**Nombre:** Josselyn Fabiola Duarte Torrres

# Carga y cruce de información de personas

Desarrollar un flujograma, donde se recibe un archivo Excel que contiene: nombre completo de persona (de la forma "Juan Antonio Garcia Hernandez"), número de DUI y otros campos demográficos.

Los nombres deben ser clasificados como: primer nombre, segundo nombre, tercer nombre, primer apellido, segundo apellido, apellido de casada.

Se requiere un reporte Excel donde estén clasificados los registros del archivo original indicando si el registro fue encontrado en alguna base interna, ya sea como cliente de banco, como persona en lista de control, ambos casos, o indicar si no encontró en ninguna (registro nuevo). Realice operaciones de limpieza de cadenas para maximizar la efectividad del cruce masivo.

Cartera de clientes del banco: información general de los clientes.

Lista de control: información de personas con las que el banco no debería establecer relación.

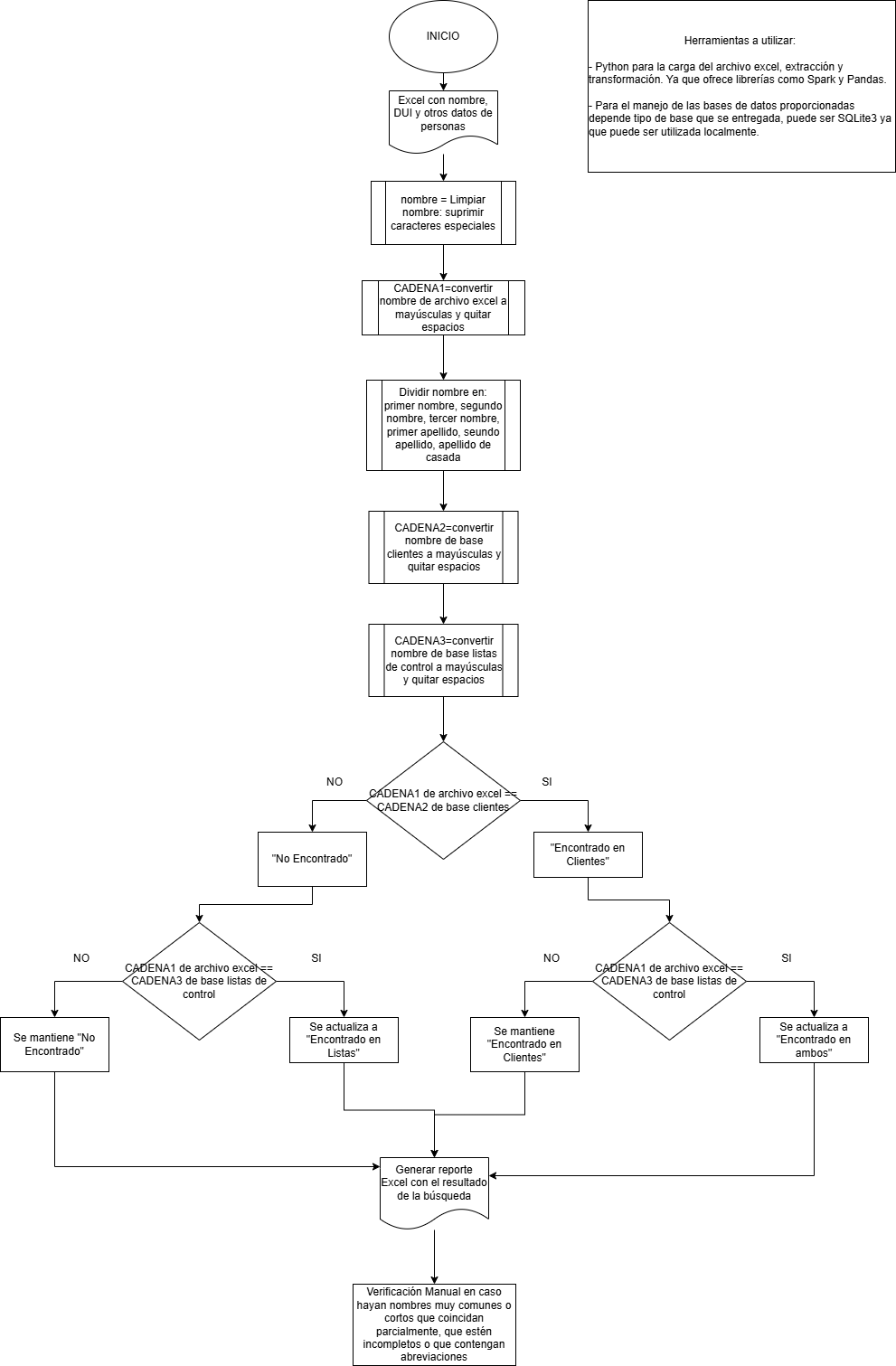
Consideraciones:

\* Plantear que herramientas se utilizarían.

\* El nombre completo puede tener caracteres especiales, espacios en blanco, números, mayúsculas, minúsculas.

\* Usar expresiones regulares de ser necesario.

\* Indicar si se detecta alguna inconsistencia. Es decir, comente si existen escenarios ambiguos en el resultado del proceso que deban ser verificados manualmente.



# Realizar el siguiente ETL

Realizar un ETL que cargue en una tabla el archivo "fuente.xlsx"

Se requiere un reporte Excel donde estén clasificados las personas basadas en la cartera donde encontró (tabla clientes, tabla lista\_control) o indicar si no encontró en ninguna cartera.

Consideraciones:

\* Utilizar las tablas de la base sqlite proporcionada (ev\_ing\_dat.db)

\* Usar python, spark, sql, Power BI, según considere necesario.

\* Documentar la solución, los pasos del proceso.

\* Usa bitácora para el proceso (log y tabla de bitácora)

\* Versionar la solución usando git.

\* Calendarizar el programa para que se ejecute todos los 1° de cada mes a las 10 a.m.

**Reporte esperado:**

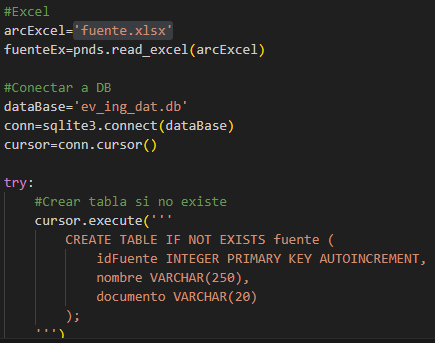
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre completo** | **Primer nombre** | **Segundo nombre** | **Tercer nombre** | **Primer apellido** | **Segundo apellido** | **Apellido de casada** | **Clasificación** |
| Juan Antonio Garcia Hernandez | Juan | Antonio |  | Garcia | Hernandez |  | Lista de control |
| María Lopez de Hernandez | María |  |  | López |  | De Hernandez | Cartera de Clientes |
| Sofía Gomez | Sofía |  |  | Gomez |  |  | No se encontró en las BD |

Pasos:

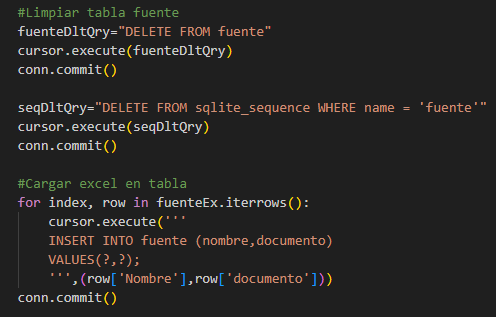
1. Primero se creó el núcleo del ETL:

* **Extracción**

Ya que se solicita que el archivo ‘fuente.xlsx’ se almacene en una tabla, y por ello se cargara en la base ‘ev\_ing\_dat.db’. Se hizo el proceso de leer el archivo conectar a la base (ya que ahí se creara la nueva tabla y ahí se encuentran las tablas de clientes y listas) y crear la tabla en caso no exista:

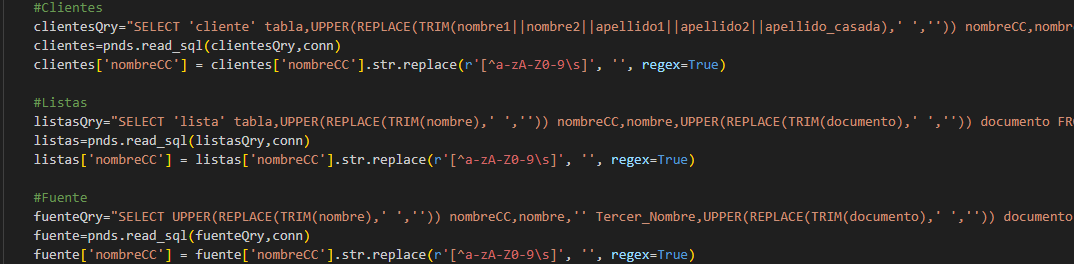


Cuando ya se tiene la tabla creada se procede a la carga.

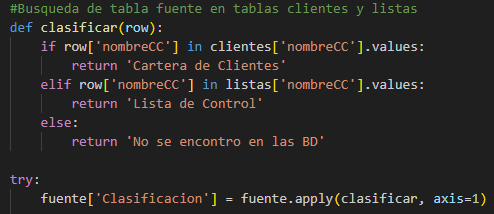


* **Transformación**

Primero se hizo una limpieza de las cadenas de texto que se iban a comparar, haciendo uso de SQL y funciones de Python.

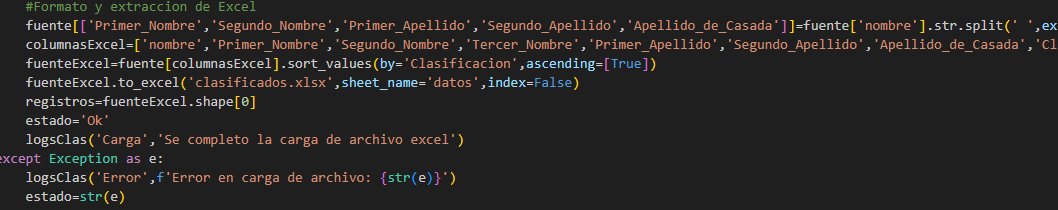


Luego se definió la función encargada de comparar las tablas y determinar si la persona existe en una de las bases: clientes o listas o si no fue encontrado.



* **Carga**

Ya que se debe generar un reporte Excel con el resultado, se usando la librería ‘pandas’ de Python de la siguiente forma:

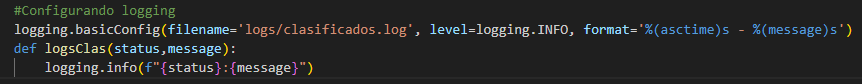


1. Creación de bitácora y logs

Para el proceso de registrar en bitácora y generar logs se hizo lo siguiente:

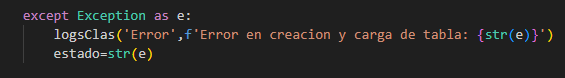
* **Logs**

Se definió una función usando la librería logging que ofrece Python.

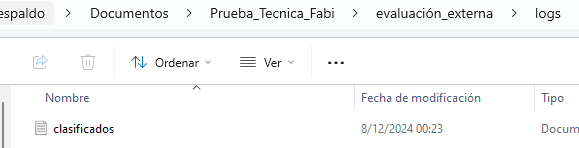


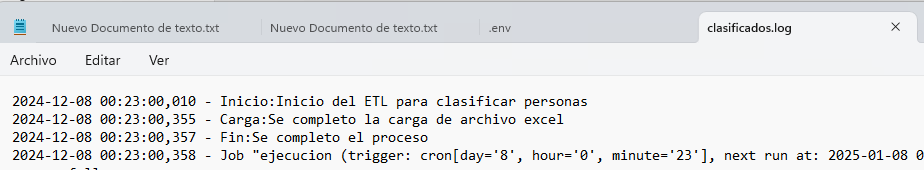
Y a lo largo del código se fue haciendo uso de esta función para registrar en el log los pasos y/o errores.





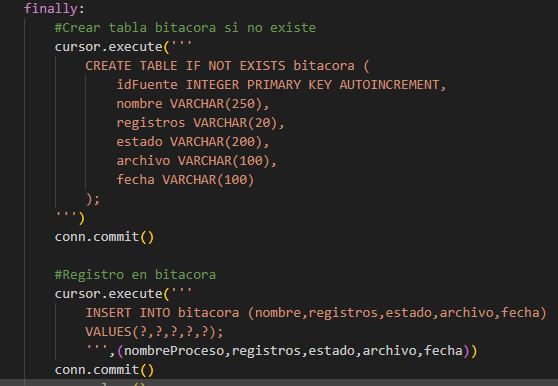
Mostrando un resultado como el siguiente.





* **Bitácora**

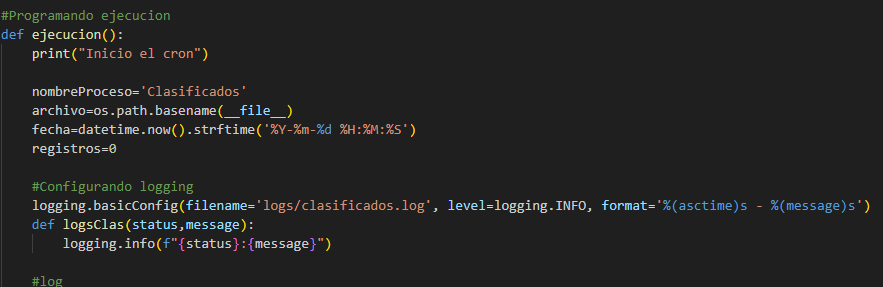
Para registro en bitácora se creo una tabla y se coloco el insert necesario para indicar el resultado de la ejecución.



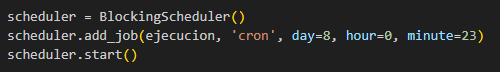
1. Calendarización

Para la calendarización se hizo uso de la librearía apschesuler de Python.

Se creo una función en la que se colocó todo el código a ejecutarse cada 1er día de cada mes a las 10 am.

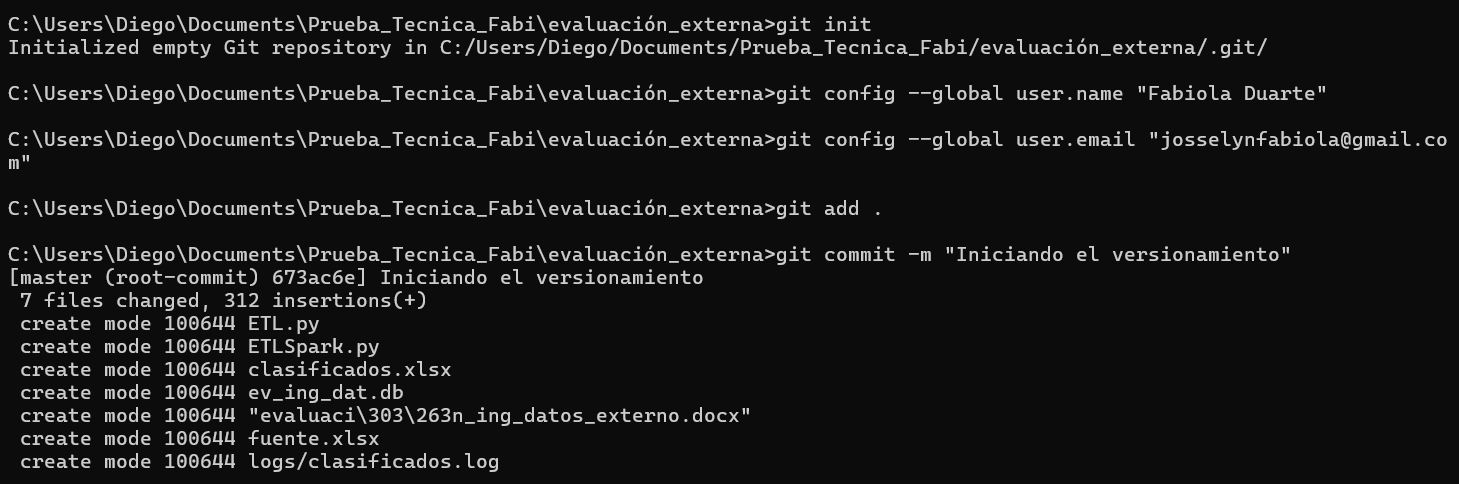


Y se creo el objeto encargado de programar la ejecución.



1. Versionamiento usando GIT

Se preparo el repositorio local para hacer uso de GIT en el versionamiento del proyecto.

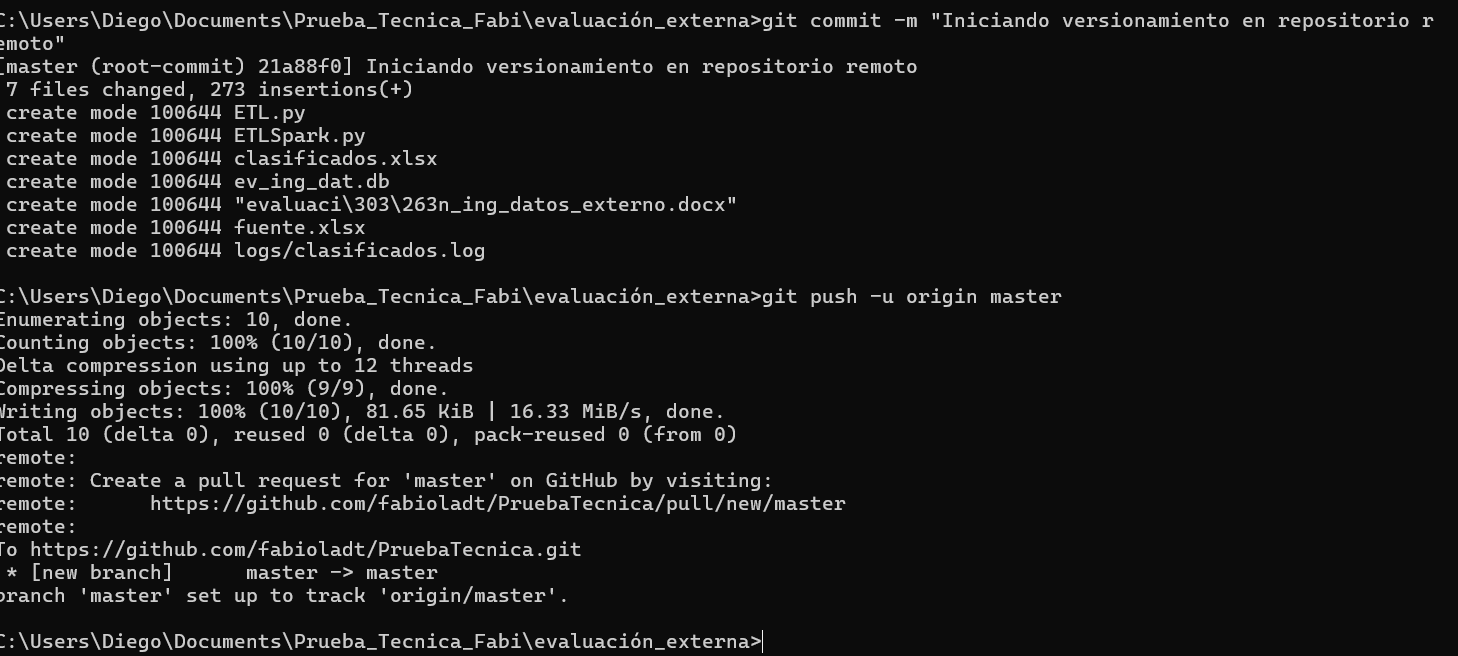


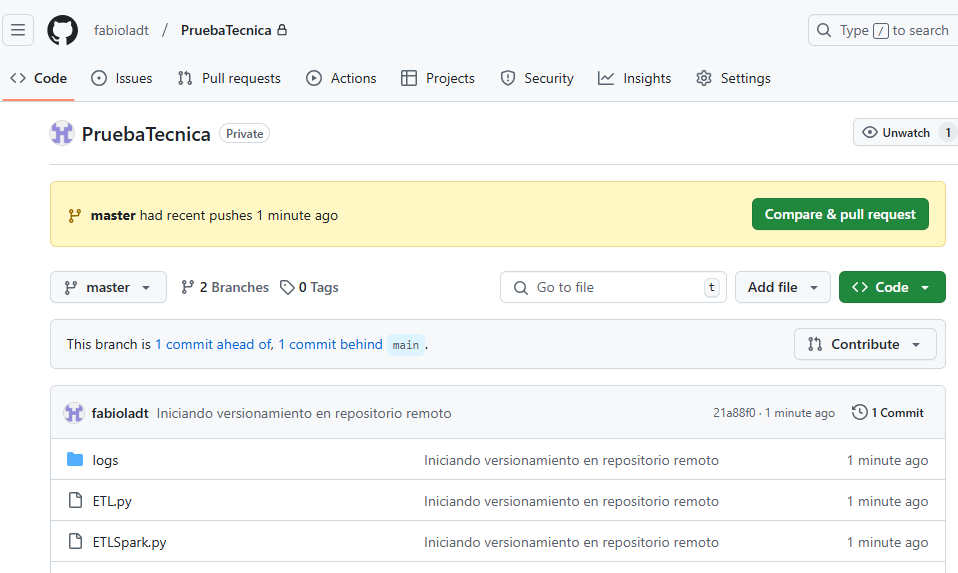
Se creo un repositorio remoto en GitHub

<https://github.com/fabioladt/PruebaTecnica.git>

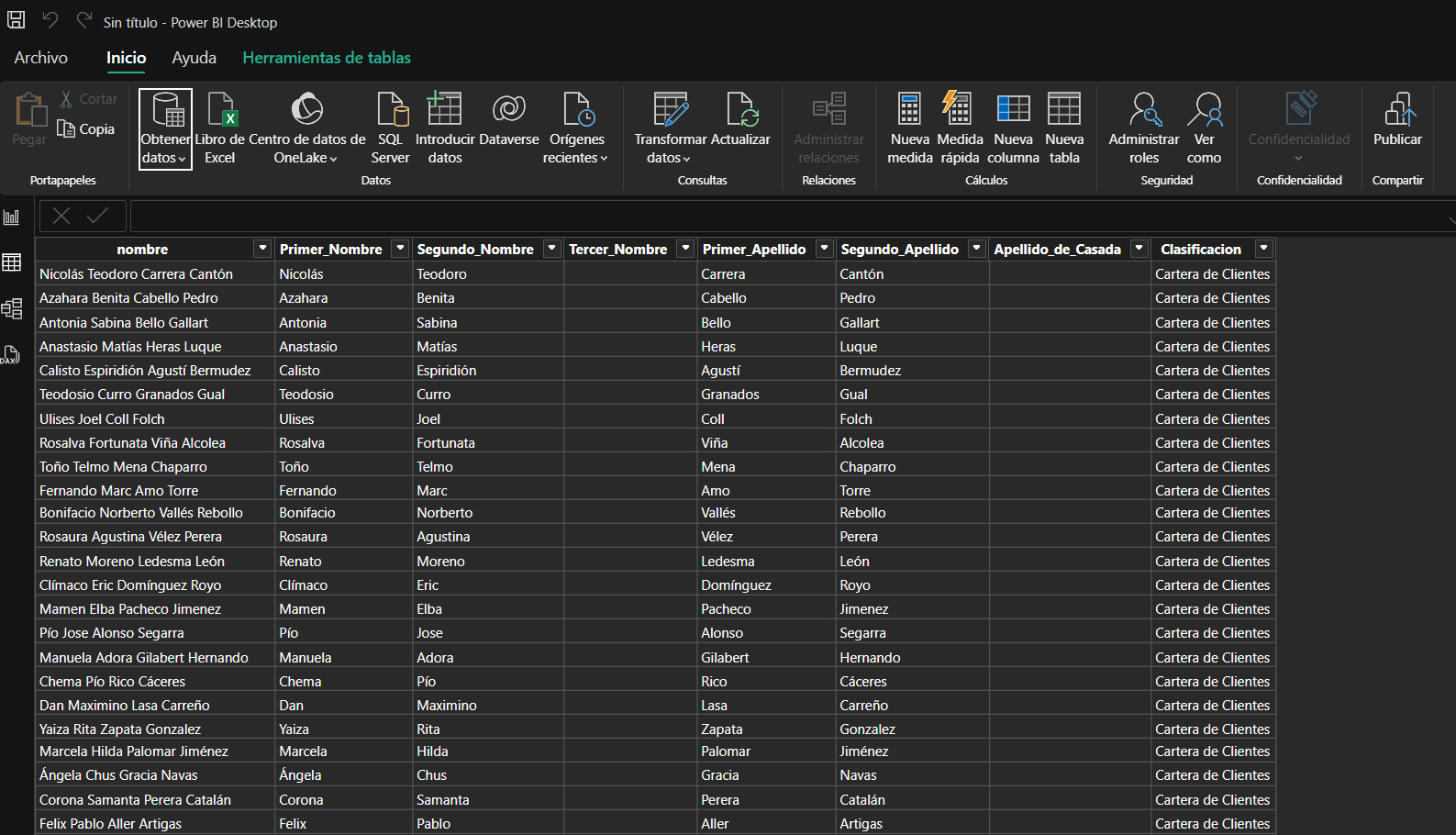
Se asocio el repositorio local con el remoto

Y se hicieron los commits en el repositorio remoto





1. Resultado



# Carga de listas de control ONU y OFAC.

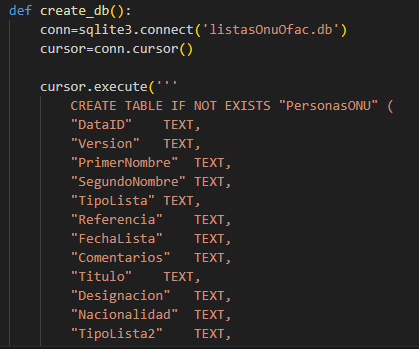
Se tienen dos URLs y se requiere un proceso que cargue esta información en diferentes tablas relacionales según considere.

ONU: <https://scsanctions.un.org/resources/xml/en/consolidated.xml>

OFAC: <https://www.treasury.gov/ofac/downloads/sdn.xml>

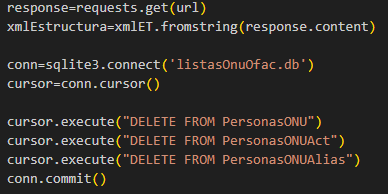
* Analizar la estructura de los archivos xml y considerar que estos son actualizados de forma periódica.
* Crear las tablas y relaciones que considere necesario.
* Utilizar Python y sqlite.
* Compartir evidencias del funcionamiento, scripts usados (Python, sql, etc)
* Versionar el proceso usando git

1. Se ha creado una base en SQLite para este proceso, en dicha base se crearan las tablas necesarias.

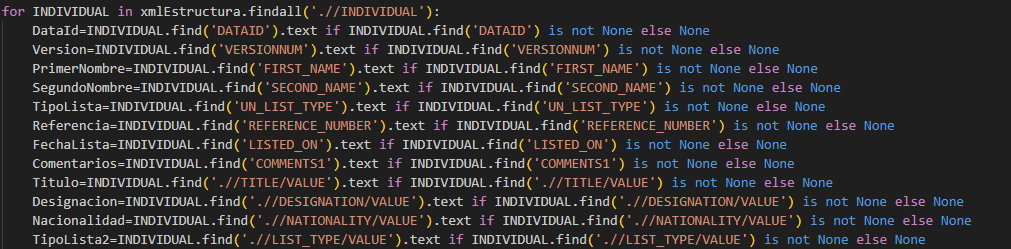


1. Se ha definido una función para cargar primero el archivo de la ONU.

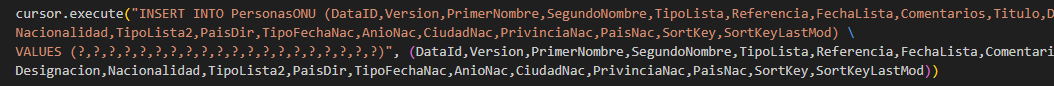
Primero se hace una limpieza de las tablas



Luego se hace un recorrido por el XML para extraer la información de cada persona



Y se procede al insert de cada registro extraído



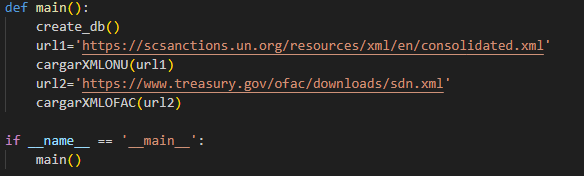
1. Se ha definido una función para cargar primero el archivo de la OFAC.

Se hace la lectura del archivo XML y limpieza de las tablas  


Luego se recorre el XML para recuperar cada registro.



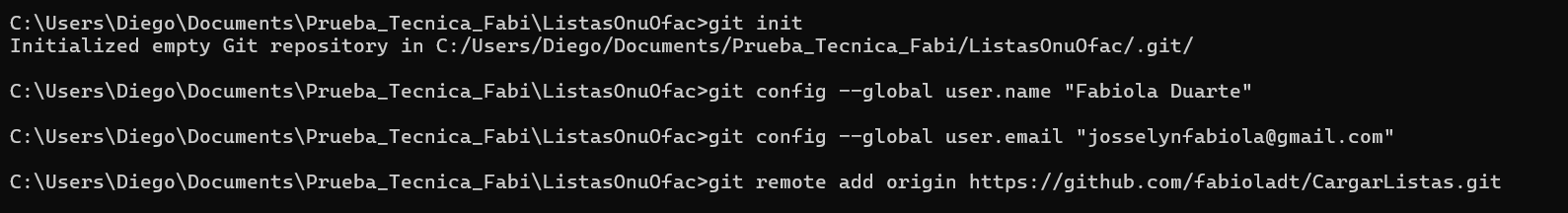
1. Luego se define la función main() para la ejecución de las funciones anteriores.

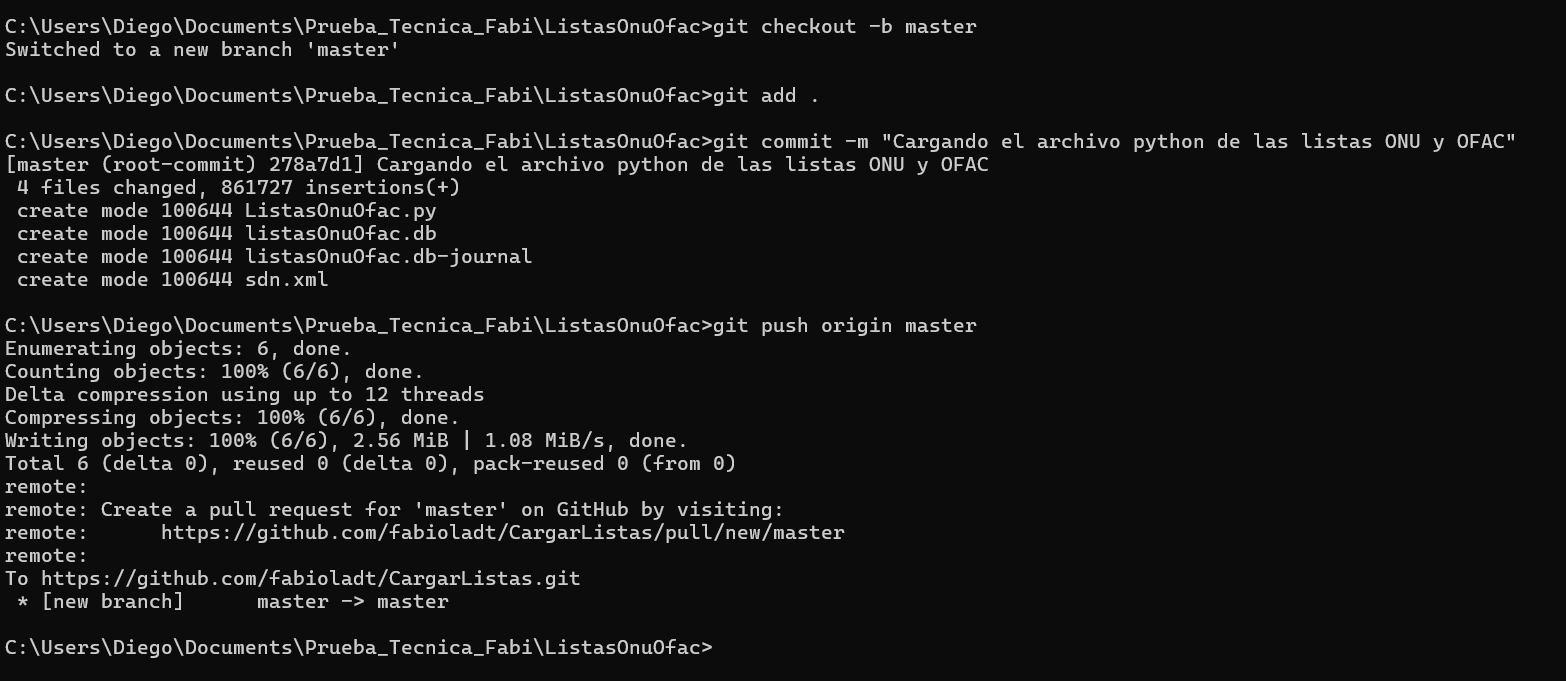


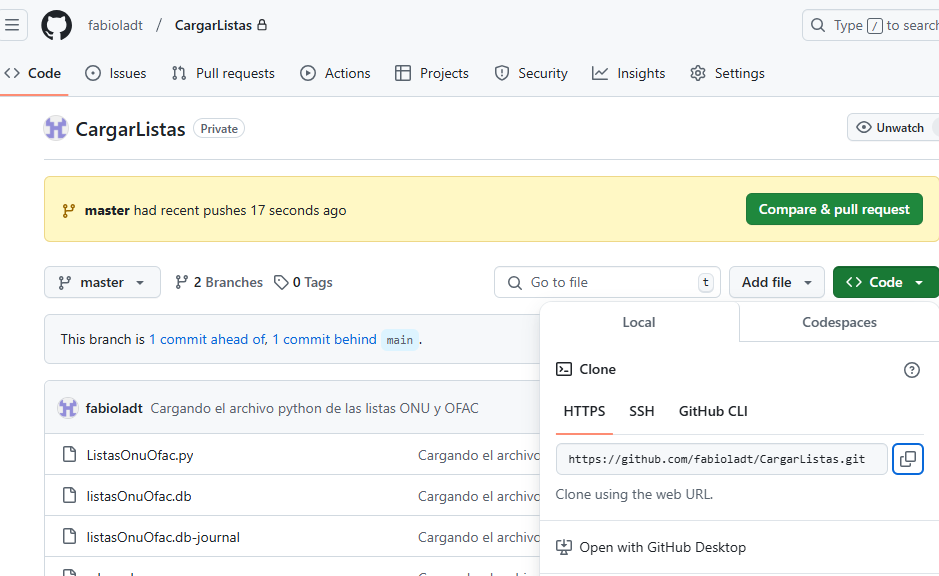
1. Se aplico versionamiento con Git

Se preparo el repositorio local y se vinculó con repositorio remoto. Y se subieron los cambios.

<https://github.com/fabioladt/CargarListas.git>

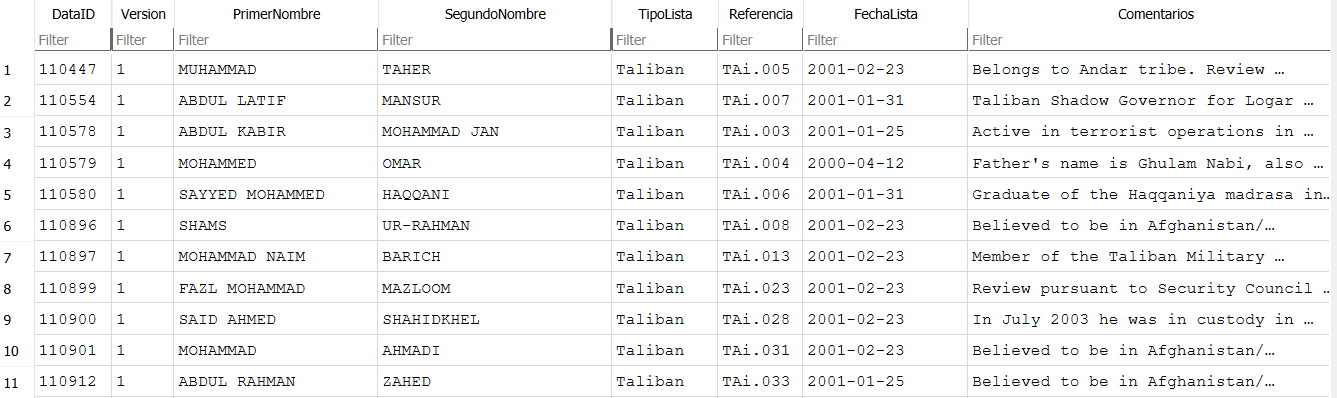


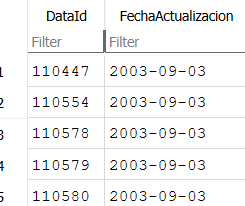


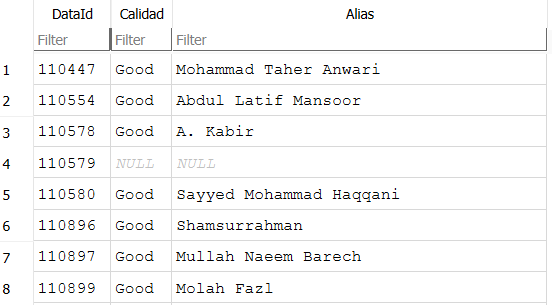


1. Resultado de la extracción.

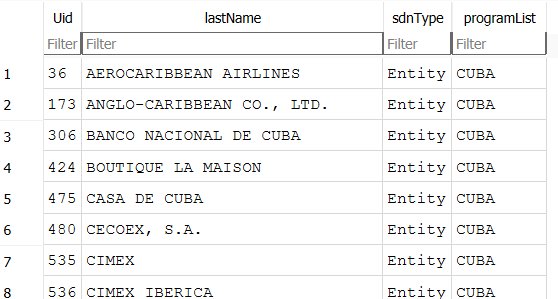
**Archivo ONU.**

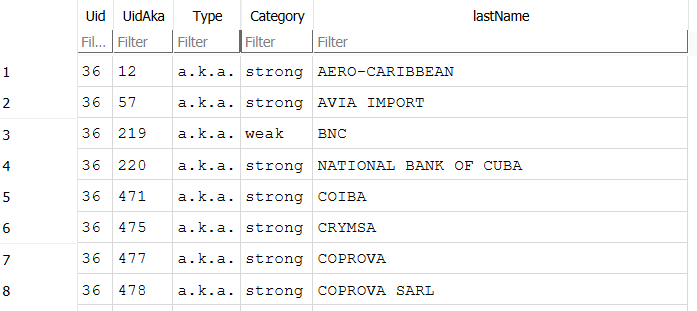
****

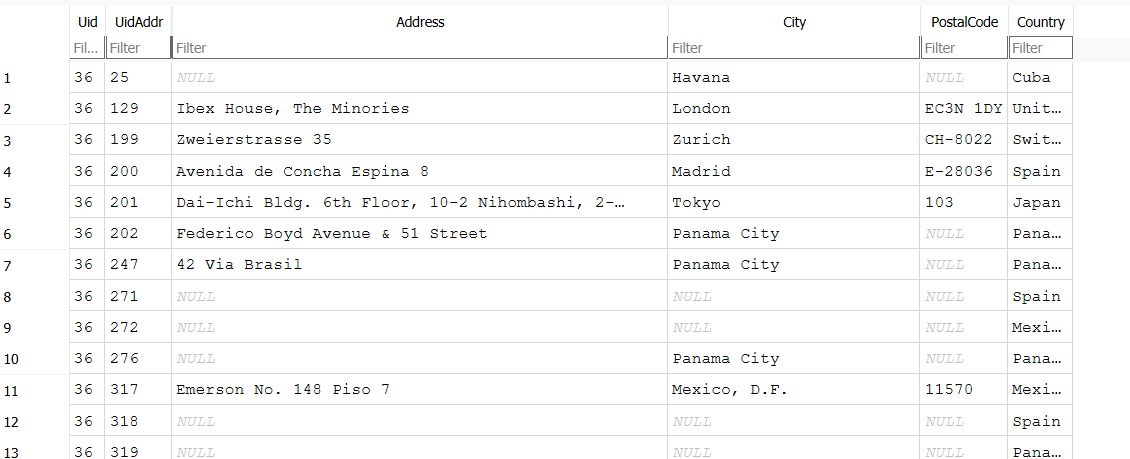
****

****

**Archivo OFAC.**

****

****

****