

Gestión de Datos y Envío de Correos Electrónicos

Este proyecto está diseñado para gestionar el cobro de comisiones de una base de datos SQLite, generar informes en formato CSV, crear archivos ZIP con los informes y enviar correos electrónicos con los informes adjuntos.

Tabla de Contenidos

1. Introducción
2. Requisitos
3. Configuración
4. Uso
5. Estructura del Proyecto

Introducción

Este proyecto automatiza la gestión de datos y el envío de correos electrónicos para reportes de facturación y transacciones. Utiliza tres módulos principales:

- `sqlManager.py`: Manejo de la base de datos.
- `fileManager.py`: Exportación de datos y creación de archivos ZIP.
- `emailManager.py`: Envío de correos electrónicos con informes adjuntos.

Requisitos

Para ejecutar el proyecto, necesitas:

- Python 3.6 o superior.
- Paquetes de Python:
 - `pandas`
 - `pywin32` (para integración con Outlook)
 - `zipfile` (incluido en la biblioteca estándar de Python)

Instala las dependencias necesarias usando `pip`:

Copiar código:

```
pip install pandas pywin32 pandas zipfile
```

Configuración

1. **Base de Datos:** Asegúrate de tener un archivo de base de datos SQLite. Modifica el valor de `db_path` en `ejecutar.py` para que apunte a tu archivo de base de datos.
2. **Directorios de Reportes:** Asegúrate de que el directorio `./recursos/reportes/` exista o cámbialo según tus necesidades.

3. **Plantilla HTML:** Asegúrate de que el archivo `./recursos/plantilla.html` exista y contenga el formato necesario para los correos electrónicos.

Uso

1. **Ejecutar el Script Principal:** Ejecuta el archivo `ejecutar.py` para realizar las siguientes tareas:
 - Conectar a la base de datos.
 - Ejecutar scripts SQL.
 - Exportar datos a archivos CSV y crear archivos ZIP.
 - Leer datos de facturación desde la base de datos y guardarlos en un archivo CSV.
 - Enviar correos electrónicos con los informes adjuntos.

Para ejecutar el script principal, usa el siguiente comando:

Copiar código

```
python ejecutar.py
```

Estructura del Proyecto

El proyecto se organiza en varios módulos:

sqlManager.py

Contiene la clase `SQLDatabaseManager` que gestiona la conexión a la base de datos SQLite y realiza operaciones como ejecutar scripts SQL y leer datos.

fileManager.py

Contiene la clase `FileManager` que exporta datos de comercios a archivos CSV y crea archivos ZIP con los reportes.

emailManager.py

Contiene la clase `EmailManager` que lee una plantilla HTML y envía correos electrónicos utilizando Outlook con los informes adjuntos.

ejecutar.py

Archivo principal que coordina el flujo de trabajo, desde la configuración hasta el envío de correos electrónicos.