# CHALLENGE TRATAMIENTO DE INFORMACIÓN

***BUENOS AIRES 29 DE JULIO DE 2022***

***Programas a instalar***

- Python 3.9 (o posterior -https://www.python.org/downloads/-)
- DB Browser for SQLite (https://sqlitebrowser.org/dl/)

***Librerías a instalar***

- pandas
- sqlite3
- cryptography

***Enfoque***

Esta aplicación está diseñada para poder consumir de manera parcial la información entregada por el proveedor. Por reglamentos de distintos entes reguladores, decidí encriptar los datos sensibles tales como "Cuenta", "Nro. de tarjeta" y "CVV" de los datos extraídos.
De igual forma, si algún equipo en particular de riesgos llegara a necesitar esta información, se generó otro programa para poder desencriptarla.

***Descripción del funcionamiento de la aplicación***

- Para el correcto funcionamiento del aplicativo, debe estar el programa "Ejecutar.py" y encrypt.py en el mismo directorio.

Para extraer los datos, se debe iniciar el programa "Ejecutar",
Lo primero que realiza este aplicativo es generar una Key "universal.key" (se usara para encriptar lo datos sensibles) y dejarla en el repositorio donde se está corriendo. Luego toma la información requerida de la siguiente URL:

- https://62433a7fd126926d0c5d296b.mockapi.io/api/v1/usuarios

Una vez que que están procesados los datos, realiza una conexión a la base de datos que llamará "DataEntry.db" en el recurso donde esté ubicado el programa. Si la base ya se encuentra generada, se conecta a la misma.
Dentro de "DataEntry.db" crea una tabla llamada "Informacion_Rescatada", donde vuelca todos los datos extraídos desde la URL consumida. Si la tabla ya estaba generada, reemplaza los datos por los nuevos descargados.

# CHALLENGE TRATAMIENTO DE INFORMACIÓN

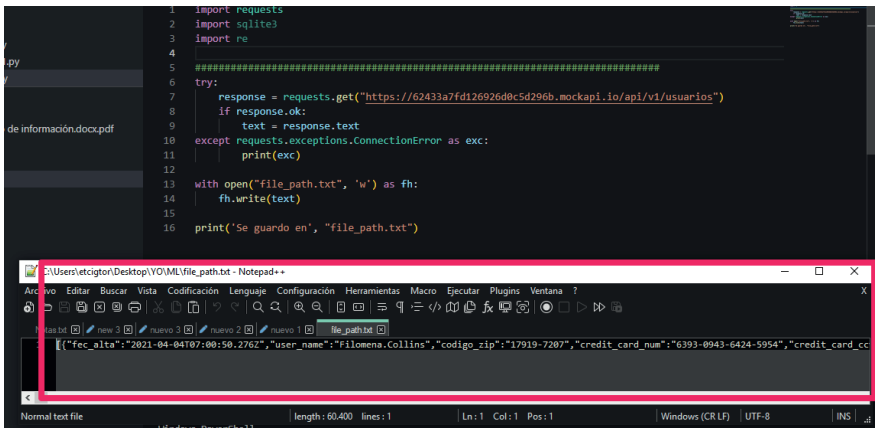
***BUENOS AIRES 29 DE JULIO DE 2022***

## *Desencripción*

Este programa se le otorgará solamente a aquellos que tengan los privilegios para visualizar los datos sensibles. Los programas "Desencriptar.py" y "decrypt.py" deben estar en el mismo directorio donde se ejecutó originalmente la extracción de datos.
Se deberá iniciar el programa "Desencriptar.py", el mismo tomará el archivo "universal.key" y lo utilizará para desencriptar los 3 campos encriptados. Luego los vuelca en una nueva tabla llamada "Datos_Desencriptados" dentro de la DB "DataEntry.db".

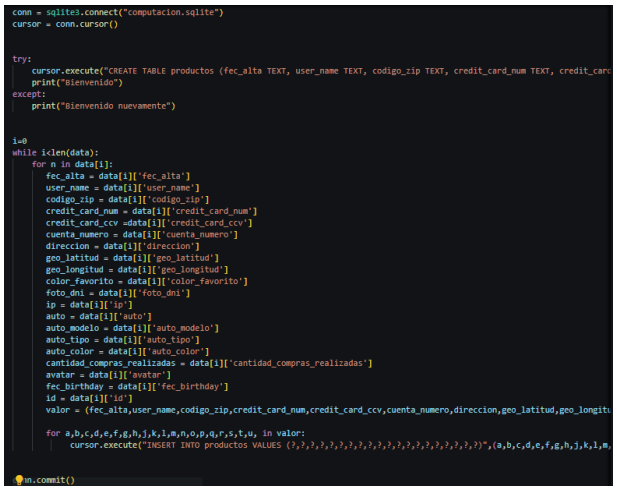
## *Obstáculos dentro del Challenge*

Al empezar a escribir el código, para la extracción de los datos utilicé varias librerías, pero cada una me trajo un inconveniente distinto.




Al utilizar la librería "request", me traía los datos, pero todos en una misma línea y como un diccionario con varias listas dentro. Me fue muy dificil volcarlos a la db.

Luego utilicé la librería "urllib", los datos los traía con un formato mas manejable, pero al armar todo el código para insertarlo en la base de datos mostraba un error que no pude solucionar.
Este indicaba que había muchos valores para procesar y el programa hacía un intento por crear la DB, sin éxito, ya que quedaba inaccesible.
Finalmente logré solucionar estos obstaculos utilizando la librería "pandas" y modificando su formato (Dataframe) a string para poder volcarlo sobre la base generada en sqlite.



# CHALLENGE TRATAMIENTO DE INFORMACIÓN

***BUENOS AIRES 29 DE JULIO DE 2022***

***Análisis de riesgo***

En un entorno productivo real, la aplicación debería residir en un server con restricción de acceso y la conexión debería ser contra un servidor de DB, también con el acceso restringido por AD.
En este challenge no lo estoy aplicando por el costo e infraestructura que esto requiere.
Se abordó equilibrando el costo-beneficio, sin olvidarse que los datos a mi entender sensibles deben permanecer seguros.