

# Laboratório – Comparar dados com um hash

---

## Objetivos

---

Use um programa de hash para verificar a integridade dos dados.

## Histórico/Cenário

---

É importante identificar quando os dados foram corrompidos ou adulterados. Um programa de hash pode ser usado para verificar se os dados foram alterados ou se permaneceram intocados. Um programa de hash executa uma função hash nos dados ou em um arquivo, o que gera um valor (geralmente mais curto). Há muitas funções hash diferentes, algumas muito simples e outras muito complexas. Quando o mesmo hash é realizado nos mesmos dados, o valor gerado é sempre o mesmo. Se for feita qualquer alteração nos dados, o valor de hash gerado será diferente.

Observação: você precisará de privilégios de instalação e algum conhecimento do processo para instalar programas no Windows.

## Recursos necessários

---

- PC com acesso à Internet

### Etapa 1: Crie um arquivo de texto

- ☒ Criar um arquivo

```
touch arquivo.txt
```

- ☒ Colocar conteúdo

```
$ nano arquivo.txt
Exemplo de conteúdo:
Antes do hash:
md5 =>
sha1 =>
sha256 =>
crc32 =>
```

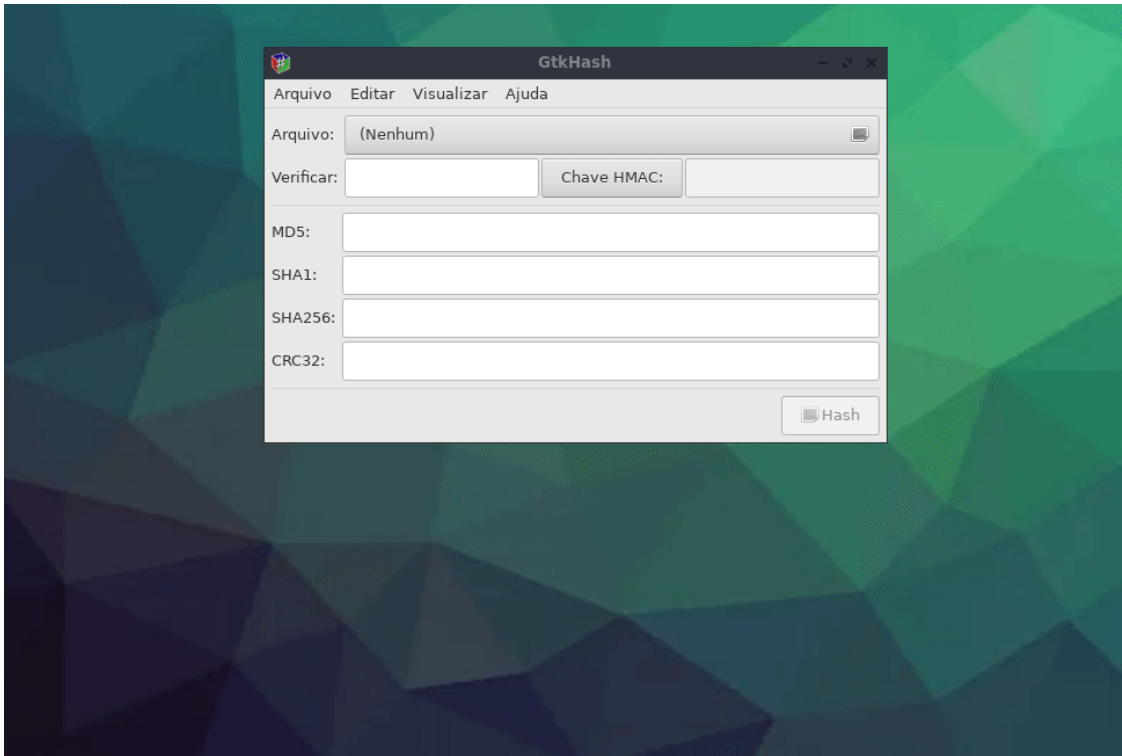
### Etapa 2: Instalar o GtkHash no ubuntu

- ☒ Basta rodar o comando no terminal

```
sudo apt install gtkhash -y
```

## Etapa 3: Gerar as chaves e testar

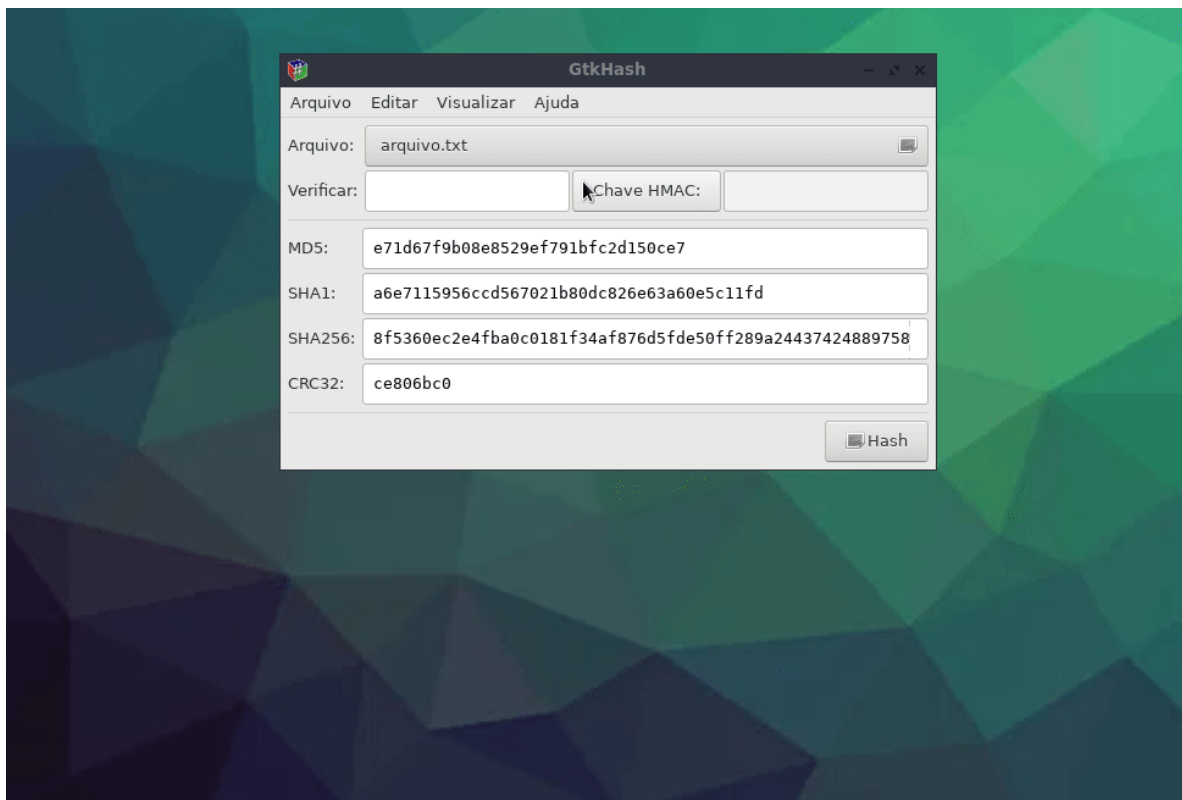
- ☒ Abra o **GtkHash**
- ☒ Abra o arquivo e clique em gerar **hash**



- ☒ Anote os **hash's** dentro do arquivo, ficando:

```
$ nano arquivo.txt
Exemplo de conteúdo:
Antes do hash:
md5 => e71d67f9b08e8529ef791bfc2d150ce7
sha1 => a6e7115956ccd567021b80dc826e63a60e5c11fd
sha256 => 8f5360ec2e4fba0c0181f34af876d5fde50ff289a24437424889758be59a3094
crc32 => ce806bc0
```

- ☒ Com o arquivo alterado, abra e gere novamente o **hash**:



☒ Adicione os hashes no arquivo depois do conteúdo anterior, ficando:

```
$ nano arquivo.txt
Exemplo de conteúdo:
Antes do hash:
md5 => e71d67f9b08e8529ef791bfc2d150ce7
sha1 => a6e7115956ccd567021b80dc826e63a60e5c11fd
sha256 => 8f5360ec2e4fba0c0181f34af876d5fde50ff289a24437424889758be59a3094
crc32 => ce806bc0

-----

Depois do hash:
md5 => 7fd8835d904efbc452915eb758f336e7
sha1 => b6f8712baac42076e0b0835cfb9634d23e1c37ad
sha256 => c52341bb0bfc2fe8f1a6e34d0e9bcb3f4ff439fa076d48493c7733be8cd2d46a
crc32 => 955a3a17
```

☒ Comparando os dois são diferentes?

Sim, são completamente diferentes.

☒ Comparando os tipos de **hash's** (md5, sha1, sha256, crc32) são diferentes? Por quê?

Sim, pois cada um dos tipos de algoritmos de hash, que calcula o hash são diferentes, logo tendo as chaves de tamanhos diferentes.

O arquivo de exemplo fica em: [Local](#)

# Criador

---

Olá me chamo Gustavo, e criei este material, para mais informações, clique nos links abaixo:

- [LinkTree](#)
- Disponível em : [Repositório de exercícios](#)