**Documentación del Proceso de Ejecución del Programa**

**1. Descripción General**

Este programa automatiza el proceso de cálculo de comisiones basadas en peticiones exitosas y no exitosas de APIs, y envía un correo electrónico con el archivo de resultados adjunto, además, se le adjunta la ruta local de ubicación del archivo. Utiliza Python con bibliotecas para manejar bases de datos SQLite, consultas SQL, procesamiento de datos en Excel, y la integración con Outlook para el envío de correos electrónicos.

**2. Requisitos Previos**

* **Python 3.11.9** instalado en el sistema.
* Bibliotecas de Python necesarias:
  + pandas
  + sqlite3
  + openpyxl (para manipular archivos Excel)
  + pywin32 (para integrar con Outlook)
  + jinja2 (para el cuerpo del html)
* Acceso a una base de datos SQLite con la estructura esperada.
* Configuración de Microsoft Outlook en el sistema.

**3. Estructura del Proyecto**

1. **conexion\_bd.py**: Script que contiene la función obtener\_datos para conectar a la base de datos SQLite, ejecutar consultas SQL, y guardar los resultados en un archivo Excel.
2. **correo.py**: Script que contiene la función enviar\_correo para enviar correos electrónicos a través de Outlook con el archivo adjunto.
3. **main.py**: Script principal que coordina la ejecución del proceso, desde la obtención de datos hasta el envío del correo.
4. **Consultas SQL**: Archivos .sql ubicados en el directorio “inputs/consultas”, que contienen las consultas para extraer y procesar los datos necesarios.
5. **Archivos de Resultados**: El archivo Excel generado se guarda en pryAutomatizacionEmpresas/results.
6. **Cuerpo del correo**: En el proyecto hay un script HTML para el cuerpo del correo, este se encuentra en la ruta: “inputs\cuerpo\_correo\00\_cuerpo\_correo.html”

**4. Proceso de Ejecución**

**4.1 Obtención de Datos**

1. **Conexión a la Base de Datos**: El script conexion\_bd.py se conecta a la base de datos SQLite ubicada en “inputs/database/database.sqlite”.
2. **Ejecución de Consultas SQL**: Lee todos los archivos SQL en el directorio “inputs/consultas” y ejecuta cada consulta.
3. **Procesamiento de Resultados**: Combina los resultados de todas las consultas en un único DataFrame utilizando pandas.
4. **Guardado en Excel**: Guarda el DataFrame consolidado en un archivo Excel en pryAutomatizacionEmpresas/results con el nombre resultados\_unificados.xlsx.

**4.2 Detección de Parámetros Destinatarios:**

Los parámetros se encuentran en la siguiente ruta: \pryAutomatizacionEmpresas\inputs\parametros

**1.** Lectura de Archivo de Parámetros: La función detectar\_destinatarios\_notificacion lee el archivo Excel parametros\_destinatarios.xlsx en el directorio inputs/parametros.

**2.** Extracción de Correos: Extrae las direcciones de correo electrónico y las combina en una cadena separada por punto y coma, compatible con Outlook.

**4.3 Envío de Correo**

1. **Preparación del Correo**: El script correo.py utiliza la biblioteca win32com.client para crear y enviar un correo electrónico a través de Outlook.
2. **Adjuntar Archivo**: Adjunto el archivo resultados\_unificados.xlsx al correo.
3. **Envío del Correo**: Envía el correo al destinatario especificado con el archivo adjunto.

**4.4 Ejecución Principal**

1. **Ejecución de Scripts**:
   * **main.py**: Este es el script principal que llama a la función obtener\_datos del módulo conexion\_bd para obtener los datos y guardar el archivo Excel.
   * Luego, llama a la función detectar\_destinatarios\_notificacion para obtener los destinatarios.
   * Finalmente, llama a la función enviar\_correo del módulo correo para enviar el archivo por correo electrónico a los destinatarios extraídos.

**5.** **Ejecución de .bat**

**1.** **PyRunEnv.bat:** Este archivo .bat se encarga de crear y activar un entorno virtual para el proyecto si aún no existe. También verifica si pip está instalado y, si no lo está, lo instala. Luego, instala las dependencias requeridas desde el archivo setup.py (si está disponible) y mantiene la ventana de la terminal abierta para confirmar que las instalaciones se realizaron correctamente.

**2.** **PyRun.bat:** Este archivo .bat activa el entorno virtual previamente creado y luego ejecuta el script principal main.py del proyecto. Este script coordina la ejecución de las funciones necesarias para obtener datos, procesarlos y enviar correos electrónicos con el archivo generado.