



Universidad Nacional Autónoma de México
Facultad de Ingeniería



Estructura de Datos y Algoritmos I

Actividad 4: Implementación de cifrado César

Sánchez Hernández Marco Antonio

Fecha: 22/marzo/2021

Pseudocódigo

INICIO

//Variables globales

alfabetoLatino[26]: CHARACTER //Arreglo de caracteres de 26 elementos de tamaño.

AlfabetoCifrado[26]: CHARACTER //Arreglo de caracteres de 26 elementos de tamaño.

Opcion, i, j: ENTERO

HACER

ESCRIBIR “Cifrado César”

ESCRIBIR “1. Cifrar mensaje”

ESCRIBIR “2. Descifrar mensaje”

ESCRIBIR “3. Salir”

LEER opcion

SELECCIONAR (opcion) EN

CASO 1 ->

FUNC cifrar (vacío) RET: vacío

//Variables locales

llave: ENTERO

mensaje: CADENA

mensajeCifrado []: CHARACTER //Arreglo dinámico

LEER llave

SI llave < 1 | llave >25 ENTONCES

ESCRIBIR “El valor ingresado no es válido”

FIN SI

DE LO CONTRARIO

FUNC alfabetos(llave: ENTERO) RET: vacío

alfabetoCifrado[26]: CHARACTER

PARA i EN alfabetoLatino //Cada elemento

alfabetoCifrado[i] := alfabetoLatino[llave]

llave := llave +1

SI llave > 25 ENTONCES

llave := 0

FIN PARA

FIN FUNC

LEER mensaje

PARA i EN mensaje //Cada carácter

j := 0

MIENTRAS mensaje[i] != alfabetoLatino[j]

j = j + 1

FIN MIENTRAS

mensajeCifrado[i] = alfabetoCifrado[j]

FIN PARA

ESCRIBIR mensajeCifrado

FIN DE LO CONTRARIO

FIN FUNC

CASO 2 ->

FUN descifrar (vacío) RET: vacío

//Variables locales

llave: ENTERO

mensaje: CADENA

mensajeDescifrado []: CHARACTER //Arreglo dinámico

LEER llave

SI llave < 1 | llave > 25 ENTONCES

ESCRIBIR "El valor ingresado no es válido"

FIN SI

DE LO CONTRARIO

FUNC alfabetos(llave: ENTERO) RET: vacío

alfabetoCifrado[26]: CHARACTER

PARA i EN alfabetoLatino //Cada elemento

alfabetoCifrado[i] := alfabetoLatino[llave]

llave := llave + 1

SI llave > 25 ENTONCES

llave := 0

FIN PARA

FIN FUNC

LEER mensaje

PARA i EN mensaje //Cada carácter

j := 0

MIENTRAS mensaje[i] != alfabetoCifrado[j]

j = j +1

FIN MIENTRAS

mensajeDescifrado[i] = alfabetoLatino[j]

FIN PARA

ESCRIBIR mensajeDescifrado

FIN DE LO CONTRARIO

FIN FUNC

CASO 3 ->

DEFECTO ->

ESCRIBIR “El valor ingresado no es válido”

FIN SELECCIONAR

MIENTRAS opcion != 3

FIN

Implementación

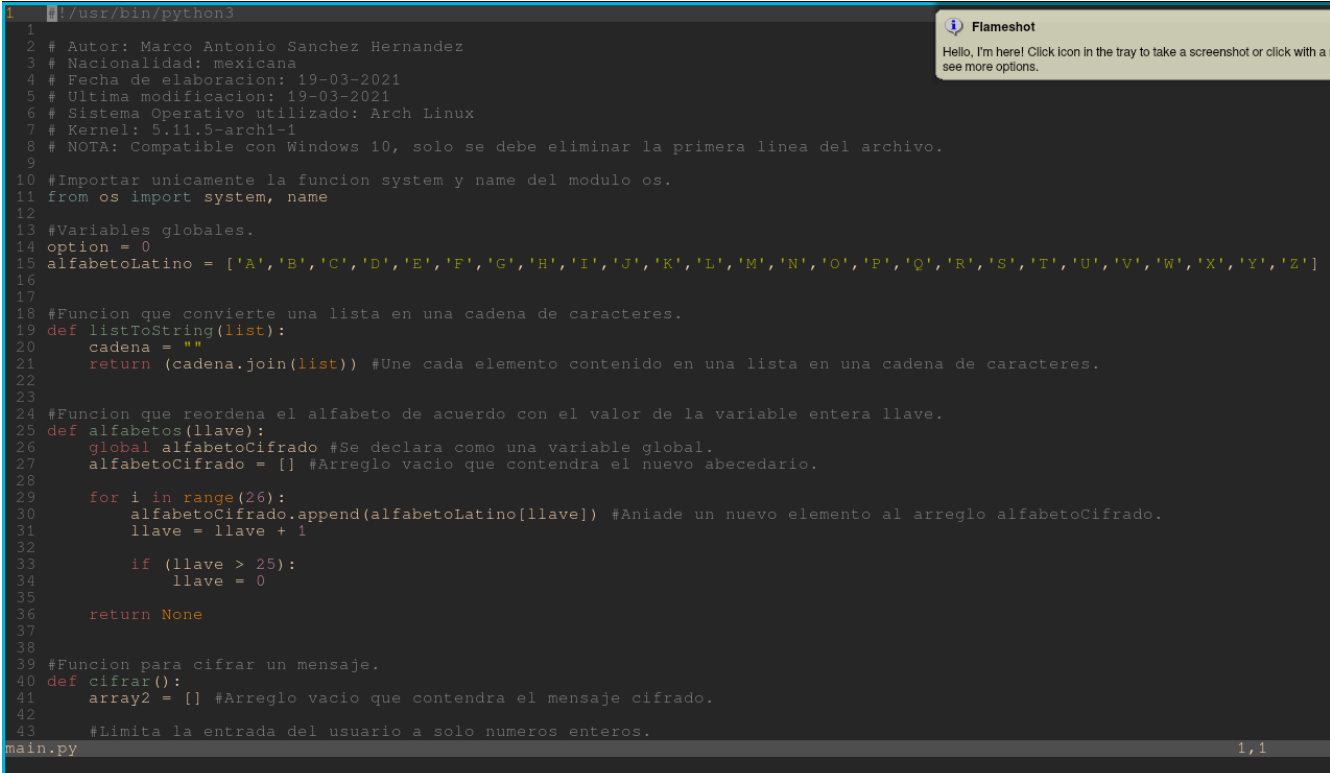
La implementación del algoritmo para cifrar un mensaje mediante el cifrado César fue realizada en el lenguaje de programación Python, en el sistema operativo Arch Linux, sin embargo, se garantiza la compatibilidad con el sistema operativo Windows 10.

Código fuente

Se utilizó el editor de texto NeoVim. La primera línea presente en este archivo corresponde a un Shell Script propio del sistema operativo donde fue creado el programa, esto con el fin de ejecutar el código fuente del programa.

NOTA: el bloque código que permite limpiar la terminal en ambos sistemas fue inspirado en un código publicado por Aditya Goyal en <https://www.codespeedy.com/clear-screen-in-python/>

Repositorio de GitHub: <https://github.com/marco-sanchez-est/EDAI/tree/main/codigos/cifradoCesar>



```
#!/usr/bin/python3
1
2 # Autor: Marco Antonio Sanchez Hernandez
3 # Nacionalidad: mexicana
4 # Fecha de elaboracion: 19-03-2021
5 # Ultima modificacion: 19-03-2021
6 # Sistema Operativo utilizado: Arch Linux
7 # Kernel: 5.11.5-arch1-1
8 # NOTA: Compatible con Windows 10, solo se debe eliminar la primera linea del archivo.
9
10 #Importar unicamente la funcion system y name del modulo os.
11 from os import system, name
12
13 #Variables globales.
14 option = 0
15 alfabetoLatino = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
16
17
18 #Funcion que convierte una lista en una cadena de caracteres.
19 def listToString(list):
20     cadena = ""
21     return (cadena.join(list)) #Une cada elemento contenido en una lista en una cadena de caracteres.
22
23
24 #Funcion que reordena el alfabeto de acuerdo con el valor de la variable entera llave.
25 def alfabetos(llave):
26     global alfabetoCifrado #Se declara como una variable global.
27     alfabetoCifrado = [] #Arreglo vacio que contendra el nuevo abecedario.
28
29     for i in range(26):
30         alfabetoCifrado.append(alfabetoLatino[llave]) #Añade un nuevo elemento al arreglo alfabetoCifrado.
31         llave = llave + 1
32
33         if (llave > 25):
34             llave = 0
35
36     return None
37
38
39 #Funcion para cifrar un mensaje.
40 def cifrar():
41     array2 = [] #Arreglo vacio que contendra el mensaje cifrado.
42
43     #Limita la entrada del usuario a solo numeros enteros.
44
45 main.py
```

```

43     try:
44         llave = int(input("Ingrese un n\u00F3mero entero del 1 al 25: "))
45     except ValueError: #Si el usuario ingresa otro valor que no sea un numero entero se ejecuta este bloque.
46         print("\nIngresar un n\u00F3mero entero")
47         print("Presione ENTER para continuar")
48         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
49         return None #Se termina anticipadamente la funcion y se regresa al menu principal.
50
51     if (llave < 1 or llave > 25):
52         print("\nEl valor ingresado no es v\u00E1lido")
53         print("Presione ENTER para continuar")
54         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
55
56     #Este bloque de codigo se ejecuta si el valor de llave esta entre 1 y 25.
57     else:
58         alfabetos(llave) #Pasa el valor de la variable llave a la funcion alfabetos
59         mensaje = str(input("Ingrese el mensaje que desea cifrar: "))
60         array1 = list(mensaje.upper()) #Convierte la cadena de caracteres ingresada en un arreglo de letras mayusculas.
61
62         for i in range(len(array1)):
63             j = 0
64
65             #Busca la posicion que ocupa una letra en alfabetoLatino.
66             while (array1[i] != alfabetoLatino[j]):
67                 j = j + 1
68
69             array2.append(alfabetoCifrado[j]) #Añade un nuevo elemento al arreglo array2.
70
71         mensajeCifrado = listToString(array2) #Convierte el arreglo array2 en una cadena de caracteres.
72         print(mensajeCifrado)
73         print("\nPresione ENTER para continuar")
74         input() #Espera a que el usuario presione ENTER o ingrese algun valor.
75
76     return None
77
78 #Funcion para descifrar un mensaje.
79 def descifrar():
80
81     array2 = [] #Arreglo vacio que contendra el mensaje descifrado.
82
83     #Limita la entrada del usuario a solo numeros enteros.
84     try:
85         llave = int(input("Ingrese un n\u00F3mero entero del 1 al 25: "))
86     except ValueError: #Si el usuario ingresa otro valor que no sea un numero entero se ejecuta este bloque.
87         print("\nIngresar un n\u00F3mero entero")
88         print("Presione ENTER para continuar")
89         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
90         return None #Se termina anticipadamente la funcion y se regresa al menu principal.
91
92     if (llave < 1 or llave > 25):
93         print("\nEl valor ingresado no es v\u00E1lido")
94         print("Presione ENTER para continuar\n")
95         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
96
97     else:
98         alfabetos(llave)
99         mensaje = str(input("Ingrese el mensaje que desea descifrar: "))
100        array1 = list(mensaje.upper()) #Convierte la cadena de caracteres ingresada en un arreglo de letras mayusculas.
101
102        for i in range(len(array1)):
103            j = 0
104
105            #Busca la posicion que ocupa una letra en alfabetoCifrado.
106            while (array1[i] != alfabetoCifrado[j]):
107                j = j + 1
108
109            array2.append(alfabetoLatino[j]) #Añade un nuevo elemento al arreglo array2.
110
111        mensajeDescifrado = listToString(array2) #Convierte el arreglo array2 en una cadena de caracteres.
112        print(mensajeDescifrado)
113        print("\nPresione ENTER para continuar")
114        input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
115
116    return None
117
118 #Ciclo principal
119 while option != 3:
120
121     option = 0
122
123     #Llamada a scripts para limpiar la terminal.
124     if name == 'nt':
125         _ = system('cls') #Script para Windows 10.
126     else:
127         _ = system('clear') #Script para MacOS y Linux.

```

main.py

88,1

```

43     except ValueError: #Si el usuario ingresa otro valor que no sea un numero entero se ejecuta este bloque.
44         print("\nIngresar un n\u00F3mero entero")
45         print("Presione ENTER para continuar")
46         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
47         return None #Se termina anticipadamente la funcion y se regresa al menu principal.
48
49     if (llave < 1 or llave > 25):
50         print("\nEl valor ingresado no es v\u00E1lido")
51         print("Presione ENTER para continuar\n")
52         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
53
54     else:
55         alfabetos(llave)
56         mensaje = str(input("Ingrese el mensaje que desea descifrar: "))
57         array1 = list(mensaje.upper()) #Convierte la cadena de caracteres ingresada en un arreglo de letras mayusculas.
58
59         for i in range(len(array1)):
60             j = 0
61
62             #Busca la posicion que ocupa una letra en alfabetoCifrado.
63             while (array1[i] != alfabetoCifrado[j]):
64                 j = j + 1
65
66             array2.append(alfabetoLatino[j]) #Añade un nuevo elemento al arreglo array2.
67
68         mensajeDescifrado = listToString(array2) #Convierte el arreglo array2 en una cadena de caracteres.
69         print(mensajeDescifrado)
70         print("\nPresione ENTER para continuar")
71         input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
72
73     return None
74
75 #Ciclo principal
76 while option != 3:
77
78     option = 0
79
80     #Llamada a scripts para limpiar la terminal.
81     if name == 'nt':
82         _ = system('cls') #Script para Windows 10.
83     else:
84         _ = system('clear') #Script para MacOS y Linux.

```

main.py

132,1

```

25
24 #Muestra el menu de opciones
23 print("\n\tCifrado C\u00E9sar\n")
22 print("1. Cifrar mensaje")
21 print("2. Descifrar mensaje")
20 print("3. Salir\n")
19
18 #Limita la entrada del usuario a solo numeros enteros.
17 try:
16     option = int(input("Seleccione una opci\u00F3n: "))
15 except ValueError: #Si el usuario ingresa otro valor que no sea un numero entero se ejecuta este bloque.
14     print("\ningresar un n\u00FAmero entero")
13
12 #Default
11 if (option < 1 or option > 3):
10     print("\nEL valor ingresado no es v\u00E1lido")
9     print("Presione ENTER para continuar")
8     input() #Espera que el usuario presione ENTER o ingrese algun valor para continuar.
7
6 #Case 1
5 elif (option == 1):
4     cifrar() #Llama a la funcion cifrar
3
2 #Case 2
1 elif (option == 2):
158     descifrar() #Llama a la funcion descifrar
main.py 158,1

```

Ejecución

Menú principal.

```

          Cifrado César
1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 

```

Manejo de errores para la selección de opciones.

```

          Cifrado César
1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 0

EL valor ingresado no es válido
Presione ENTER para continuar

```

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 7

EL valor ingresado no es válido
Presione ENTER para continuar
█
```

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: g

Ingresar un número entero

EL valor ingresado no es válido
Presione ENTER para continuar
█
```

Manejo de errores para la opción “Cifrar mensaje”

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 1
Ingrese un número entero del 1 al 25: 0

El valor ingresado no es válido
Presione ENTER para continuar
█
```

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Selección una opción: 1
Ingrese un número entero del 1 al 25: 28

El valor ingresado no es válido
Presione ENTER para continuar
█
```



```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 1
Ingrese un número entero del 1 al 25: f

Ingrese un número entero
Presione ENTER para continuar
█
```

Manejo de errores para la opción “Descifrar mensaje”

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 2
Ingrese un número entero del 1 al 25: -1

EL valor ingresado no es válido
Presione ENTER para continuar
█
```

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 2
Ingrese un número entero del 1 al 25: 30

EL valor ingresado no es válido
Presione ENTER para continuar
█
```

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 2
Ingrese un número entero del 1 al 25: cdf

Ingrese un número entero
Presione ENTER para continuar
█
```

Cifrado de mensaje

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 1
Ingrese un número entero del 1 al 25: 3
Ingrese el mensaje que desea cifrar: hola
KROD

Presione ENTER para continuar
█
```

Descifrado de mensaje con la misma llave que fue cifrado

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 2
Ingrese un número entero del 1 al 25: 3
Ingrese el mensaje que desea descifrar: krod
HOLA

Presione ENTER para continuar
█
```

Descifrado de mensaje con una llave diferente

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 2
Ingrese un número entero del 1 al 25: 5
Ingrese el mensaje que desea descifrar: krod
FMJY

Presione ENTER para continuar
█
```

Funcionamiento tercera opción “Salir”

```
Cifrado César  
1. Cifrar mensaje  
2. Descifrar mensaje  
3. Salir  
Seleccione una opción: 3  
[marko@marko-inspition-5567 cifradoCesar]$
```

Referencias

- Goyal A. (s. f.). CodeSpeedy. “*Clear screen in Python*”. Recuperado el 20 de marzo de 2021, de <https://www.codespeedy.com/clear-screen-in-python/>
- TutorialsPoint (s. f.). “*Python String upper() Method*”. Recuperado el 19 de marzo de 2021, de https://www.tutorialspoint.com/python/string_upper.htm
- w3school. (s. f.). “*Python List append() Method*”. Recuperado el 19 de marzo de 2021, de https://www.w3schools.com/python/ref_list_append.asp