



**Instituto Politécnico Nacional**  
***Escuela Superior de Cómputo***



## *Práctica 2*

# **“Implementación del cifrado Affin”**

**3CM14**

**Corte Aguirre Isaí Misael**  
**Hernández Velázquez Emmanuel Alejandro**  
**Vargas Hernandez Carlo Ariel**

<b>Introducción.....</b>	<b>3</b>
<b>Desarrollo.....</b>	<b>4</b>
Función para cifrar.....	4
Función para descifrar.....	5
<b>Pruebas.....</b>	<b>5</b>
Carlo Ariel.....	11
Misael Corte.....	11
Emmanuel Alejandro.....	11
<b>Código.....</b>	<b>12</b>
<b>Referencias.....</b>	<b>14</b>

## Introducción

El cifrado afín también se le llama cifrado de transformación afín o cifrado monoalfabético genérico. Es un tipo de cifrado por sustitución en el que cada símbolo del alfabeto en claro (el alfabeto del texto en claro) es sustituido por un símbolo del alfabeto cifrado (el alfabeto del texto cifrado) siendo el número de símbolos del alfabeto en claro igual que el número de símbolos del alfabeto cifrado. Para hallar el símbolo del alfabeto cifrado que sustituye a un determinado símbolo del alfabeto en claro, se usa una función matemática afín en aritmética modular. Para poder aplicar la función matemática lo primero que hay que hacer es asignar un orden que a cada símbolo de cada uno de los alfabeto le asocie un número de orden [1].

De acuerdo a lo visto en clase, la fórmula para cifrar es la siguiente:

$$c_i = (\alpha * p_i) + \beta \bmod n$$

Donde:

$c_i$  : Es el símbolo  $i$  del texto cifrado

$\alpha$  : Es la constante de decimación

$m_i$  : Identifica el símbolo  $i$  del texto sin cifrar

$\beta$  : Es la constante de desplazamiento

$n$  : Es el número de símbolos del alfabeto (para inglés 26)

La fórmula para descifrar es la siguiente:

$$p_i = ((-\alpha) * c_i) + (-\beta) \bmod n$$

Donde:

–  $\alpha$  : Es el inverso multiplicativo de  $\alpha$

–  $\beta$  : Es el inverso aditivo de  $\beta$

De forma que:

$$\alpha * (-\alpha) \bmod n = 1 \text{ y } \beta + (-\beta) \bmod n = 0$$

## Desarrollo

Para crear el programa debemos tener en cuenta que se usará el valor ASCII para cada letra del mensaje, a diferencia con como el cifrador le asigna valor a las letras del alfabeto inglés que es 0 para 'a', 1 para 'b' hasta 25 para 'z'. La entrada será a través de un archivo de texto (.txt) y la salida será a través de un archivo de texto cuyo nombre debe ser el mismo que el del mensaje original más "\_c", asimismo para el archivo descifrado deberá ser el mismo nombre del archivo de texto de entrada más "\_d". Finalmente el programa debe tener interfaz.

Para la implementación del cifrado utilizamos el lenguaje de programación Python ya que es conocido por los miembros del equipo. A grandes rasgos hay funciones para cargar archivos de texto para lectura (*load\_file()*), para crear y escribir en archivos de texto (*save\_file(content, file\_path)*), funciones de utilidad para el MCD y para el inverso del módulo (*gcd(a, b)* y *mod\_inverse(a, m)*), entre otras.

Las funciones que más destacan ya que es donde está implementado lo visto en clase son las siguientes:

### Función para cifrar

```
def affine_encrypt(text, a, b):
    result = ""
    m = 26 # Tamaño del alfabeto inglés
    for char in text:
        if char.isalpha():
            if char.islower():
                result += chr(((a * (ord(char) - ord('a'))) + b) % m) + ord('A'))
            elif char.isupper():
                result += chr(((a * (ord(char) - ord('A'))) + b) % m) + ord('A'))
            else:
                result += char
    return result
```

## Función para descifrar

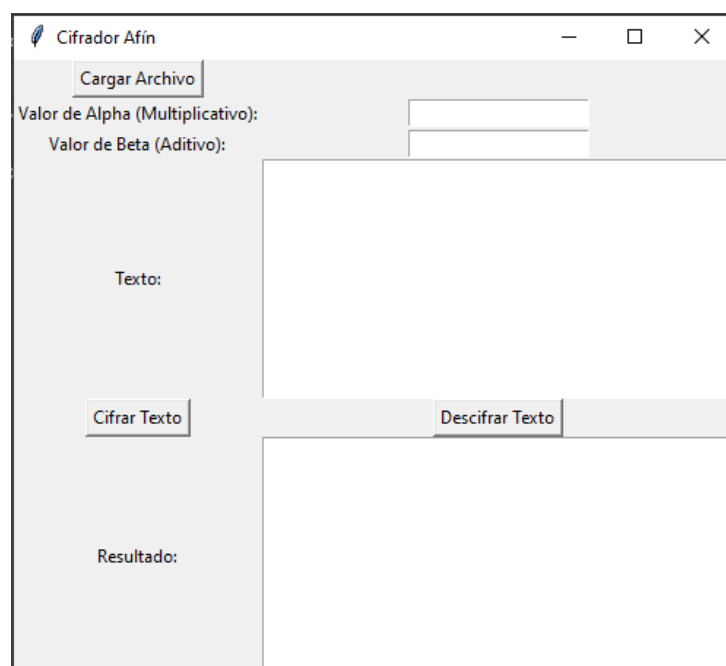
```
def affine_decrypt(ciphertext, a, b):
    result = ""
    m = 26 # Tamaño del alfabeto inglés
    a_inverse = mod_inverse(a, m)

    for char in ciphertext:
        if char.isalpha():
            if char.islower():
                result += chr(((a_inverse * (ord(char) - ord('a') - b)) % m) +
ord('a'))
            elif char.isupper():
                result += chr(((a_inverse * (ord(char) - ord('A') - b)) % m) +
ord('A'))
            else:
                result += char

    return result
```

## Pruebas

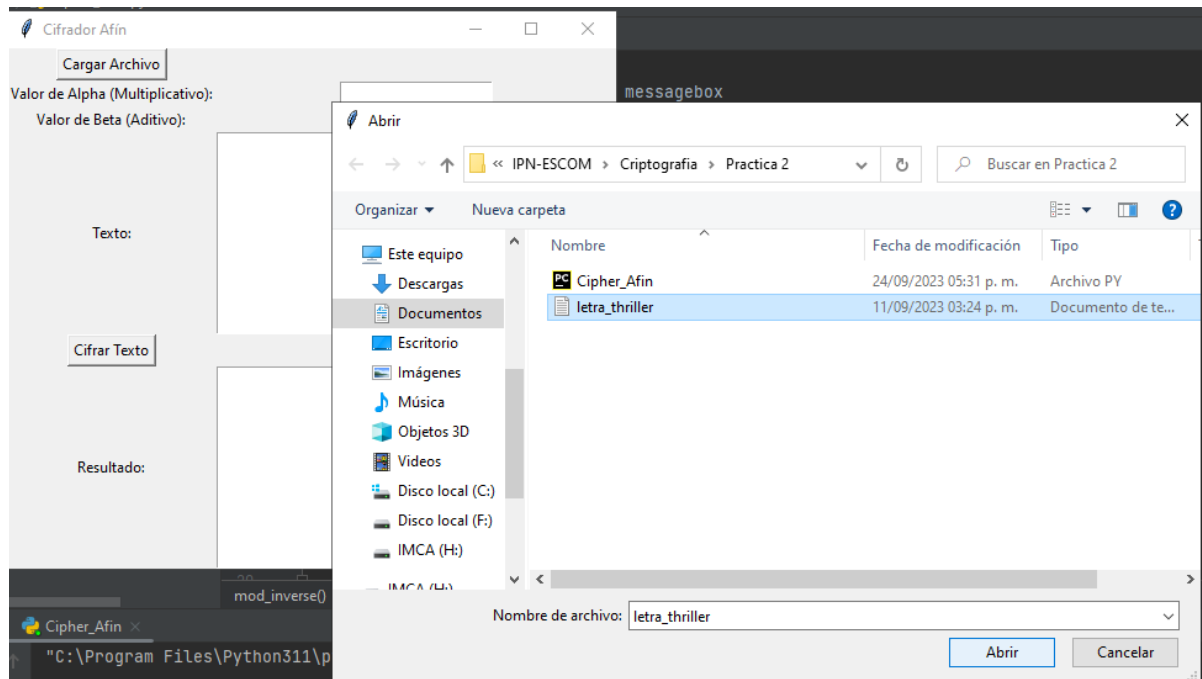
Al correr el programa aparecerá una ventana como esta, en la que podemos ingresar los parámetros Alfa y Beta, podemos cargar archivos de texto, y seleccionar las opciones de cifrar y descifrar texto.



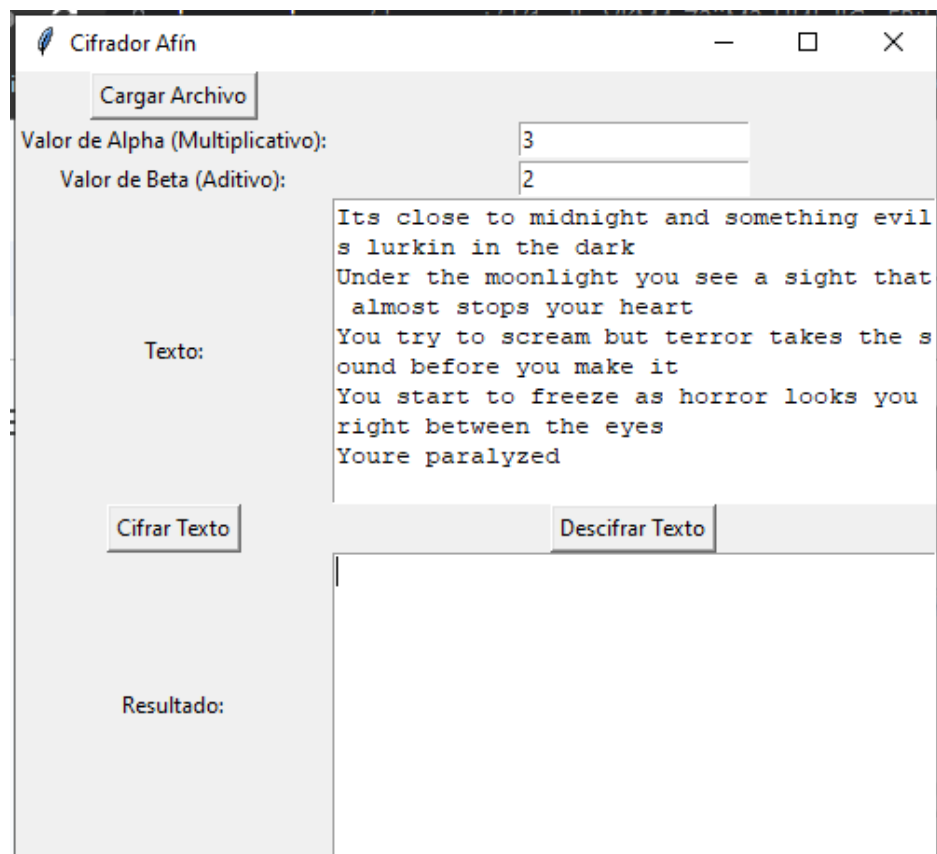
The screenshot shows a window titled "Cifrador Afin" with standard Windows window controls (minimize, maximize, close). The interface is divided into several sections:

- Top Section:** Contains a button labeled "Cargar Archivo".
- Parameters Section:** Below the button, there are two labels: "Valor de Alpha (Multiplicativo):" and "Valor de Beta (Aditivo):", each followed by a text input field.
- Text Input Section:** A large text area labeled "Texto:" for entering the message to be encrypted or decrypted.
- Action Buttons:** Two buttons, "Cifrar Texto" and "Descifrar Texto", are positioned below the text input area.
- Result Section:** A large text area labeled "Resultado:" for displaying the output of the encryption or decryption process.

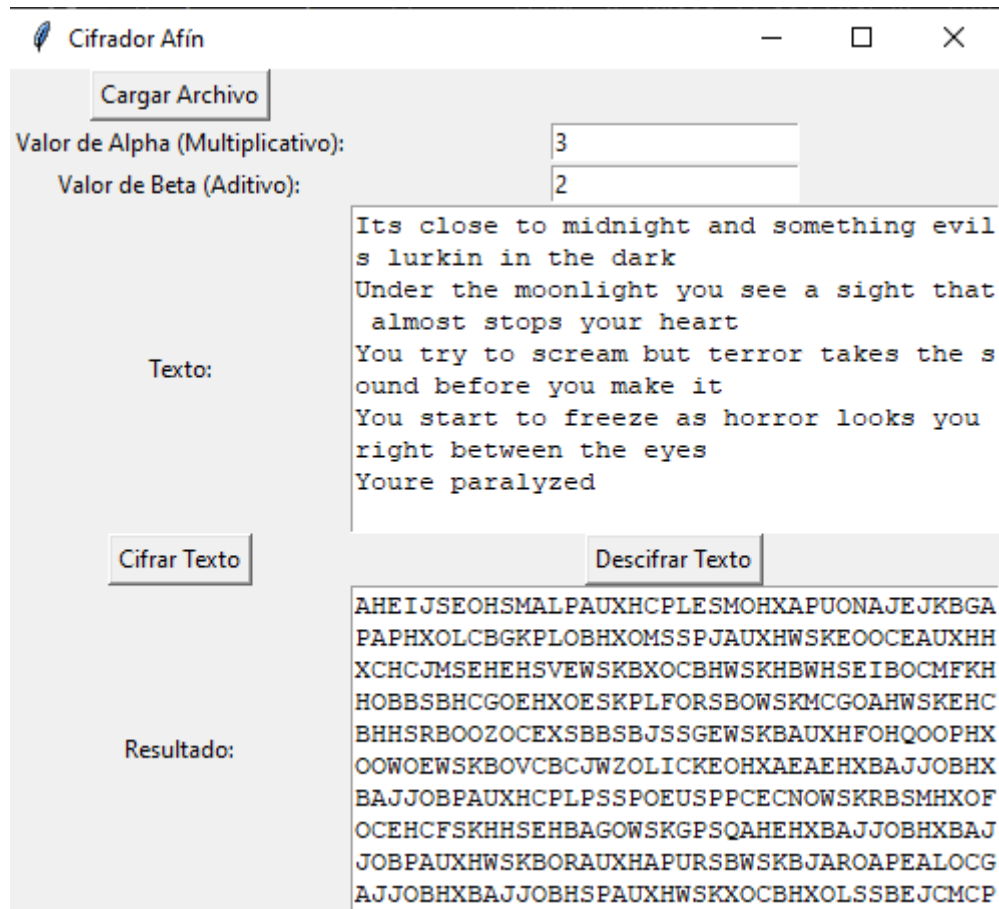
Al darle la opción de “Cargar Archivo” podemos seleccionar un archivo de texto para cifrarlo.



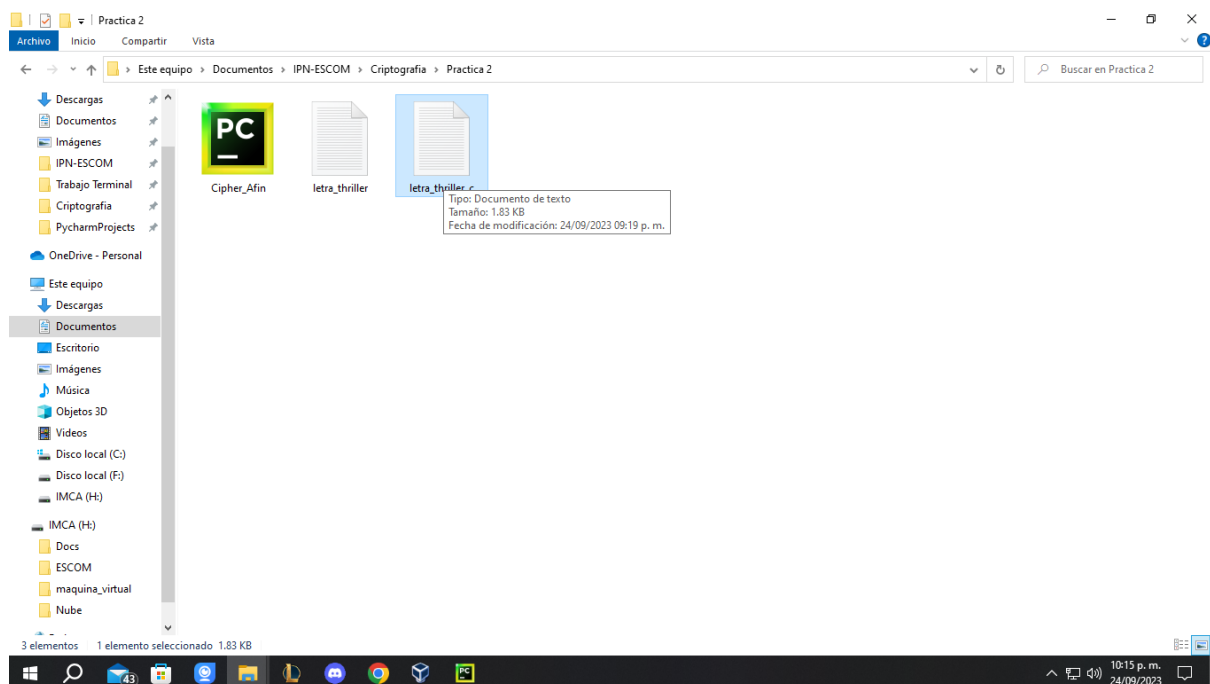
Metemos los parámetros multiplicativo y aditivo (alfa y beta) y aparecerá el texto del archivo en la ventana para poder cifrarlo.



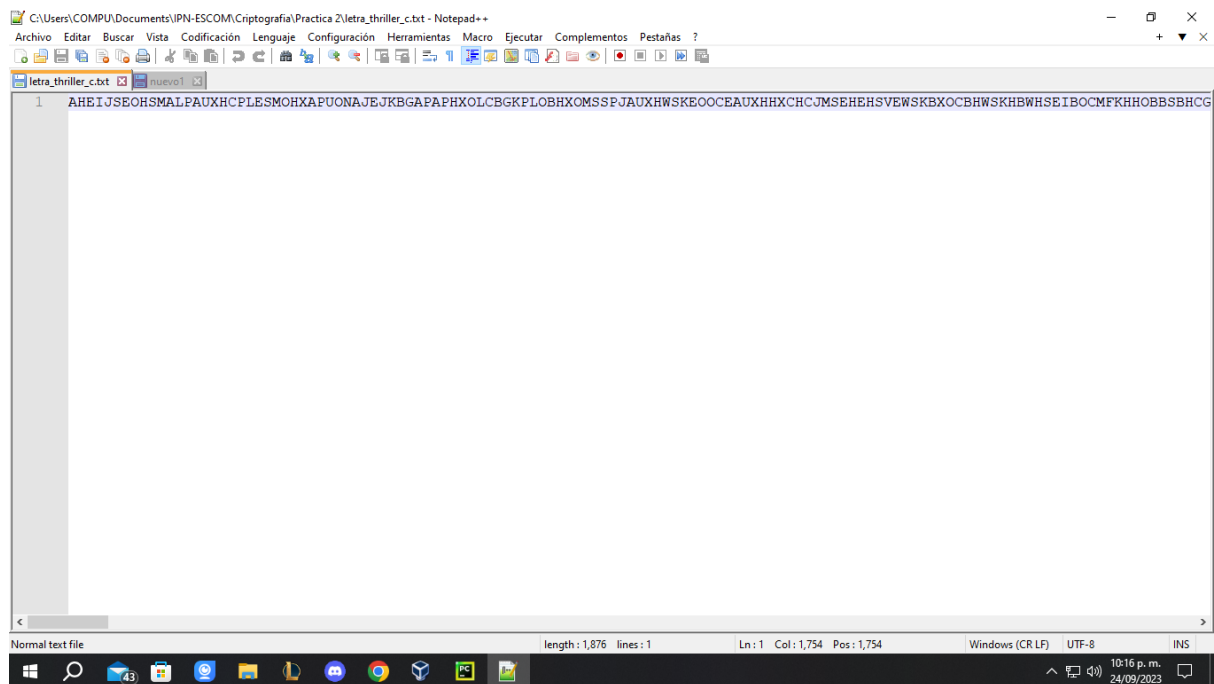
Nos mostrará el texto cifrado y este se guardará en un archivo de texto, se eliminaron los espacios y saltos de línea y se pusieron en mayúsculas.



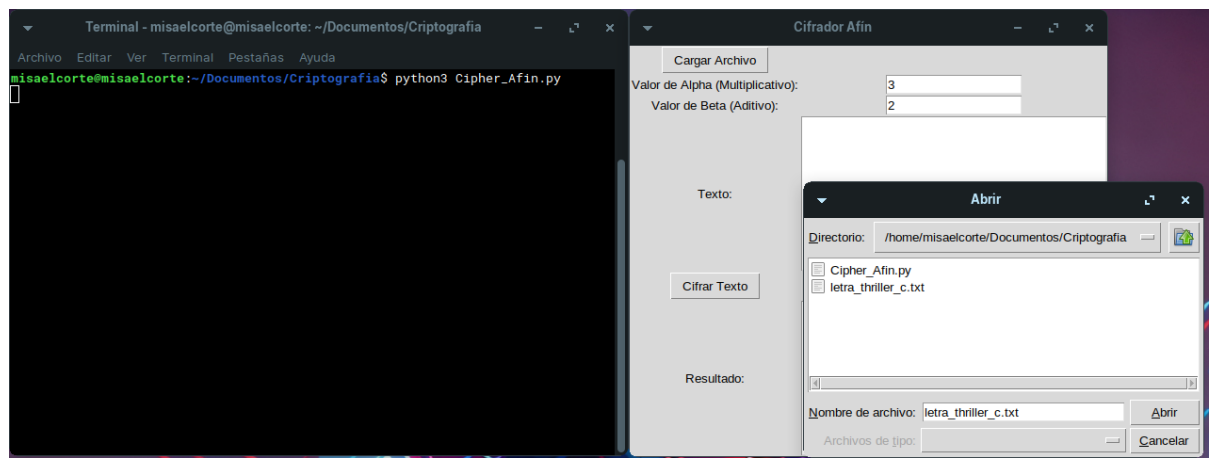
Se puede ver que se creó el archivo de texto con el texto cifrado.



Al abrirlo se puede ver todo el texto cifrado del archivo.



El archivo de texto cifrado se mandó a otra computadora donde se ejecutó el programa con los mismos parámetros Alfa y Beta, y se cargó el archivo cifrado.





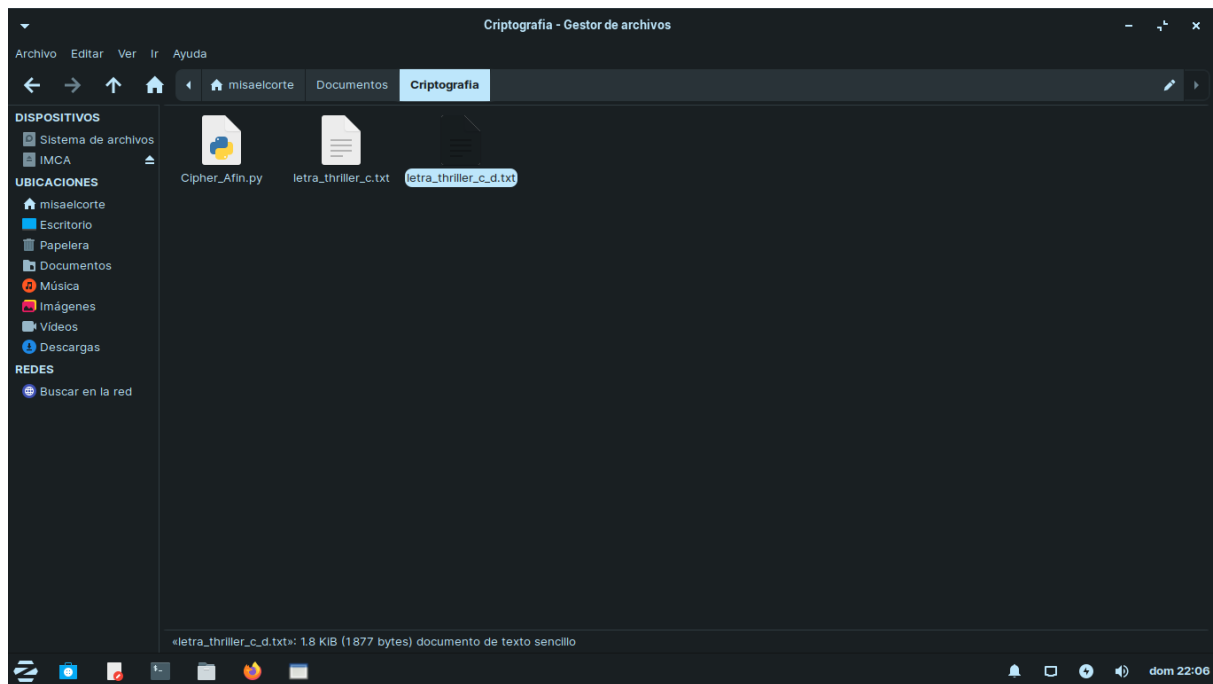
Se muestra en la ventana el texto cifrado en mayúsculas y sin espacios.

The screenshot shows the 'Cifrador Afin' application window. At the top, there is a title bar with the text 'Cifrador Afin' and standard window controls. Below the title bar, there is a 'Cargar Archivo' button. Underneath, there are two input fields: 'Valor de Alpha (Multiplicativo):' with the value '3' and 'Valor de Beta (Aditivo):' with the value '2'. To the left of the text area is the label 'Texto:'. The text area itself contains a block of encrypted text in uppercase letters without spaces. Below the text area are two buttons: 'Cifrar Texto' and 'Descifrar Texto'. At the bottom, there is a label 'Resultado:' followed by a large empty text box for the output.

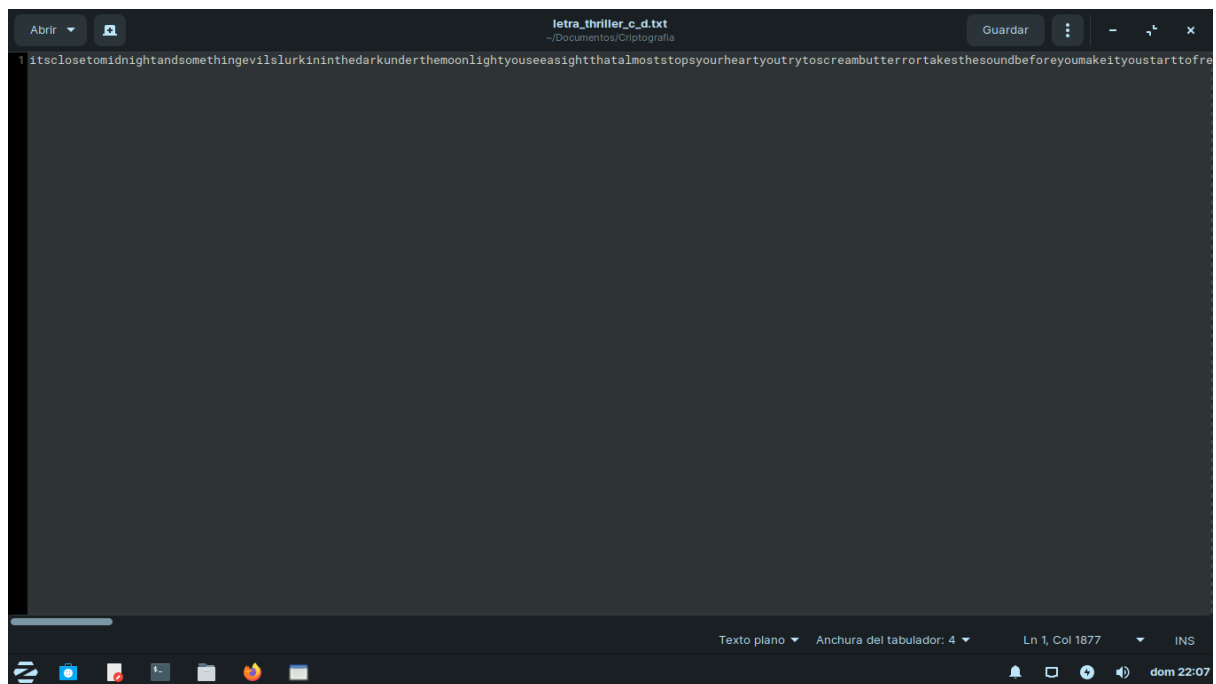
Al darle en la opción "Descifrar Texto" se mostrará el texto descifrado en minúsculas y sin espacios, y este se guardará dentro de un archivo de texto.

This screenshot shows the same 'Cifrador Afin' application window, but with the 'Descifrar Texto' button clicked. The 'Resultado:' text box at the bottom now displays the decrypted text in lowercase letters without spaces. The rest of the interface, including the title bar, input fields for Alpha (3) and Beta (2), and the encrypted text in the 'Texto:' area, remains the same as in the previous screenshot.

Como se puede ver se creó el archivo de texto que contiene el texto descifrado.



Al abrir el archivo, se puede observar el texto descifrado, en minúsculas y sin espacios.



## **Conclusiones**

### **Carlo Ariel**

En la realización de esta práctica, aplicamos el conocimiento adquirido en las clases de criptografía para implementar el cifrador afín en Python. Hemos utilizado el valor ASCII para cada letra del mensaje, lo que diferencia a nuestro cifrador de otros que asignan valores del 0 al 25 para las letras del alfabeto inglés. Esto nos permitió trabajar con un rango más amplio de caracteres, lo cual es un punto de mejora con respecto a otras implementaciones.

Además, tuvimos la oportunidad de practicar la lectura y escritura de archivos en Python, ya que la entrada y salida de nuestro programa se realiza a través de archivos de texto. De esta manera, nos familiarizamos con funciones para cargar archivos de texto para lectura y para crear y escribir en archivos de texto.

Finalmente, nuestro programa también incluye funciones de utilidad para el Máximo Común Divisor y para el inverso del módulo, lo que nos ha permitido refinar nuestras habilidades en la manipulación de números en Python.

### **Misael Corte**

En esta práctica aplicamos el Cifrador Afín haciendo manejo de archivos, fue una práctica interesante de implementar ya que se deben tomar cosas en cuenta, como que los parámetros alfa y beta deben ser coprimos, porque sino a la hora de descifrar los mensajes estos no se pueden recuperar correctamente y el mensaje se perdería.

Se ha mejorado un poco la forma de programar, ya que se ha hecho el manejo de errores para que el programa no termine en caso de un error, por ejemplo si los parámetros que se ingresaron no son correctos o no son coprimos.

### **Emmanuel Alejandro**

Para esta práctica, mejoramos la interfaz respecto a la práctica anterior, incrementando el espacio para observar el texto cifrado y descifrado, y modificando los datos que el usuario debe introducir, siendo estos los valores del factor aditivo y multiplicativo del cifrado afín.

Además, se realizó una gran mejora respecto a la modularidad de la base de código, lo que no solo lo hace más sencillo de entender, pues cada función ahora realiza “una sola acción”, sino que también nos permitirá añadir nuevos métodos de cifrado/descifrado, funciones nuevas, o mejorar la captura de errores, sentando con ello la base de código para las siguientes prácticas del semestre..

## Código

```
1 import tkinter as tk
2 from tkinter import filedialog, messagebox
3 import os
4
5 def gcd(a, b):
6     while b:
7         a, b = b, a % b
8     return a
9
10 def mod_inverse(a, m):
11     for x in range(1, m):
12         if (a * x) % m == 1:
13             return x
14     return None
15
16 def affine_encrypt(text, a, b):
17     result = ""
18     m = 26 # Tamaño del alfabeto inglés
19
20     for char in text:
21         if char.isalpha():
22             if char.islower():
23                 result += chr((a * (ord(char) - ord('a')) + b) % m + ord('a'))
24             elif char.isupper():
25                 result += chr((a * (ord(char) - ord('A')) + b) % m + ord('A'))
26             else:
27                 result += char
28
29     return result
30
31 def affine_decrypt(ciphertext, a, b):
32     result = ""
33     m = 26 # Tamaño del alfabeto inglés
34     a_inverse = mod_inverse(a, m)
35
36     for char in ciphertext:
37         if char.isalpha():
38             if char.islower():
39                 result += chr((a_inverse * (ord(char) - ord('a') - b)) % m + ord('a'))
40             elif char.isupper():
41                 result += chr((a_inverse * (ord(char) - ord('A') - b)) % m + ord('A'))
42             else:
43                 result += char
44
45     return result
46
47 def load_file():
48     file_path = filedialog.askopenfilename()
49     with open(file_path, 'r') as file:
50         text.delete('1.0', tk.END)
51         text.insert(tk.END, file.read())
52     global original_file_path
53     original_file_path = file_path
54
55 def save_file(content, file_path):
56     with open(file_path, 'w') as file:
57         file.write(content)
58
59
60
```

```

61
62 def get_output_path(file_path, suffix):
63     file_dir, file_name = os.path.split(file_path)
64     base_name, ext = os.path.splitext(file_name)
65     output_file_name = f"{base_name}{suffix}{ext}"
66     output_file_path = os.path.join(file_dir, output_file_name)
67     return output_file_path
68
69
70 def encrypt_text():
71     a = int(a_entry.get())
72     b = int(b_entry.get())
73     message = text.get('1.0', tk.END)
74
75     # Eliminar espacios y saltos de línea
76     message = ''.join(message.split())
77
78     try:
79         encrypted_message = affine_encrypt(message, a, b)
80         result.delete('1.0', tk.END)
81         result.insert(tk.END, encrypted_message)
82
83         # Guardar el texto cifrado en un archivo en la misma carpeta que el archivo original
84         output_path = get_output_path(original_file_path, "_c")
85         save_file(encrypted_message, output_path)
86     except Exception as e:
87         messagebox.showerror("Error", f"Error durante el cifrado: {str(e)}")
88
89
90 def decrypt_text():
91     a = int(a_entry.get())
92     b = int(b_entry.get())
93     ciphertext = text.get('1.0', tk.END)
94
95     try:
96         decrypted_message = affine_decrypt(ciphertext, a, b)
97         result.delete('1.0', tk.END)
98         result.insert(tk.END, decrypted_message)
99
100         # Guardar el texto descifrado en un archivo en la misma carpeta que el archivo original
101         output_path = get_output_path(original_file_path, "_d")
102         save_file(decrypted_message, output_path)
103     except Exception as e:
104         messagebox.showerror("Error", f"Error durante el descifrado: {str(e)}")
105
106 # Crear la ventana principal
107 root = tk.Tk()
108 root.title("Cifrador Afín")
109
110 # Crear elementos de la interfaz gráfica
111 original_file_path = ""
112 load_button = tk.Button(root, text="Cargar Archivo", command=load_file)
113 a_label = tk.Label(root, text="Valor de Alpha (Multiplicativo):")
114 a_entry = tk.Entry(root)
115 b_label = tk.Label(root, text="Valor de Beta (Aditivo):")
116 b_entry = tk.Entry(root)
117 text_label = tk.Label(root, text="Texto:")

```

```

118 text = tk.Text(root, height=10, width=40)
119 encrypt_button = tk.Button(root, text="Cifrar Texto", command=encrypt_text)
120 decrypt_button = tk.Button(root, text="Descifrar Texto", command=decrypt_text)
121 result_label = tk.Label(root, text="Resultado:")
122 result = tk.Text(root, height=10, width=40)
123
124 # Colocar elementos en la ventana
125 load_button.grid(row=0, column=0)
126 a_label.grid(row=1, column=0)
127 a_entry.grid(row=1, column=1)
128 b_label.grid(row=2, column=0)
129 b_entry.grid(row=2, column=1)
130 text_label.grid(row=3, column=0)
131 text.grid(row=3, column=1)
132 encrypt_button.grid(row=4, column=0)
133 decrypt_button.grid(row=4, column=1)
134 result_label.grid(row=5, column=0)
135 result.grid(row=5, column=1)
136
137 # Ejecutar la aplicación
138 root.mainloop()
139

```

## Referencias

[1] Colaboradores de los proyectos Wikimedia. “Cifrado afín - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. Accedido el 25 de septiembre de 2023. [En línea]. Disponible: [https://es.wikipedia.org/wiki/Cifrado\\_afin](https://es.wikipedia.org/wiki/Cifrado_afin)