Universidad de Nacional de San Agustín de Arequipa
Facultad de Ingeniería Producción y Servicios
Escuela Profesional Ciencia de la Computación

Laboratorio Nro. 02

Cifrado Polialfabético

Integrantes: Castillo Caccire, Kemely
Chullunquía Rosas, Sharon Rossely
Lipe Brayan, Brayan Alexander
Docente: Antonio Cruz Gamero, Franklin Luis

Fecha de entrega: 25 de septiembre de 2021

Arequipa, Perú

1. Actividades

El código fuente de cada actividad está desarrollado en **Python**, y se encuentra en el siguiente repositorio: GitHub

1.1. Cifrado de Vignere

1. Implementar un cifrador de Vignere, donde se pueda seleccionar el módulo, alfabeto módulo 27 o módulo 191 (ASCII), ingresar el texto claro (en archivo o por interface) y genere la cifra resultante

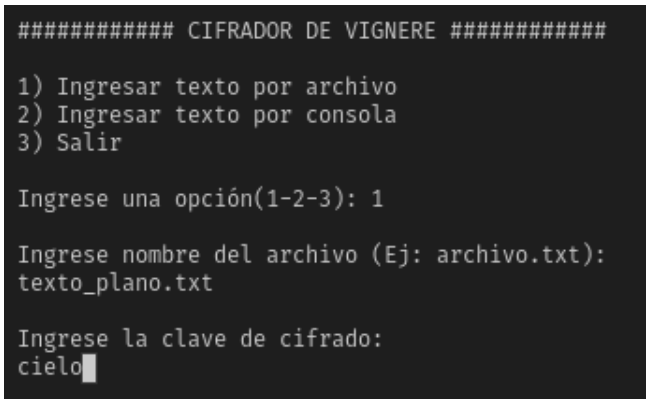

Figura 1: Menú para ingreso de texto claro

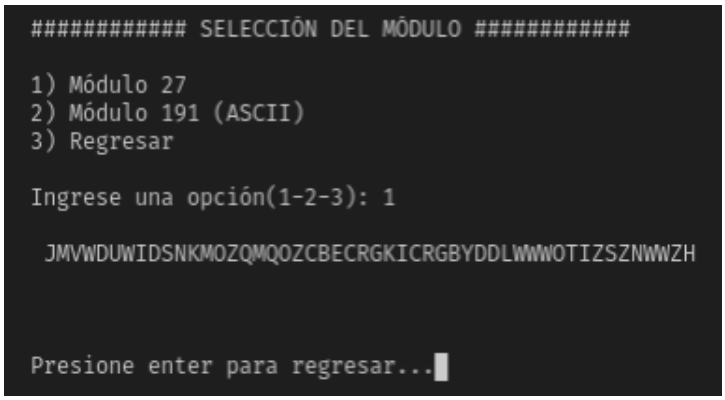

Figura 2: Menú para seleccion de módulo

- Para el módulo 27:

Antes de cifrar, preprocesamos el texto, eliminamos saltos de línea, espacios en blanco y convertimos a mayúsculas todo. Luego definimos nuestro alfabeto como se ve en la figura 11

```python
# Para cifrar con módulo 27
if modulo == 27:
  # Preprocesando texto plano
  texto = preprocesar(texto)
  texto = a_mayusculas(texto)
  clave = a_mayusculas(clave)
  texto_cifrado = ""
  alfabeto = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
  ind = 0
  for car in texto:
    caracterText_index = alfabeto.index(car)
    caracterClav_index = alfabeto.index(clave[ind])
    posicion_cifra = (caracterText_index + caracterClav_index) % 27
    texto_cifrado += alfabeto[posicion_cifra]
    ind += 1
    if ind == len(clave):
      ind = 0
  with open('texto-cifrado_27.txt', 'w') as output:
    output.write(texto_cifrado)
  print("\n",texto_cifrado,'\n')
```

Figura 3: Definiendo alfabeto módulo 27.

Después aplicamos matemáticas discretas con módulo 27 para cifrar:

```python
caracterText_index = alfabeto.index(car)
caracterClav_index = alfabeto.index(clave[ind])
posicion_cifra = (caracterText_index + caracterClav_index) % 27
texto_cifrado += alfabeto[posicion_cifra]
```

Figura 4: Formula para hallar cada carácter cifrado.

Siendo este un ejemplo.

Texto claro:

Hermoso es el cielo en el atardecer de tus ojos maravillosos

Resultado:

JMVWDUWIDSNKMOZQMQOZCBECRGKICRGBYDDLWWWOTIZSZNWWZH

- Para el módulo 191:

Igualmente, antes de cifrar preprocesamos el texto y definimos nuestro alfabeto, en este caso ASCII módulo 191.

```
# Para cifrar con módulo 191
elif modulo == 191:
  # Solo quitamos espacios y saltos de linea
  texto = texto.replace(' ','').replace('\n','')
  clave = a_mayusculas(clave)
  texto_cifrado = ""
  # De esta lista se excluirá carácteres con \
  alfabeto = [chr(i) for i in range(33, 256)]
  ind = 0
  for car in texto:
    caracterText_index = alfabeto.index(car)
    caracterClav_index = alfabeto.index(clave[ind])
    posicion_cifra = (caracterText_index + caracterClav_index) % 191

    # Para no contar carácteres con \
    if posicion_cifra < 94:
      texto_cifrado += alfabeto[posicion_cifra]
    elif posicion_cifra >= 94:
      posicion_cifra += 32
      if posicion_cifra >= len(alfabeto):
        posicion_cifra = posicion_cifra % 191
        texto_cifrado += alfabeto[posicion_cifra]
      else:
        # En el peor de los casos podria volver a caer en un caracter /
        if posicion_cifra == 126 or posicion_cifra == 127 or posicion_cifra == 140:
          posicion_cifra %= 32
          texto_cifrado += alfabeto[posicion_cifra]
        else:
          texto_cifrado += alfabeto[posicion_cifra]
```

Figura 5: Definiendo alfabeto ASCII moódulo 191.

Excluimos caracteres que tiene de prefijo ’´, y ciframos mediante matemáticas discretas el texto claro con módulo 191. A continuación, se muestra un ejemplo:

Texto claro:

Hermoso es el cielo en el atardecer de tus ojos maravillosos

Resultado:

j-¶¸½µ · ©¾³®«-°º±-²ºº£¼¥½²§«©½²§¼¹¾½¬ · · ¸¯´©º´º® · · ºÁ

2. Verificar cifrando “Creer que es posible es el paso número uno hacia el éxito. Despertarse y pensar en algo positivo puede cambiar el transcurso de todo el día. No eres lo suficientemente viejo como para no iniciar un nuevo camino hacia tus sueños. Levántate cada mañana creyendo que vas a vivir el mejor día de tu vida”. Usando la clave POSITIVO

El resultado de nuestro programa es:

Texto claro:

Creer que es posible es el paso número uno hacia el éxito. Despertarse y pensar en algo positivo puede cambiar el transcurso de todo el día. No eres lo suficientemente viejo como para no iniciar un nuevo camino hacia tus sueños. Levántate cada mañana creyendo que vas a vivir el mejor día de tu vida.

Resultado con módulo 27:

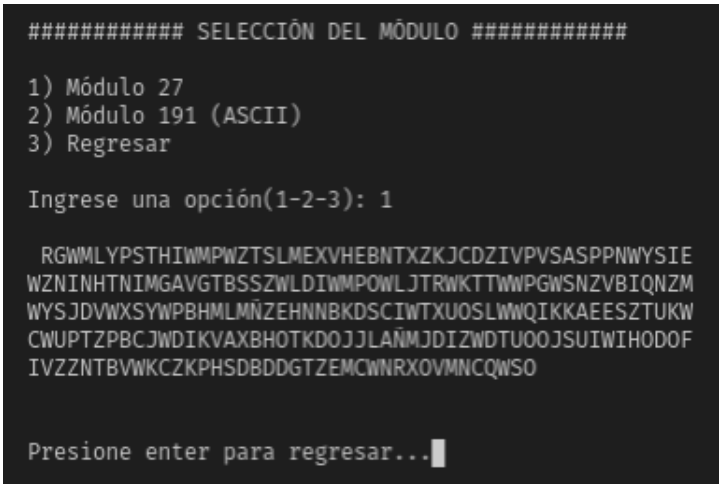

Figura 6: Resultado con módulo 27.

Resultado con módulo 191:

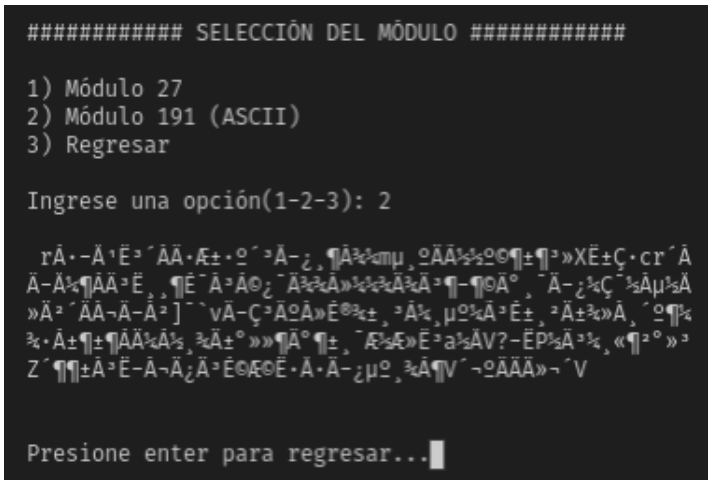

Figura 7: Resultado con módulo 191.

3. Verifica el resultado obtenido a partir del cifrador Criptoclásicos v2.1 (http://www.criptored.

upm.es/software/sw_m001c.htm) haciendo las capturas de pantalla respectivas para los módulos 27 y 191

Resultado con módulo 27

Clave usada: POSITIVO

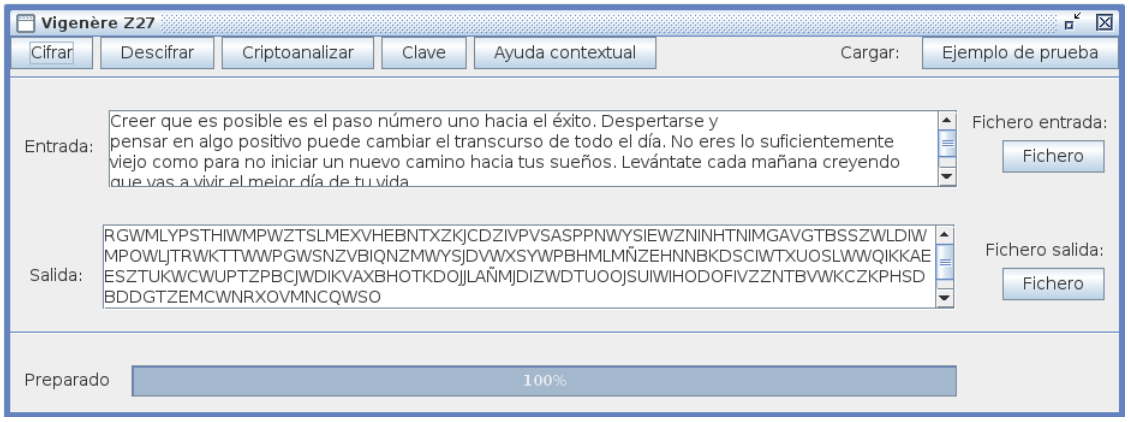

Figura 8: Resultado con módulo 27.

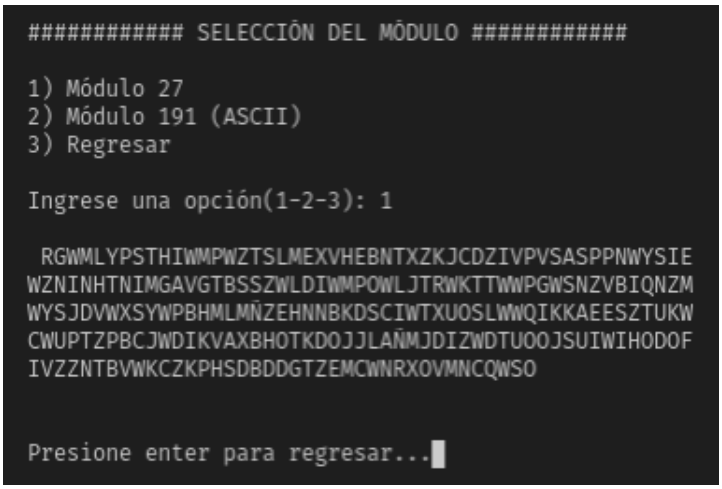

Figura 9: Resultado con módulo 27.

Resultado con módulo 191:

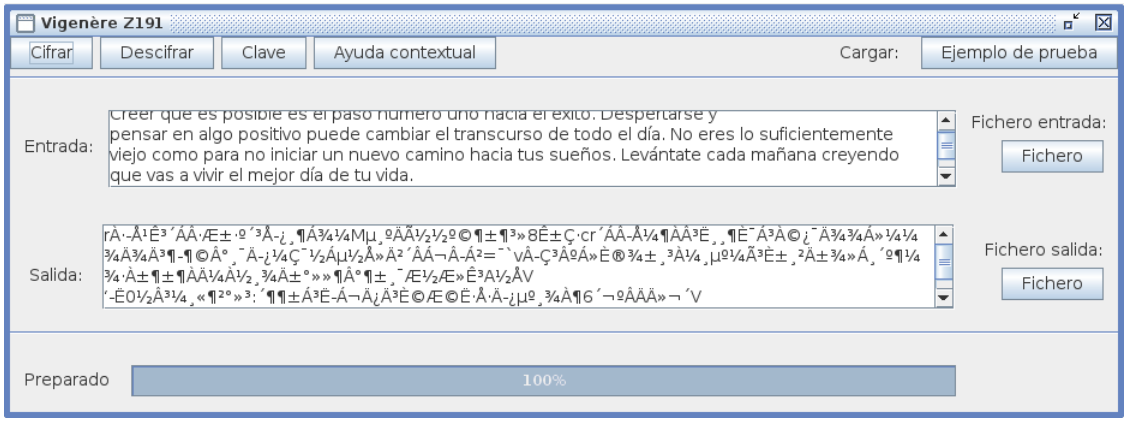

Figura 10: Resultado con módulo 191.

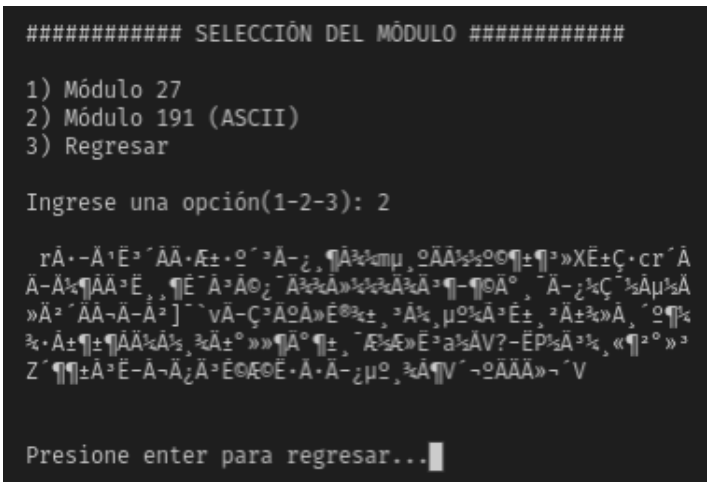

Figura 11: Resultado con módulo 191.

4. Mostrar el resultado de cifrar usando al menos otros dos métodos disponibles (deberá explicar el principio de dicho método de cifrado)

   - **Cifrado de Gronsfeld:** El método de Gronsfeld utiliza más de un alfabeto cifrado para poner en clave el mensaje y que se cambia de uno a otro según se pasa de una letra del texto en claro a otra. Es decir que deben tenerse un conjunto de alfabetos cifrados y una forma de hacer corresponder cada letra del texto original con uno de ellos.
     Por ejemplo, la letra más común en el castellano es la letra E, la cual se cifrará de forma diferente según su posición en el texto original. Al igual que Vignere es de tipo polialfabético, el cual usa más de un alfabeto para poner en clave el mensaje y que cambiar de uno a otro según se pasa de una letra del texto claro a otra.
     Para ello, utiliza una clave numérica, de preferencia corta, tal como vamos a hacer con el texto **GRONSFELD**. Utilizando la CLAVE **734173417** con los siguientes desplazamientos **4, 8, 4, 4, 4, 4, 4, 3, 3, 5**.

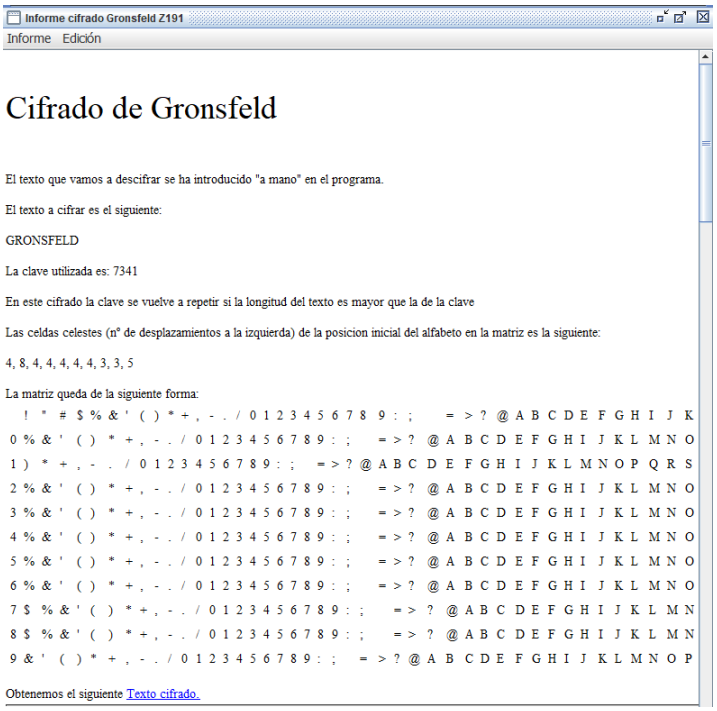

Figura 12: Cifrando el texto con Gronsfeld

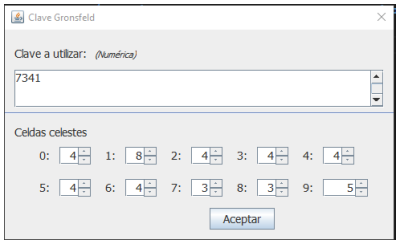

Figura 13: Clave y números de desplazamientos en cada celda

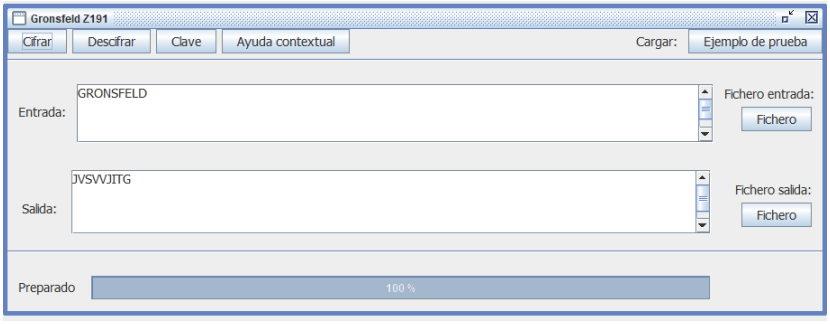

Figura 14: Texto cifrado por Gronsfeld

- **Cifrado de César o Desplazamiento Puro:** El cifrado César es uno de los primeros métodos de cifrado conocidos históricamente. Julio César lo usó para enviar órdenes a

sus generales en los campos de batalla.
Este consiste en desplazar tres posiciones a la derecha el alfabeto latino, quedando de esta manera.

Alfabeto en claro:	A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z
Alfabeto cifrado:	D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

Figura 15: Alfabeto cifrado y sin cifrar.

El receptor del mensaje conocía la clave secreta de éste (es decir, que estaba escrito con un alfabeto desplazado tres posiciones a la derecha), y podía descifrarlo fácilmente haciendo el desplazamiento inverso con cada letra del mensaje. Básicamente está dado por la fórmula:

$$E_{(a,b)}\left(M\right) = \left(aM + b\right) mod\ N \tag{1}$$

Donde **M** es el valor numérico de un caracter del alfabeto, **a** y **b** dos números enteros que son números que el cardinal **N** (longitud o cantidad total del alfabeto original). Por otro lado, **a** es una constante que determina el intervalo de separación entre dos letras del alfabeto cifrado cuando estas son consecutivas en el alfabeto original(posición), Mientras que **b** es la que determina el desplazamiento entre las letras del mensaje claro y las correspondientes en el cifrado.

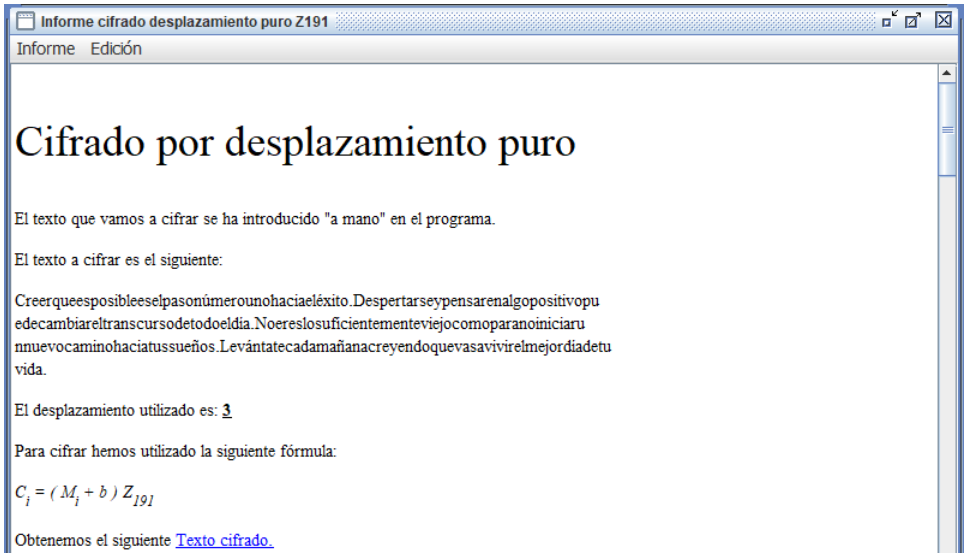

Figura 16: Resultados al aplicar el algoritmo de César

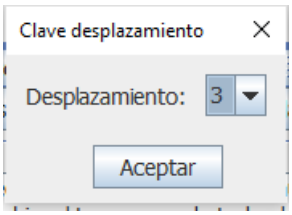

Figura 17: Cantidad de desplazamiento

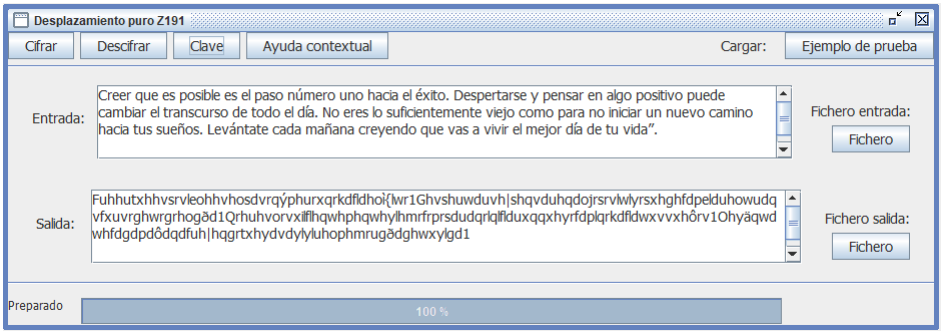

Figura 18: Texto cifrado con un desplazamiento igual a 3

5. Muestre las frecuencias de cada letra del mensaje original usando como claves POSITIVO, HIELO y MAR, compare y concluya sobre la variación de las frecuencias en base a la longitud de la clave. Verifique el resultado usando la aplicación desarrollada en la práctica anterior

   Ojo: Se está utilizando el cifrado con el módulo 27, ya que sólo funciona con letras que van desde la A hasta la Z (mayúsculas)

   - Usando la clave **POSITIVO**

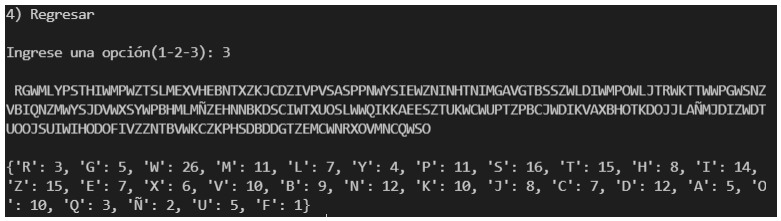

Figura 19: Frecuencias del texto cifrado con la clave POSITIVO

   - Usando la clave **HIELO**

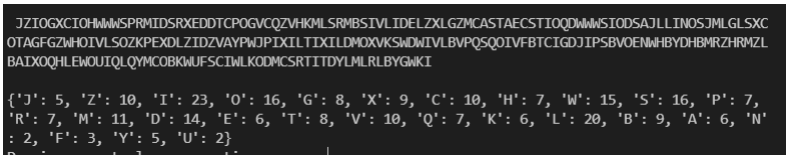

Figura 20: Frecuencias del texto cifrado con la clave HIELO

- Usando la clave **MAR**

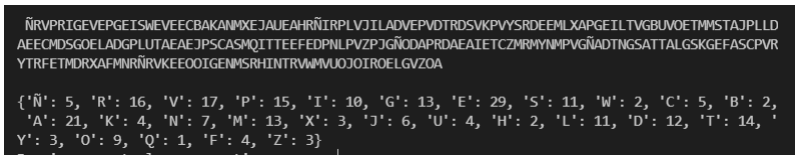

Figura 21: Frecuencias del texto cifrado con la clave MAR

Clave	Longitud	Frecuencias
POSITIVO	8	R=3, G=5,...
HIELO	5	R=7, G=8,...
MAR	3	R=16, G=13,...

Se puede concluir, que entre más grande sea la longitud de la **clave** que recibe el algoritmo, más alta será frecuencia obtenida de entre todas las letras del texto.

6. Desarrolle un algoritmo que encuentre el texto claro si recibió la cifra WPIXHVYYOSR-TECSZBEEGHUUFWRWTZGRWUFSRIWESSXVO
HAIHOHWWHCWH UZOBOZEAOYBMCRLTEYOTI, y se sabe que ha cifrado con la clave HIELO

```python
abc = 'ABCDEFGHIJKLMNÑOPQRSTUVWXYZ'
texto = 'WPIXHVYYOSRTECSZBEEGHUUFWRWTZGRWUFSRIWESSXVOHAIHOHWWHCWHUZOBOZEAOYBMCRLTEYOTI'
clave = 'HIELO'

def descifrar(texto,clave):
    k=0;
    descifrado=''
    for i in texto:
        TamClave = len(clave)
        a = abc.find(i)
        if (k<TamClave):
            b= abc.find(clave[k])
            k=k+1
            if k == TamClave:
               k=0
        resta = (a-b)% 27
        descifrado = descifrado + abc[resta]
    print(descifrado)

descifrar(texto,clave)
```

Figura 22: Algoritmo

Resultado:

PIENSOQUEELMARESTATRANQUILOPORLOQUELASTEMPRESTADESPOD
RIANVENIRAPARTIRDEMAÑANA

PIENSO QUE EL MAR ESTA TRANQUILO POR LO QUE LAS TEMPRESTADES P ODRIAN VENIR APARTIR DE MAÑANA

7. Usando el software anterior, verifique el resultado, eligiendo el cifrado Vignere con módulo 27

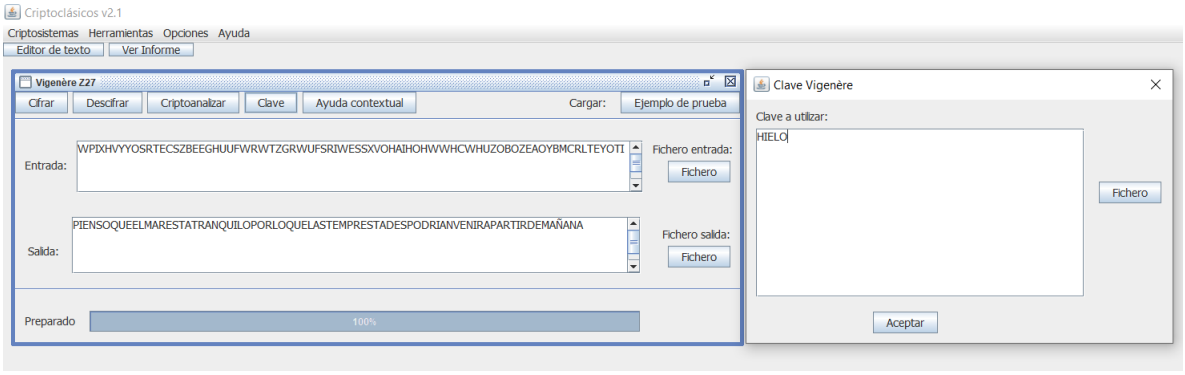

Figura 23: Resultado

8. Usando matemáticas discretas, descifre manualmente YGVMSSKKOX si la clave fue FORTALEZA en un alfabeto de 27 caracteres

YGVMSSKKOX
Clave: FORTALEZA

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z

Y	G	V	M	S	S	K	K	O	X
25	6	22	12	19	19	10	10	15	24
F	O	R	T	A	L	E	Z	A	F
5	15	18	20	0	11	4	26	0	5
25-5=20	6-15=-9	22-18=4	12-20=-8	19-0=19	19-11=8	10-4=6	10-26=-16	15-0=15	24-5=19
mod 27	mod 27	mod 27	mod 27	mod 27	mod 27	mod 27	mod 27	mod 27	mod 27
20	18	4	19	19	8	6	11	15	21
T	R	E	S	S	I	G	L	O	S

Figura 24: Resultado

1.2. Cifrado Autoclave

1. Descifre el texto, usando la clave UNODELOSMASGRANDESCRIPTOGRAFOS: XHGDQESDMPKÑDEEDKNGJZPFJSUIFZOLFCINFJCESVZTGBFXCIUDAYNUUDIZYW WZBEYNVQWIVUNKZEPHDODQUZZLBDNDRWTHQSERÑIVMLERCMGIFLSORZXTSDIGLO XQSDJHWVCIWQXQJCKMBPOKMPSKMUVIMNJDNBLCSZHXHNYYUI XDBSO

XHZLXWVGDJGXHWLTDWKÑSAQIMZLNBVMLXHUOQQXIQGWGUFTWKZK MOKUDNINSIFJDUOZIJBSVVOWFAIEÑGYOWPSOAP

```
==== RESTART: C:/Users/Kemely/Desktop/CSUNSA2021-B/SEguridad/LAB02/Ejer18.py ===
Descifrado
DURANTELAPRIMERAGUERRAMUNDIALWILLIAMFREDERICKFRIEDMANSIRVIOCOMOTENIENTEENLAUNIDA
DDECRIPTOLOGIADELAFUERZAEXPEDICIONARIAESTADOUNIDENSEDISTINGUIENDOSEPORSUSTRABAJO
SYPROEZASENELANALISISDECODIGOSENEMIGOSINVIOLABLESPORLOCUALRECIBIOELRECONOCIMIENT
ODELGOBIERNOESTADOUNIDENSE
>>>
```

Figura 25: Resultado

1.3. Ataque de Kasiski

1. Criptoanalizar el siguiente criptograma mod 27, encontrar la clave y el texto en claro.

MAXYHGAVAPUUGZHEGZQOWOBNIPQKRNÑMEXIGONIICUCAWIGCTEAGMNOL
RSZJNLWÑAWWIGLDDZSNIZDNBIXGZLAYMXÑCVEKIETMOEOPBEWPTNIXCXUI
HMECXLNOCECYXEQPBWUFANIICÑJIKISCZUAILBGSOANKBFWUAYWNSCHLCW
YDZHDZAQVMPTVGFGPVAJWFVPUOYMXCWERVLQCZWECIFVITUZSNCZUAIKBF
MÑALIEGLBSZLQUXÑOHWOCGHNYWÑQKDANZUDIFOIMXNPHNUWQOKLMVBN
NKRMKONDPDPNMIKAWOXMEEIVEKGBGSFHVADWPGOYMHOIUEEIPGOLENZBS
CHAGKQTZDRÑMÑNWTUZIÑCMÑAXKQUWDLVANNIHLÑCQNWGEHIPGZDTZTÑN
WÑEEWFUMGIÑXNTWXNVIXCZOAZSOQUVENDNFWUSZYHGLRACPGGUGIYWH
OTRMZUGQQDDZIZFWHVVSHCUGOGIFKBXAXPBOBRDVDUCMVTKGIKDRSZLUQ
SDVPMXVIVEYMFGTEANIMQLHLGPQOHRYWCFEWFOISNÑPUAYINNÑXNÑPGKW
GOILQGAFOILQTAHEIIDWMÑEÑXNEPRCVDQTURSK

```python
def decifrar(texto, clave):
    i = 0
    alfabeto = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
    texto_plano = ""
    for c in texto:
        caracterText_index = (alfabeto.index(c) - alfabeto.index(clave[i])) % 27
        texto_plano += alfabeto[caracterText_index]
        i += 1
        if i > len(clave) - 1: i = 0
    return texto_plano
```

Resultado

Texto Descifrado:

JACQUESSAUNIEREELRENOMBRADOCONSERVADORAVANZABATAMBALEAN
DOSEBAJOLABOVEDADELAGRANGALERIADELMUSEOARREMETIOCONTRAL
APRIMERAPINTURAQUEVIOUNCARAVAGGIOAGARRANDOELMARCODORADO
AQUELHOMBREDESETENTAYSEISAÑOSTIRODELAOBRADEARTEHASTAQUE
LAARRANCODELAPAREDYSEDESPLOMOCAYENDOBOCAARRIBACONELLIEN
ZOENCIMATALCOMOHABIAPREVISTOCERCASEOYOELCHASQUIDODEUNAR
EJADEHIERROQUEALCERRARSEBLOQUEABAELACCESOALASALAELSUELO
DEMADERATEMBLOLEJOSSEDISPAROUNAALARMAELCONSERVADORSEQUE
DOAHITENDIDOUNMOMENTOJADEANDOEVALUANDOLASITUACIONTODAVI
AESTOYVIVOSEDIOLAVUELTASEDESEMBARAZODELLIENZOYBUSCOCONL
AMIRADAALGUNSITIODONDEESCONDERSEENAQUELESPACIOCAVERNOSO

La clave de cifrado es: DAVINCI

2. Conclusiones

- Antes de realizar el cifrado a cualquier texto plano, siempre debe pasar por un preprocesamiento para un mejor resultado en el cifrado.
- El método Vignere no es tan eficiente, ya que en el proceso de cifrado su clave se repite periódicamente de acuerdo a la longitud del texto preprocesado, convirtiéndose así en una gran debilidad.
- El cifrado Autoclave es mejor que el cifrado Vignere, nos ayuda a ser mas indescifrable que el cifrado de Vignere que es repetitivo.
- Generalmente, los métodos polialfabéticos hacen uso de las matemáticas discretas, donde se aplica la aritmética modular para el cifrado de cada carácter.
- Las letras que se repite mayormente en un determinado lenguaje(inglés o español, como A E O para el español) son de gran importancia para el descifrado de un texto plano del mismo lenguaje de las letras con mayor frecuencia.
- Cuando queremos hallar la longitud de la clave, al realizarse el calculo de las distancias entre trigramas repetidos se deben de evitar los números primos, porque pueden causar que el máximo común divisor sea 1.
- Al cifrar con Vignere con un módulo de 191, se debe tener en cuenta que no todos los caracteres ASCII son válidos, por lo que no deben tomarse en cuenta a la hora de cifrar.
- Para el cifrado Vignere con módulo 191, el texto claro debe de quitarse solo espacios en blanco y saltos de línea, ya que en el alfabeto ASCII hay todas las letras con tilde y sin tilde, también están los signos de puntuación.

3. Cuestionario Final

1. **Trabajando en módulo 191 (un subconjunto imprimible del código ASCII del software Criptoclásicos), cifra el siguiente texto en claro con la clave: El ingenioso hidalgo.**
   En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda. El resto della concluían sayo de velarte, calzas de velludo para las fiestas con sus pantuflos de lo mismo, los días de entre semana se honraba con su vellori de lo más fino. Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que no llegaba a los veinte, y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la podadera.

   Cifrado usando ASCII191, ver figura 26.

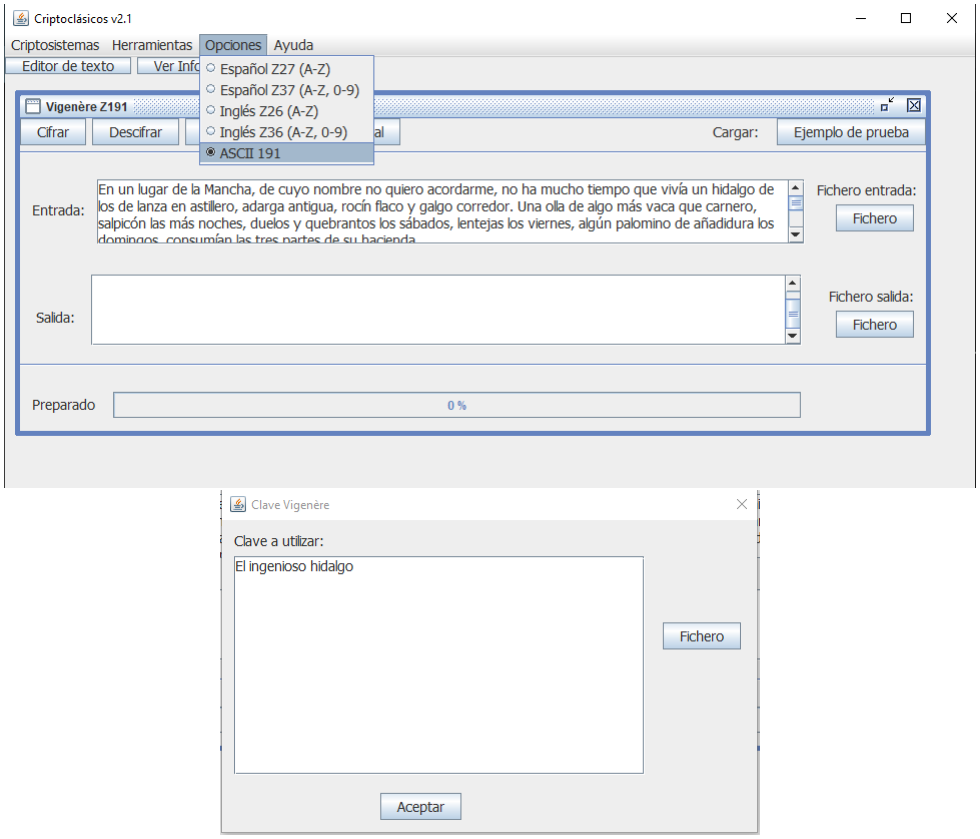

Figura 26: Ingresando el texto y la clave

Texto cifrado:

¡ÙÝÛÒÙÔÉàÖÓÓÉ°ÁÙÉÖ¥wÌÒÉÙæ×ÜáÛÉÚÈÎÚ×ãÐÚÜÇÇÜÚÒÓàÔÍoÎÚÎÏ±à
ËÕÕØÖÍÛàÝØÝÈÖÔÜ\¥àÖÕÏÈÎÔÕáÒÌÔÒÓÓÏËÚ¥ÙàÎËÒÎÛâÛÚÓÍÕÏwÇÒ¥ÝÏÎ

ÇÒáÑÕçÏsÚÒÃYÔÔ°ÌËÜßËÎÔÕáÑÖÚÕÅÏÕàRÀÖÎÕÐÙÉÒ×ÏÓÏÒÍMÙä¥ÎÉÞÛÉ ÐÉààÓÙ×oÓÌÒÞÎ\ÛÒÅàÕPåÜÖËËÅÞrÒ¹ÐÔÜÙÝÞÝÓÔàÈÖ×ÏÞÒÝ · ÞJÏÇÈÜÛ zÞÓÕÜÈÊÌÙÚ³ÞÞÖËÖÛÍá~ÏÓÏ^ÎÛÇÚ³ØÑÛÕÈÒÉ‘ÓÒÐÌØÒÌÒÝ · Ï×ÚÏÒÔ×á~ÑÖ ÖÖÕØTÏ²×ÉàÚÖÒÛÞÓàÛÍÖÄÐÙã¬ÌËÖËÒÑÉ| · ÚÙÍÖÔÚÊÓ°×ÉÐÕÒÐÔã‘ÏÕ ÛÄÙÚÊÓºÐÔÎØØÒtÑÓÚáÉÖÄÐÜÓ°×ÝÑÕÔÎÚÏÞÏÚÎÌÅÞÚÏ · Î×ÛÙÙàØÏàâÜÎÏ ÏÞÊÓ°ÚÕÖÙÑÜtÚááËVÄÓÏËÓ²ßÚÒÙÉÚÉÜÓáÌÐÒÎÝÇÐ¥Î×ÛÙÙãÍÚÞÝÙÑÇÅ× ÕÛ&ÞÎÖÔÓ{¼Óà\ÈÍÑÓàÉÏ · ÌÝÛÇÅÚÉßçÓ×ÉÖÁÍÇÒ©××àÉÙÎÚÓàâÈtÜÕÙÇá ³ÍÚÖÔÅÞÝÓàÝÓÔÈÇÌÈÏ¥××àÜÉÖÖâ×zàÝÑÍÚàÝ¨ÐËÎÓÔÜáÞÞÏáÉoÑàËÏ · Y ÍÛÙÍÙÔÏÔÏÎÔÕÏÎTÜ§ÚÕÜÚÓÚÉÐÓÚÈØÒÄÌÊÓ¶Ìv

2. **Descifre el criptograma en el mismo software ¿Por qué crees que el software no permite hacer un criptoanálisis?**

   En la figura 28 se observa que si se pudo realizar el descifrado con con la clave *El ingenioso hidalgo*.

Figura 27: Descifrado exitoso en el software

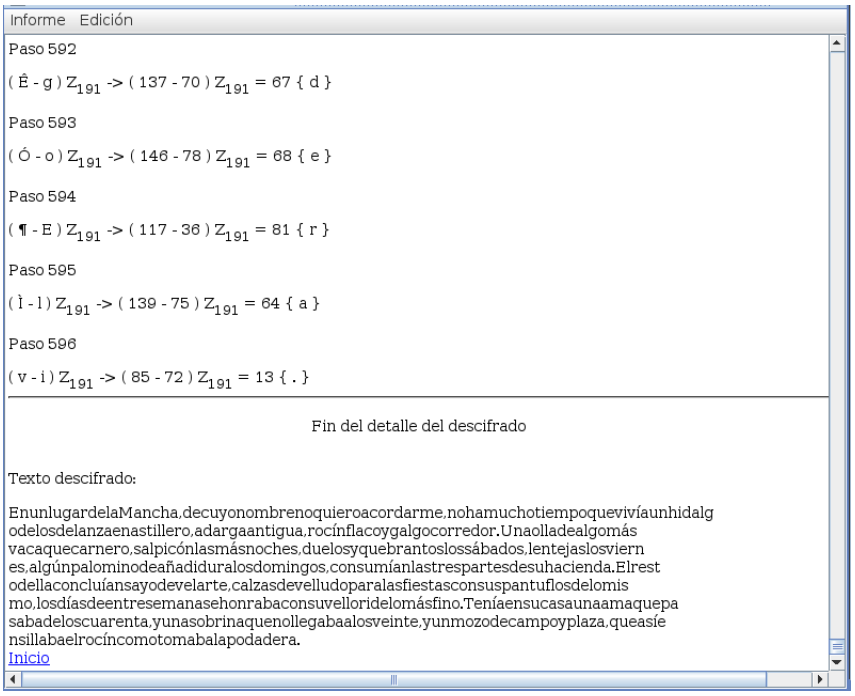

Figura 28: Informe del descifrado

3. **Si el cifrado de Vigenere es IZLQOD y la clave SOL, ¿cuál era el mensaje en claro?**

   En la figura 29 se observa que el mensaje claro es **PLAYAS**.

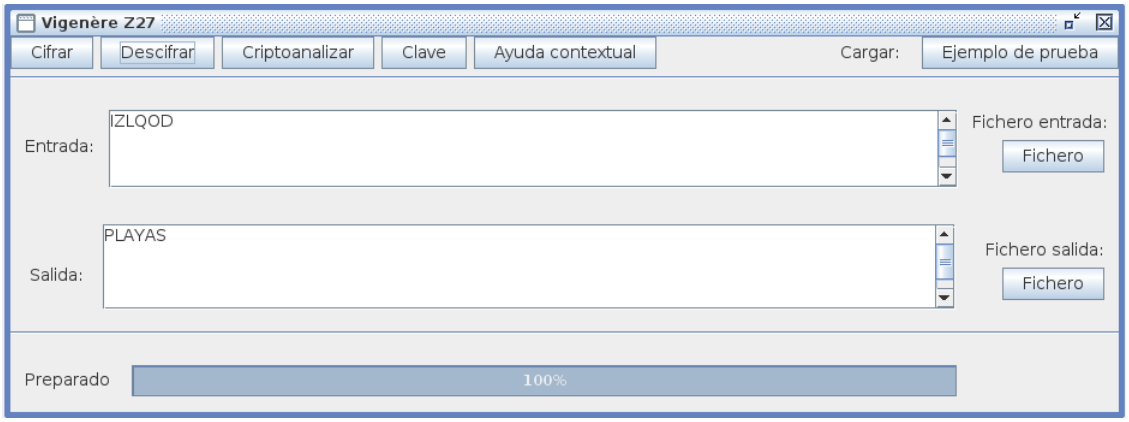

Figura 29: Descifrando el mensaje claro

4. **¿Cuál será la cifra con autoclave del texto HABIA UNA VEZ, con la clave CIRCO?**

   La cifra autoclave es **JISKOBNBDET**.

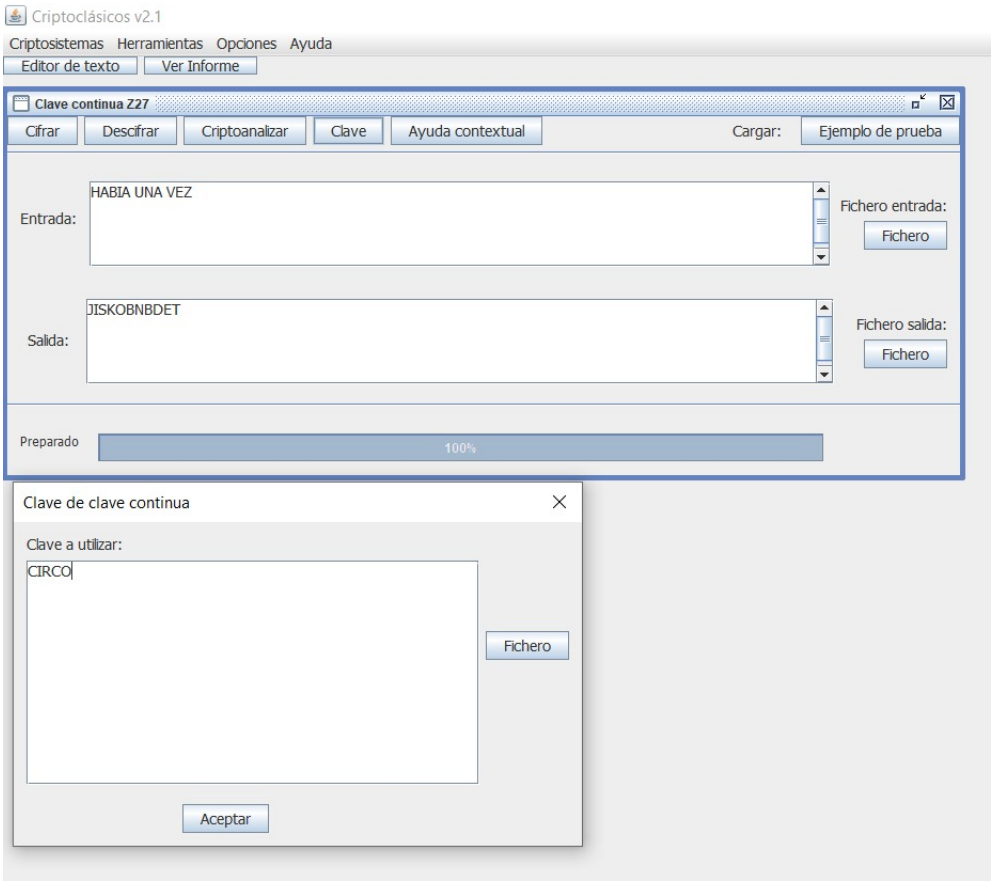

Figura 30: Cifrado con autoclave

5. **En el ataque a Vigenere por Kasiski ¿Qué buscamos preferentemente?**

   En el ataque a Vigenere por Kasiski se busca la distribución 0, +4, +11; en cada subcriptograma obtenido. Esta distribución viene de las letras A E O S, las cuales tienen mayor frecuencia en el lenguaje español [1]. Al encontrar las distribuciones en cada subcriptograma podemos hallar la clave, con la cual podemos realizar nuestro ataque.

6. **Encontradas las cadenas repetidas en el criptograma, con separación d1, d2, d3 y d4 ¿Cuál sería la longitud L de la clave?**

   El máximo común divisor de las distancias entre cadenas repetidas es la longitud de la clave más probable[2]. Entonces, para hallar la longitud de la clave, se calcula el máximo común divisor de las distancias. Es decir, **L = m.c.d(d1, d2, d3, d4)**.

7. **Si las distancias entre repeticiones de cadenas en un criptograma son 35, 112, 70. ¿Cuál sería la longitud L de la clave?**

   Según, lo mencionado anteriormente, la longitud L la calculamos del máximo común divisor. Siendo el resultado **7**, ya que m.c.d(35, 11, 70) = 7.

8. **¿Qué diferencia la regla AEOS de AEO en Kasiski?**

   La diferencia entre la AEOS y AEO, es la precisión con la que encontramos la clave.

Referencias

[1] Universidad Politécnica de Madrid.(2017).*Introducción a la seguridad informática y criptografía clásica*. http://www.criptored.upm.es/crypt4you/temas/criptografiaclasica/leccion9.html

[2] Friedrich W. Kasiski.(1863).*Kasiski's Method*. https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-Kasiski.html