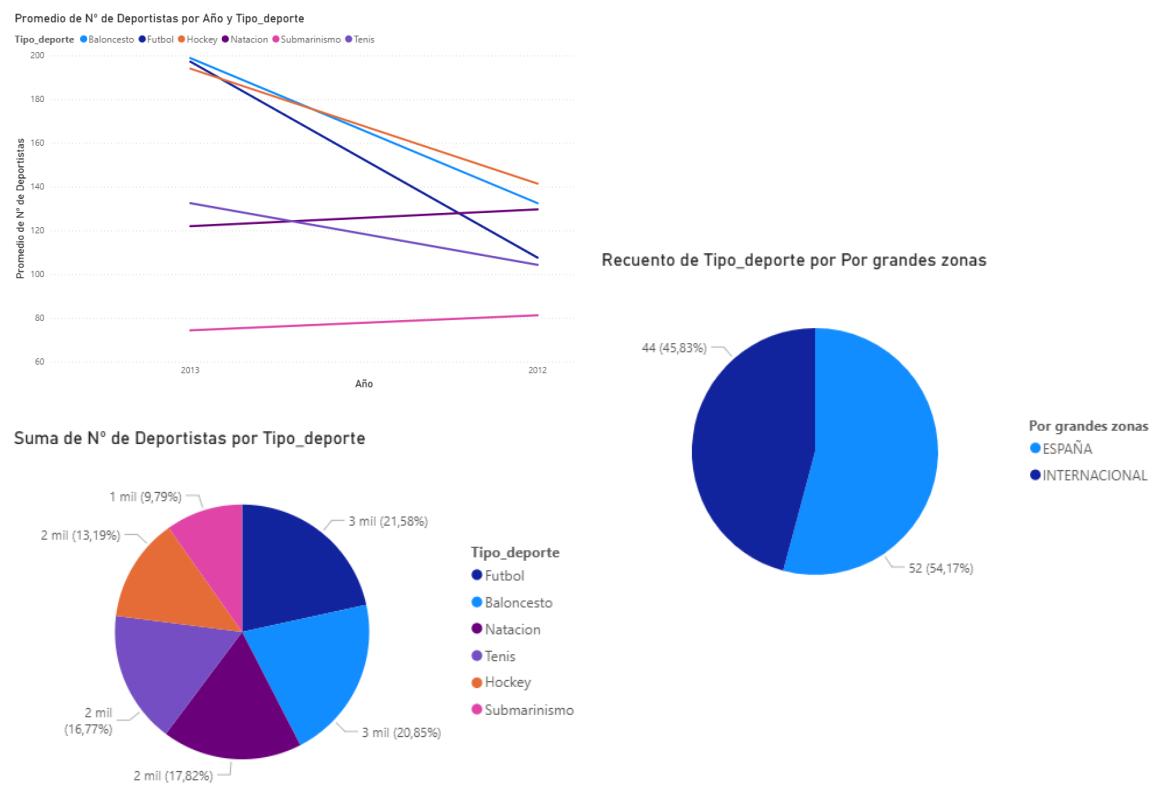
Así, si los datos suelen llegarnos en este formato, es posible automatizar su transformación y hacerla más eficiente, porque simplemente le pasaremos el script creado para el primer año, y el resto será aplicar y anexar.

The screenshot shows the Power BI Data Editor interface. On the left, there's a navigation pane with 'Consultas [2]' and two items: '2012' and '2013'. The main area displays a table titled 'Table.Combine({"#Personalizada agregada", "#2013"})'. The table has columns: 'Nº de Grandes zonas', 'Nº de Pequeñas zonas', 'Tipo_deporte', 'NP de Deportistas', 'Eventos', and 'Año'. The data includes rows for various regions like ANDALUCÍA, ASTURIAS, CANTABRIA, EXTREMADURA, and MADRID, across sports like Baloncesto, Fútbol, Hockey, Natación, and Submarinismo, with counts ranging from 5 to 200. On the right, there's a 'Configuración de la consulta' pane with sections for 'PROPIEDADES' (Nombre: 2012) and 'PASOS APLICADOS'. The 'PASOS APLICADOS' section lists various steps like 'Origen', 'Navegación', and 'Personalizada agregada', with some being checked.

Después de las transformaciones, podemos desde el panel de análisis de PowerBI, realizar gráficas y crear dashboards valiosos, y si nuestros datos nos lo permiten, crear medidas DAX para resultados más completos.

De este modo, Power BI no solo facilita la creación de dashboards, sino que asegura que los datos estén siempre listos para el análisis, optimizando el tiempo y mejorando la calidad de las decisiones basadas en información mediante la **automatización** de todo el proceso.

Ejemplos de gráficos rápidos en PowerBI:



Proceso de transformación con Python

Adicionalmente, este mismo proceso de transformación puede realizarse **mediante Python**, creando funciones reutilizables e integrándolas dentro de un flujo **ETL automatizado** previo a la carga en Power BI u otra herramienta de visualización.

El uso de Python aporta ventajas adicionales como mayor escalabilidad, versionado del código, pruebas automatizadas y facilidad de integración con otros sistemas, complementando perfectamente las capacidades de Power Query. De este modo, Power BI puede centrarse en la capa de análisis y visualización, mientras que Python se encarga de la preparación avanzada de los datos dentro de un proceso ETL robusto y reproducible.

Utilizando la librería pandas, como se puede ver en el siguiente ejemplo:

Se ha creado esta función:

```
import pandas as pd

def transformar_deportes_excel(ruta_excel: str, hoja: str, anio: int) -> pd.DataFrame:
    # 1. Cargamos el Excel sin procesar nada todavía [cite: 104]
    df = pd.read_excel(ruta_excel, sheet_name=hoja, header=None)

    # 2. LIMPIEZA DE CABECERAS (El truco maestro)
    # En la fila 0 están los nombres de los deportes
    # Usamos ffill() para que el nombre del deporte se repita en la columna de 'Eventos'
    #Rellenar hacia abajo
    deportes = df.iloc[0].ffill()

    # En la fila 2 están las métricas: "N de Deportistas", "Eventos"
    metricas = df.iloc[2]

    # 3. UNIÓN
    # Solo unimos si es un deporte real. Si es una zona, la dejamos igual.
    nuevos_nombres = []
    for dep, met in zip(deportes, metricas):
        # Si la columna es de zonas, no le pegamos ningún deporte
        if "zona" in str(met).lower() or pd.isna(dep) or dep == "TIPO DE DEPORTE":
            nuevos_nombres.append(met)
        else:
            # Esto creará algo como "N de Deportistas_Futbol"
            nuevos_nombres.append(f"{met}_{dep}")
```

```

# 4. QUITAR LAS FILAS QUE YA NO SIRVEN
# Quitamos las primeras 3 filas que eran los títulos [cite: 107]
df = df.iloc[3: ].reset_index(drop=True)

# 5. RELLENAR ZONAS (España, Internacional...) [cite: 119]
df['Por grandes zonas'] = df['Por grandes zonas'].ffill()

# 6. QUITAR FILAS DE TOTALES
df = df[~df['Por pequeñas zonas'].str.contains('Total', na=False)]

# 7. DESPIVOTAR (Pasar de horizontal a vertical) [cite: 123]
"""Escalabilidad: Al usar el melt sin listar todos los deportes a mano,
si el archivo pasa de tener 6 deportes a tener 20, tu función seguirá
funcionando sin tocar ni una coma."""

df_unpivot = df.melt(
    id_vars=['Por grandes zonas', 'Por pequeñas zonas'],
    var_name='Atributo',
    value_name='Valor'
)

# 8. Introducimos una nueva función:

# Si la columna es "N de Deportistas_Fitoor", divide por el "_"
def separar_deporte(texto):
    if "_" in str(texto):
        partes = str(texto).split("_")
        return pd.Series([partes[0], partes[1]]) # [Metrica, Deporte]
    return pd.Series([texto, None])

df_unpivot[['metrica', 'Tipo_deporte']] = df_unpivot['Atributo'].apply(separar_deporte)

# 9. PIVOTAR PARA TENER COLUMNAS LIMPIAS
df_final = df_unpivot.pivot_table(
    index=['Por grandes zonas', 'Por pequeñas zonas', 'Tipo_deporte'],
    columns='metrica',
    values='Valor',
    aggfunc='first'
).reset_index()

# 10. AÑADIR EL AÑO
df_final['Año'] = año

return df_final.head()

```

Este proyecto demuestra la capacidad de transformar y preparar datos partiendo de archivos Excel con estructuras no óptimas, aplicando un proceso ETL mediante **Excel, Power Query en Power BI y Python**, con el objetivo de obtener un modelo de datos limpio, automatizado y listo para el análisis. El resultado permite la creación de visualizaciones y medidas DAX, evidenciando el manejo de distintas herramientas y tecnologías clave orientado a la calidad del dato y al análisis eficiente.

Isabel Mañero