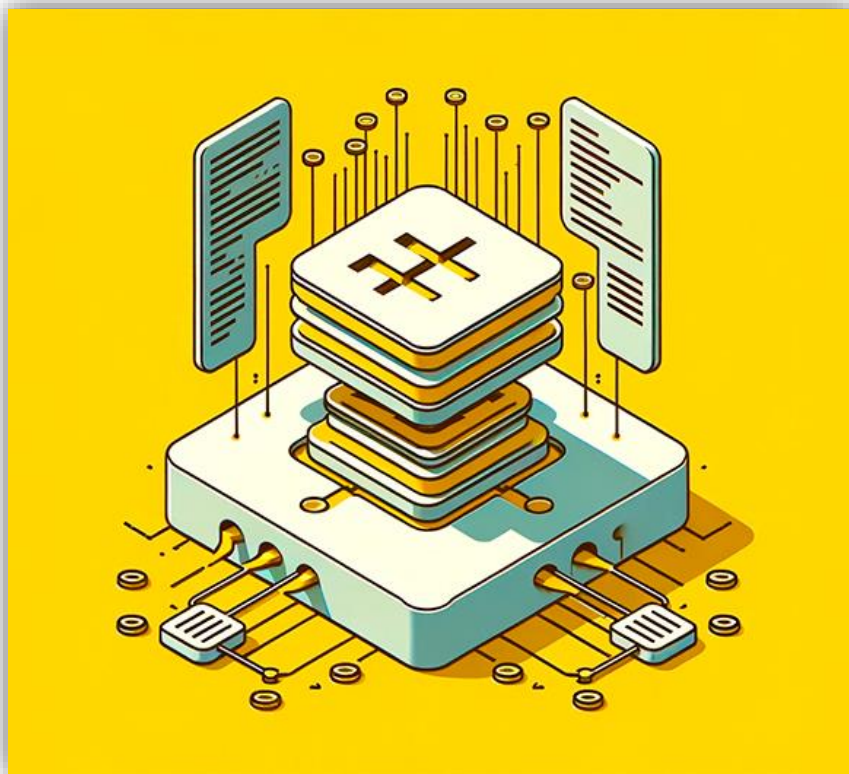


VERIFICACIÓN DE LA **INTEGRIDAD DE ARCHIVOS** **MEDIANTE FUNCIONES HASH**



Leidy Pasaca Herrera

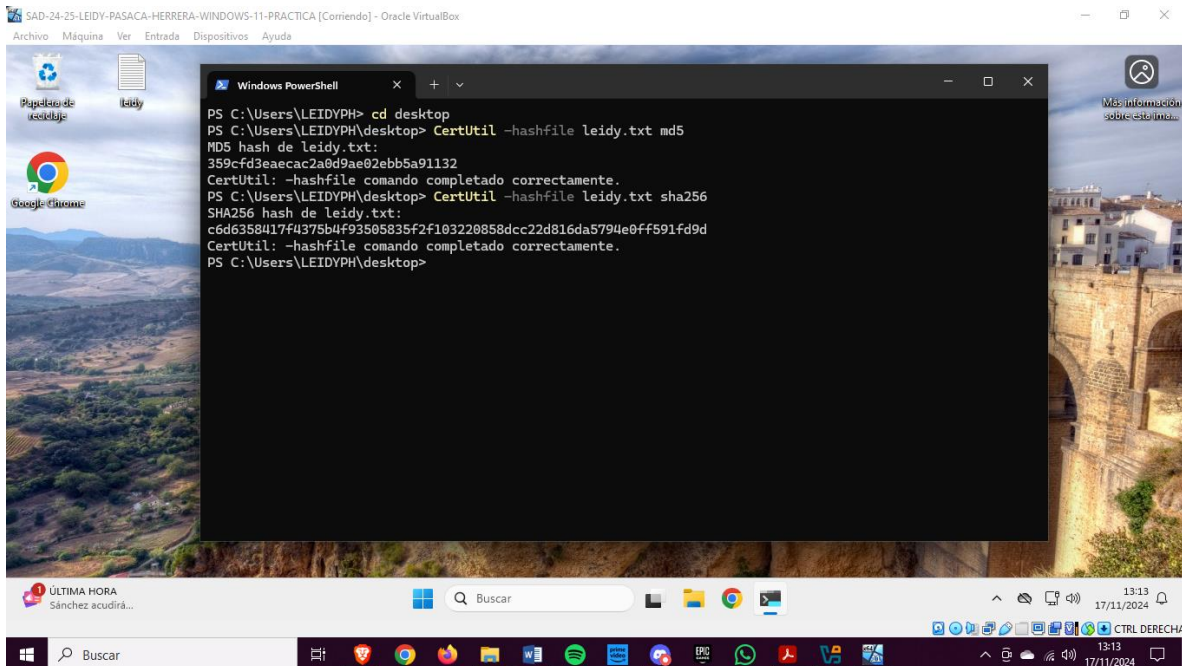
2º Administración de Sistemas Informáticos en Redes



ÍNDICE

1.- PRÁCTICA EN WINDOWS	3
2.- PRÁCTICA EN LINUX.....	6
3.- COMPARACIÓN DE ALGORITMOS	7
4.- INFORME FINAL	7

1.- PRÁCTICA EN WINDOWS



```
PS C:\Users\LEIDYPH> cd desktop
PS C:\Users\LEIDYPH\desktop> CertUtil -hashfile leidy.txt md5
MD5 hash de leidy.txt:
359cfd3eacac2a0d9ae02ebb5a91132
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\LEIDYPH\desktop> CertUtil -hashfile leidy.txt sha256
SHA256 hash de leidy.txt:
c6d6358417f4375b4f93505835f2f103220858dcc22d816da5794e0ff591fd9d
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\LEIDYPH\desktop>
```

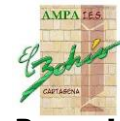
1) Para empezar la parte en Windows, abriremos un terminal y entraremos en la ruta donde hayamos creado el archivo en mi caso es “Desktop” y desde allí ejecutaremos los siguientes comandos:

- CertUtil -hashfile leidy.txt md5

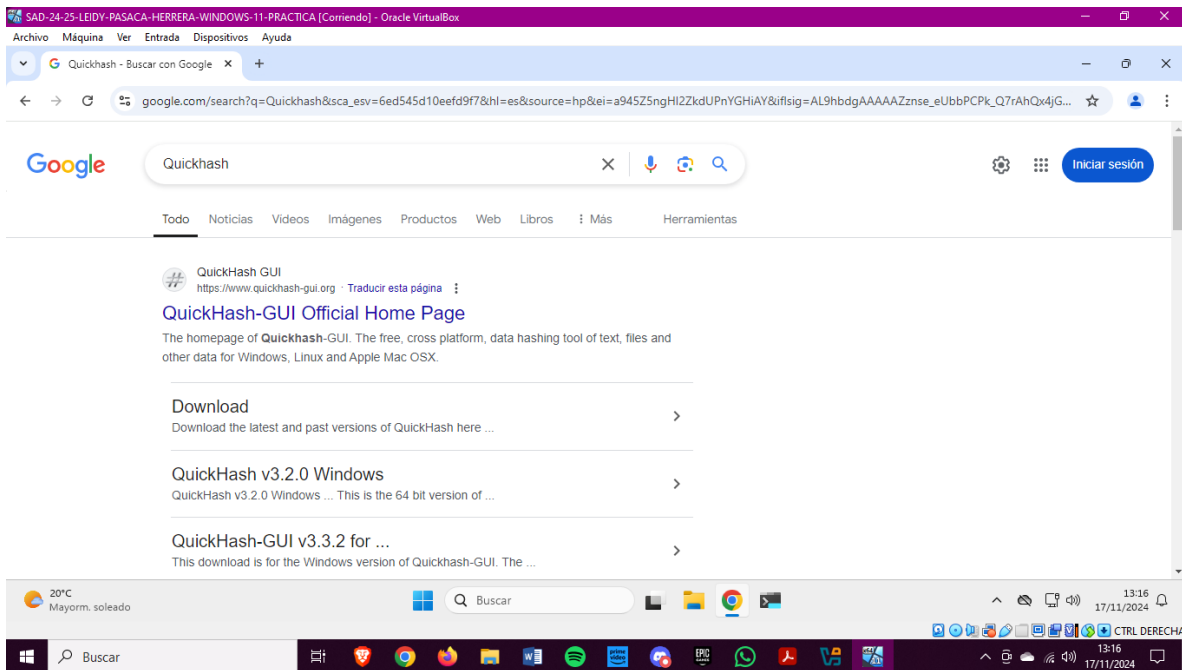
- CertUtil -hashfile leidy.txt sha256

NOTA: Al crear o descargar un archivo que queramos comprobar, nunca dejarlo en blanco pues nos dará error.

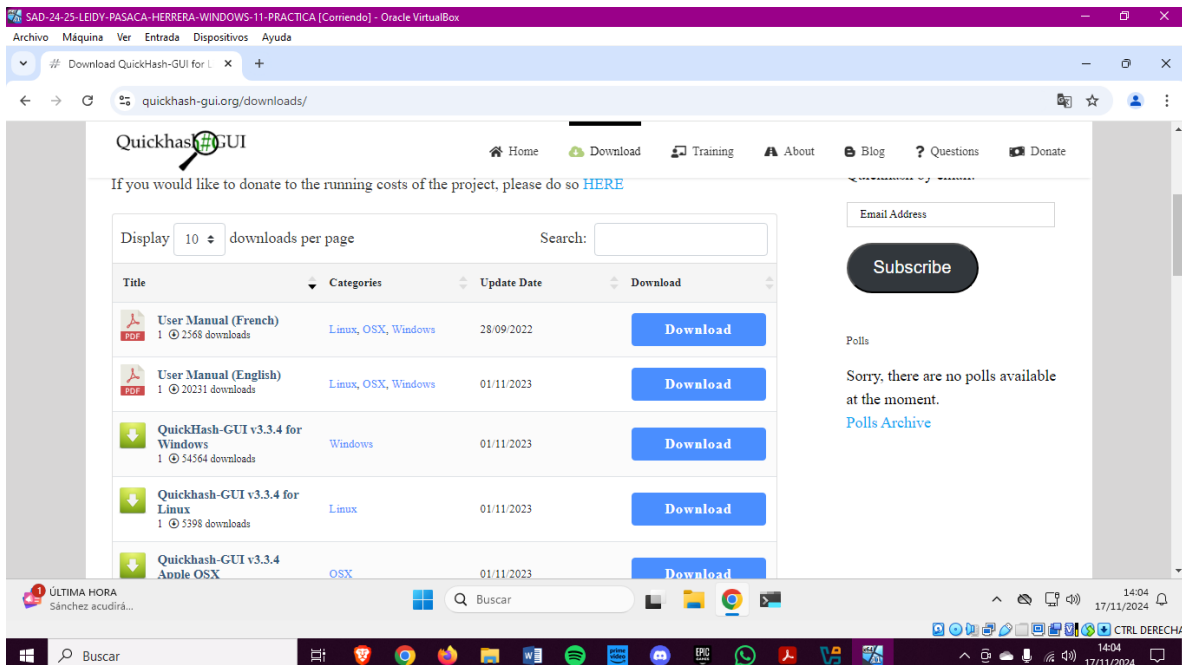
Verificación de la integridad de archivos mediante funciones hash Seguridad y Alta Disponibilidad



Leidy Pasaca Herrera

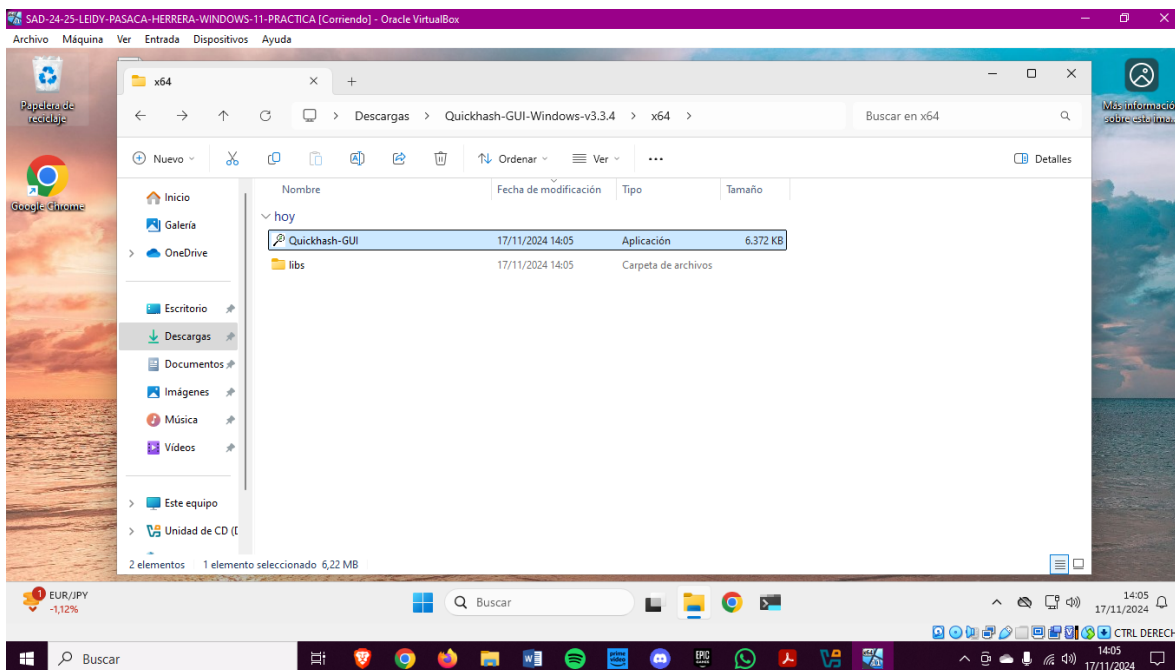


2) Para la parte opcional he elegido Quickhash, para utilizarlo primero debemos ir al buscador y ponemos “Quickhash” y seleccionamos la primera opción.

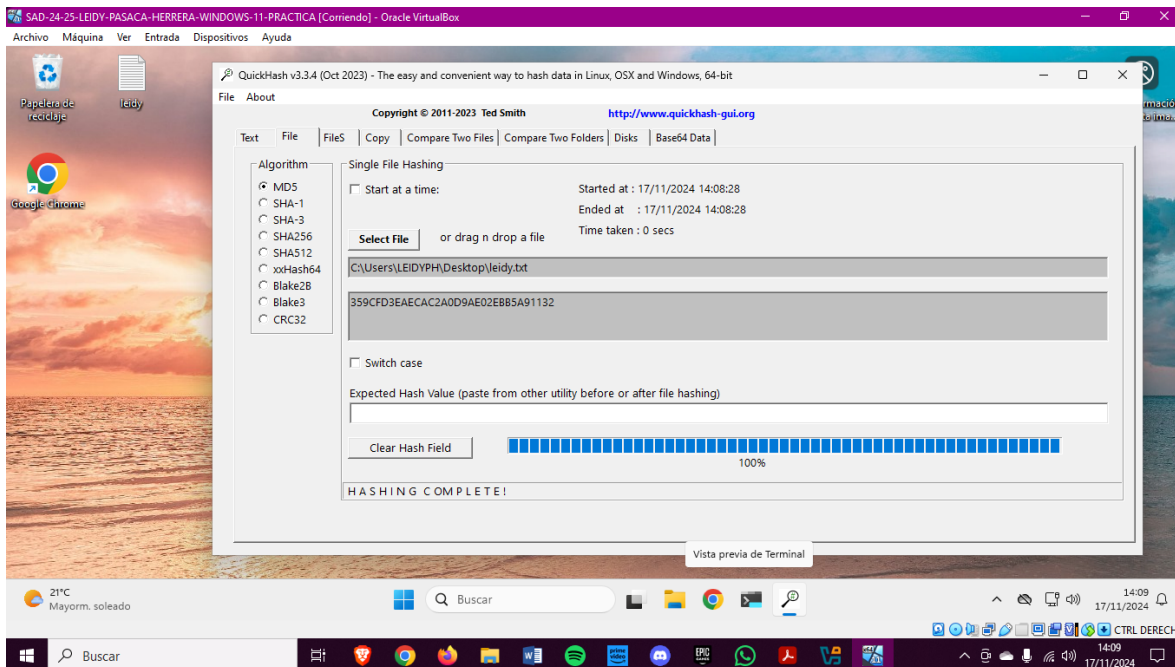


3) Dentro nos iremos a “Download” y nos descargaremos la versión más actual para Windows.

Verificación de la integridad de archivos mediante funciones hash Seguridad y Alta Disponibilidad

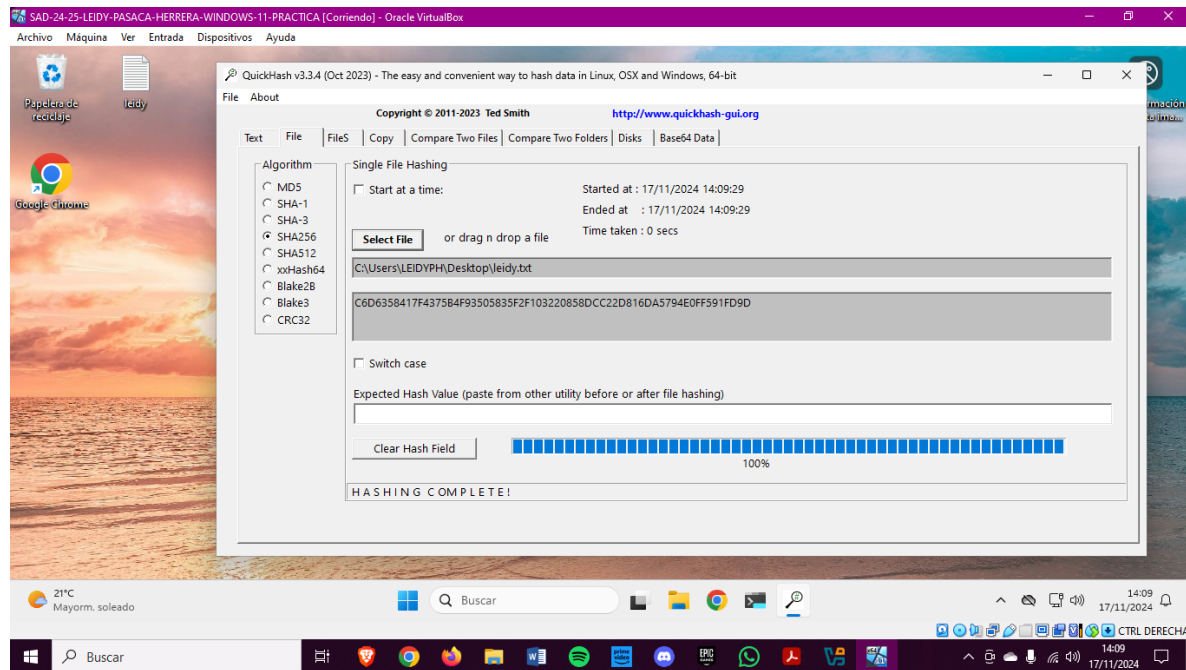


4) Una vez descargado, nos vamos a descargas y lo descomprimos y ejecutamos la primera opción.



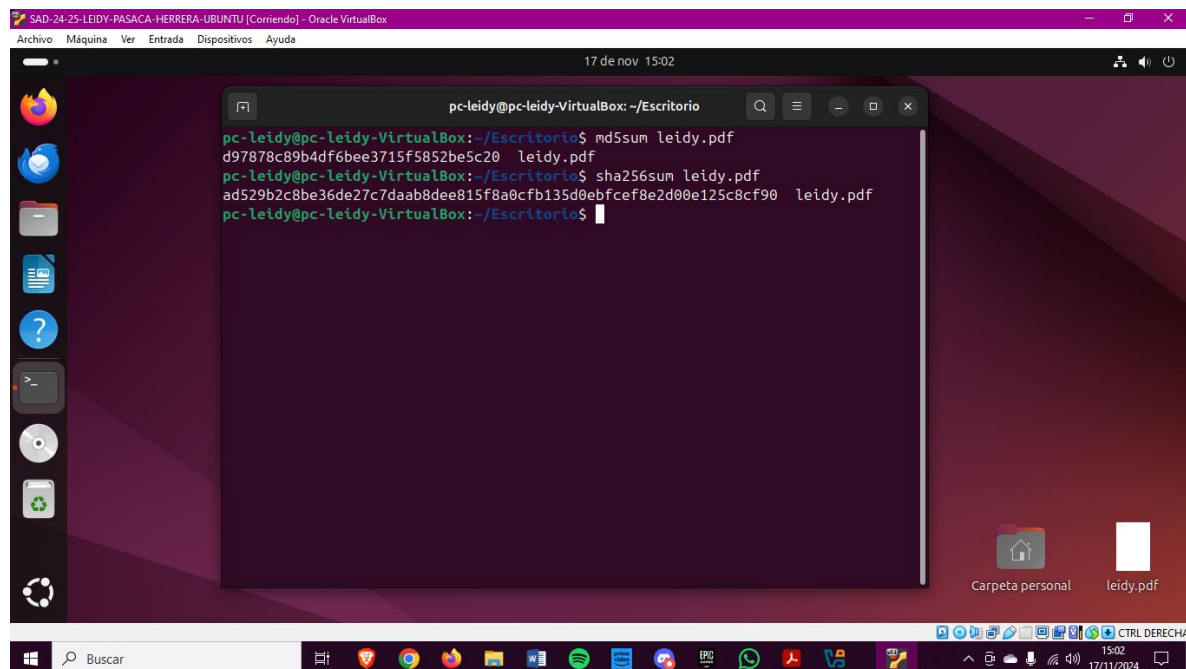
5) Cuando se nos inicie veremos esta pestaña y le daremos a "File" y después a "MD5" y seleccionaremos el archivo y esperamos a que termine.

Verificación de la integridad de archivos mediante funciones hash Seguridad y Alta Disponibilidad



6) Cuando termine seleccionamos la opción de “SHA256” y volvemos a seleccionar el archivo y esperamos a que termine.

2.- PRÁCTICA EN LINUX



1) Para empezar en la practica de Linux, elegimos un archivo y abrimos una terminal en la ruta donde se encuentra dicho archivo. Despues ejecutaremos los siguientes comandos:

- md5sum leidy.pdf

- sha256sum leidy.pdf

3.- COMPARACIÓN DE ALGORITMOS

¿Por qué los algoritmos MD5 y SHA-1 ya no son recomendados para aplicaciones críticas?

- Debido a una vulnerabilidad de colisión criptográfica, los algoritmos MD5 y SHA-1 ya no se consideran seguros para aplicaciones críticas. Estas colisiones ocurren cuando dos entradas diferentes producen el mismo hash, comprometiendo la integridad y autenticidad de los datos.

Una situación real que demuestra el riesgo fue el ataque a los certificados digitales. Un atacante puede crear un certificado falso con la misma firma hash que un certificado legítimo, lo que hace que un navegador o un sistema confiable acepte el certificado falso. Este tipo de ataque se puede utilizar para interceptar mensajes cifrados o robar información personal.

Aunque MD5 no es seguro para aplicaciones de misión crítica, puede ser aceptable en entornos donde la integridad o la seguridad no son importantes. Algunos escenarios incluyen:

1. Comprobaciones de integridad básicas:

MD5 se puede utilizar para detectar corrupción de datos en archivos descargados, por ejemplo, comparando el hash MD5 con el hash proporcionado por el servidor. Esto resulta útil si no se esperan ataques activos.

2. Índices o hashes no cifrados:

En aplicaciones que utilizan hashes como identificadores únicos (como sistemas de bases de datos o de duplicación de archivos), la velocidad y la simplicidad son más importantes que las colisiones. elasticidad.

3. Diseño tradicional:

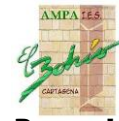
En sistemas más antiguos donde no se requieren altos estándares de seguridad y la actualización de algoritmos tendría un costo o impacto significativo.

4.- INFORME FINAL

Aunque MD5 no es seguro para aplicaciones de misión crítica, puede ser aceptable en entornos donde la integridad o la seguridad no son importantes. Algunos escenarios incluyen:

Comprobaciones de integridad básicas:

MD5 se puede utilizar para detectar corrupción de datos en archivos descargados, por ejemplo, comparando el hash MD5 con el hash proporcionado por el servidor. Esto resulta útil si no se esperan ataques activos.



Índices o hashes no cifrados:

En aplicaciones que utilizan hashes como identificadores únicos (como sistemas de bases de datos o de duplicación de archivos), la velocidad y la simplicidad son más importantes que las colisiones. elasticidad.

Diseño tradicional:

En sistemas más antiguos donde no se requieren altos estándares de seguridad y la actualización de algoritmos tendría un costo o impacto significativo.