

Ejercicio Hashing and Password Cracking

Marco Jurado 20308

```
In [ ]: import hashlib
import logging
import random
```

```
In [ ]: logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(mes
```

```
In [ ]: def generar_hash_sha1(texto):
    return hashlib.sha1(texto.encode()).hexdigest()
```

```
In [ ]: def leer_archivo(path, codificaciones=['utf-8', 'latin-1']):
    for codificacion in codificaciones:
        try:
            with open(path, 'r', encoding=codificacion) as file:
                contenido = file.read().splitlines()
                logging.info(f"Archivo {path} leído correctamente con codificación")
                return contenido
        except UnicodeDecodeError as e:
            logging.warning(f"Fallo al leer {path} con {codificacion}: {e}")
            continue
    error_msg = f"No se pudo leer el archivo {path} con las codificaciones proporci
    logging.error(error_msg)
    raise ValueError(error_msg)
```

```
In [ ]: def encontrar_contraseñas(hashes_objetivo, palabras):
    encontrados = {}
    contador = 0
    total_palabras = len(palabras)
    intervalo_progreso = int(total_palabras * 0.05)
    if intervalo_progreso == 0:
        intervalo_progreso = 1

    for palabra in palabras:
        contador += 1
        hash_generado = generar_hash_sha1(palabra)
        if hash_generado in hashes_objetivo:
            encontrados[hash_generado] = palabra
            logging.info(f"Encontrado: {palabra} -> {hash_generado}")

        if contador % intervalo_progreso == 0:
            porcentaje = (contador / total_palabras) * 100
            logging.info(f"{porcentaje:.2f}% completado")

    if contador % intervalo_progreso != 0: # Capturar el último segmento de progre
        porcentaje = (contador / total_palabras) * 100
        logging.info(f"{porcentaje:.2f}% completado")
```

```
return encontrados
```

```
In [ ]: def mutar_palabras(palabras):
    mutaciones = set(palabras)
    logging.info(f"Comenzando la mutación de {len(palabras)} palabras.")

    for palabra in palabras:
        mutaciones.add(palabra.lower())
        mutaciones.add(palabra.upper())
        mutaciones.add(palabra.capitalize())
        logging.debug(f"Variantes de mayúsculas/minúsculas añadidas para {palabra}")

        mutaciones.add(palabra[::-1])
        logging.debug(f"Palabra invertida añadida: {palabra[::-1]}")

        for numero in range(10):
            mutaciones.add(f"{numero}{palabra}")
            mutaciones.add(f"{palabra}{numero}")
        logging.debug(f"Prefijos y sufijos numéricos añadidos para {palabra}")

        leet = palabra.replace('a', '4').replace('e', '3').replace('l', '1').replace('o', '0')
        mutaciones.add(leet)
        logging.debug(f"Variante Leet añadida para {palabra}: {leet}")

    logging.info(f"Total de mutaciones generadas: {len(mutaciones)}")
    return list(mutaciones)
```

```
In [ ]: hashes_objetivo = set(Leer_archivo('target_hashes.txt'))
palabras_basicas = Leer_archivo('words.txt')
# palabras_completas = Leer_archivo('realhuman_phill.txt')
# palabras_1 = Leer_archivo('engDictionary.txt')
# palabras_2 = Leer_archivo('engDictionaryEdited.txt')
# palabras_3 = Leer_archivo('engDictionaryLarge.txt')
# palabras_4 = Leer_archivo('engDictionaryLargeEdited.txt')
# palabras_5 = Leer_archivo('common-and-spanish.txt')
# palabras_6 = Leer_archivo('spanich2.txt')
# palabras_7 = Leer_archivo('top-10000-spanish-words.txt')

# EN caso de tener dataset generado en txt correr esto
palabras_totales = Leer_archivo('palabras_totales.txt')
```

```
2024-04-29 00:05:36,639 - INFO - Archivo target_hashes.txt leído correctamente con codificación utf-8
2024-04-29 00:05:36,653 - INFO - Archivo words.txt leído correctamente con codificación utf-8
2024-04-29 00:05:48,667 - INFO - Archivo palabras_totales.txt leído correctamente con codificación utf-8
```

```
In [ ]: # palabras_totales = palabras_basicas + palabras_completas + palabras_1 + palabras_2 + palabras_3 + palabras_4 + palabras_5 + palabras_6 + palabras_7
```

```
In [ ]: #mutaciones = mutar_palabras(palabras_totales)
```

Con el dataset de palabras generado vamos a generar un nuevo txt que vamos a cargar para poder replicar el funcionamiento del ejercicio.

```
In [ ]: def guardar_palabras_en_txt(palabras, nombre_archivo):  
        """Guarda una lista de palabras en un archivo de texto, cada palabra en una nueva  
        with open(nombre_archivo, 'w', encoding='utf-8') as archivo:  
            for palabra in palabras:  
                archivo.write(palabra + '\n')
```

```
In [ ]: # de no tener el txt generado correr esto  
        # guardar_palabras_en_txt(palabras_totales, 'palabras_totales.txt')
```

```
In [ ]: len(palabras_totales)
```

```
Out[ ]: 55999697
```

```
In [ ]: resultados = encontrar_contraseñas(hashes_objetivo, palabras_totales)
```

```
2024-04-29 00:05:58,465 - INFO - 5.00% completado  
2024-04-29 00:06:01,857 - INFO - 10.00% completado  
2024-04-29 00:06:04,528 - INFO - Encontrado: abcdefghijklmnopqrstuvwxyz -> 32d10c7b8  
cf96570ca04ce37f2a19d84240d3a89  
2024-04-29 00:06:05,358 - INFO - 15.00% completado  
2024-04-29 00:06:09,016 - INFO - 20.00% completado  
2024-04-29 00:06:12,759 - INFO - 25.00% completado  
2024-04-29 00:06:15,894 - INFO - Encontrado: easypass111 -> d5bdd5ec4b1ef2da23b22dc2  
db99731d9cb2823d  
2024-04-29 00:06:16,259 - INFO - 30.00% completado  
2024-04-29 00:06:19,608 - INFO - 35.00% completado  
2024-04-29 00:06:23,246 - INFO - 40.00% completado  
2024-04-29 00:06:26,606 - INFO - 45.00% completado  
2024-04-29 00:06:29,952 - INFO - 50.00% completado  
2024-04-29 00:06:33,428 - INFO - 55.00% completado  
2024-04-29 00:06:35,771 - INFO - Encontrado: ANTIGUAGUATEMALA! -> ce00a097dc3ff6d558  
9d8c130f14c9fe0084e4fe  
2024-04-29 00:06:36,850 - INFO - 60.00% completado  
2024-04-29 00:06:40,315 - INFO - 65.00% completado  
2024-04-29 00:06:43,730 - INFO - 70.00% completado  
2024-04-29 00:06:47,260 - INFO - 75.00% completado  
2024-04-29 00:06:50,609 - INFO - Encontrado: P4ssw0rd -> 264bc0768362a68984faea923ef  
aa21f67f4d10a  
2024-04-29 00:06:50,734 - INFO - 80.00% completado  
2024-04-29 00:06:54,214 - INFO - 85.00% completado  
2024-04-29 00:06:54,709 - INFO - Encontrado: &*() -> 5ea777cd7ac9149f843fc39386570b2  
b5c2ce586  
2024-04-29 00:06:57,723 - INFO - 90.00% completado  
2024-04-29 00:07:01,086 - INFO - 95.00% completado  
2024-04-29 00:07:04,546 - INFO - 100.00% completado  
2024-04-29 00:07:04,547 - INFO - 100.00% completado
```

```
In [ ]: for hash_val, pass_val in resultados.items():  
        print(f'Hash: {hash_val}, Contraseña: {pass_val}')
```

Hash: 32d10c7b8cf96570ca04ce37f2a19d84240d3a89, Contraseña: abcdefghijklmnopqrstuvwx
yz
Hash: d5bdd5ec4b1ef2da23b22dc2db99731d9cb2823d, Contraseña: easypass111
Hash: ce00a097dc3ff6d5589d8c130f14c9fe0084e4fe, Contraseña: ANTIGUAGUATEMALA!
Hash: 264bc0768362a68984faea923efaa21f67f4d10a, Contraseña: P4ssw0rd
Hash: 5ea777cd7ac9149f843fc39386570b2b5c2ce586, Contraseña: &*()