Documentación: Automatización de Procesamiento y Conciliación de Datos Financieros.

L David Villegas Pavez

Este script realiza el procesamiento y limpieza de un archivo Excel almacenado en Google Drive, utilizando Python y la biblioteca 'pandas'. Su objetivo principal es filtrar datos relevantes, realizar transformaciones específicas y guardar los resultados en un nuevo archivo.

A continuación, se detalla cada parte del script para facilitar su comprensión.

Objetivos:

I. Filtrado y Normalización de Datos Financieros
- Objetivo Especifico: Limpiar y estandarizar los datos, eliminando registros irrelevantes y duplicados.
- Razón: Optimiza el análisis al asegurar que solo se utilicen datos precisos y relevantes.

II. Comparación de Descripciones Financieras entre Fuentes
- Objetivo Especifico: Comparar transacciones de dos fuentes distintas para identificar coincidencias.
- Razón: Automatiza la conciliación de transacciones, reduciendo el tiempo y errores manuales.

Procesos Script 1: Filtrado y Normalización de Datos Financieros.

1. Montaje de Google Drive.
   Se monta el sistema de archivos de Google Drive para acceder al archivo de entrada y almacenar el resultado procesado.

2. Lectura y Validación.
   Se especifican las rutas de entrada y salida para el archivo a procesar. Utiliza pandas para cargar los datos desde Excel, manejando errores de lectura que puedan interrumpir el flujo. Con ello se puede validar automáticamente la disponibilidad y formato de los datos.

3. Filtrado y Limpieza.
   Se eliminan registros donde la columna Ingresos sea nula o igual a 0, con el propósito de conservar únicamente transacciones financieras realizadas.

4. Análisis de Frecuencia.
   Se genera un conteo de ocurrencias únicas en la columna Detalle para identificar patrones y clasificar datos, con el fin de identificar y corroborar los tipos recurrentes de transacciones.

5. Eliminación de Duplicados.
   Se retienen únicamente las primeras ocurrencias en Detalle, eliminando duplicados. Facilitar análisis únicos para auditorías y reportes.

6. Guardado del Archivo Procesado
   El DataFrame final se guarda en un archivo Excel en Google Drive.

[1]Fecha: 31 de diciembre de 24

Procesos Script 2: Comparación de Descripciones Financieras entre Fuentes.

Automatizar la comparación de descripciones entre dos conjuntos de datos financieros (INFORME 5 y mov fondos) para identificar coincidencias y optimizar el análisis.

1. Montaje de Drive.
   Se utiliza Google Drive como sistema de almacenamiento para acceder a los archivos fuente y guardar el archivo procesado. Este paso asegura que los datos sean fácilmente accesibles desde la nube y centraliza el flujo de trabajo.

2. Definición de Rutas.
   Se especifican las rutas de los archivos de entrada y salida. Esto incluye la ubicación del archivo INFORME 5, el archivo de referencia mov fondos, y el destino para el archivo procesado. La separación clara de rutas permite gestionar múltiples versiones de los archivos sin confusiones.

3. Carga de Archivos.
   Los archivos INFORME 5 y mov fondos se cargan en estructuras de datos (DataFrames) utilizando la biblioteca pandas. Este paso verifica que los datos estén disponibles y listos para ser procesados, manejando errores en caso de archivos faltantes o formatos incompatibles.

4. Normalización de la Columna Detalle.
   Del archivo mov fondos, se extraen los valores únicos de la columna Detalle, eliminando nulos, espacios innecesarios y unificando todo en minúsculas. Este paso es crítico para garantizar que la comparación entre datos sea precisa y libre de inconsistencias causadas por variaciones en el formato.

5. Comparación de Descripciones.
   Se implementa un método para comparar cada entrada de la columna DESCRIPCION en INFORME 5 con los valores normalizados de Detalle en mov fondos. El resultado de esta comparación es binario: SI si hay coincidencia y NO si no la hay. Esto automatiza lo que normalmente sería una tarea manual repetitiva.

6. Generación de la Columna Banco 20.
   Se crea una nueva columna en INFORME 5 llamada Banco 20, que contiene los resultados de la comparación para cada fila. Esta nueva columna permite identificar de forma rápida y clara cuáles transacciones están relacionadas entre las dos fuentes de datos.

7. Exportación de Resultados.
   El archivo procesado se guarda con el nombre especificado en la ruta de salida. Este archivo incluye el INFORME 5 original más la columna Banco 20, lo que facilita su uso en auditorías, conciliaciones o análisis adicionales.

[1]Fecha: 31 de diciembre de 24

Procesos Script 3: Unificación de Archivos Excel.

El propósito del código es procesar varios archivos en formato Excel, buscar un encabezado específico dentro de cada archivo, extraer todas las filas siguientes a dicho encabezado, y combinar los datos en un único archivo Excel de salida.

1. Definición de Archivos.
   Se especifican las rutas de los archivos Excel que contienen los datos a procesar (input_files) y la ubicación del archivo combinado que se generará (output_file).

2. Procesamiento de Archivos.
   Para cada archivo en la lista:
   - Se carga el archivo sin encabezado inicial (header=None).
   - Se identifica la fila que contiene el texto clave "Fecha Venta", que marca el inicio de los datos relevantes.
   - Se establecen las columnas utilizando esta fila como encabezado y se seleccionan las filas posteriores, descartando las anteriores.

3. Combinación de Datos.
   Los datos procesados de cada archivo se almacenan en una lista (combined_data). Al finalizar el recorrido, se combinan en un único DataFrame utilizando la función pd.concat.

4. Exportación de Datos.
   El DataFrame combinado se guarda como un archivo Excel en la ruta especificada por output_file. Si no se encuentran datos válidos en ningún archivo, se notifica al usuario.

[1]Fecha: 31 de diciembre de 24

Anexo – Scripts Mencionados:

Script 1 - Filtro Filas:

```
# Importar librerías
import pandas as pd
from google.colab import drive
import os

# Montar Google Drive
drive.mount('/content/drive', force_remount=True)
# Definir rutas de archivos
archivo_entrada = '/content/drive/MyDrive/Archivos Conciliaciones/mov.fondos dic.23.xlsx'
archivo_salida_excel = '/content/drive/MyDrive/Archivos Conciliaciones/mov.fondos
dic.23_lista.xlsx'

# Procesar archivo
try:
    # Verificar existencia del archivo
    if not os.path.exists(archivo_entrada):
        raise FileNotFoundError(f"Archivo no encontrado: {archivo_entrada}")

    # Leer archivo Excel
    df = pd.read_excel(archivo_entrada, engine="openpyxl")

    # Filtrar filas: columna H no nula y distinta de 0
    df = df[df.iloc[:, 7].notna() & (df.iloc[:, 7] != 0)]

    # Limpiar y eliminar duplicados basados en columna G
    df.iloc[:, 6] = df.iloc[:, 6].str.replace(r"^- ", "", regex=True)
    df = df.drop_duplicates(subset=df.columns[6])

    # Exportar columnas G y H a Excel
    columnas_necesarias = [df.columns[6], df.columns[7]]
    df[columnas_necesarias].to_excel(archivo_salida_excel, index=False, engine="openpyxl")
    print(f"Archivo procesado y guardado en: {archivo_salida_excel}")

except Exception as e:
    print(f"Error: {e}")
```

Script 2 - Comparación de Descripciones Financieras:

```
import pandas as pd
from google.colab import drive

# Montar Google Drive
drive.mount('/content/drive')

# Definir las rutas de entrada y salida en Google Drive
ruta_informe5 = '/content/drive/MyDrive/Concilia Docs/INFORME5.xlsx'  # Ruta del archivo
INFORME 5
ruta_movfondos = '/content/drive/MyDrive/Archivos
Conciliaciones/movfndic23_listado222.xlsx'  # Ruta del archivo mov fondos
ruta_salida_informe5 = '/content/drive/MyDrive/Concilia
Docs/INFORME_5_con_banco201.xlsx'  # Ruta de salida para INFORME 5

# Leer el archivo INFORME 5
try:
    df_informe5 = pd.read_excel(ruta_informe5)  # Cargar el archivo INFORME 5
```


[1]Fecha: 31 de diciembre de 24

```
except Exception as e:
    print(f"Error al leer el archivo INFORME 5: {e}")
    exit()

# Leer el archivo mov fondos
try:
    df_movfondos = pd.read_excel(ruta_movfondos)  # Cargar el archivo mov fondos
except Exception as e:
    print(f"Error al leer el archivo mov fondos: {e}")
    exit()

# Crear una lista de todos los valores en la columna 'Detalle' del archivo mov fondos,
normalizada
detalles_movfondos = df_movfondos['Detalle'].dropna().apply(lambda x:
str(x).strip().lower()).unique()
# Función para comparar si un string de la columna 'DESCRIPCION' en INFORME 5 existe en
'Detalle' de mov fondos
def comparar_descripcion(descripcion):
    descripcion_normalizada = str(descripcion).strip().lower()
    # Comprobamos si la descripción está en la lista de detalles
    if descripcion_normalizada in detalles_movfondos:
        return 'SI'
    else:
        return 'NO'

# Crear la nueva columna 'Banco 20' en INFORME 5 comparando las descripciones
df_informe5['Banco 20'] = df_informe5['DESCRIPCION'].apply(comparar_descripcion)
# Guardar el archivo procesado en Google Drive
try:
    df_informe5.to_excel(ruta_salida_informe5, index=False)
    print(f"Archivo procesado y guardado como '{ruta_salida_informe5}'")
except Exception as e:
    print(f"Error al guardar el archivo: {e}")
```

Script 3 - Unificación de Archivos:

```
import pandas as pd

# Definir las rutas de entrada y salida
input_files = [
    '/content/drive/MyDrive/Archivos Conciliaciones/Fechas_Excel/transbank dic 23.xlsx',
    '/content/drive/MyDrive/Archivos Conciliaciones/Fechas_Excel/transbank ene.24.xlsx',
    '/content/drive/MyDrive/Archivos Conciliaciones/Fechas_Excel/transbank feb.24.xlsx']
output_file =
'/content/drive/MyDrive/ArchivosConciliaciones/Fechas_Excel/filas_combined.xlsx'
combined_data = [] #Crear una lista para almacenar los datos combinados

#Procesar cada archivo de entrada
for file in input_files:
    # Leer el archivo Excel
    df = pd.read_excel(file, header=None)  # Leer sin encabezado inicial
    # Buscar la fila que contiene el encabezado deseado
    header_row_index = df.apply(lambda row: row.astype(str).str.contains('Fecha
Venta').any(), axis=1)
```


[1]Fecha: 31 de diciembre de 24

```
    header_index = header_row_index.idxmax() if header_row_index.any() else None
    # Configurar las columnas usando el encabezado encontrado
    df.columns = df.iloc[header_index]
    df = df.iloc[header_index + 1:].reset_index(drop=True)  # Tomar solo las filas debajo
del encabezado
    combined_data.append(df) #se agregan los datos al archivo combinado

#Combinar los datos y guardar en un archivo Excel
if combined_data:
    combined_df = pd.concat(combined_data, ignore_index=True)
    combined_df.to_excel(output_file, index=False, engine='openpyxl')
    print(f"Archivo combinado guardado como: {output_file}")
else:
    print("No se encontraron datos en ninguno de los archivos de entrada.")
```


[1]Fecha: 31 de diciembre de 24