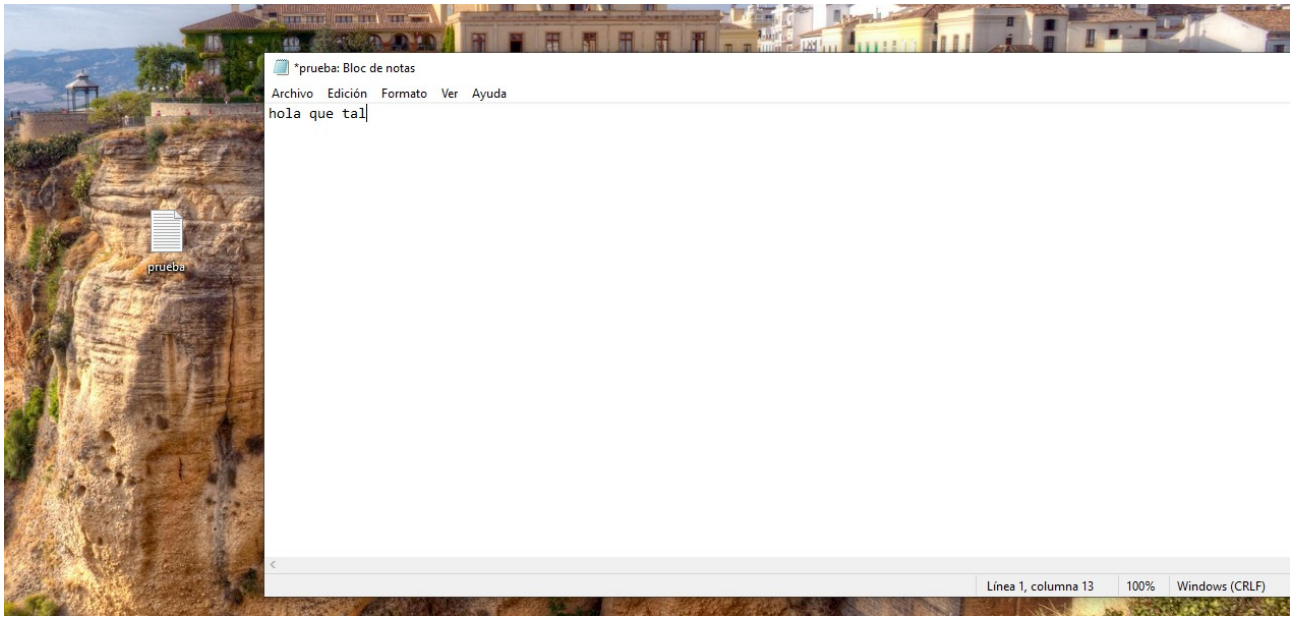
David Martínez 2ASIR

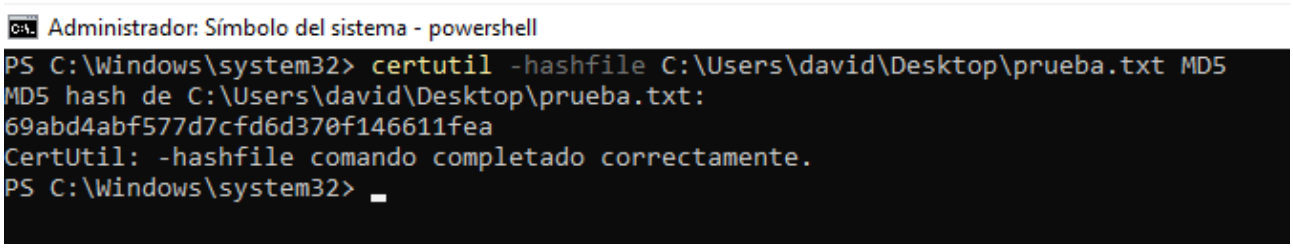
VERIFICACIÓN DE IDENTIDAD DE ARCHIVOS MEDIANTE FUNCIONES HASH

WINDOWS

Para calcular el hash de un archivo tenemos que usar la herramienta certUtil, vamos a probar con un archivo de texto con un poco de texto



Primero entramos en cmd y powershell, usamos -hashfile y la ruta donde tenemos el fichero y el algoritmo que vamos a utilizar, primero utilizamos MD5



certutil -hashfile C:\Users\david\Desktop\prueba.txt MD5
MD5 hash de C:\Users\david\Desktop\prueba.txt:
69abd4abf577d7cfd6d370f146611fea
CertUtil: -hashfile comando completado correctamente.

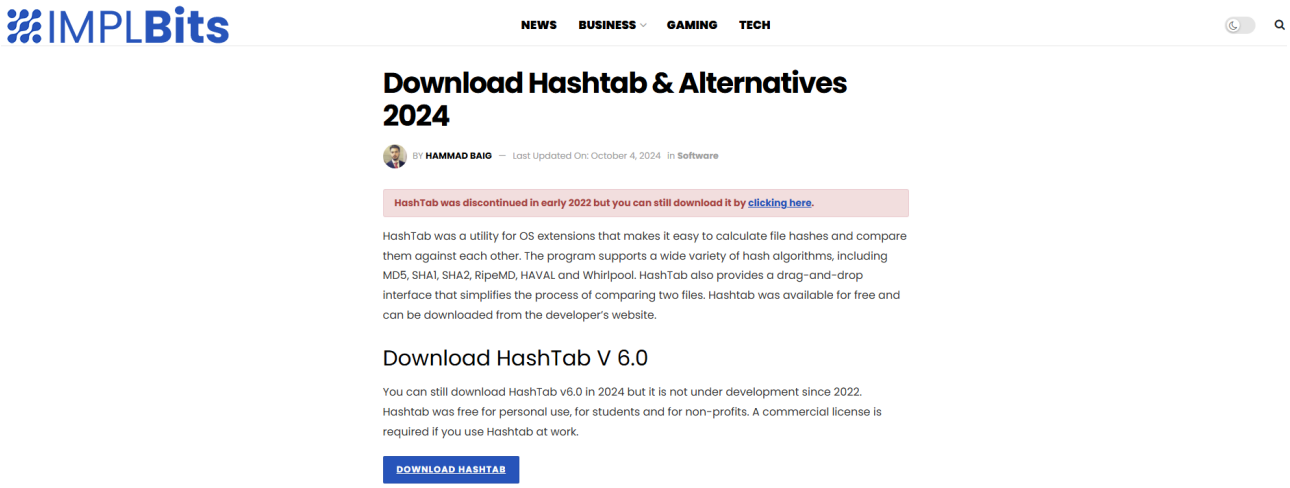
David Martínez 2ASIR

Ahora hacemos lo mismo pero con SHA256

```
PS C:\Windows\system32> certutil -hashfile C:\Users\david\Desktop\prueba.txt SHA256
SHA256 hash de C:\Users\david\Desktop\prueba.txt:
bc498a23877e37cac0086557a9eb90d0e3a657c1dd3db7bfdc4d736d5c017fd2
CertUtil: -hashfile comando completado correctamente.
PS C:\Windows\system32>
```

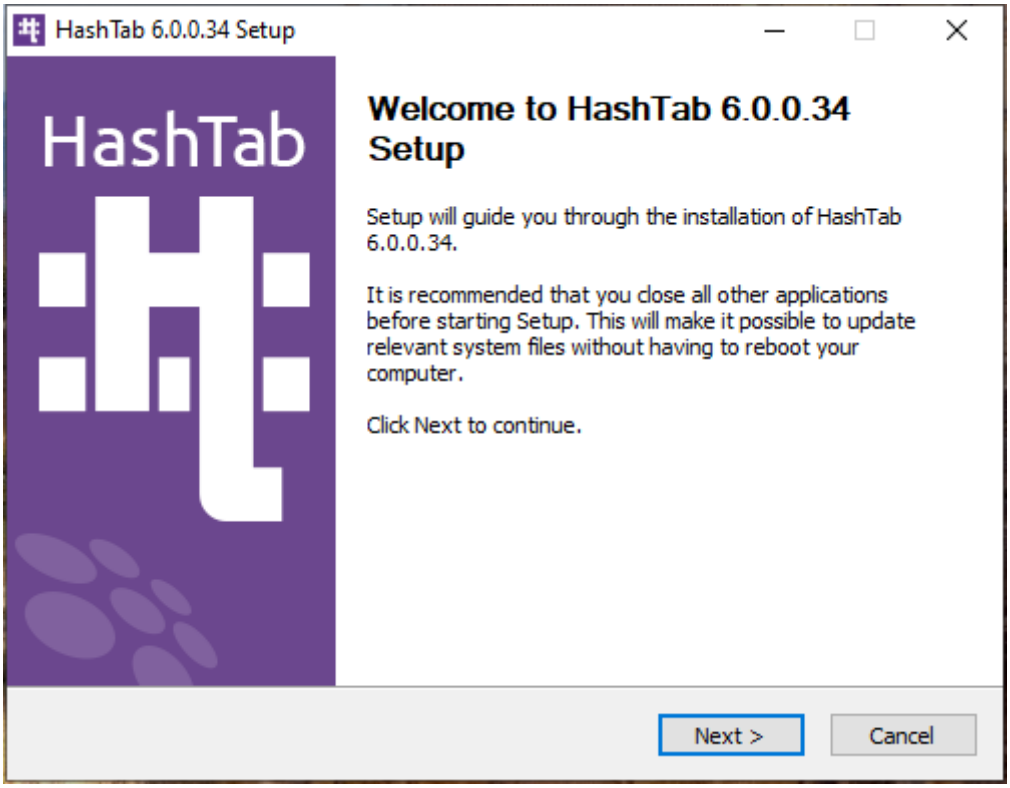
SHA256 hash de C:\Users\david\Desktop\prueba.txt:
bc498a23877e37cac0086557a9eb90d0e3a657c1dd3db7bfdc4d736d5c017fd2
CertUtil: -hashfile comando completado correctamente.

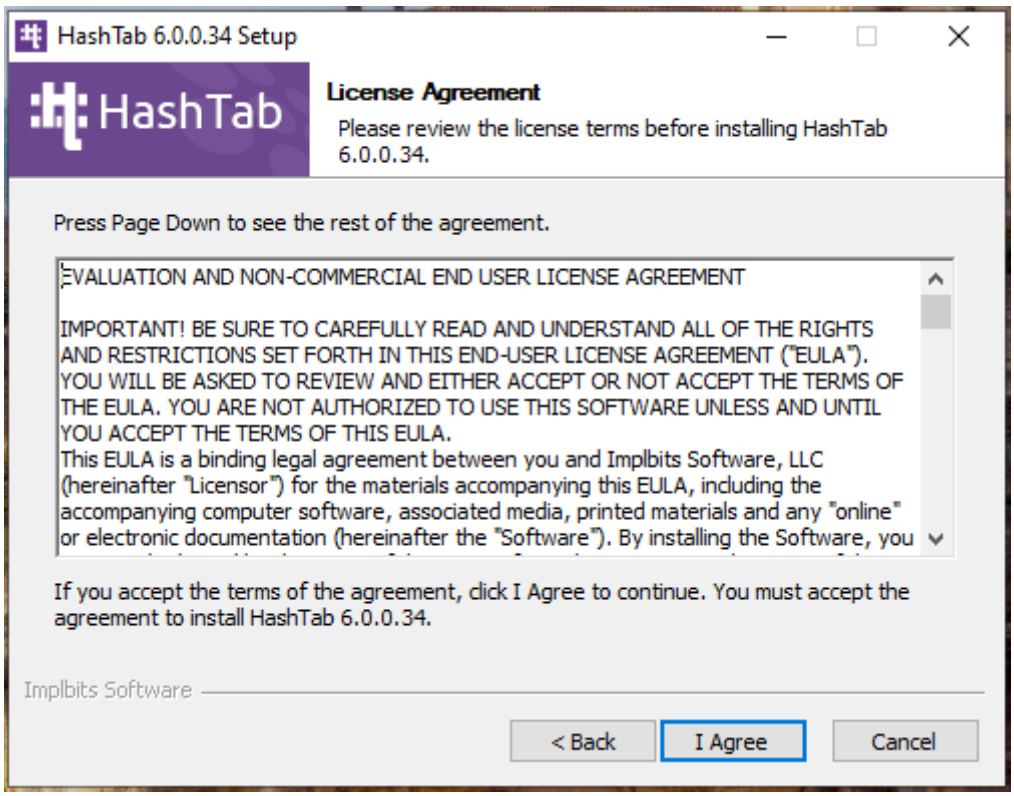
Vamos a usar el programa hastab para crear el hash del archivo, lo descargamos



David Martínez 2ASIR

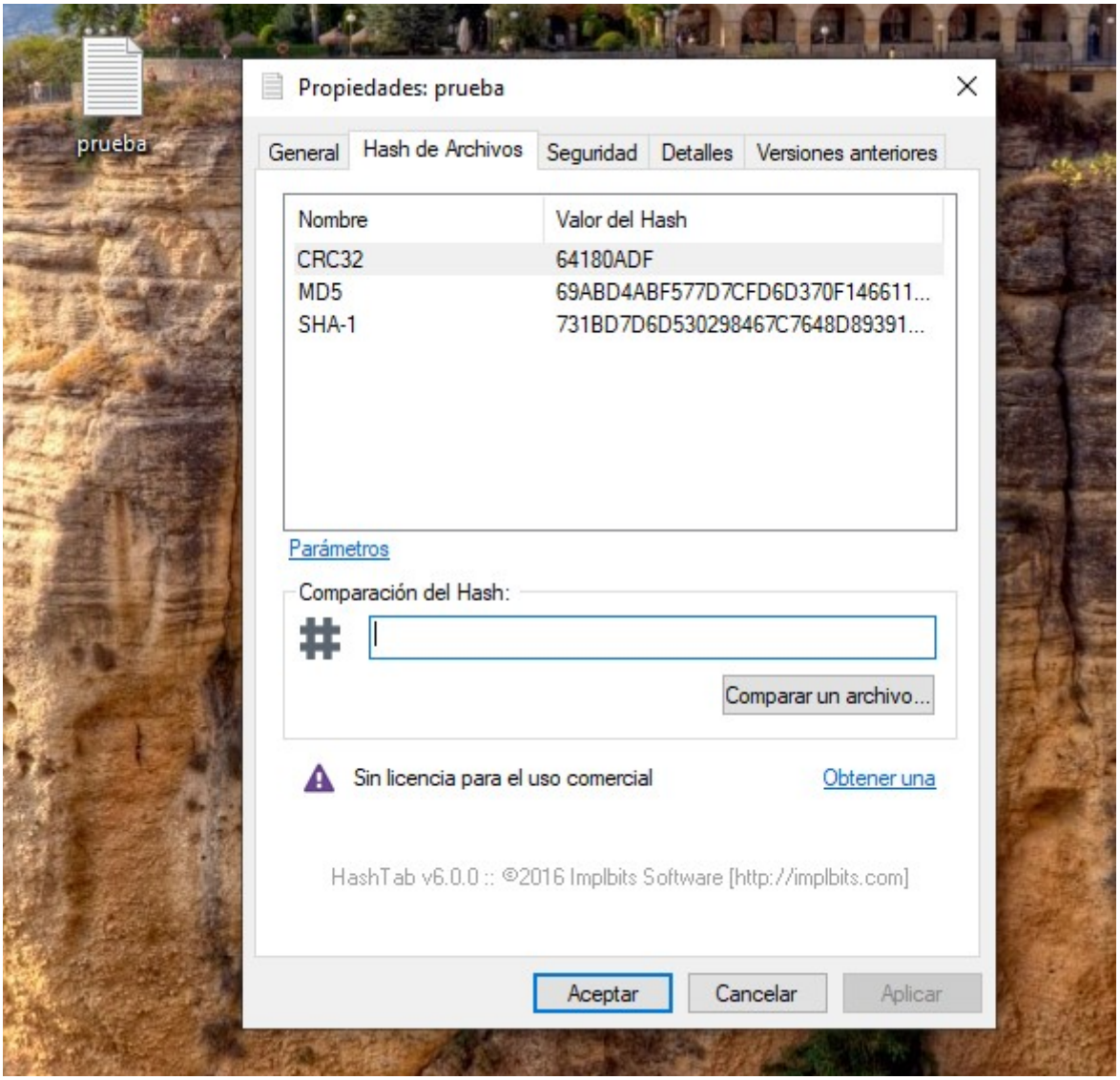
Le damos a siguiente, i agree e instalar





David Martínez 2ASIR

Ahora al hacer click derecho y propiedades en le archivo habrá una pestaña donde se muestran los hash con distintos algoritmos

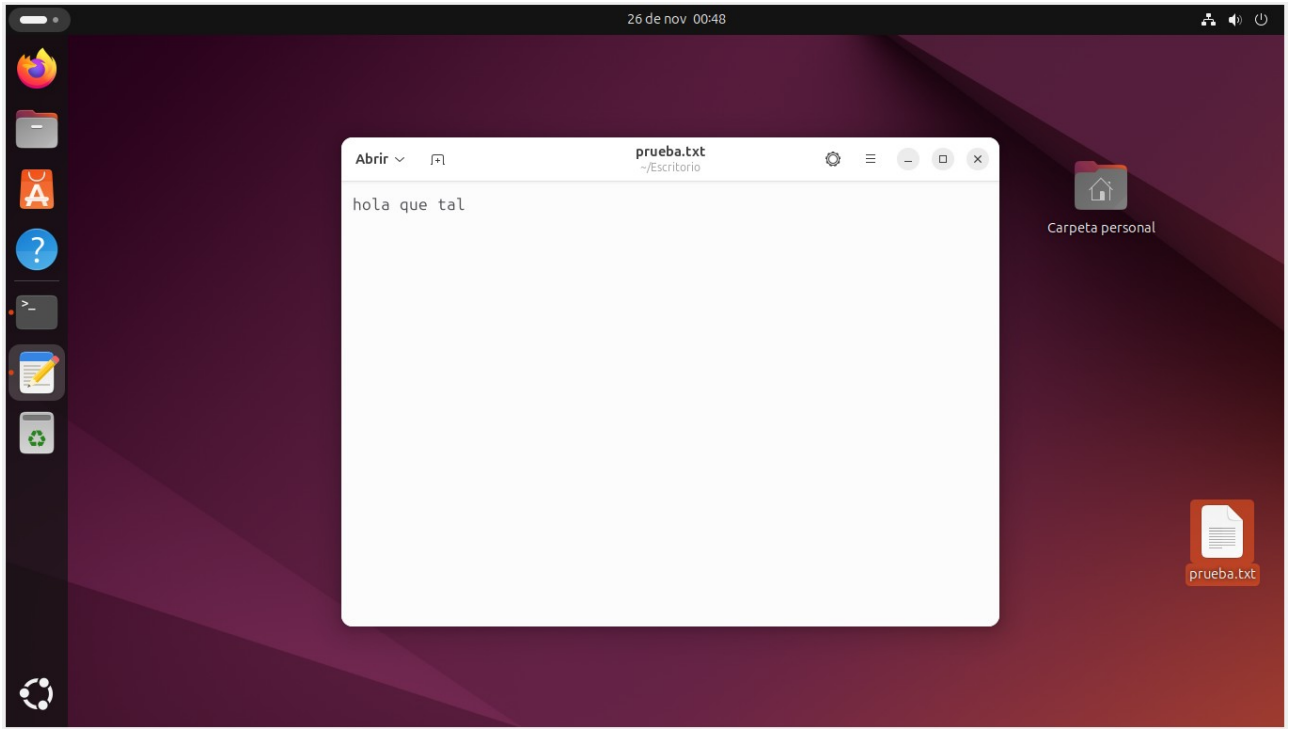


La herramienta me pareció fácil de usar ya que es una especie de extensión, la instalas y ya aparece la pestaña de los hash en las propiedades del archivo, la ventaja de que tenga interfaz grafica es que se ve todo con un simple click

David Martínez 2ASIR

LINUX

Ahora vamos a hacer lo mismo pero en linux creamos un fichero igual que en windows



Ponemos el algoritmo y la ruta del archivo primero probamos con MD5

```
david@david-VirtualBox:~/Escritorio$ md5sum /home/david/Escritorio/prueba.txt
f6940e22c9e2eccb5e1d0251a984ddf8  /home/david/Escritorio/prueba.txt
```

md5sum /home/david/Escritorio/prueba.txt
f6940e22c9e2eccb5e1d0251a984ddf8 /home/david/Escritorio/prueba.txt

Ahora probamos el sha256

```
david@david-VirtualBox:~/Escritorio$ sha256sum /home/david/Escritorio/prueba.txt
98651386247062b83afcd68d294251c95d02f5e4313999d164b70593ac1278d2  /home/david/Escritorio/prueba.txt
```

David Martínez 2ASIR

Esto te puede servir si la web donde descargas un archivo te proporciona el hash, así puedes compararlos y saber si te has descargado el correcto

COMPARACIÓN DE ALGORITMOS

¿Por qué los algoritmos MD5 y SHA-1 ya no son recomendados para aplicaciones críticas? Da un ejemplo de una situación en la que el uso de estos algoritmos podría representar un riesgo.

Los algoritmos MD5 y SHA-1 ya no son recomendados para aplicaciones críticas debido a sus vulnerabilidades conocidas frente a ataques de colisión y preimagen. Estas vulnerabilidades permiten que un atacante manipule datos sin ser detectado o genere múltiples entradas con el mismo hash, comprometiendo la seguridad.

En aplicaciones HTTPS, un certificado digital asegura que un sitio web es auténtico y protege la comunicación entre el navegador y el servidor. Si el algoritmo de hash usado para firmar el certificado es vulnerable (como MD5 o SHA-1), un atacante podría generar un certificado falso con el mismo hash que uno legítimo.

En 2008, investigadores demostraron cómo se podía usar una colisión en MD5 para crear un certificado digital falso que parecía emitido por una Autoridad de Certificación confiable (CA). Este ataque mostró cómo MD5 permitía a los atacantes hacerse pasar por sitios legítimos y violar la confianza en el sistema de certificados.

Indica en qué situaciones podría ser aceptable utilizar MD5 en lugar de algoritmos más seguros como SHA-256 o SHA-512.

Verificación de integridad para archivos no sensibles
MD5 sigue siendo rápido y eficiente para generar un hash, por lo que puede utilizarse para comprobar la integridad de archivos en entornos donde:

-No se requiere alta seguridad.
-La integridad es solo para detectar errores accidentales, como los causados por fallos en la transferencia o almacenamiento de archivos.

REFLEXIÓN

Las funciones hash son pilares fundamentales en la seguridad informática, ya que garantizan la integridad y autenticidad de la información en numerosos sistemas y aplicaciones. Estas funciones convierten datos de cualquier tamaño en una cadena fija de caracteres, lo que permite comparar fácilmente versiones de datos para detectar cambios o manipulaciones.