Instituto politécnico nacional

Centro de Estudios Científicos y Tecnológicos 9

“Juan de dios Bátiz”

**INVOCACIÓN DE SERVICIOS: SOAP**

**Nombre del Alumno:**

Mendoza Sánchez Carlos Eduardo

**Nombre del Profesor:**

Alma Erika Vázquez

**Unidad de Aprendizaje:**

Ciberseguridad

**Grupo:** 6IV8

**Fecha de entrega:** 21 de marzo de 2025

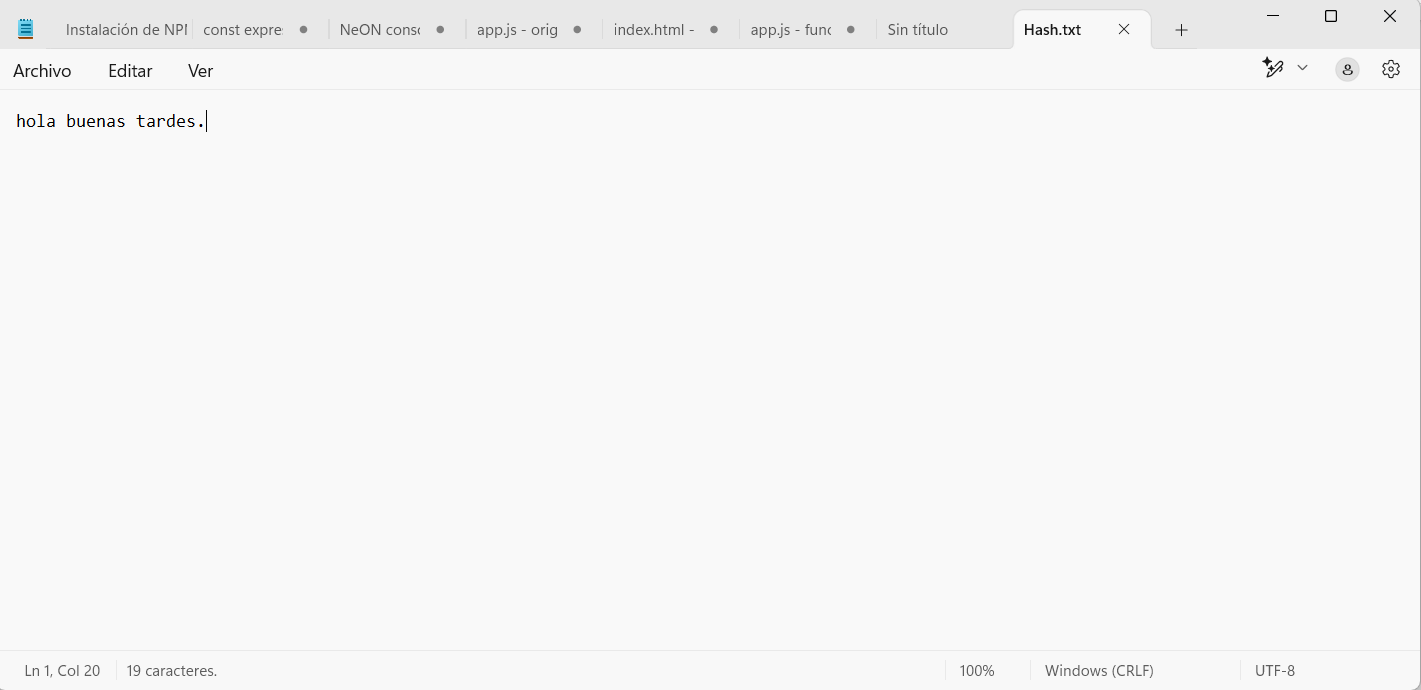
**Primero del hash, ¿Cuál es el valor junto a MD5?**

b8d1bbc48df8730a52b596b6e86107c2

eso me dio el hash de MD5 pero se me olvido tomar captura xd

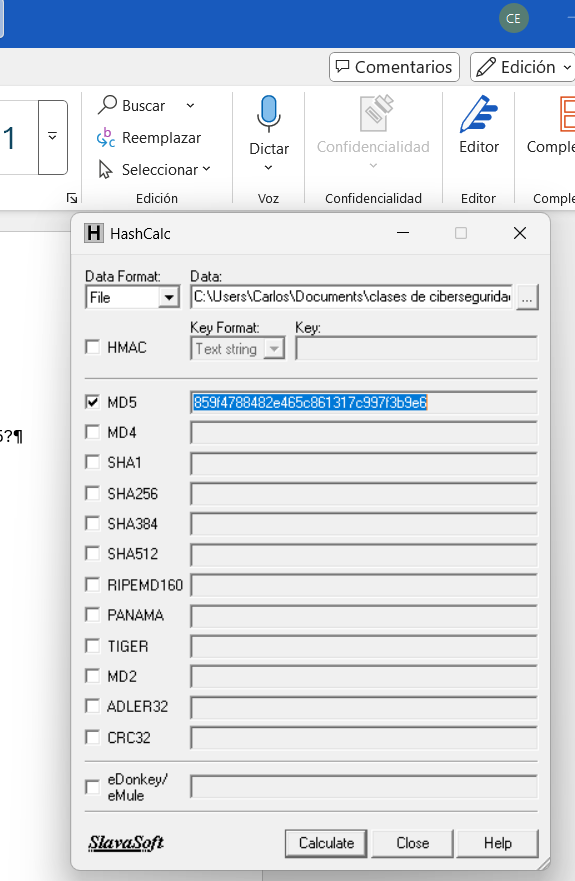
**Segundo hash despues de poner un punto al final ¿Cuál es el valor junto a MD5?**

859f4788482e465c861317c997f3b9e6

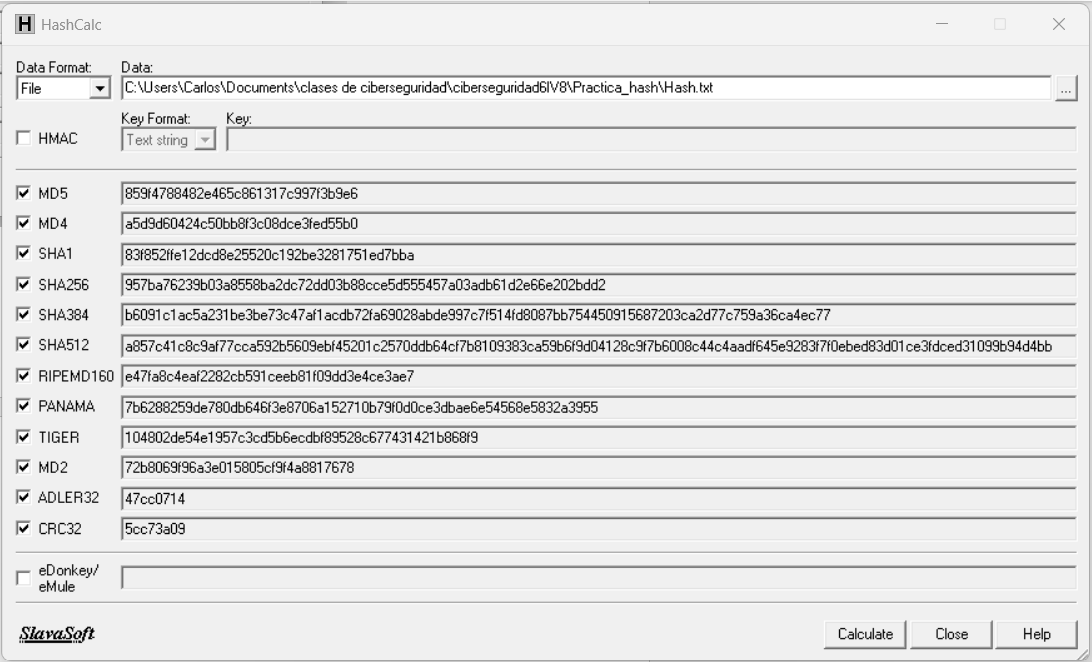


**¿El valor es diferente del valor registrado en el paso 3?**

Si porque cambio el cifrado al cambiar su valor, por mínimo que sea el cambio como poner un punto al final del texto, eso si no cambio la longitud



Fíjese cómo muchos de los tipos de hash crean un hash de longitud diferente. ¿Por qué?



**Fíjese cómo muchos de los tipos de hash crean un hash de longitud diferente. ¿Por qué?**

La razón es porque cada cifrado es diferente, refiriéndome a que, por ejemplo MD5 usa valores de 128 bits osea 32 caracteres en hexadecimal, por eso cuando cambie el hash.txt mantuvo la longitud, pero valores y caracteres diferentes, SHA-1 usa 160 bits, esto se traduce en 40 caracteres hexadecimales, SHA-256, como dice usa 256 bits y son 64 valores hexadecimales, cada algoritmo tiene su propia compresión y tamaño de salida diferente, por eso son de diferentes longitudes

Estos son los hash que se usaron :U :

 **MD5 (Message-Digest 5)**

* Genera un hash de 128 bits (32 caracteres hexadecimales).
* Muy popular en el pasado para verificar integridad de archivos.
* Se considera **inseguro** para fines críticos por vulnerabilidades de colisión.

 **MD4 (Message-Digest 4)**

* Antecesor de MD5, también produce un hash de 128 bits.
* Rápidamente dejó de usarse debido a problemas de seguridad.

 **SHA-1 (Secure Hash Algorithm 1)**

* Produce un hash de 160 bits (40 caracteres hexadecimales).
* Amplio uso histórico, pero **ya no recomendado** por colisiones descubiertas.

 **SHA-256 (Secure Hash Algorithm 256 bits)**

* Pertenece a la familia **SHA-2**.
* Muy utilizada y considerada **segura** para la mayoría de aplicaciones actuales.

 **SHA-384 (Secure Hash Algorithm 384 bits)**

* También de la familia **SHA-2**.
* Versión recortada de SHA-512, con 384 bits de salida.
* Ofrece un alto nivel de seguridad.

 **SHA-512 (Secure Hash Algorithm 512 bits)**

* El más extenso de la familia **SHA-2** de uso común (512 bits).
* Muy seguro y usado en aplicaciones que requieren máxima protección.

 **RIPEMD160 (RACE Integrity Primitives Evaluation Message Digest 160 bits)**

* Función desarrollada por la comunidad académica como alternativa a SHA.
* Produce 160 bits de salida y es razonablemente segura.

 **PANAMA**

* Es un algoritmo criptográfico menos común.
* Diseñado para ser versátil y rápido, pero no tan extendido como SHA o MD5.

 **TIGER**

* Produce un hash de 192 bits.
* Conocido por su buen rendimiento y velocidad, aunque de uso menos generalizado.

 **MD2 (Message-Digest 2)**

* Una de las primeras funciones de hash de 128 bits.
* Está obsoleta debido a avances y descubrimiento de vulnerabilidades.

 **ADLER32**

* Un **checksum** de 32 bits, no es criptográficamente seguro.
* Más rápido y sencillo que CRC32, pero solo sirve para detectar errores simples.

 **CRC32 (Cyclic Redundancy Check 32 bits)**

* Otro **checksum** de 32 bits muy común en detección de errores en archivos y transmisiones.
* No se considera seguro frente a ataques, solo verifica integridad básica.