Transformación de datos con Pandas

El siguiente proyecto, tiene como objetivo unificar varios archivos, almacenados en un directorio, con el fin de hacer un solo archivo unificado y, luego de eso, limpiar y transformar los datos, hasta tener una base final limpia y bien estructurada. Este es un proyecto real, realizado para una empresa de telecomunicaciones. Sin embargo, para la práctica, se usará una cantidad de archivos reducido, con data simplificada y con información ficticia, con el objetivo de proteger los datos privados de la empresa.

Parte I: Importación de librerías y carga de datos al dataframe

Antes que nada, nos conectaremos a nuestro Google Colab para extraer los archivos.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/dri
```

```
pip install datetime
```

```
Collecting datetime
  Downloading DateTime-5.2-py3-none-any.whl (52 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 52.2/52.2 kB 1.5 MB/s eta 0:00:00
Collecting zope.interface (from datetime)
  Downloading zope.interface-6.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 247.0/247.0 kB 11.0 MB/s eta 0:00:00
Requirement already satisfied: pytz in /usr/local/lib/python3.10/dist-packages (from datetime) (2023.3.
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from zope.interfa
Installing collected packages: zope.interface, datetime
Successfully installed datetime-5.2 zope.interface-6.0
```

Ahora importaremos las librerías necesarias y cargaremos el archivo a un df, utilizando la librería os, para crear listas de directorios y acceder a todos los archivos, para unificarlos.

```
import pandas as pd
import os
import glob
```

```
# Crear lista con los archivos presentes en el directorio y asignarlos a la variable "files"
files = os.listdir('/content/drive/My Drive/python_projects/XLS')
```

```
# Acceder a cada archivo con el ciclo 'for' y agregarlo a un dataframe
for i in files:
  df = pd.read_excel('/content/drive/My Drive/python_projects/XLS/'+i)
```

Realizaremos una exploración inicial de los datos.

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46 entries, 0 to 45
Data columns (total 26 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   ID_ORDEN                    46 non-null     object 
 1   ID_ORDEN_DE_DESPACHO        46 non-null     object 
 2   ID_DE_ACCION_DE_ORDEN       46 non-null     object 
 3   NOMBRE_DEL_CLIENTE          46 non-null     object 
 4   DOCUMENTO_CLIENTE_TIPO      46 non-null     object 
 5   DOCUMENTO_CLIENTE           46 non-null     object 
 6   DOCUMENTO_CLIENTE_DV        46 non-null     object 
 7   USUARIO_CREACION            46 non-null     object 
 8   TIPO_DE_REQUERIMIENTO       46 non-null     object 
 9   ESTADO                      46 non-null     object 
 10  FECHA_HORA_DEL_ESTADO       46 non-null     object 
 11  COMENTARIO_FLAG_ERROR       46 non-null     object 
 12  PRODUCTO_1                  46 non-null     int64  
 13  PRODUCTO_2                  46 non-null     int64  
 14  SISTEMA_DE_ORIGEN           46 non-null     object 
 15  CANAL                       46 non-null     object 
 16  FECHA_ESTADO_AGENDADO       46 non-null     object 
 17  FECHA_ALTA                  26 non-null     object 
 18  MONTH_FINALIZATION          0 non-null      float64
 19  SERIE                       42 non-null     object 
 20  MSISDN                      3 non-null      object 
 21  NUMERO_DE_GUIA_DE_DESPACHO  41 non-null     object 
 22  ID_ASIGNACION_OTR           46 non-null     object 
 23  TIPO_CONFIRMACION           3 non-null      float64
 24  CODIGO_CLIENTE              43 non-null     object 
 25  ESTADO_ANTERIOR             46 non-null     object 
dtypes: float64(2), int64(2), object(22)
memory usage: 9.5+ KB
None
```

```
df.head(2)
```

	ID_ORDEN	ID_ORDEN_DE_DESPACHO	ID_DE_ACCION_DE_ORDEN	NOMBRE_DEL_CLIENTE	DOCUMENTO_CLIENTE_TIPO	DO
0	eenapsr	entcssrap	entcssray	Cliente_1	RUT	
1	eenapsr	entcssrap	entcssray	Cliente_1	RUT	

2 rows × 26 columns

Para comenzar con la transformación de los datos, uniremos las columnas "DOCUMENTO_CLIENTE" y "DOCUMENTO_CLIENTE_DV" para crear una nueva columna llamada "RUT". Actualmente se encuentran de la siguiente forma:

```
print(df[["DOCUMENTO_CLIENTE","DOCUMENTO_CLIENTE_DV"]].head(2))
```

```
  DOCUMENTO_CLIENTE DOCUMENTO_CLIENTE_DV
0          scsptocc                    a
1          scsptocc                    a
```

Aplicamos la transformación

```
df["RUT"] = df[["DOCUMENTO_CLIENTE","DOCUMENTO_CLIENTE_DV"]].apply("-".join, axis=1)
print(df["RUT"].head(2))
```

```
0    scsptocc-a
1    scsptocc-a
Name: RUT, dtype: object
```

Ahora, eliminaremos las columnas que no utilizaremos.

```
df = df.drop(["ID_ORDEN","ID_DE_ACCION_DE_ORDEN","DOCUMENTO_CLIENTE_TIPO","DOCUMENTO_CLIENTE","DOCUMENTO_CL
```

Verificaremos que se hayan eliminado las columnas

```
print(df.columns)
```

```
Index(['ID_ORDEN_DE_DESPACHO', 'NOMBRE_DEL_CLIENTE', 'USUARIO_CREACION',
       'TIPO_DE_REQUERIMIENTO', 'ESTADO', 'FECHA_HORA_DEL_ESTADO',
       'COMENTARIO_FLAG_ERROR', 'PRODUCTO_1', 'PRODUCTO_2',
       'SISTEMA_DE_ORIGEN', 'FECHA_ESTADO_AGENDADO', 'TIPO_CONFIRMACION',
       'CODIGO_CLIENTE', 'ESTADO_ANTERIOR', 'RUT'],
      dtype='object')
```

Guardaremos el archivo transformado, en un nuevo archivo, en este caso csv para mejorar el rendimiento de las próximas transformaciones. No se sobreescribirá el archivo original para evitar perder información valiosa.

```
df.to_csv('/content/drive/My Drive/python_projects/CSV/' + i[:12] + '_new.csv')
```

Ahora, consolidaremos todos los archivos en uno solo, con toda la información, con un pd.concat

```
consolidado_anual = os.path.join("/content/drive/My Drive/python_projects/CSV/","*.csv")
list_files_2 = glob.glob(consolidado_anual)
df = pd.concat(map(pd.read_csv, list_files_2), ignore_index=True)
print(df.info())
df.to_csv("/content/drive/My Drive/python_projects/CSV/Consolidado.csv",sep=",",encoding="utf-8")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9152 entries, 0 to 9151
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Unnamed: 0             9152 non-null   int64 
 1   ID_ORDEN_DE_DESPACHO   9152 non-null   object
 2   NOMBRE_DEL_CLIENTE     9152 non-null   object
 3   USUARIO_CREACION       9152 non-null   object
 4   TIPO_DE_REQUERIMIENTO  9152 non-null   object
 5   ESTADO                 9152 non-null   object
 6   FECHA_HORA_DEL_ESTADO  9152 non-null   object
 7   COMENTARIO_FLAG_ERROR  9128 non-null   object
 8   PRODUCTO_1             9152 non-null   int64 
 9   PRODUCTO_2             9152 non-null   int64 
 10  SISTEMA_DE_ORIGEN      9152 non-null   object
 11  FECHA_ESTADO_AGENDADO  9152 non-null   object
```

```
 12  TIPO_CONFIRMACION      8616 non-null   float64
 13  CODIGO_CLIENTE         536 non-null    object
 14  ESTADO_ANTERIOR        9152 non-null   object
 15  RUT                    9152 non-null   object
 16  Unnamed: 0.7           8008 non-null   float64
 17  Unnamed: 0.6           6864 non-null   float64
 18  Unnamed: 0.5           5720 non-null   float64
 19  Unnamed: 0.4           4576 non-null   float64
 20  Unnamed: 0.3           3432 non-null   float64
 21  Unnamed: 0.2           2288 non-null   float64
 22  Unnamed: 0.1           1144 non-null   float64
dtypes: float64(8), int64(3), object(12)
memory usage: 1.6+ MB
None
```

Ahora, importaremos el archivo consolidado a un nuevo dataframe

```
df_consolidado = pd.read_csv('/content/drive/My Drive/python_projects/CSV/Consolidado.csv')
print(df_consolidado.head(2))
```

```
   Unnamed: 0.8  Unnamed: 0 ID_ORDEN_DE_DESPACHO NOMBRE_DEL_CLIENTE  \
0             0           0           mcccoyporc          Cliente_1
1             1           1           mcccoyporc          Cliente_1

  USUARIO_CREACION TIPO_DE_REQUERIMIENTO     ESTADO FECHA_HORA_DEL_ESTADO  \
0        usuario_1               Tipo_5  Estado_14   2023-01-23 16:15:16
1        usuario_1               Tipo_5  Estado_14   2023-01-23 16:15:16

  COMENTARIO_FLAG_ERROR  PRODUCTO_1  ...  CODIGO_CLIENTE ESTADO_ANTERIOR  \
0            En Gestion           0  ...             NaN        Estado_9
1            En Gestion           0  ...             NaN        Estado_9

          RUT  Unnamed: 0.7 Unnamed: 0.6 Unnamed: 0.5 Unnamed: 0.4  \
0  ssntrsrt-K           NaN          NaN          NaN          NaN
1  ssntrsrt-K           NaN          NaN          NaN          NaN

   Unnamed: 0.3  Unnamed: 0.2  Unnamed: 0.1
0           NaN           NaN           NaN
1           NaN           NaN           NaN

[2 rows x 24 columns]
```

Convertiremos la columna FECHA_HORA_DEL_ESTADO al tipo de dato 'datetime', para luego ordenar al dataframe por esta columna, del valor más reciente al más antiguo

```
df_consolidado['FECHA_HORA_DEL_ESTADO'] = pd.to_datetime(df_consolidado['FECHA_HORA_DEL_ESTADO'])
print(df_consolidado['FECHA_HORA_DEL_ESTADO'].head(2))
```

```
0   2023-01-23 16:15:16
1   2023-01-23 16:15:16
Name: FECHA_HORA_DEL_ESTADO, dtype: datetime64[ns]
```

```
df_consolidado = df_consolidado.sort_values('FECHA_HORA_DEL_ESTADO', ascending=False)
print(df_consolidado.head(5))
```

```
      Unnamed: 0.8  Unnamed: 0 ID_ORDEN_DE_DESPACHO NOMBRE_DEL_CLIENTE  \
1040          1040         531           mcccoysspc          Cliente_6
3374          3374         531           mcccoysspc          Cliente_6
7950          7950         531           mcccoysspc          Cliente_6
```

```
9094          9094         531           mcccoysspc          Cliente_6
1041          1041         532           mcccoysspc          Cliente_6

     USUARIO_CREACION TIPO_DE_REQUERIMIENTO     ESTADO FECHA_HORA_DEL_ESTADO  \
1040        usuario_5                Tipo_5  Estado_14   2023-02-07 15:41:00
3374        usuario_5                Tipo_5  Estado_14   2023-02-07 15:41:00
7950        usuario_5                Tipo_5  Estado_14   2023-02-07 15:41:00
9094        usuario_5                Tipo_5  Estado_14   2023-02-07 15:41:00
1041        usuario_5                Tipo_5  Estado_14   2023-02-07 15:41:00

     COMENTARIO_FLAG_ERROR  PRODUCTO_1  ...  CODIGO_CLIENTE ESTADO_ANTERIOR  \
1040            En Gestion           1  ...             NaN        Estado_9
3374            En Gestion           1  ...             NaN        Estado_9
7950            En Gestion           1  ...             NaN        Estado_9
9094            En Gestion           1  ...             NaN        Estado_9
1041            En Gestion           1  ...             NaN        Estado_9

             RUT  Unnamed: 0.7 Unnamed: 0.6 Unnamed: 0.5 Unnamed: 0.4  \
1040  aoncnccc-c           NaN          NaN          NaN          NaN
3374  aoncnccc-c        2230.0       1086.0          NaN          NaN
7950  aoncnccc-c        6806.0       5662.0       4518.0       3374.0
9094  aoncnccc-c        7950.0       6806.0       5662.0       4518.0
1041  aoncnccc-c           NaN          NaN          NaN          NaN

      Unnamed: 0.3  Unnamed: 0.2  Unnamed: 0.1
1040           NaN           NaN           NaN
3374           NaN           NaN           NaN
7950        2230.0        1086.0           NaN
9094        3374.0        2230.0        1086.0
1041           NaN           NaN           NaN

[5 rows x 24 columns]
```

A partir de esto, se eliminarán los registros que en la columna "ID_ORDEN_DE_DESPACHO", se encuentren duplicadas

```
df_consolidado = df_consolidado.drop_duplicates("ID_ORDEN_DE_DESPACHO")
print(df_consolidado.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 36 entries, 1040 to 2243
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0.8           36 non-null     int64
 1   Unnamed: 0             36 non-null     int64
 2   ID_ORDEN_DE_DESPACHO   36 non-null     object
 3   NOMBRE_DEL_CLIENTE     36 non-null     object
 4   USUARIO_CREACION       36 non-null     object
 5   TIPO_DE_REQUERIMIENTO  36 non-null     object
 6   ESTADO                 36 non-null     object
 7   FECHA_HORA_DEL_ESTADO  36 non-null     datetime64[ns]
 8   COMENTARIO_FLAG_ERROR  34 non-null     object
 9   PRODUCTO_1             36 non-null     int64
 10  PRODUCTO_2             36 non-null     int64
 11  SISTEMA_DE_ORIGEN      36 non-null     object
 12  FECHA_ESTADO_AGENDADO  36 non-null     object
 13  TIPO_CONFIRMACION      9 non-null      float64
 14  CODIGO_CLIENTE         27 non-null     object
 15  ESTADO_ANTERIOR        36 non-null     object
 16  RUT                    36 non-null     object
 17  Unnamed: 0.7           32 non-null     float64
```

```
 18  Unnamed: 0.6          29 non-null     float64
 19  Unnamed: 0.5          23 non-null     float64
 20  Unnamed: 0.4          21 non-null     float64
 21  Unnamed: 0.3          18 non-null     float64
 22  Unnamed: 0.2          12 non-null     float64
 23  Unnamed: 0.1          10 non-null     float64
dtypes: datetime64[ns](1), float64(8), int64(4), object(11)
memory usage: 7.0+ KB
None
```

Guardamos el nuevo dataframe en un archivo excel final. Con esto, terminamos la transformación de los datos de este proyecto.

```
df_consolidado.to_excel('/content/drive/My Drive/python_projects/CSV/Consolidado_sin_duplicados.xlsx')
```

✓ 0s completed at 5:51 PM