¿QUE ES UN HASH?

Una función hash es un algoritmo matemático que recibe una entrada de longitud variable, por ejemplo, un archivo, una imagen, un mensaje, una contraseña o incluso una imagen forense de un disco y la transforma mediante operaciones matemáticas en una salida de longitud fija denominada **valor hash, digest** o **resumen criptográfico**.

Esta transformación posee cuatro características fundamentales:

1. ES DETERMINÍSTICA
La misma entrada siempre produce exactamente el mismo resultado.

2. TIENE LONGITUD FIJA
Independientemente del tamaño de los datos procesados, el resultado tiene una longitud determinada por el algoritmo.
Por ejemplo, SHA-256 genera 256 bits, que normalmente se representan como 64 caracteres hexadecimales.

3. PRESENTA EFECTO AVALANCHA
Una modificación mínima en los datos de entrada puede producir un cambio radical en el resultado del hash.

4. ES DE UNA SOLA VÍA
En una función hash criptográfica segura, debe ser computacionalmente inviable reconstruir la entrada original a partir de su hash.

EN OTRAS PALABRAS
La función hash procesa los datos de entrada y genera un resultado que depende de su contenido.
Si los datos son exactamente los mismos, el resultado será el mismo.
Si los datos cambian, aunque sea mínimamente, el resultado cambiará.

¿QUE SIGNIFICA QUE CAMBIE UN SOLO BIT?

Supongamos que tenemos el siguiente texto:
“Consejo de Abogados”

Calculamos su hash SHA-256 y obteneos un determinado valor:
0FB09CF7C0733BE0E848C4FF25835CABB6EE48BF0821F41CFC96564827FFE33A

Ahora modificamos únicamente una sola letra:
“Consejo De Abogados”
Visualmente pasa desapercibido…
Pero para el algoritmo no son los mismos datos.
La secuencia de bits que representa la información cambió y, como consecuencia, obtenemos un hash completamente diferente:

1F0EA1A59A44D0FDC1C63AD97F08A8EBCD819F01B57803025EBBBDCFB9F18CC6

Esto permite detectar modificaciones en la información que incluso serían imperceptibles para una persona.

¿PARA QUE SIRVE UN HASH?

Su función principal es verificar la integridad de la información, es decir, comprobar rápidamente que un archivo sigue siendo exactamente el mismo y que nadie lo modificó accidental o intencionalmente.

Actualmente se utiliza entre otras cosas para:

a) Verificación de archivos descargados.
b) Detección de modificaciones.
c) Copias de seguridad y sincronización.
d) Identificación de archivos.
e) Análisis de malware.
f) Firmas digitales.
g) Certificados digitales.
h) Sistemas de control de versiones.
i) Blockchain.
j) Distribución de software.
k) Informática forense.

l) Diferentes mecanismos internos de sistemas operativos y aplicaciones.

Prácticamente todos los sistemas modernos utilizan hashes continuamente, aunque el usuario nunca los vea.

UN EJEMPLO DE USO COTIDIANO…

SUBIR ARCHIVOS A LA NUBE

Cuando subís un archivo a un servicio en la nube, el sistema puede calcular un **hash del contenido** para identificarlo y trabajar con él de forma más eficiente.

Por ejemplo, si subís exactamente el mismo archivo dos veces, el sistema puede detectar que el contenido ya existe porque produce el mismo hash. Esto puede ayudar a **evitar procesar o almacenar nuevamente datos idénticos**, dependiendo de cómo esté diseñado el servicio.

El hash también puede utilizarse para comprobar que los datos recibidos coinciden con los datos originales y detectar posibles modificaciones o corrupción durante el proceso.

Lo interesante es que el sistema no necesita confiar solamente en el **nombre del archivo.**
Aunque tengan nombres diferentes, si contienen exactamente los mismos datos, producirán el mismo hash.
Así, una simple cadena matemática puede ayudar a la nube a **identificar, verificar y gestionar millones de archivos de forma automática.**

¿CUÁLES SON LOS ALGORITMOS MAS CONOCIDOS?

Los algoritmos hash más conocidos son:

MD5 - 128 bits
Fue uno de los algoritmos más populares durante muchos años.
Hoy ya no se considera seguro para aplicaciones criptográficas porque existen colisiones conocidas.
Aun así, todavía puede encontrarse para verificaciones simples de integridad.

SHA-1 - 160 bits
Fue ampliamente utilizado hasta que se demostraron ataques capaces de generar colisiones.
Actualmente también está obsoleto para aplicaciones de seguridad.

SHA-2 — 224, 256, 384 y 512 bits
Es una familia de algoritmos, incluye:

- **SHA-224**
- **SHA-256**
- **SHA-384**
- **SHA-512**
- **SHA-512/224**
- **SHA-512/256**

SHA-256 es probablemente el miembro que más vas a encontrar en aplicaciones prácticas de verificación de integridad, software, firmas digitales y numerosos sistemas de seguridad.

SHA-3 — 224, 256, 384 y 512 bits
Es una familia de funciones hash con un diseño criptográfico diferente al de SHA-2, Incluye:

- **SHA3-224**

- **SHA3-256**
- **SHA3-384**
- **SHA3-512**

También incluye **SHAKE128 y SHAKE256**, que permiten obtener salidas de longitud variable.

BLAKE2 y BLAKE3

Son funciones hash criptográficas modernas diseñadas para ofrecer alta seguridad y un elevado rendimiento.

¿QUÉ ES UNA COLISION?

Una colisión ocurre cuando dos datos completamente diferentes generan exactamente el mismo hash.

Por ejemplo:

Archivo A
SHA-1 - GENERA EL HASH: **A35F98**...

Archivo B (Completamente distinto a A)
SHA-1 - GENERA EL HASH: **A35F98**...

AMBOS PRODUCEN EXACTAMENTE EL MISMO RESULTADO.

Aunque pueda parecer extraño, **matemáticamente las colisiones son inevitables**. La razón es sencilla:
podemos tener una cantidad prácticamente infinita de archivos diferentes, pero cada algoritmo hash tiene una cantidad **limitada de resultados posibles**.

Por ejemplo, **SHA-256 siempre genera 256 bits**, independientemente de que estés calculando el hash de una foto de 2 MB o de un disco de 1 TB. Por lo tanto, en algún momento necesariamente existirán datos diferentes que produzcan el mismo resultado.

El verdadero desafío está en **encontrar esa coincidencia de forma intencional**.

Un algoritmo criptográfico seguro está diseñado para que encontrar dos datos diferentes con el mismo hash sea tan difícil que, en la práctica, resulte computacionalmente inviable con la tecnología disponible.

Ahí es donde MD5 y SHA-1 quedaron expuestos: **ya existen técnicas capaces de construir colisiones de manera práctica**. Por eso dejaron de considerarse

adecuados para usos criptográficos donde la resistencia a colisiones es importante.

En otras palabras: **que una colisión pueda existir es normal...**
Pero que alguien pueda provocarla fácilmente… ese es el problema.

MUY IMPORTANTE COMPRENDER:

UN HASH NO DEPENDE DEL NOMBRE DEL ARCHIVO.
Si renombras un archivo sin modificar su contenido, el hash seguirá siendo exactamente el mismo.

EL HASH TAMPOCO DEPENDE DE LA UBICACIÓN DEL ARCHIVO:
Moverlo de carpeta o copiarlo a otro disco no modifica el resultado.

¿CUÁL ES MÁS SEGURO?
Para verificar integridad actualmente conviene priorizar algoritmos modernos como **SHA-256, SHA-512 o SHA-3**, según el contexto.

¿CÓMO CALCULAR UN HASH?

1. WINDOWS:
Utilizando PowerShell:
ya viene integrado y no necesitas instalarlo.
Get-FileHash "archivo.pdf" -Algorithm SHA256

También en cmd utilizando CertUtil:
certutil -hashfile Archivo.pdf SHA256

2. LINUX:
incorpora herramientas para calcular hashes directamente desde la terminal:
sha256sum archivo.pdf

3. 7-ZIP:
Si utilizas 7-Zip, podes calcular hashes mediante su interfaz gráfica.
Botón derecho sobre el archivo → CRC SHA → seleccionar el algoritmo.
Sitio oficial: https://www.7-zip.org/download.html

4. HASHMYFILES:
Permite seleccionar uno o varios archivos y obtener diferentes valores hash.
También permite exportar los resultados y trabajar cómodamente con múltiples archivos.

Sitio oficial: https://www.nirsoft.net/utils/hash_my_files.html

5. ONLINE:
Para experimentar con **texto**, una herramienta muy interesante es CyberChef.
Podes introducir: **“Consejo de Abogados”**
y aplicar: **Hashing → SHA2 → SHA-256**

Es excelente para aprender porque permite observar cómo los datos se transforman y encadenar diferentes operaciones.
Pero para información sensible, documentos privados, evidencia digital o datos personales, **no es recomendable enviar información a servicios online** cuando podés realizar el cálculo localmente.

Sitio oficial: https://gchq.github.io/CyberChef

HASH – EL ADN MATEMATICO DE UN ARCHIVO

Un hash no demuestra que un archivo sea verdadero.
Demuestra algo más concreto: **que sigue siendo exactamente el mismo.**
Porque un solo cambio en sus datos puede transformar por completo su hash.
Y ahí está su verdadero poder: convertir una pregunta difícil:
¿este archivo fue alterado?
en una comprobación matemática…
Mismo hash: mismo contenido.
Hash diferente: algo cambió….

LOS ARCHIVOS PUEDEN CAMBIAR SIN QUE NADIE LO NOTE.
EL HASH NO MIRA. EL HASH CALCULA. Y CUANDO CALCULA…
EL CAMBIO APARECE.

Así de simple. Así de preciso.
Así de PODEROSO!

Eduardo E. Lecce
Abogado

Ciberseguridad & Ethical Hacker
3624628578 - eelciberseguridad@gmail.com