

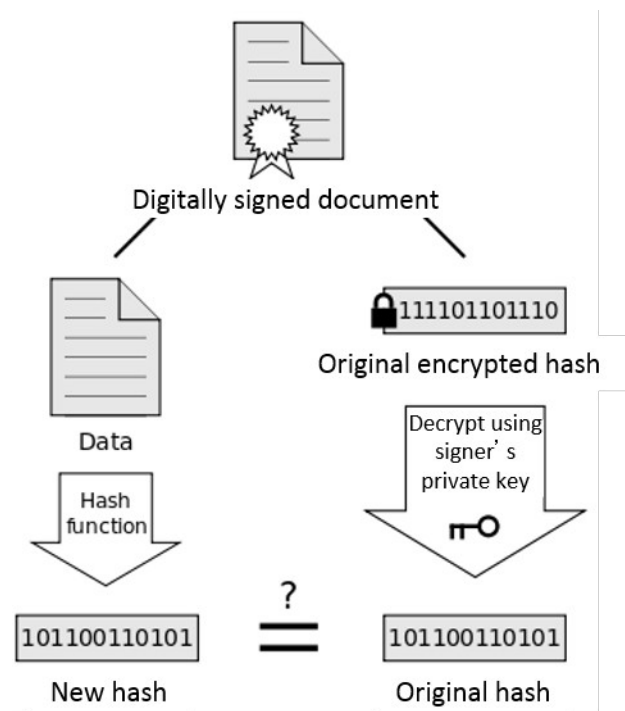
SAD

UNIDAD 3 TAREA 0

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

GONZALO MARTOS CONESA

2º ASIR



ÍNDICE

1. PRÁCTICA EN WINDOWS.....	<u>3</u>
2. PRÁCTICA EN LINUX.....	<u>5</u>
3. COMPARACIÓN DE ALGORITMOS.....	<u>7</u>
4. REFLEXIÓN.....	<u>7</u>

1. PRÁCTICA EN WINDOWS:

Empiezo por el CertUtil. Lo voy a hacer en mi máquina física windows. Voy a usar los dos algoritmos que puso el profesor de ejemplo: **MD5 y SHA256**:

Para ello el comando de PowerShell es ***certutil -hashfile*** seguido de la ***ruta absoluta o relativa*** al fichero y por último ***escribimos el algoritmo, ya sea MD5 o SHA256***. Aporto captura mía ejecutando ambos comandos en mi pc:

```
Administrator: PowerShell
PS C:\Users\gonzalo\taller> certutil -hashfile 'C:\users\gonzalo\taller\Documento no guardado 1.txt' MD5
MD5 hash de C:\users\gonzalo\taller\Documento no guardado 1.txt:
5a1b3207358f35cc8caae89ec708477b
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\gonzalo\taller> certutil -hashfile 'C:\users\gonzalo\taller\Documento no guardado 1.txt' SHA256
SHA256 hash de C:\users\gonzalo\taller\Documento no guardado 1.txt:
d08393a83c452400044d67be52c650643e160080d88b584d47a262f557a2f45a
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\gonzalo\taller>
```

Copio y pego los comandos utilizados y el resultado obtenido. El comando está en negrita y con itálica, el resultado está normal:

certutil -hashfile 'C:\users\gonzalo\taller\Documento no guardado 1.txt' MD5

MD5 hash de C:\users\gonzalo\taller\Documento no guardado 1.txt:

5a1b3207358f35cc8caae89ec708477b

CertUtil: -hashfile comando completado correctamente.

certutil -hashfile 'C:\users\gonzalo\taller\Documento no guardado 1.txt' SHA256

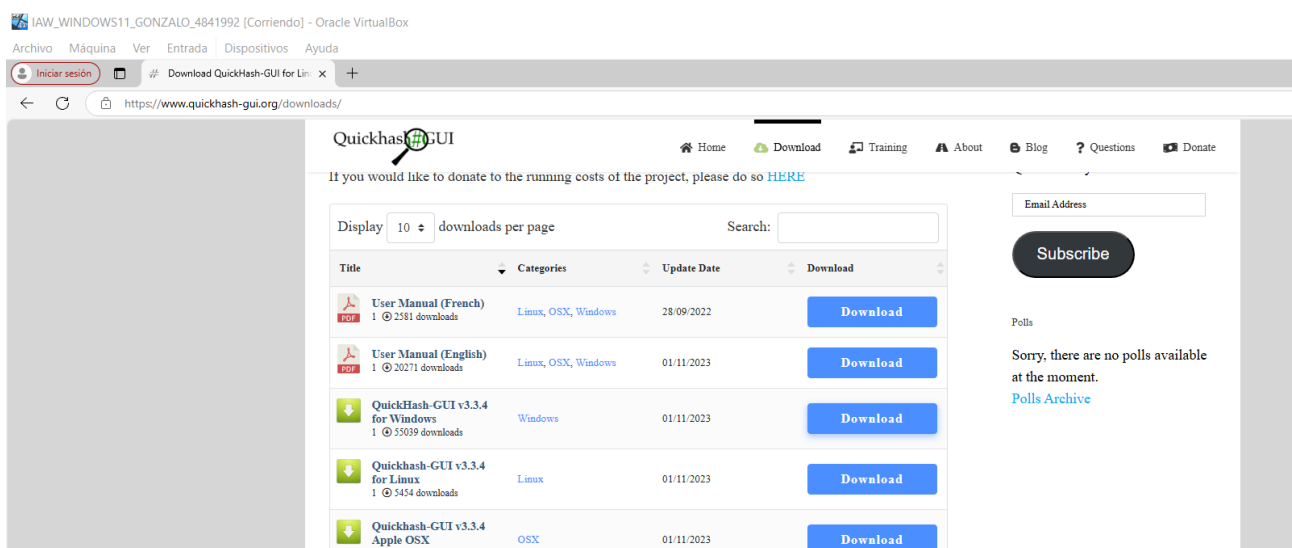
SHA256 hash de C:\users\gonzalo\taller\Documento no guardado 1.txt:

d08393a83c452400044d67be52c650643e160080d88b584d47a262f557a2f45a

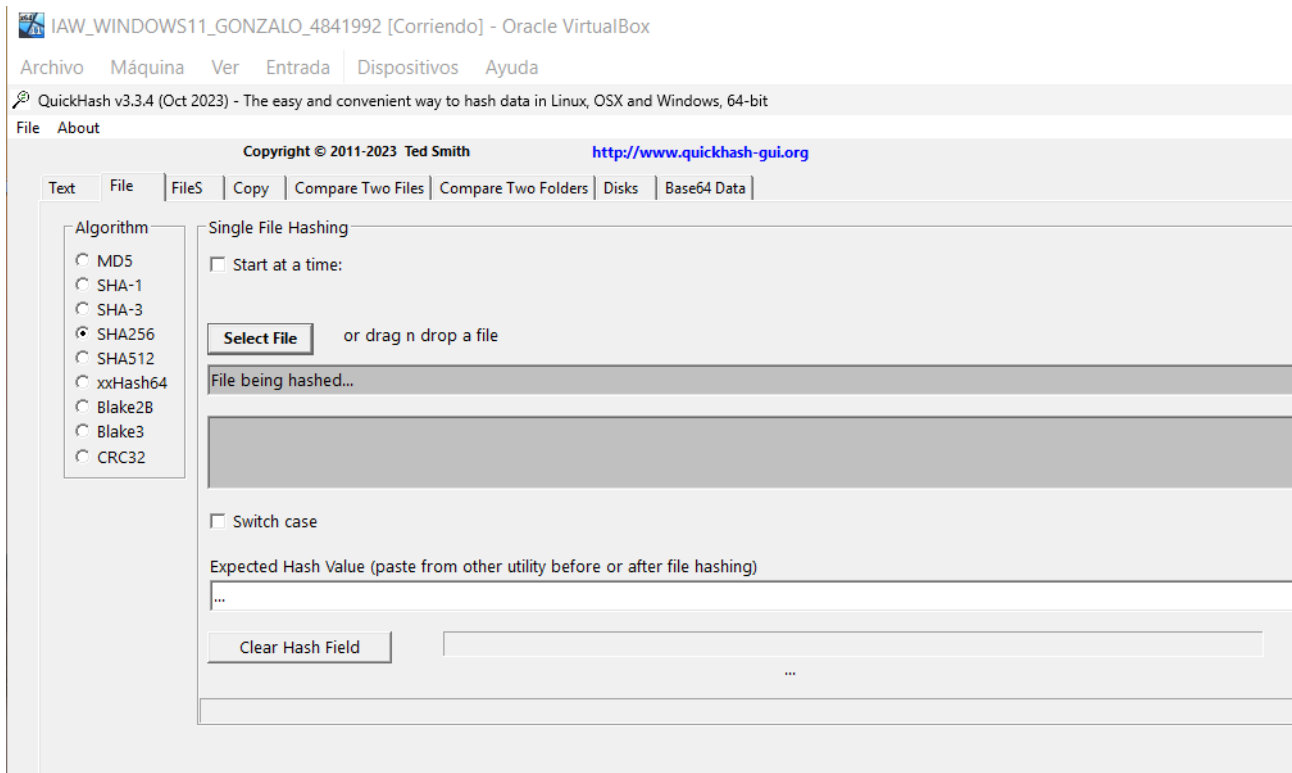
CertUtil: -hashfile comando completado correctamente.

El ejercicio opcional ya lo hago en máquina virtual para no instalar cosas que no usaré en la máquina física:

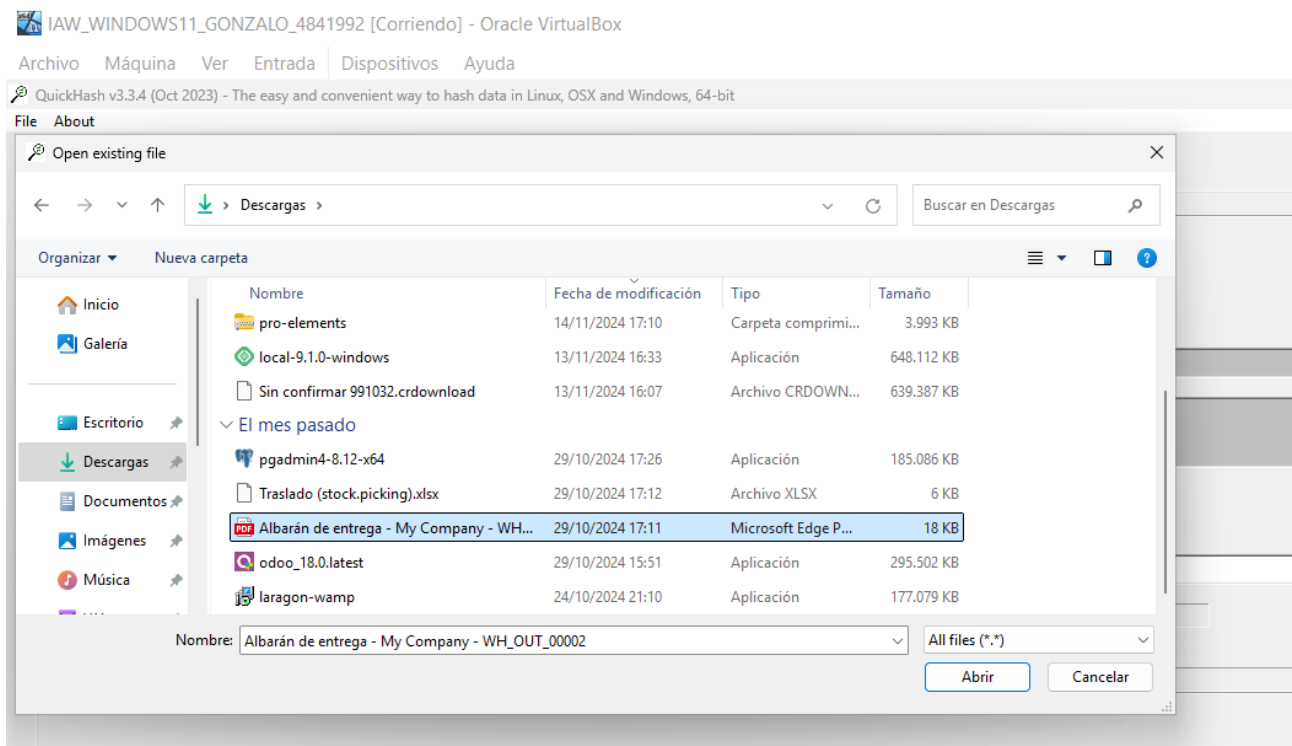
Me descargo la herramienta **QuickHash GUI** desde la web oficial:



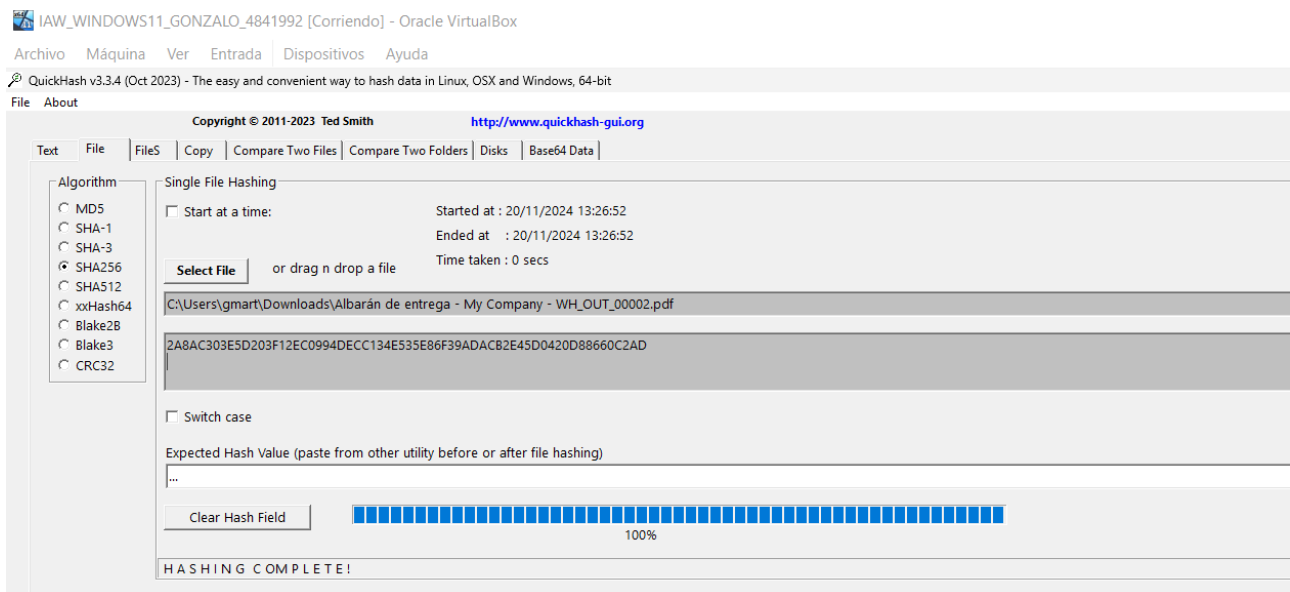
Para usarla, marcamos el algoritmo que queremos, le damos a file, seleccionamos select file y elegimos la que queremos sacar el hash:



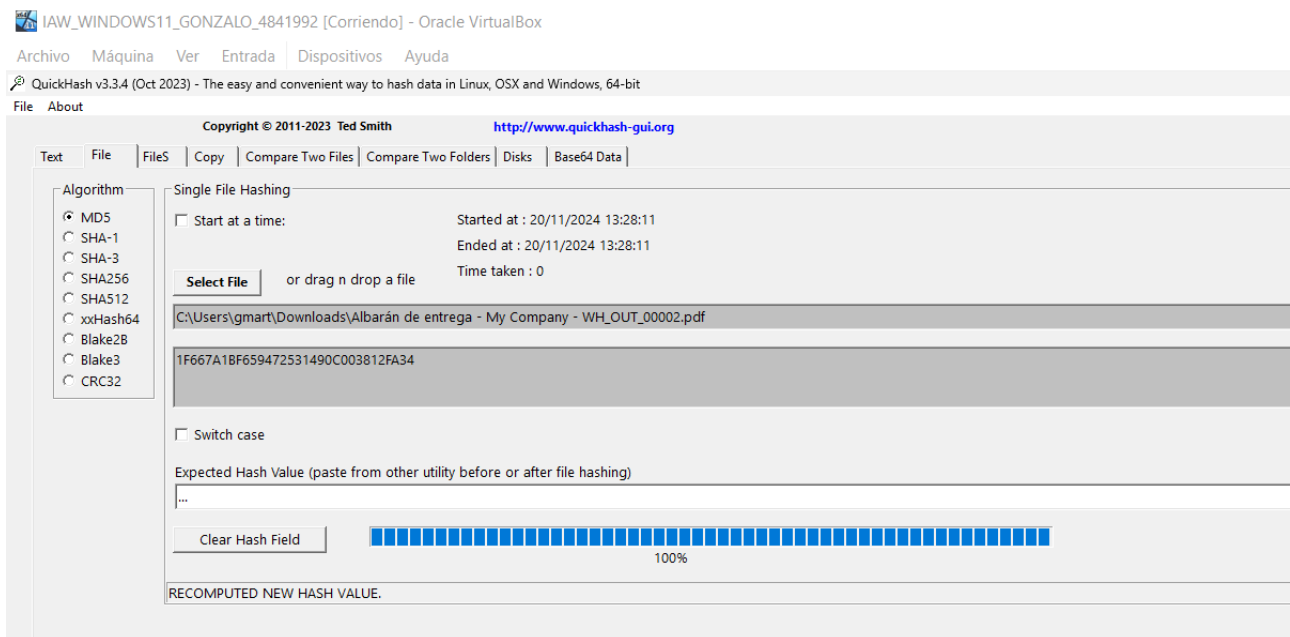
Elegimos algún archivo y lo abrimos:



Y automáticamente te genera el hash:



De echo genera automáticamente todos los algoritmos, si selecciono por ejemplo MD5, automáticamente me genera otro hash:

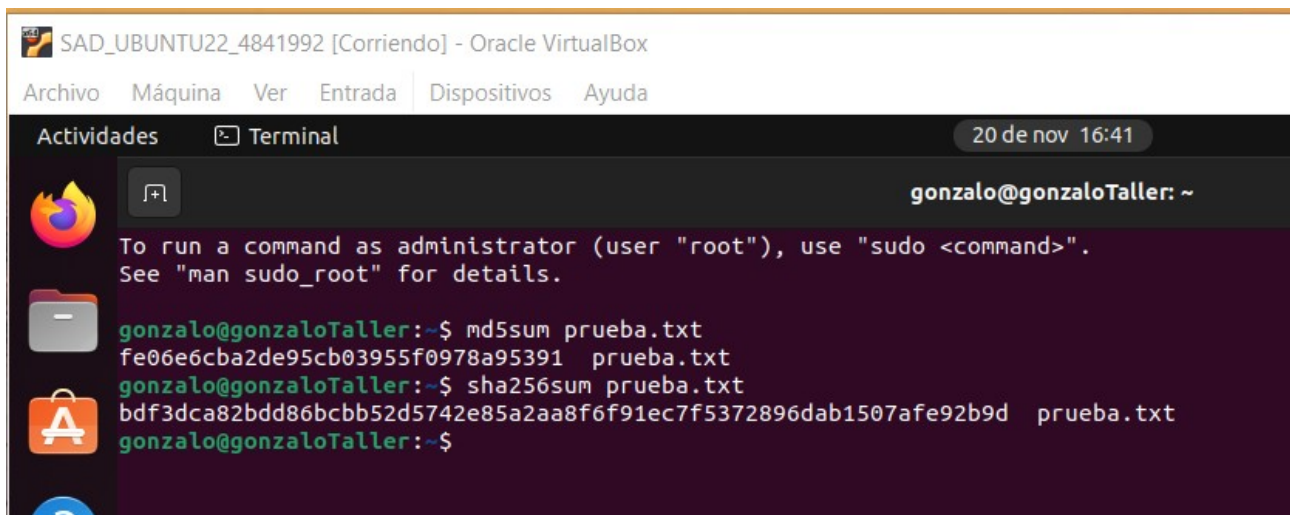


¿Es fácil de usar? Sí, no tiene mucho misterio. ¿Qué ventaja tiene? Que no tienes que memorizar o buscar en Google el comando.

Pasamos a Linux.

2. PRÁCTICA EN LINUX:

Es lo mismo que con Powershell, al menos en Linux el comando es más corto. Copio y pego los comandos utilizados y su salida. El comando está en **negrita y cursiva**, la salida no:



```
SAD_UBUNTU22_4841992 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Actividades  Terminal  20 de nov 16:41

gonzalo@gonzaloTaller: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

gonzalo@gonzaloTaller:~$ md5sum prueba.txt
fe06e6cba2de95cb03955f0978a95391  prueba.txt
gonzalo@gonzaloTaller:~$ sha256sum prueba.txt
bdf3dca82bdd86bcbb52d5742e85a2aa8f6f91ec7f5372896dab1507afe92b9d  prueba.txt
gonzalo@gonzaloTaller:~$
```

gonzalo@gonzaloTaller:~\$ md5sum prueba.txt

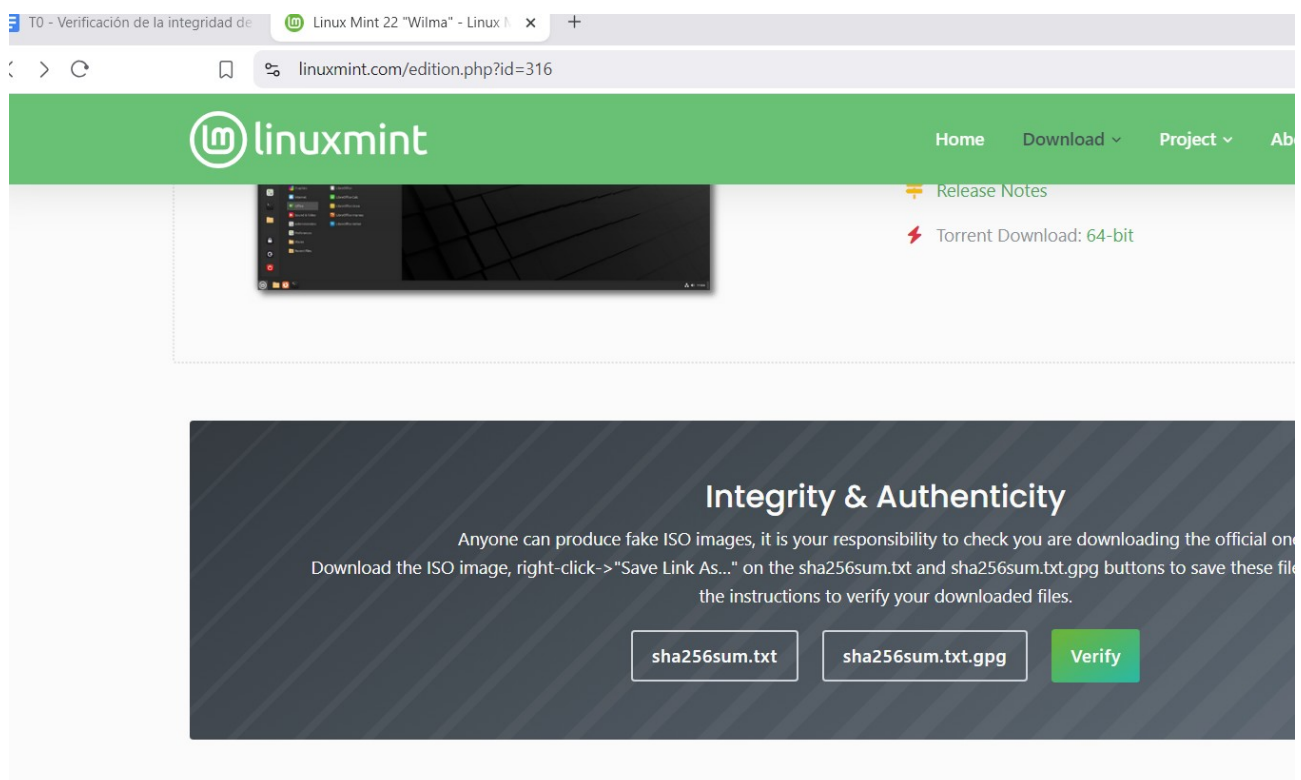
fe06e6cba2de95cb03955f0978a95391 prueba.txt

gonzalo@gonzaloTaller:~\$ sha256sum prueba.txt

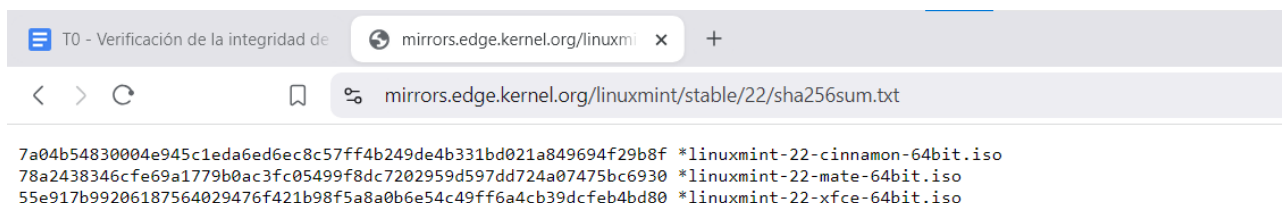
bdf3dca82bdd86bcbb52d5742e85a2aa8f6f91ec7f5372896dab1507afe92b9d prueba.txt

¿Cómo podría utilizarlos para verificar la integridad de un archivo descargado de Internet?

Necesitas que la web desde la que lo descargaste proporcione los hashes. Por ejemplo, queremos descargar Linux Mint, nos vamos a la web oficial y en la página de descarga vemos esto:



Hacemos click en **sha256sum.txt** y el fichero es este:



The screenshot shows a web browser window with the address bar displaying `mirrors.edge.kernel.org/linuxmint/stable/22/sha256sum.txt`. The page content lists three SHA256 hashes, each followed by an asterisk and the filename of a Linux Mint 22 ISO image:

```
7a04b54830004e945c1eda6ed6ec8c57ff4b249de4b331bd021a849694f29b8f *linuxmint-22-cinnamon-64bit.iso
78a2438346cfe69a1779b0ac3fc05499f8dc7202959d597dd724a07475bc6930 *linuxmint-22-mate-64bit.iso
55e917b99206187564029476f421b98f5a8a0b6e54c49ff6a4cb39dcfeb4bd80 *linuxmint-22-xfce-64bit.iso
```

Aquí nos dan el hash sha256 de las distintas versiones de la distro. Simplemente comprobamos el sha256, eso ya sabemos hacerlo, y comprobamos que coincide con el de la versión que tenemos, por ejemplo si hemos descargado la versión de cinnamon pues comprobamos que el hash coincide, si coincide el fichero está íntegro, si no, entonces su integridad ha sido comprometida.

3. COMPARACIÓN DE ALGORITMOS:

MD5 y SHA-1 son inseguros porque son vulnerables a ataques de colisión. Un ataque de colisión es cuando se consigue que dos ficheros distintos den el mismo resultado al hash. Eso lo vuelve inútil, porque el punto de hash es que ningún valor puede repetirse, no puede haber dos ficheros con el mismo hash, si puedes conseguir que otro fichero replique el hash entonces puedes violar la integridad de un fichero sin que el algoritmo lo detecte.

Sería una pésima idea usar ese algoritmo para por ejemplo firmar un certificado SSL. En general para cualquier situación en la que la falsificación de un archivo sería fatal.

Sin embargo, para casos en los que la seguridad no es tan importante, por ejemplo si queremos que nuestra base de datos pueda verificar rápidamente si dos ficheros tienen la misma información, sin necesidad de comparar la totalidad de los mismos, MD5 o SHA-1 pueden servir para hacerlo más rápido, es extremadamente improbable que tengas dos ficheros distintos con el mismo hash de casualidad, por lo que para eso sí sirve.

Dejo una fuente a una explicación técnica de por que son vulnerables a ataques de colisión y lo que eso supone. Es para MD5, pero SHA-1 es igual de inseguro por las mismas razones que MD5.

[Fuente](#)

4. REFLEXIÓN

Las funciones hash son claves para asegurar la integridad de los archivos, y son imprescindibles en cualquier ámbito que maneje datos delicados, como la banca, las bases de datos, o incluso para tú como usuario si vas a descargar ficheros que, en caso de ser alterados, podrían ser un grave riesgo de seguridad. Hay que pensar que sin algoritmos o algún tipo de metodología para verificar la integridad de los datos. Los ciberataques se multiplicarían, la banca online sería mucho más insegura, nunca sabrías si lo que te estás descargando viene con “regalitos”. Las funciones hash, por tanto, son absolutamente necesarias para el correcto funcionamiento de los sistemas informáticos.