**AUTOMATIZACION DOCUMENTOS**

Este script de Python automatiza la generación de **documentos Word y Excel personalizados** y la creación de **estructuras de carpetas** a partir de un archivo Excel con parámetros. A continuación te explico por secciones:

**🔧 1. Importación de librerías**

python

CopiarEditar

import pandas as pd # Leer y manipular datos en Excel

import os, shutil # Manejo del sistema de archivos (crear/eliminar carpetas)

from docxtpl import DocxTemplate # Rellenar plantillas Word con datos

from datetime import datetime

from pathlib import Path

import openpyxl # Manipular archivos Excel (.xlsx)

import xlwings # Control de Excel local (no se usa en el código)

import warnings

**🗂 2. Parámetros iniciales**

python

CopiarEditar

OUTPUT\_PATH = './Outputs' # Carpeta de salida

EXCEL\_PATH = './Parameters/Formato Nombres.xlsx' # Archivo fuente

WORD\_PATHS = {...} # Plantillas Word disponibles

EXCEL\_TEMPLATES = {...} # Plantillas Excel disponibles

FECHA = datetime.today().strftime("%d/%m/%Y") # Fecha actual

warnings.simplefilter("ignore", UserWarning) # Ocultar warnings

**🧹 3. Eliminar y recrear carpeta de salida**

python

CopiarEditar

def eliminar\_y\_crear\_carpeta(path):

if os.path.exists(path):

shutil.rmtree(path) # Elimina carpeta si existe

os.makedirs(path, exist\_ok=True) # Crea la carpeta

**📁 4. Crear subcarpetas por OC (orden de cambio)**

python

CopiarEditar

def crear\_sub\_carpetas(df\_datos, output\_path, carpetas=None):

if carpetas is None:

carpetas = ["01\_Diagnostico", "02\_Solucion", "03\_Pruebas", "04\_Instalacion"]

**Para cada fila en el Excel:**

1. Lee OC y NOMBRE.
2. Crea una carpeta como: OC\_1234\_JuanPerez
3. Dentro de esa carpeta crea las subcarpetas mencionadas arriba.

**📄 5. Generar documentos Word y Excel**

python

CopiarEditar

def crear\_documentos(df\_datos):

todos\_los\_tipos = {\*\*WORD\_PATHS, \*\*EXCEL\_TEMPLATES}

df\_datos = df\_datos[df\_datos['TIPO'].isin(todos\_los\_tipos.keys())]

**Para cada TIPO en el archivo Excel:**

1. Usa la plantilla correspondiente (WORD\_PATHS o EXCEL\_TEMPLATES)
2. Genera:
   * Si es .docx, llena campos como OC, NOMBRE, ROL, FECHA, etc.
   * Si es .xlsx, modifica celdas específicas de hojas como Portada, Caso 1, 1-Est. y Planeación, etc.
3. Guarda el nuevo archivo con nombre NOMBRE\_OC.docx o .xlsx en ./Outputs/.

**🚀 6. Función principal (main)**

python

CopiarEditar

def main():

eliminar\_y\_crear\_carpeta(OUTPUT\_PATH)

df = pd.read\_excel(EXCEL\_PATH, sheet\_name='Datos') # Lee Excel

crear\_sub\_carpetas(df, OUTPUT\_PATH) # Crea carpetas

crear\_documentos(df) # Genera archivos

print("Proceso finalizado")

**🏁 7. Ejecución del script**

python

CopiarEditar

if \_\_name\_\_ == '\_\_main\_\_':

main()

Esto permite ejecutar el script directamente como programa.

**✅ RESUMEN FUNCIONAL**

Este script:

1. **Lee un Excel de parámetros** con datos de órdenes de cambio.
2. **Crea carpetas por cada OC con subcarpetas estándar**.
3. **Llena documentos Word y/o Excel con esos datos**.
4. **Guarda todo en una carpeta Outputs limpia y nueva cada vez.**

**FUNCIONALIDAD**

**🧹 eliminar\_y\_crear\_carpeta(path)**

**✅ ¿Qué hace?**

Elimina la carpeta especificada (si existe) y la vuelve a crear vacía.

**🔧 Parámetros:**

* path: Ruta de la carpeta a eliminar/crear.

**🧠 Detalles:**

python

CopiarEditar

if os.path.exists(path): # Verifica si la carpeta ya existe

shutil.rmtree(path) # La elimina con todo su contenido

os.makedirs(path, exist\_ok=True) # La crea de nuevo

**📁 crear\_sub\_carpetas(df\_datos, output\_path, carpetas=None)**

**✅ ¿Qué hace?**

Crea carpetas y subcarpetas para cada fila en el DataFrame df\_datos.

**🔧 Parámetros:**

* df\_datos: DataFrame con columnas como OC y NOMBRE.
* output\_path: Carpeta raíz donde se crearán las subcarpetas.
* carpetas: Lista de nombres de subcarpetas (opcional, tiene un valor por defecto).

**🧠 Detalles:**

python

CopiarEditar

for \_, fila in df\_datos.iterrows(): # Recorre cada fila

v\_oc = fila['OC']

v\_nombre = fila['NOMBRE']

nombre\_carpeta = f"OC\_{v\_oc}\_{v\_nombre}"

ruta\_final = os.path.join(output\_path, nombre\_carpeta)

for carpeta in carpetas:

os.makedirs(os.path.join(ruta\_final, carpeta), exist\_ok=True)

**📝 Devuelve:**

* Lista de mensajes de error (si hay problemas al crear carpetas).

**📄 crear\_documentos(df\_datos)**

**✅ ¿Qué hace?**

Genera documentos Word o Excel a partir de plantillas, usando los datos de cada fila.

**🔧 Parámetros:**

* df\_datos: DataFrame con información de OC, nombre, desarrollador, tipo, etc.

**🧠 Pasos importantes:**

1. **Unifica** las plantillas (WORD\_PATHS + EXCEL\_TEMPLATES)
2. **Filtra** por tipo de plantilla (APP\_, EDLLO\_, etc.)
3. **Itera** por cada grupo de NOMBRE\_OC.
4. **Para cada tipo de plantilla**:
   * Si es Word:
     + Usa DocxTemplate para rellenar campos como OC, NOMBRE, APLICACION, etc.
   * Si es Excel:
     + Abre con openpyxl, modifica celdas específicas en hojas como 'Portada', 'Caso 1', etc.
     + Guarda el archivo con el nombre correspondiente.

**✏️ Ejemplo de contexto para Word:**

python

CopiarEditar

contexto = {

'Numero\_OC': fila['OC'],

'Nombre\_OC': fila['NOMBRE'],

'Desarrollador': fila['DESARROLLADOR'],

'Fecha': FECHA,

}

doc.render(contexto)

doc.save(output\_path)

**🚀 main()**

**✅ ¿Qué hace?**

Es la **función principal** del programa. Controla el flujo de ejecución.

**🧠 Pasos:**

1. Muestra mensaje inicial.
2. Elimina y crea carpeta Outputs.
3. Lee el Excel de parámetros (Formato Nombres.xlsx).
4. Llama a crear\_sub\_carpetas().
5. Llama a crear\_documentos().
6. Imprime mensaje final de éxito.

**🔚 if \_\_name\_\_ == '\_\_main\_\_': main()**

**✅ ¿Qué hace?**

Es una **estructura estándar en Python**. Significa:

* Solo ejecuta main() **si el script se ejecuta directamente** (no si se importa como módulo desde otro script).

**✅ Funciones auxiliares usadas (de librerías)**

**os.path.exists(path)**

Verifica si existe una carpeta o archivo.

**shutil.rmtree(path)**

Elimina una carpeta completa.

**os.makedirs(path, exist\_ok=True)**

Crea una carpeta (y sus padres si no existen).

**pandas.read\_excel()**

Lee un archivo Excel y lo convierte en un DataFrame.

**openpyxl.load\_workbook()**

Abre un archivo Excel (.xlsx) para edición.

**DocxTemplate.render(context)**

Llena una plantilla Word con datos del diccionario context.

**DocxTemplate.save(ruta)**

Guarda el documento Word generado.