Nombre: Ana Paula Vertedor Conde **Fecha:** *30/10/2025*

Ciclo: *Desarrollo de Aplicaciones Web* / ***Asignatura:*** *BLOCKCHAIN*

UNIDAD 1: BLOCKCHAIN - INTRODUCCIÓN
CASO PRÁCTICO 1: CIFRADO

1. Introducción

El objetivo de este caso práctico era crear una herramienta capaz de cifrar y descifrar texto de forma reversible, aplicando el concepto de integridad de datos.
Para ello desarrollé dos programas en Python 3 basados en el cifrado César, un método sencillo que desplaza las letras del alfabeto según una clave numérica.

El proyecto fue también una excusa para experimentar y divertirme un poco: lo ambienté con estética *hacker noventera* inspirada entre *Hackers (1995)* y Matrix (1999).

- El codificador (codif.py) representa al hacker *CrashOverride*, encargado de cifrar los datos.

- El decodificador (decod.py) representa a *AcidBurn*, quien logra revertir el cifrado y recuperar la información.

Así, además de cumplir con el ejercicio de cifrado, el resultado tiene un toque más personal y creativo.

2. Descripción del proyecto

El proyecto consta de dos scripts:

- codif.py → cifra el texto usando una clave numérica.
    - Permite modo normal (sin animación) y modo cinematográfico (CrashOverride).
    - Guarda los resultados en la carpeta mision/:
        - input.txt (texto original)
        - output.txt (texto cifrado)
        - log.txt (fecha, clave y hash SHA-256)
- decod.py → descifra el texto cifrado.
    - También tiene un modo con animación de *AcidBurn*, que simula el proceso de rastreo y recuperación de datos.

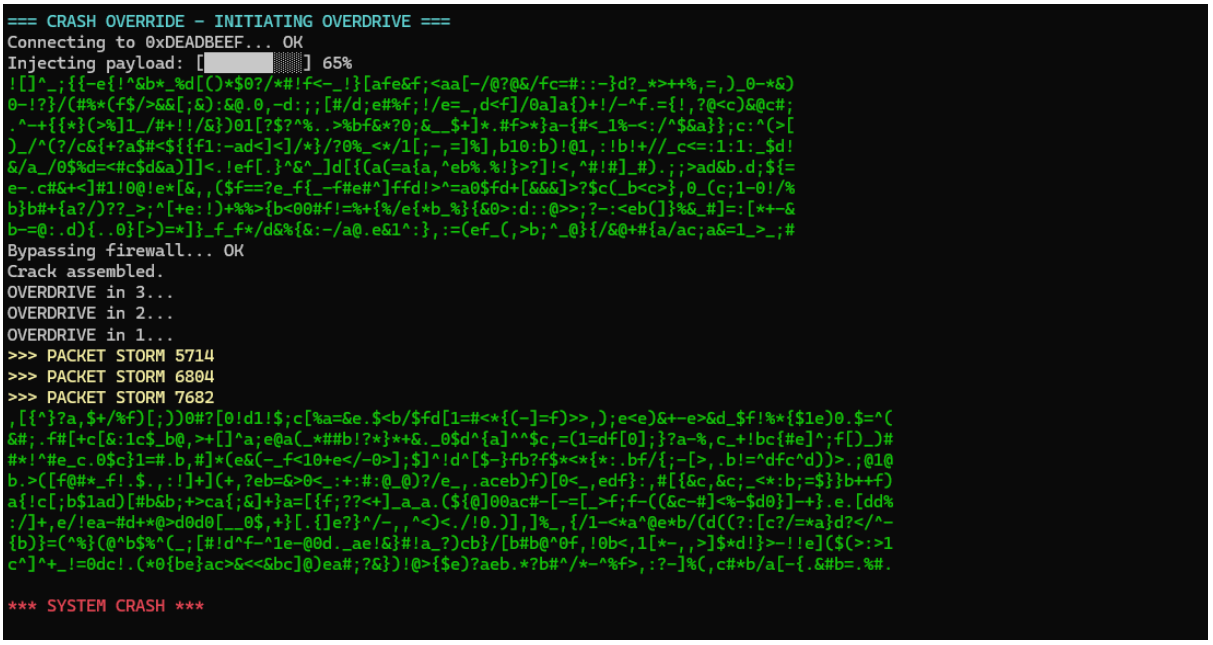
Ambos scripts aceptan argumentos desde terminal mediante argparse y pueden leer mensajes directos (-m) o desde archivos (-f).

3. Funcionamiento del codificador

Al ejecutar el programa con:

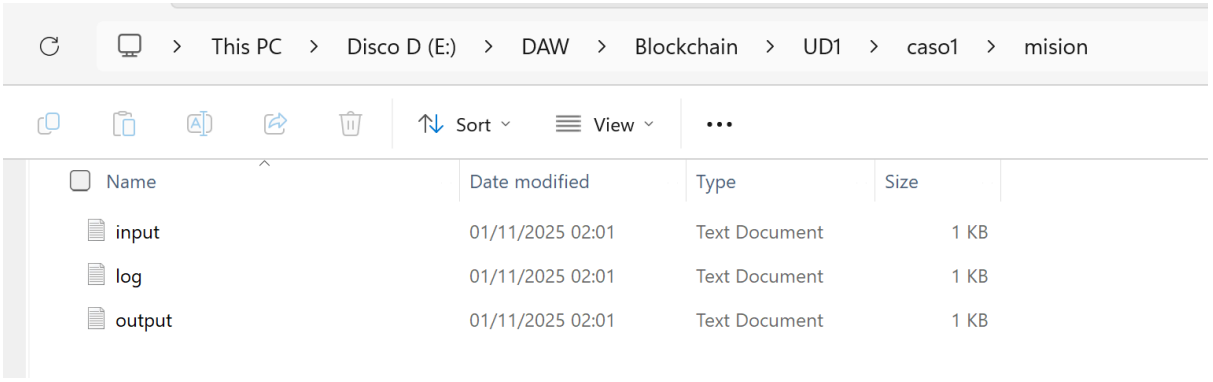
```
python codif.py -m "crashoverride" -k 707
```

se inicia la secuencia visual de *CrashOverride*.
El texto se cifra con una clave (707 ≡ 5 en el cifrado César) y se guardan los archivos correspondientes en mision/.



Captura 1: Ejecución del codificador con animación.

Captura 2: Ejecución del codificador sin animación.



Captura 3: Contenido de la carpeta mision/.

```
[2025-10-31 23:33:13.823715] Clave=707 | Hash=3a69cd3af12b5275
[2025-11-01 00:15:36.369142] Clave=707 | Hash=0f5c1c2f5581c49f
[2025-11-01 00:20:19.076968] Clave=707 | Hash=0f5c1c2f5581c49f
[2025-11-01 01:24:47.872079] Clave=707 | Hash=0f5c1c2f5581c49f
[2025-11-01 01:25:09.572961] Clave=707 | Hash=3a69cd3af12b5275
[2025-11-01 01:26:12.842206] Clave=707 | Hash=0f5c1c2f5581c49f
[2025-11-01 01:27:30.502767] Clave=707 | Hash=3a69cd3af12b5275
[2025-11-01 01:45:58.376273] Clave=707 | Hash=3a69cd3af12b5275
[2025-11-01 02:01:40.725443] Clave=707 | Hash=3a69cd3af12b5275
```

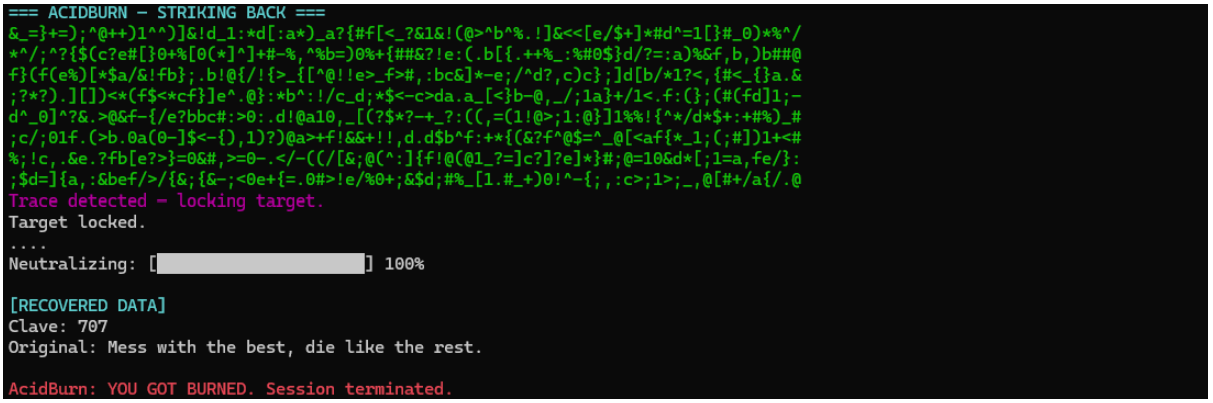
Captura 4: Extracto del archivo log.txt.

4. Funcionamiento del decodificador

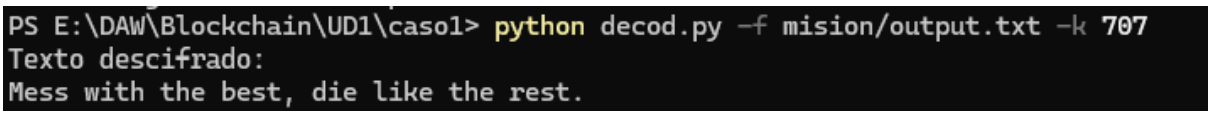
El proceso inverso se realiza con:

```
python decod.py -f mision/output.txt -k 707 --anim
```

Esto muestra la animación de *AcidBurn*, quien "rastreando el ataque" recupera el texto original.



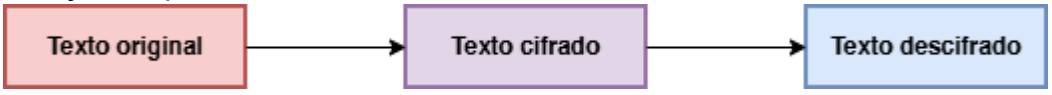
Captura 5: Ejecución del decodificador con animación y texto de salida.



Captura 6: Ejecución del decodificador sin animación y texto de salida.

5. Resultados

El flujo completo demuestra un cifrado reversible:



Captura 7: Proceso de cifrado y descifrado del mensaje.

y cómo puede registrarse un **hash** para comprobar la integridad del mensaje, concepto que también se aplica en sistemas basados en blockchain.

Video del código en acción en: https://youtu.be/urLft32FYwE

6. Conclusión

Este ejercicio me permitió comprender de forma práctica la diferencia entre cifrado y hashing, y cómo ambos contribuyen a la integridad de los datos.

Además de reforzar el uso de Python para leer y escribir archivos, fue una oportunidad para practicar argumentos por terminal y darle un toque creativo al resultado con animaciones tipo consola.

7. Recursos consultados

- Curso Python de Udemy.
- *¿Cómo programar un cifrado de César en Python? Aprende en 9 minutos* (Youtube): https://www.youtube.com/watch?v=3jeFGqLWeE8
- *Cifrado de César en Python* (Youtube): https://www.youtube.com/watch?v=DhSInnOlK1c