UNIVERSIDAD DEL VALLE DE GUATEMALA

Cifrado de Información

Sección 10

Ludwing Cano

Laboratorio 1 B

Encriptado y Decriptado de Texto

José Daniel Gómez Cabrera 21429

Repositorio

https://github.com/JDgomez2002/cipher/tree/main/lab1/lab1-b

Implementar un análisis de fuerza bruta por frecuencias. Para cada uno de los siguientes archivos y determinar cual fue la clave utilizada en cada caso, y decriptar el mensaje. (100 pts).

En los archivos de texto cipher1.txt, cipher2.txt y cipher3.txt se encuentran tres textos cifrados, en los que se usaron
diferentes métodos, como sigue:

- cipher1.txt, cifrado Caesar.
- cipher2.txt, cifrado afín.
- cipher3.txt, cifrado Vigenére

Sugerencias:

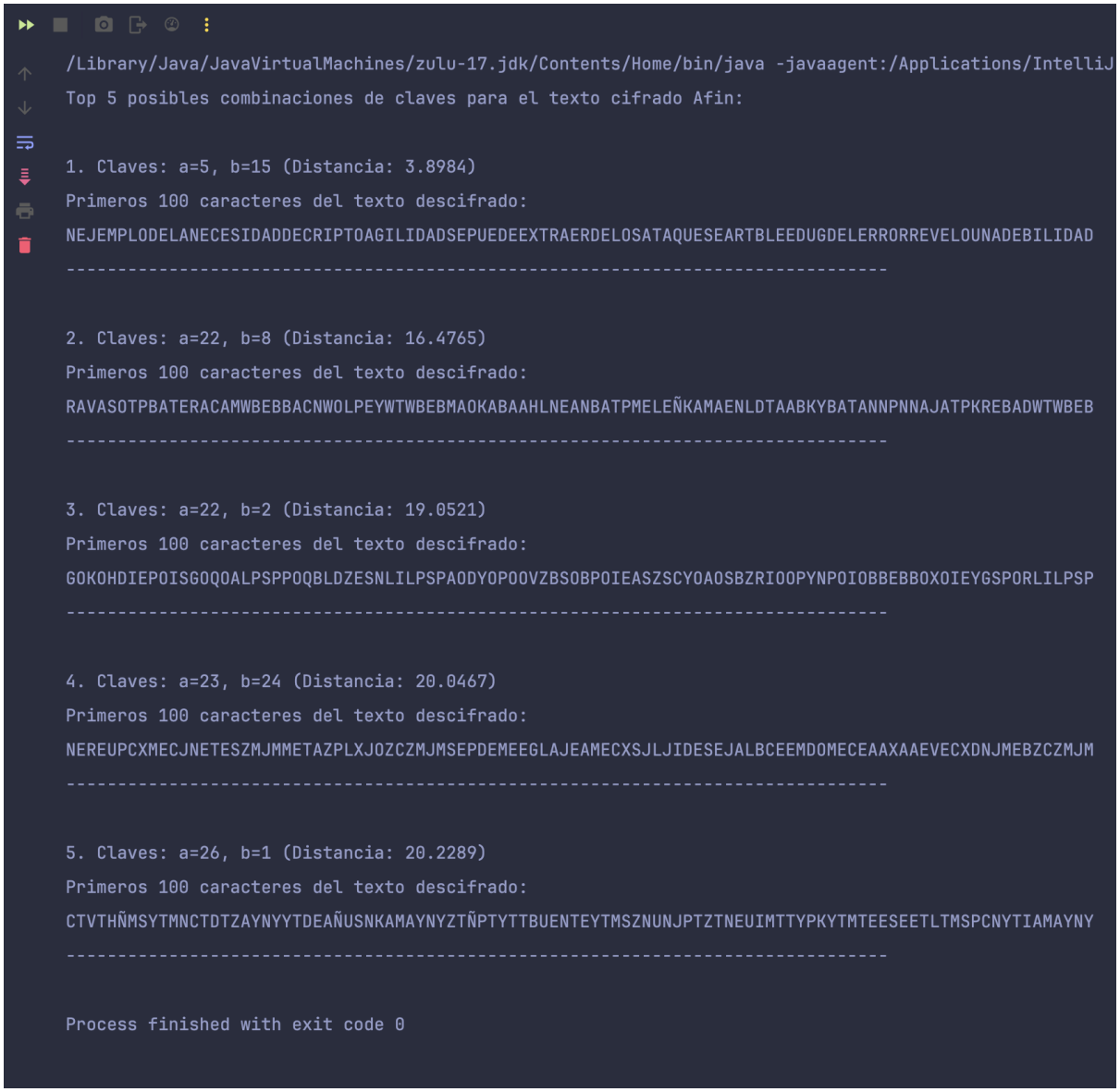
- Construir una función que haga el proceso de fuerza bruta. Un ciclo que barre todas las llaves posibles en el espacio de claves.
- Por cada clave se almacena un arreglo de la métrica obtenida y se ordena de mejor a peor. La función debe de devolver las k mejores claves, se entiende que la clase más probable que descifra el texto es la que tiene mejor métrica

Caesar

```
Top 5 posibles claves para el texto cifrado Caesar:

1. Clave: 23 (Distancia: 8.8975)
Primeros 100 caracteres del texto descifrado:
NNUESTROLABERINTODIGITALENCONSTANTEEVOLUCIONLAAGILIDADCRIPTOGRAFICACRIPTOAGILIDADPARAABREVIARESUNMEC
--------------------------------------------------------------------------------

2. Clave: 12 (Distancia: 24.0885)
Primeros 100 caracteres del texto descifrado:
XXFODECZVLMOCSXEZÑSQSELVOXNZXDELXEOOGZVFNSZXVLLQSVSÑLÑNCSAEZQCLPSNLNCSAEZLQSVSÑLÑALCLLMCOGSLCODFXWON
--------------------------------------------------------------------------------

3. Clave: 4 (Distancia: 25.0979)
Primeros 100 caracteres del texto descifrado:
FFNWLMKHDSTWKAFMHVAYAMSDWFUHFLMSFMWWÑHDNUAHFDSSYADAVSVUKAIMHYKSXAUSUKAIMHSYADAVSVISKSSTKWÑASKWLNFEWU
--------------------------------------------------------------------------------

4. Clave: 19 (Distancia: 25.3562)
Primeros 100 caracteres del texto descifrado:
QQYIWXVSOEFIVMQXSHMKMXEOIQGSQWXEQXIIZSOYGMSQOEEKMOMHEHGVMTXSKVEJMGEGVMTXSEKMOMHEHTEVEEFVIZMEVIWYQPIG
--------------------------------------------------------------------------------
```

NNUESTROLABERINTODIGITALENCONSTANTEEVOLUCIONLAAGILIDADCRIPTOGRAFI
CACRIPTOAGILIDADPARAABREVIARESUNMEC

Afin



NEJEMPLODELANECESIDADDECRIPTOAGILIDADSEPUEDEEXTRAERDELOSATAQUESEARTBLEEDUGDELERRORREVELOUNADEBILIDAD

Vigenere

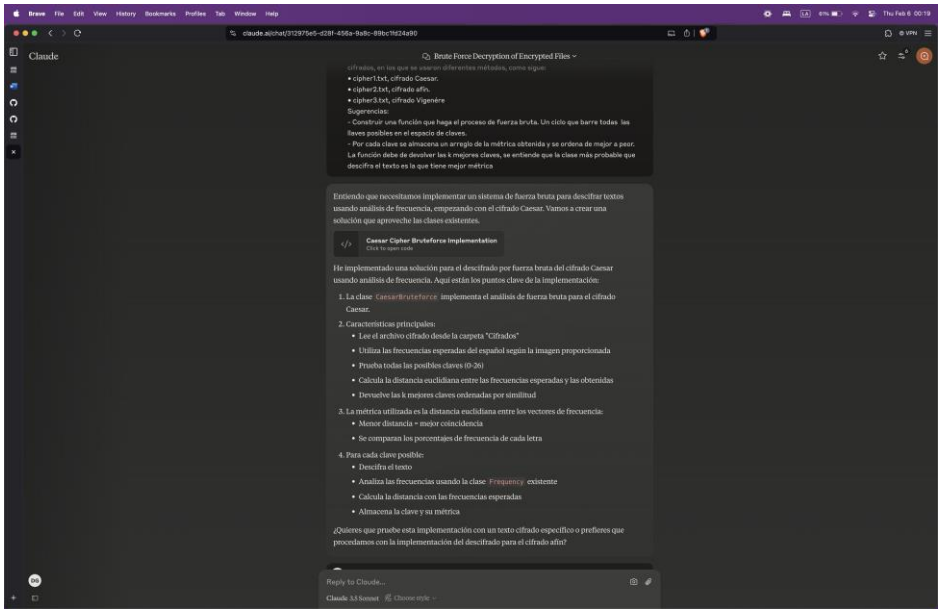
```
/Library/Java/JavaVirtualMachines/zulu-17.jdk/Contents/Home/bin/java -javaagent:/Applica
Longitudes de clave más probables: [2, 3, 6]

1. Clave: PAYASO (Distancia: 0.0179)
Primeros 200 caracteres del texto descifrado:
AUNQUEELERRORHEARTBLEEDSEHASOLUCIONADOSIEMPREHAYUNANUEVAAMENAZAENELHORIZONTEHOYLACUANTIC
--------------------------------------------------------------------------------

2. Clave: PAY (Distancia: 0.1498)
Primeros 200 caracteres del texto descifrado:
AUNBNUELECKFRHELKKBLEOVJEHADHBUCIZFQDOSSWCPRERSOUNAXNUVAAWWDAZAOFULHOCAPONTOZFYLANNQNTIN
--------------------------------------------------------------------------------

3. Clave: YA (Distancia: 0.1553)
Primeros 200 caracteres del texto descifrado:
RUNQOSVLERMDJHEAMISLEEXHVHASJZMCIOHOUOSIYAHREHUNMNANOSNAAMYBRZAEHSCHORCÑGNTEBDPLACOOETIC
--------------------------------------------------------------------------------
```

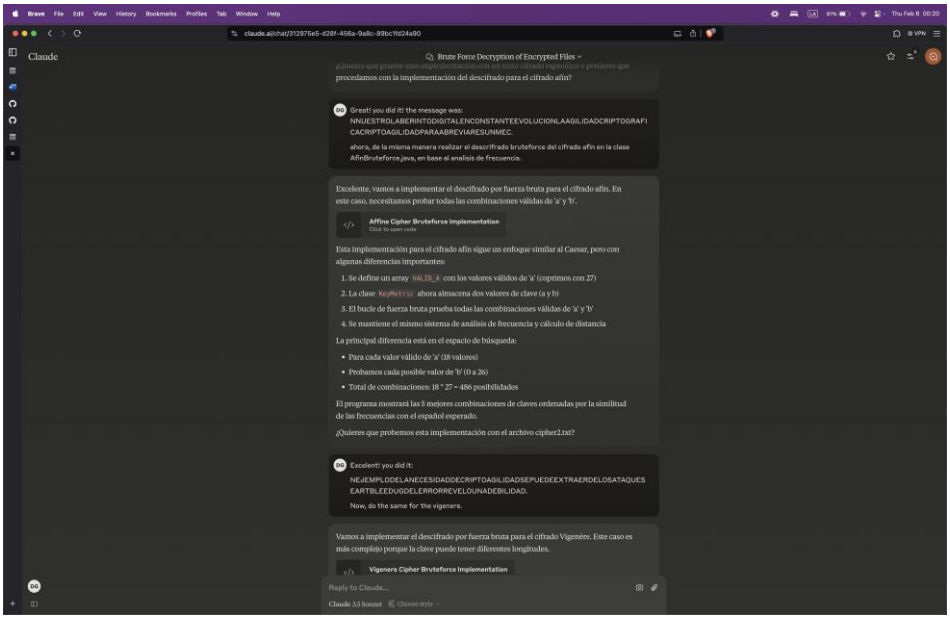
AUNQUEELERRORHEARTBLEEDSEHASOLUCIONADOSIEMPREHAYUNANUEVAAMENAZAENELHORIZONTEHOYLACUANTICAESESAAMENAZAQUEPUEDEATRAVESARTODASNUESTRASDEFENSASANTESDEQUETODOESTEPERDIDODEBEMOSADOPTARLACRIPTOAGILIDADPARADE

Ayuda de Claude con el modelo Sonnet 3.5

Utilice el modelo Sonnet 3.5 para poder obtener las sugerencias y estrategias más prometedoras para poder implementar los algoritmos de descrifrado por fuerza bruta. Claude no permite compartir el chat o sesión, pero en este documento se presetan algunas capturas de pantalla.

Para el último fuerza bruta de Vigenere, fue clave realizar un previo analisis de la longitud de la llave, ya que sin este análisis, fue imposible encontrar la llave únicamente por medio del análisis de frecuencia. Este análisis previo de la longitud de a llave permitió descifrar el mensaje a fuerza bruta.


Claude
Brute Force Decryption of Encrypted Files
cifrados, en los que se usaron diferentes métodos, como sigue:
• cipher1.txt, cifrado Caesar.
• cipher2.txt, cifrado afín.
• cipher3.txt, cifrado Vigenére
Sugerencias:
- Construir una función que haga el proceso de fuerza bruta. Un ciclo que barre todas las llaves posibles en el espacio de claves.
- Por cada clave se almacena un arreglo de la métrica obtenida y se ordena de mejor a peor. La función debe de devolver las k mejores claves, se entiende que la clase más probable que descifra el texto es la que tiene mejor métrica
Entiendo que necesitamos implementar un sistema de fuerza bruta para descifrar textos usando análisis de frecuencia, empezando con el cifrado Caesar. Vamos a crear una solución que aproveche las clases existentes.
Caesar Cipher Bruteforce Implementation
Click to open code
He implementado una solución para el descifrado por fuerza bruta del cifrado Caesar usando análisis de frecuencia. Aquí están los puntos clave de la implementación:
1. La clase CaesarBruteforce implementa el análisis de fuerza bruta para el cifrado Caesar.
2. Características principales:
• Lee el archivo cifrado desde la carpeta "Cifrados"
• Utiliza las frecuencias esperadas del español según la imagen proporcionada
• Prueba todas las posibles claves (0-26)
• Calcula la distancia euclidiana entre las frecuencias esperadas y las obtenidas
• Devuelve las k mejores claves ordenadas por similitud
3. La métrica utilizada es la distancia euclidiana entre los vectores de frecuencia:
• Menor distancia = mejor coincidencia
• Se comparan los porcentajes de frecuencia de cada letra
4. Para cada clave posible:
• Descifra el texto
• Analiza las frecuencias usando la clase Frequency existente
• Calcula la distancia con las frecuencias esperadas
• Almacena la clave y su métrica
¿Quieres que pruebe esta implementación con un texto cifrado específico o prefieres que procedamos con la implementación del descifrado para el cifrado afín?
Reply to Claude...
Claude 3.5 Sonnet
Choose style


Claude
Brute Force Decryption of Encrypted Files
procedamos con la implementación del descifrado para el cifrado afín?
Great! you did it! the message was:
NNUESTROLABERINTODIGITALENCONSTANTEEVOLUCIONLAAGILIDADCRIPTOGRAFICACRIPTOAGILIDADPARAABREVIARESUNMEC.
ahora, de la misma manera realizar el descifrado bruteforce del cifrado afín en la clase AfinBruteforce.java, en base al analisis de frecuencia.
Excelente, vamos a implementar el descifrado por fuerza bruta para el cifrado afín. En este caso, necesitamos probar todas las combinaciones válidas de 'a' y 'b'.
Affine Cipher Bruteforce Implementation
Click to open code
Esta implementación para el cifrado afín sigue un enfoque similar al Caesar, pero con algunas diferencias importantes:
1. Se define un array VALID_A con los valores válidos de 'a' (coprimos con 27)
2. La clase KeyMetric ahora almacena dos valores de clave (a y b)
3. El bucle de fuerza bruta prueba todas las combinaciones válidas de 'a' y 'b'
4. Se mantiene el mismo sistema de análisis de frecuencia y cálculo de distancia
La principal diferencia está en el espacio de búsqueda:
• Para cada valor válido de 'a' (18 valores)
• Probamos cada posible valor de 'b' (0 a 26)
• Total de combinaciones: 18 * 27 = 486 posibilidades
El programa mostrará las 5 mejores combinaciones de claves ordenadas por la similitud de las frecuencias con el español esperado.
¿Quieres que probemos esta implementación con el archivo cipher2.txt?
Excelent! you did it:
NEJEMPLODELANECESIDADDECRIPTOAGILIDADSEPUEDEEXTRAERDELOSATAQUESEARTBLEEDUGDELERRORREVELOUNADEBILIDAD.
Now, do the same for the vigenere.
Vamos a implementar el descifrado por fuerza bruta para el cifrado Vigenère. Este caso es más complejo porque la clave puede tener diferentes longitudes.
Vigenere Cipher Bruteforce Implementation
Reply to Claude...
Claude 3.5 Sonnet
Choose style