

HASHES, COLISIONES DE HASHES Y ACTUALIDAD

Bryan Velasco Pachuca



OVERVIEW

¿Qué es una función hash?

Tipos de hashes

La actualidad del SHA

Several black lines of varying lengths and orientations are drawn on the right side of the slide, extending from the right edge of the text boxes.

Si en algún momento
tienes una duda, por
favor, házmela saber



A meme featuring Squidward Tentacles from the animated show The Simpsons. He is shown from the chest up, wearing his signature brown shirt, and is positioned behind a chain-link fence. He has a wide, toothy grin and is looking towards the right. His right arm is extended, with his hand pressed against the fence. In the background, a blue sky with light clouds is visible. A butterfly is perched on a horizontal chain of the fence, and a small portion of a white sailboat with a black anchor is visible in the upper left corner. The text 'PROGRAMMER' is overlaid in the top left, 'HASH FUNCTION' is overlaid in the top right, and 'IS THIS COOL?' is overlaid in the bottom right, all in a bold, white, sans-serif font with black outlines.

PROGRAMMER

**HASH
FUNCTION**

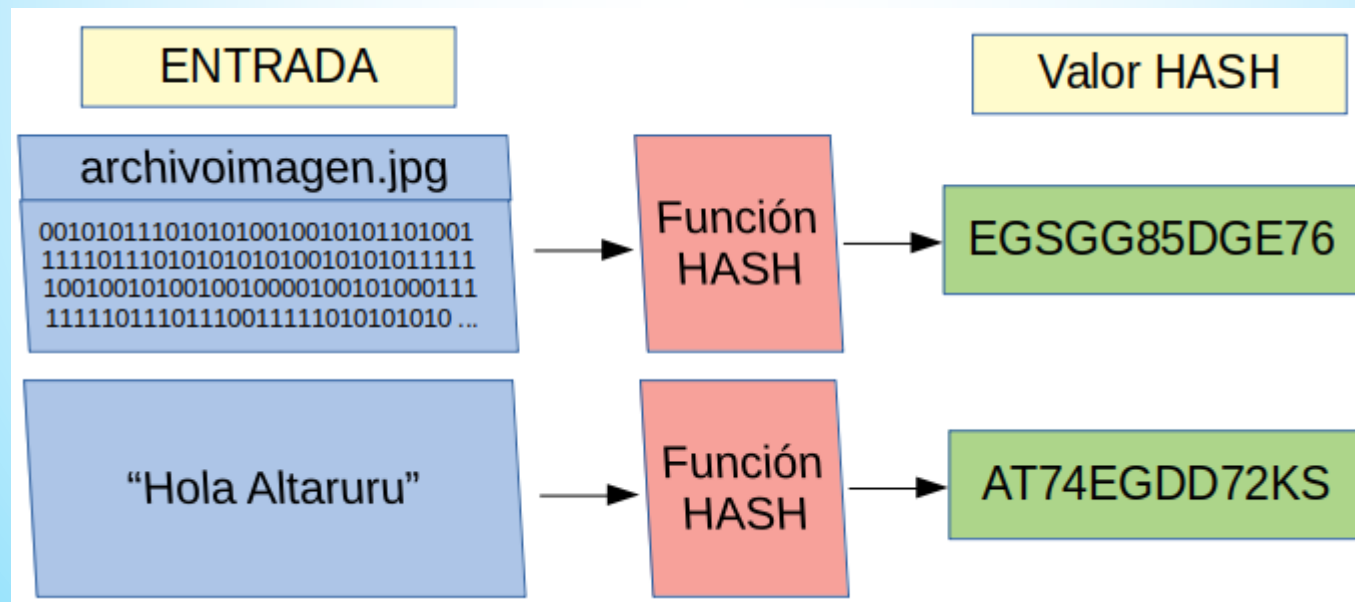
IS THIS COOL?

¿QUÉ ES UNA
FUNCIÓN HASH?



¿QUÉ ES UNA FUNCIÓN HASH?

Una función hash o función digestora es una función que te permite transformar cualquier cadena arbitraria de información en una cadena de bits con longitud fija.



EJEMPLO

Convertimos las cadenas 'Sistemas operativos' y 'Sistemas operativ0s' usando el hash SHA-1 cuya salida es una cadena de 160 bits de longitud:

'Sistemas operativos' \Rightarrow (SHA-1) 24651F4BC0FBC5F705F2429B7521BCD0673CAE9D_{hex}

'Sistemas operativ0s' \Rightarrow (SHA-1) 7684C1749D5E15022C58B95AE47DB8B60D555CBE_{hex}

*La longitud de la cadena de salida dependerá de la complejidad técnica y la función de compresión utilizada.

ELF HASH

- ▶ Función no criptográfica.
- ▶ Útil para almacenar valores en una tabla de hashes.
- ▶ Bloque de entrada de 8 bits
- ▶ Devuelve un arreglo de 32 bits (unsigned 32-bit integer)

```
public static uint ELFHash(string str)
{
    uint hash = 0;
    uint x;
    uint i;

    for (i = 0; i < str.Length; i++)
    {
        hash = (hash << 4) + ((byte)str[(int)i]);

        if ((x = hash & 0xF0000000) != 0)
        {
            hash ^= (x >> 24);
        }

        hash &= ~x;
    }

    return hash;
}
```


EJEMPLO

Se convirtieron las cadenas 'Espero aprobar SO' y 'Espero @probar SO' usando el hash ELF:

'Espero aprobar SO' \Rightarrow (ELF) **CD2642F**_{hex}

'Espero @probar SO' \Rightarrow (ELF) **CD0942F**_{hex}

-¿Existe alguna forma de robustecer el algoritmo?

-La neta si, pero no te voya decir.

ELF HASH DE 2 RONDAS

Rondas: Cuando la retroalimentación sucede n-veces por bloque de entrada, se dice que es de n-rondas.

```
public static uint ELFHash(string str)
{
    uint hash = 0;
    uint x;
    uint i;

    for (i = 0; i < str.Length; i++)
    {
        hash = (hash << 4) + ((byte)str[(int)i]);

        1ra if ((x = hash & 0xF0000000) != 0)
        {
            hash ^= (x >> 24);
        }
        hash &= ~x;

        hash = (hash << 4) + hash / 2;

        2da if ((x = hash & 0xF0000000) != 0)
        {
            hash ^= (x >> 24);
        }
        hash &= ~x;
    }
    return hash;
}
```

Al utilizar ELF hash de 2 rondas para el ejemplo anterior se obtienen las siguientes salidas:

'Espero aprobar SO' \Rightarrow (ELF) **F99084F**_{hex}

'Espero @probar SO' \Rightarrow (ELF) **B447827**_{hex}

-¿Así o más robusto?

-Creo que así está bueno.

¿QUÉ SE ESPERA DE UN BUEN HASH?

- ▶ NO reversible.
- ▶ Con efecto avalancha.
- ▶ Determinista.
- ▶ No predecible.
- ▶ Resistente a colisiones.

¿QUÉ ES UNA COLISIÓN?



TIPOS DE HASHES

Several thin, white, parallel diagonal lines are located in the bottom right corner of the slide, extending from the right edge towards the center.

Adler32

Create your Adler32 hash or calculate a checksum of your file with this free online converter.

CRC-32

Create a CRC-32 checksum of an uploaded file with this free online hash calculator.

Gost

Create a GOST hash from your data with this free online encryption tool.

MD5

Encrypt your data like passwords and files with this free online MD5 hash generator.

SHA-1

Generate a SHA-1 hash from your sensitive data like passwords with this free online SHA-1 hash generator. Optionally create a SHA-1 checksum of your files.

SHA-512

Generate a SHA-512 hash from your data or upload a file to create a SHA-512 checksum with this free online converter.

Tiger-160

Create a Tiger hash with 160 Bit to protect your data with this free online converter.

htpasswd Apache

Calculate your passwords for Apache's .htpasswd file with this free online encryption tool.

CRC-32B

Calculate the CRC-32B checksum with this free online checksum tool.

Haval-128

Generate a Haval-128 hash with this free online hash generator. Additionally upload a file to create a Haval-128 checksum.

RIPEMD-128

Generate a RIPEMD-128 hash with this free online converter. Optionally upload a file to create a checksum or provide a shared key for the HMAC variant.

SHA-256

Calculate a SHA-256 hash with this free online converter. Additionally create a checksum of your file.

Snefru

Encrypt your data with the Snefru hash algorithm. Optionally upload a file to create a Snefru checksum.

Tiger-192

Encrypt your sensitive data like passwords with this free online converter using the Tiger hashing algorithm with 192 Bits.

Blowfish

Encrypt and hash your data using the Blowfish encryption algorithm with this free online tool.

DES

Calculate a DES hash from your passwords or files with this free online encryption tool.

MD4

Create a MD4 hash with this free online encryption tool.

RIPEMD-160

Encrypt your data with this free online RIPEMD-160 hash converter. Optionally upload a file to create a RIPEMD-160 checksum or provide a HMAC shared key.

SHA-384

Generate a SHA hash with 384 Bits with this free online hash generator. Optionally upload a file to calculate a SHA-384 checksum.

Tiger-128

Create a Tiger hash with 128 Bit using this free online hash converter.

Whirlpool

Generate a Whirlpool hash with this free online hash calculator. Optionally create the Whirlpool checksum of a file you can upload.

FAMILIA SHA (Secure Hash Algorithm)

Familia de 4 hashes respaldada por el NIST (*National Institute of Standards and Technolgy*) y la NSA (*National Security Agency*).

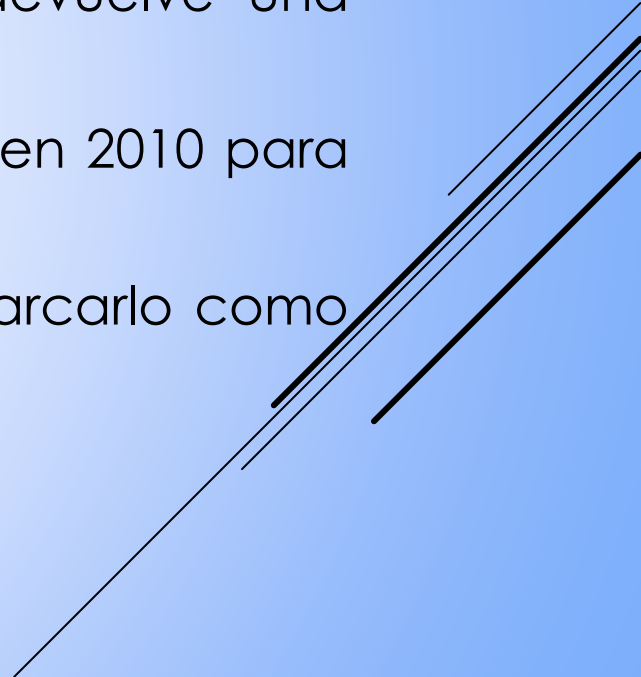


SHA-0

- ▶ Publicado en 1993 como SHA por el NIST y la NSA.
- ▶ Es de 80 rondas.
- ▶ Desde 1998 fueron encontradas las primeras colisiones.
- ▶ En 2004 se hallaron colisiones en su versión reducida de 65 rondas y colisiones en 142 de los 160 bits de la salida en su versión completa.



SHA-1

- ▶ Debido a las colisiones confirmadas en SHA-0, el NIST publica en 1995 una versión “mejorada” (solo se agregan rotaciones de bit), el SHA-1.
 - ▶ También emplea 80 rondas en su versión completa y devuelve una cadena de 160 bits.
 - ▶ En 2005 se hallan colisiones en una versión de 53 rondas y en 2010 para una de 73 rondas.
 - ▶ Debido a la cercanía con las 80 rondas, el NIST decide marcarlo como obsoleto (e inseguro) desde el 2010.
- 



- En 2017 el SHA-1 es brutalmente asesinado por el ataque SHattered que demuestra es posible generar colisiones en 2 PDF's con información diferente.

SHattered

The first concrete collision attack against SHA-1
<https://shattered.io>



Marc Stevens
Pierre Karpman



Elie Bursztein
Ange Albertini
Yarik Markov

SHattered

The first concrete collision attack against SHA-1
<https://shattered.io>



Marc Stevens
Pierre Karpman



Elie Bursztein
Ange Albertini
Yarik Markov

```
└─ sha1sum *.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a 1.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a 2.pdf
```

```
└─ /tmp/sha1
```

```
└─ sha256sum *.pdf
```

```
2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0 1.pdf
d4488775d29bdef7993367d541064dbdda50d383f89f0aa13a6ff2e0894ba5ff 2.pdf
```

0.64G 8-11h

- ▶ Finalmente, en 2020 al SHA-1 le echaron cal al documentar un ataque practico en donde resultó redituable producir colisiones en la herramienta GnuPG.
- ▶ Para lograr esto se implementaron ataques de prefijo.
- ▶ El costo aproximado por colisión fue de \$45,000 dólares, aprox. \$890,000 pesos.





**SHA-1 BIEN
TOSTADO EN 2020**

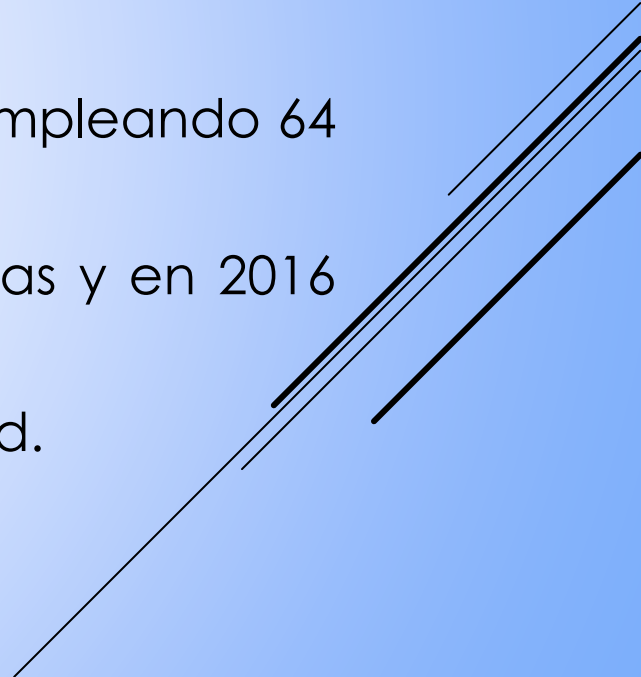


**SHA-1 USADO
PARA VALIDAR
TODO OBJETO EN GIT**

SHA-2



SHA-2 Y COMPAÑÍA

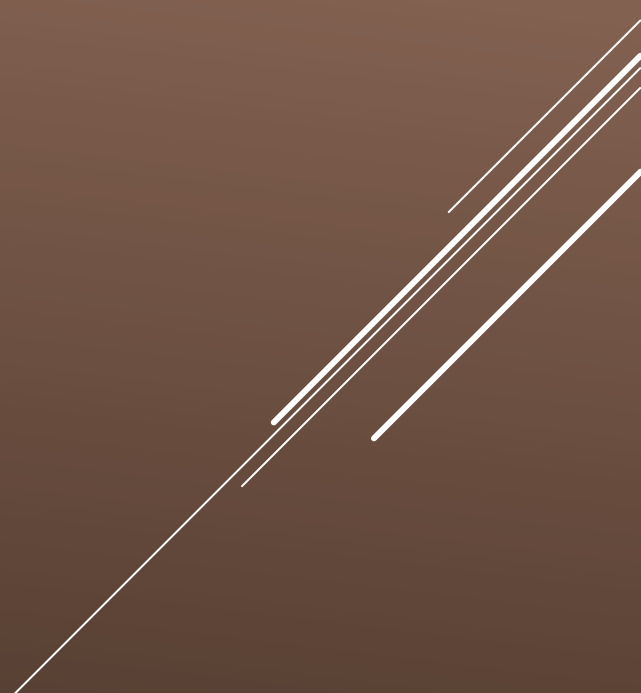
- ▶ Conjunto de funciones hash (SHA2-224, SHA2-256, SHA2-384 y SHA2-512) presentados en 2001 con cambios más significativos respecto a sus predecesores.
 - ▶ Sus tamaños de bloque de entrada van de 512 a 1024 bits empleando 64 u 80 rondas (según la versión).
 - ▶ En 2008 se hallaron colisiones para un SHA2-256 de 24 rondas y en 2016 para un SHA2-512 de 27 rondas.
 - ▶ Hasta el día de hoy no se ha visto comprometida su seguridad.
- 

SHA-3 (KECCAK) Y COMPAÑÍA

- ▶ Último miembro de la familia SHA, adoptado oficialmente el 5 de agosto de 2015.
- ▶ Ganador de la NIST hash function competition.
- ▶ Sus tamaños de bloque de entrada varían de 576 a 1152 bits empleando siempre 24 rondas.
- ▶ En 2012 se hallaron colisiones en un SHA3-256 de 4 rondas y en 2019 en uno de 5 rondas.



LA ACTUALIDAD DEL SHA



CERTIFICADOS SSL

- ▶ Desde 2017 Google anunció que dejaría de dar soporte a certificados SSL (Secure Sockets Layer) basados en SHA-1.

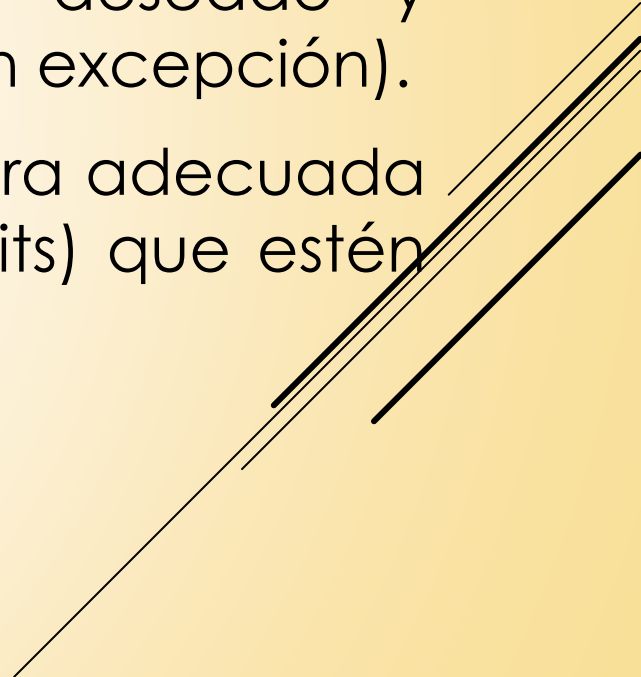
MICROSOFT DOWNLOAD CENTER

- ▶ El 3 de agosto de 2020, todo software descargable de Microsoft firmado por SHA-1 fue removido del Microsoft Download Center.

GIT

- ▶ Es de conocimiento popular que GIT usa SHA-1 prácticamente para todo (Validación de la integridad de un objeto).
- ▶ Debido al ataque SHAttered toda su infraestructura se vio comprometida.
- ▶ En 2018 se decidió como sucesor el SHA-256



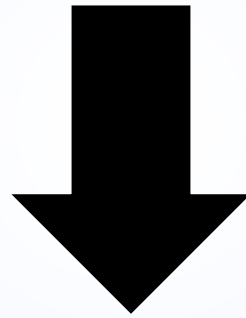
- ▶ Tener repositorios con diferente hash lo volvería imposible de operar entre distintos clientes.
 - ▶ Migrar a SHA-2 requeriría clonar el repositorio deseado y reescribirlo para que fuese compatible con SHA-2 (sin excepción).
 - ▶ Viendo más a futuro, se espera crear la infraestructura adecuada para llegar implementar hashes más largos (>256 bits) que estén preparados para la seguridad cuántica.
- 

ESPERANDO

A photograph of a human skeleton sitting on a wooden bench in a grassy field. The skeleton is looking upwards and to the right. The background shows a green field and a pile of hay on the right.

**ESPERANDO A QUE LAS
COMPUTADORAS CUÁNTICAS
LLEGUEN A LOS 1,000 QUBITS**

54DCF947960C71781BDB69A766A03A799FAEF47CAC9
0AA242CCDC95564B59BCFAF0FD6A715D389D8490E7
33B32C9F9B5D44354750F634DEB12280F8518AB2BBC



Gracias por su atención, buenas
noches :)