# Analisis de Frecuencia Implementado en Java

Andrés Cruz Chipol
Facultad De Ciencias de la Computaicon - Criptografia
Benemerita Universidad Autonoma de Puebla
Veracruz, México
andres.cruzc@alumno.buap.mx

***Abstract*— Análisis de frecuencia es una técnica de análisis que consiste en el aprovechamiento de estudios sobre la frecuencia de las letras o de los grupos de letras en algún idioma para descifrar el texto sin la necesidad de tener la llave de cifrado. Se desarrolla e implementa el algoritmo.**

***Keywords: cripto, cifrado, analisis de frecuencia, descifrado***

## I. INTRODUCCIÓN

El análisis de frecuencias se basa por el hecho de que los textos, ciertas letras o combinaciones de letras suelen repetirse a menudo, estas mismas aparecen en un texto repetidamente, dependiendo del idioma cambiaran de frecuencia.

Tenemos en cuenta que al menos para el español las letras más comunes son la letra A y la E, sin embargo, como consideramos que estamos escribiendo este algoritmo para caracteres de 255 en la computadora, vamos a tomar como el más común el espacio. Mientras tanto que algunas letras menos comunes podrían ser k,w,x. El objetivo en si es contar y ordenar la frecuencia de las letras para poder reemplazar con alguna configuración dada por su frecuencia.

Describiremos el algoritmo e implementación en Java utilizando NetBeans 8.2 Desarrollando una interfaz

## II. HISTORIA

La primera explicación que se tiene documentada sobre el análisis de frecuencias, se dio en el siglo IX por el filosofo árabe Al-kindi que desarrollo en un manuscrito para el descifrado de mensajes criptográficos.

Su uso se extendió bastante que fue usado en toda Europa durante la época del renacimiento, aun que claro, muchos intentaron “bloquear” el análisis de frecuencias. Como utilizar letras menos comunes en las letras mas comunes remplazándolas.

También el uso del cifrado polialfabetico que es el uso de varios alfabetos para el cifrado.

La sustitución poligráfica que son esquemas donde pares o tríos de letras eran cifrados como una unidad única.

Claro que al final de cuentas como este mismo algoritmo, tienen ciertas desventajas, como el hecho que complicaban demasiado el cifrado como el descifrado que provocaba errores.

El análisis de frecuencias solo necesita el conocimiento básico de la estadística de un texto plano, para poder obtener el texto descifrado, en su tiempo claramente era un reto hacer estas cuestiones, pero este tipo de análisis hoy en día es dejado para los informáticos, ya que todas estas cuentas pueden realizar en un par de segundos, dado a la potencia de computación algunos métodos de cifrado clásico ya no son un estándar o incluso son simplemente nulos a la hora de cifrar el texto.

## III. DESARROLLO

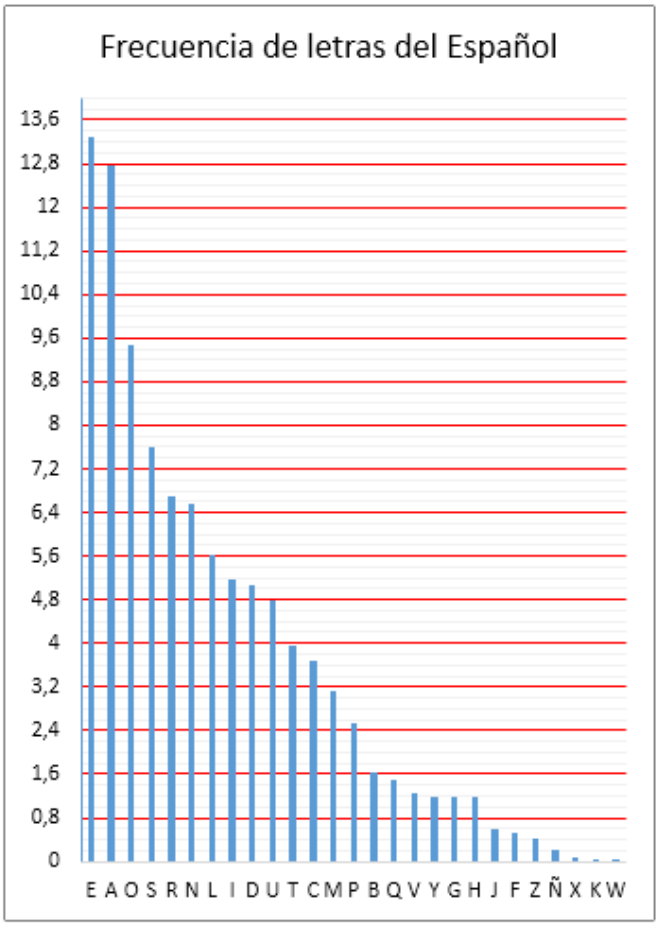
El desarrollo del algoritmo prácticamente es muy sencillo, sin embargo, hay algunas inconveniencias a la hora de tratar un texto plano.

Pensemos en algo… los diferentes estudios que existen para demostrar que letras son las mas frecuentes en el español.

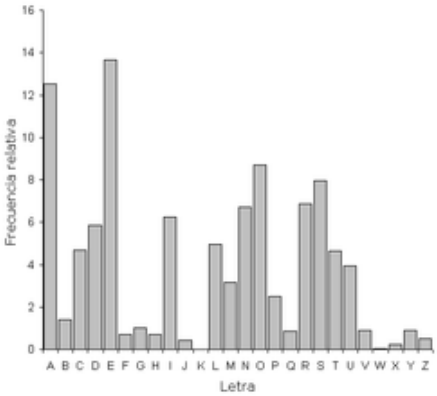
| Letra | Porcentaje | Letra | Porcentaje |
|---|---|---|---|
| a | 8.2 | n | 6.7 |
| b | 1.5 | o | 7.5 |
| c | 2.8 | p | 1.9 |
| d | 4.3 | q | 0.1 |
| e | 12.7 | r | 6.0 |
| f | 2.2 | s | 6.3 |
| g | 2.0 | t | 9.1 |
| h | 6.1 | u | 2.8 |
| i | 7.0 | v | 1.0 |
| j | 0.2 | w | 2.4 |
| k | 0.8 | x | 0.2 |
| l | 4.0 | y | 2.0 |
| m | 2.4 | z | 0.1 |

| Letra | Porcentaje | Letra | Porcentaje |
|---|---|---|---|
| A | 12,53 | O | 8,68 |
| B | 1,42 | P | 2,51 |
| C | 4,68 | Q | 0,88 |
| D | 5,86 | R | 6,87 |
| E | 13,68 | S | 7,98 |
| F | 0,69 | T | 4,63 |
| G | 1,01 | U | 3,93 |
| H | 0,70 | V | 0,90 |
| I | 6,25 | W | 0,02 |
| J | 0,44 | X | 0,22 |
| K | 0,01 | Y | 0,90 |
| L | 4,97 | Z | 0,52 |
| M | 3,15 | | |
| N | 6,71 | | |
| Ñ | 0,31 | | |

**Figura 1. Porcentaje de aparición de letras**




Existen muchas tablas que nos indican la frecuencia de las palabras que posiblemente estén bien o estén mal. Pero claro, depende demasiado de lo que conocemos del texto plano. Sin embargo, podemos aclarar que incluso en la mayoría de estas graficas podemos notar que las letras A, E, O son las letras más frecuentes que se pueden encontrar en nuestro alfabeto español.




Lo primero que tenemos que hacer es contabilizar nuestras letras del texto plano. Lo principal es estar haciendo el procesado de texto tenemos que separar nuestras letras. Mientras las vamos contabilizando. Como nuestros símbolos totales son 256 pues tenemos que contabilizar al menos los que encontremos en el texto, aun que claro, encontraremos normalmente lo que es nuestro abecedario en español.

Una vez que tengamos nuestras letras contabilizadas tenemos que crear una lista de mayor a menor para saber que letras estuvieron están en ese puesto.

Comenzamos entonces a probar nuestras tablas de frecuencias que encontramos en nuestras investigaciones.

```
Similitud Palabras: 0 Total: 29 Igualdad: 0.0%
Similitud Palabras: 1 Total: 29 Igualdad: 3.4482758%
Similitud Palabras: 0 Total: 29 Igualdad: 0.0%
Similitud Palabras: 1 Total: 29 Igualdad: 3.4482758%
Similitud Palabras: 2 Total: 29 Igualdad: 6.8965516%
Similitud Palabras: 2 Total: 29 Igualdad: 6.8965516%
Similitud Palabras: 3 Total: 29 Igualdad: 10.344828%
Similitud Palabras: 3 Total: 29 Igualdad: 10.344828%
Similitud Palabras: 2 Total: 29 Igualdad: 6.8965516%
Similitud Palabras: 2 Total: 29 Igualdad: 6.8965516%
PS C:\Users\andy_\Desktop>
```

Empezamos con texto que apenas y tiene 29 letras, lo cual obviamente será difícil de descifrar el resultado que tenemos es el esperado.

Probemos entonces con un poco más de letras.

```
Similitud Palabras: 10 Total: 146 Igualdad: 6.849315%
Similitud Palabras: 4 Total: 146 Igualdad: 2.739726%
Similitud Palabras: 7 Total: 146 Igualdad: 4.7945204%
Similitud Palabras: 3 Total: 146 Igualdad: 2.0547945%
Similitud Palabras: 15 Total: 146 Igualdad: 10.2739725%
Similitud Palabras: 17 Total: 146 Igualdad: 11.643836%
Similitud Palabras: 18 Total: 146 Igualdad: 12.328767%
Similitud Palabras: 20 Total: 146 Igualdad: 13.69863%
Similitud Palabras: 8 Total: 146 Igualdad: 5.479452%
Similitud Palabras: 8 Total: 146 Igualdad: 5.479452%
PS C:\Users\andy_\Desktop>
```

Ahora tenemos 146 Caracteres

Puede parecer bastante caracteres, pero realmente no lo es, puede que incluso algunas letras no comunes estén más repetidas que otras de lo común que no es. Por lo cual nuestro máximo porcentaje fue del 13.69%

Bien pues empecemos a colocar texto cifrado con una cantidad considerable de texto con nuestras distintas tablitas que se encuentran listas para descifrar.

```
Similitud Palabras: 110 Total: 695 Igualdad: 15.827338%
Similitud Palabras: 84 Total: 695 Igualdad: 12.086331%
Similitud Palabras: 123 Total: 695 Igualdad: 17.697842%
Similitud Palabras: 102 Total: 695 Igualdad: 14.676259%
Similitud Palabras: 141 Total: 695 Igualdad: 20.28777%
Similitud Palabras: 173 Total: 695 Igualdad: 24.892086%
Similitud Palabras: 217 Total: 695 Igualdad: 31.223022%
Similitud Palabras: 249 Total: 695 Igualdad: 35.82734%
Similitud Palabras: 163 Total: 695 Igualdad: 23.453238%
Similitud Palabras: 163 Total: 695 Igualdad: 23.453238%
PS C:\Users\andy_\Desktop>
```

Máximo fue de 35.82%

```
Similitud Palabras: 556 Total: 2155 Igualdad: 25.800465%
Similitud Palabras: 374 Total: 2155 Igualdad: 17.354988%
Similitud Palabras: 532 Total: 2155 Igualdad: 24.686775%
Similitud Palabras: 375 Total: 2155 Igualdad: 17.401392%
Similitud Palabras: 874 Total: 2155 Igualdad: 40.556843%
Similitud Palabras: 874 Total: 2155 Igualdad: 40.556843%
Similitud Palabras: 569 Total: 2155 Igualdad: 26.403713%
Similitud Palabras: 569 Total: 2155 Igualdad: 26.403713%
Similitud Palabras: 1012 Total: 2155 Igualdad: 46.960556%
Similitud Palabras: 1018 Total: 2155 Igualdad: 47.23898%
PS C:\Users\andy_\Desktop>
```

Máximo fue de 47.2389%

```
Similitud Palabras: 2863 Total: 4951 Igualdad: 57.826702%
Similitud Palabras: 1606 Total: 4951 Igualdad: 32.437893%
Similitud Palabras: 2237 Total: 4951 Igualdad: 45.182793%
Similitud Palabras: 1264 Total: 4951 Igualdad: 25.530195%
Similitud Palabras: 2166 Total: 4951 Igualdad: 43.748737%
Similitud Palabras: 2338 Total: 4951 Igualdad: 47.222782%
Similitud Palabras: 1904 Total: 4951 Igualdad: 38.45688%
Similitud Palabras: 2076 Total: 4951 Igualdad: 41.930923%
Similitud Palabras: 1449 Total: 4951 Igualdad: 29.266815%
Similitud Palabras: 1461 Total: 4951 Igualdad: 29.50919%
```

```
Similitud Palabras: 209 Total: 909 Igualdad: 22.9923%
Similitud Palabras: 165 Total: 909 Igualdad: 18.151815%
Similitud Palabras: 209 Total: 909 Igualdad: 22.9923%
Similitud Palabras: 167 Total: 909 Igualdad: 18.371838%
Similitud Palabras: 589 Total: 909 Igualdad: 64.79648%
Similitud Palabras: 522 Total: 909 Igualdad: 57.425743%
Similitud Palabras: 719 Total: 909 Igualdad: 79.09791%
Similitud Palabras: 652 Total: 909 Igualdad: 71.72717%
Similitud Palabras: 324 Total: 909 Igualdad: 35.643566%
Similitud Palabras: 324 Total: 909 Igualdad: 35.643566%
PS C:\Users\andy_\Desktop>
```

Máximo fue de 79.097%

Para el hecho de que no tenemos un simple abecedario, si no del hecho que tenemos muchos símbolos ascii en la computadora, podemos decir que sus porcentajes están muy bien, ya que seria absurdo que algún porcentaje de esos llegara al 90% sin antes un análisis mucho mas profundo y un algoritmo mas complejo del que se tiene planteado.

Podemos observar que de 10 tablas de frecuencias en promedio algunas mucho mejores que otras, pero que algunas raras veces alcanzaban un porcentaje “alto” y otros se mantenían en un porcentaje “alto” dependiendo del texto.

Se podría pensar entonces… que la mejor solución seria utilizar el que mejor promedio tenga será escogido. Sin embargo, no decidí hacer eso.

El hecho esta que las palabras de un documento varían temas, las áreas de investigación de la minería de datos nos dicen que podemos proyectar documentos de manera semántica, lo cual nos dice que la frecuencia de las palabras seguirá cambiando de acuerdo con el tema que se habla. Por ejemplo. Si el tema en específico habla de “Twitter”. La frecuencia de la ‘t’,’w’ y la ‘i’ como la ‘r’. Causaría ruido si usara solamente una tabla de frecuencia. Dándonos márgenes de Errores.

Dado a esta problemática. Sin salirme de el algoritmo de análisis de frecuencia, decidí usar estas 10 tablas de análisis de frecuencia para poder descifrar. Por lo cual de las 10 tablas se analizarán, obtendremos un % y el mas alto de ellos será el escogido para mostrar el texto descifrado.

Así quitaríamos un margen de error en vez de solamente utilizar una sola tabla de frecuencia. Pero también no nos estamos saliendo del algoritmo original, simplemente comparándolos entre las distintas tablas que se crearon de las tablas ya mostradas en este artículo.

## IV. IMPLEMENTACION DEL ALGORITMO

La interfaz de forma general.


Se extrae la dirección absoluta de los archivos Arrastrando el archivo y llevándolos hacia las cajitas de archivo.


El botón que limpia los textos y las direcciones.

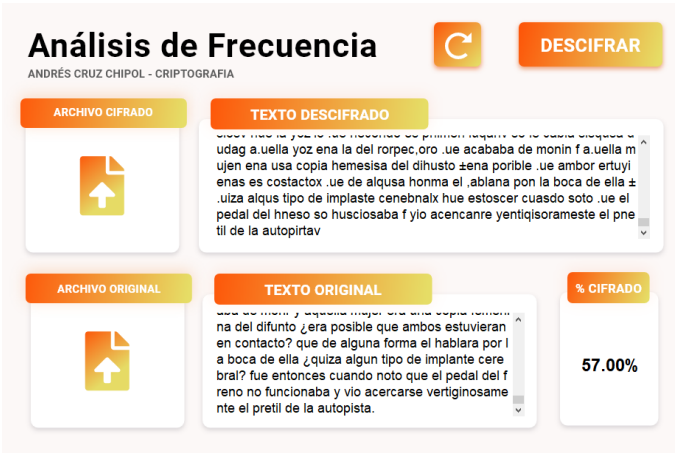
El texto se descifra por el código.

Hice una nueva clase con el nombre estructuraCifrado la cual almacenara el texto descifrado junto con el porcentaje obtenido a la hora dedescifrarlo.
Este mismo porcentaje ser va a comparar con los nueve demás descifrados para poder obtener el mejor resultado, lo que queremos es que en nuestra nueva clase estructuraCifrado se guarde el resultado que tenga mejor porcentaje para poder mostrar ese texto en la interfaz.

Una vez cargado los dos archivos se podrán descifrar. Por el hecho de que primero descifra y después compara. Pero todos los resultados se obtienen al final. Nunca se utiliza el texto original para el análisis de frecuencia. Solo para comparar los resultados y obtener un porcentaje.

```
int contador = 0;
for(int i =0; i < textDescifrado.length();i++){
    if(textDescifrado.charAt(i) == textOriginal.charAt(i)){
        contador++;
       // System.out.println(textDescifrado.charAt(i)+"="+textPrue
    }
}
//System.out.println("Similitud Palabras: " + contador + " Total:
retornar.porciento = (float)((contador*100)/textPrueba.length());
retornar.txt = textDescifrado;
return retornar;
```

La variable retornar pertenece a la clase estructura Cifrado, ya que guarda el porcentaje y el texto descifrado.




Asi es como podemos ver nuestro programa descifrado.

## V. CONCLUSIONES

El análisis de frecuencia parece ser el algoritmo de descifrado menos complejo de realizar sin la necesidad de tener la llave. El problema puede surgir por el hecho de que este tipo de técnica ya no nos podría ser útil en el presente.

Si bien los cifrados anteriores como el cifrado de cesar nacen en tiempos donde el poder computacional no existía, tratar de evitar el análisis de frecuencia era mas complejo. Hoy en día se pueden burlar de manera muy fácil este análisis, pues solamente hace falta compilar nuestro cifrado a maneras muy complejas, cálculos que solamente las computadoras de hoy en día pueden hacer. Lo cual complicaría demasiado descifrar el texto llano.

## VI. REFERENCIAS


[1] Como desencriptar un texto usando análisis de frecuencia (nahuelbrandan.com)

[2] Análisis de Frecuencias – Numerentur.org

[3] Análisis de frecuencia (cryptomex.org)

[4] Análisis de frecuencias - Wikipedia, la enciclopedia libre

[5] Cesar decryption algorithm, by the method of frequency points in the Spanish language . Bárbara E. Sánchez Rinza, Sergio R. Cruz –Góme