

Origen de la diferencia de datos nuevos en Ubicaciones y por ende de las demas tablas que dependen de ubicaciones:



Mara Destefanis 7:06 PM



Te paso de nuevo ubicaciones. El excel tuvo que corregirse afuera, ya que arrastraba un dato en donde el código tenía pegado un nombre y atrataba ese error. Se podía corregir en el código.- Pero miramos antes de subir.- Específicamente uno.- así lo otro hablado.- Ahora el siguiente script de proyectos va también con menos ruido.- Seguimos

2 files ▾

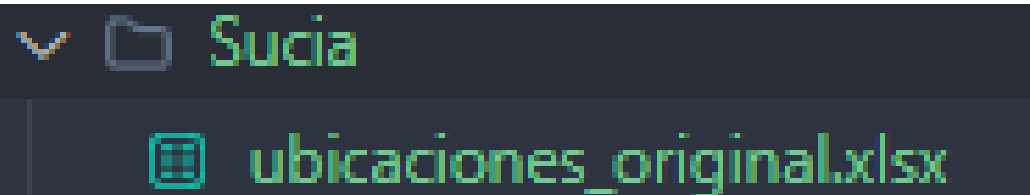
01

01_OneDrive_Ubicaciones_Cliente...
Binary

X

Ubicaciones.xlsx
Excel Spreadsheet

archivo ORIGINAL



aca tareas viene CON
codigo y usamos esos
codigos para restar
212 filas al ubicaciones
limpio

ubicaciones_original.xlsx U X		
	G	H
	TAREAS	
	80313	
	80315	
	80309	

cambiado mara 11/2

aca el archivo ubicaciones nuevo
que me das, viene tareas vacio, sin
codigos

Sucia		
Ubicaciones.xlsx		
Ubicaciones.xlsx X		
	G	H
IC...	TAREAS	
3		
5		
9		
4		

archivo ORIGINAL

esta es la parte que borraste

cambiado mara 11/2

reemplazaste
todo el bloque 1
por esta linea

01_OneDrive_Ubicaciones_Clientes_ORIGINAL.ipynb U X

```
BLOQUE 1: CARGA INICIAL Y LIMPIEZA BÁSICA

print("===== BLOQUE 1: CARGA Y LIMPIEZA BÁSICA =====")
print("-----")

# 1.1. Carga inicial del archivo
try:
    ruta = "../Sucia/ubicaciones_original.xlsx"
    df_ubicaciones = pd.read_excel(ruta, sheet_name='Ubicacion', engine='openpyxl')
    print(f"✓ Archivo de ubicaciones cargado exitosamente")
    total_filas_inicial = df_ubicaciones.shape[0]
    print(f"Número total de filas al inicio: {total_filas_inicial}")
except Exception as e:
    print(f"✗ Error al cargar archivo: {e}")
    raise

# 1.2. Eliminar filas completamente vacías
print("\n1.2. Eliminando filas completamente vacías...")
filas_antes = df_ubicaciones.shape[0]
df_ubicaciones = df_ubicaciones.dropna(how='all').copy()
print(f"Filas eliminadas: {filas_antes - df_ubicaciones.shape[0]}")
print(f"Filas restantes: {df_ubicaciones.shape[0]}")

# 1.3. Eliminar filas mayoritariamente vacías (80% o más)
print("\n1.3. Eliminando filas mayoritariamente vacías...")
columnas_verificar = df_ubicaciones.columns.tolist()
umbral_vacias = 0.8 # 80% de columnas vacías o con valor 0

condicion_mayoritariamente_vacia = (
    (df_ubicaciones[columnas_verificar].isnull().sum(axis=1) > len(columnas_verificar) * umbral_vacias) |
    (df_ubicaciones[columnas_verificar].isnull() | df_ubicaciones[columnas_verificar] == 0).sum(axis=1) > len(columnas_verificar) * umbral_vacias
)

total_filas_mayoritariamente_vacias = condicion_mayoritariamente_vacia.sum()
print(f"Filas mayoritariamente vacías a eliminar (>80% vacías o 0.0): {total_filas_mayoritariamente_vacias}")

df_ubicaciones = df_ubicaciones[~condicion_mayoritariamente_vacia].copy()
print(f"Filas restantes: {df_ubicaciones.shape[0]}")

# 1.4. Verificación de columna TAREAS
print("\n1.4. Verificando columna 'TAREAS'...")
if 'TAREAS' in df_ubicaciones.columns:
    filas_vacias_tareas = df_ubicaciones['TAREAS'].isna() | (df_ubicaciones['TAREAS'] == 0)
    total_filas_vacias_tareas = filas_vacias_tareas.sum()
    print(f"Filas con 'TAREAS' vacías: {total_filas_vacias_tareas}")
    df_ubicaciones = df_ubicaciones[~filas_vacias_tareas].copy()
    print(f"Filas restantes: {df_ubicaciones.shape[0]}")
else:
    print("La columna 'TAREAS' no existe en el DataFrame")

[3] ✓ 0%
```

```
# Dejo solo los que tienen ubicaciones.

df_ubicaciones = df_ubicaciones.dropna(subset=['UBICACIÓN'])

df_ubicaciones.info()

<class 'pandas.core.frame.DataFrame'>
Index: 747 entries, 0 to 747
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   CODIGO      747 non-null    int64
1   CODIGO2     747 non-null    object
2   UBICACIÓN   747 non-null    object
3   FECHA       747 non-null    datetime64[ns]
4   ORIGEN      747 non-null    object
5   CODCLI2     747 non-null    int64
6   TAREAS      0 non-null      float64
dtypes: datetime64[ns](1), float64(1), int64(2), object(3)
memory usage: 46.7+ KB
```

se nota
claramente que
son 747 registros
vs 536 que
teníamos antes

```
1.4. Verificando columna 'TAREAS'...
Filas con 'TAREAS' vacías: 212
Filas restantes: 536
```

1.4. Verificando columna 'TAREAS'...
Filas con 'TAREAS' vacías: 212
Filas restantes: 536

archivo ORIGINAL

**Ahi arriba se ve
claramente que
borrabamos 212
lineas, usando
como filtro los
CODIGOS VACIOS
que venian en la
columna TAREAS
del archivo
ubicaciones viejo**

cambiado mara 11/2

**Con el nuevo
archivo
Ubicaciones que
me diste, sin
codigos en la
columna TAREAS,
y habiendo
eliminado el
bloque 1, estamos
agregando 212
datos que
soliamos eliminar.**

repercusiones del cambio que consideras que no existe

archivo ORIGINAL

Ubicaciones_direcciones_original.xlsx

512	13090	DANIEL MERCADO SALINAS
513	82164	LOS SALTEÑOS

Ubicaciones_Limpias

513 lineas antes

04_02_Datos_Table_Proyectos_Clientes.ipynb
Tabla_Proyectos_Clientes_original.xlsx

1014	82164	Ruta Hugo Sabado
1015	82164	\$ Creditos Hugo

Sheet1

1015 lineas antes

cambiado mara 11/2

Ubicaciones_direcciones.xlsx

720	12893	AUT.77
721	82164	LOS SALTEÑOS

Ubicaciones_Limpias

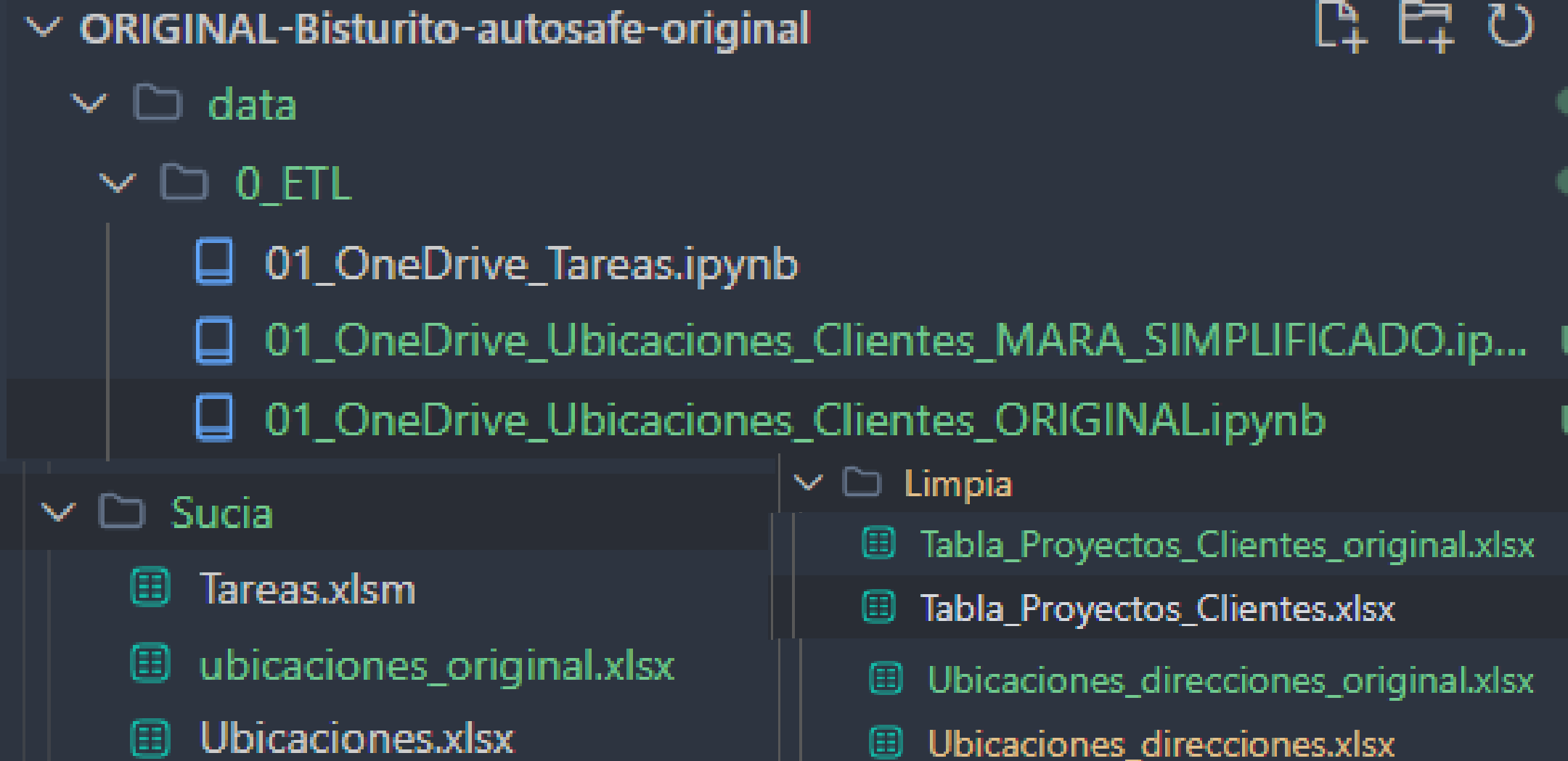
vs 721 ahora con tus cambios

04_02_Datos_Table_Proyectos_Clientes.ipynb
Tabla_Proyectos_Clientes.xlsx

1228	82164	Ruta Hugo Sabado
1229	82164	\$ Creditos Hugo

Sheet1

1229 lineas con tus cambios



dejo los ipynb y excell en el repositorio para que revises si esta BIEN que agreguemos estos datos que soliamos borrar.

Asumo que si vos lo cambiaste, haciendo calidad de datos, deberia ser al pedo que yo te diga esto, pero me sorprende y me deja en duda que al nombrarte esto e intentar explicartelo, no sepas de que te hablo y te confunda tanto.

Espero que se entienda que este cambio nos afecta AHORA y no es nada viejo ni antiguo ni del pasado, es de ahora ya, es nuevo, now, right now, `sys.date(now)`