



Universidad Nacional Autónoma de México

Facultad de Ingeniería



Asignatura:

Estructura de Datos y Algoritmos I

Actividad #3 | Pseudocódigo e Implementación del Cifrado César

Nombre del Alumno:

Sánchez Estrada Angel Isaac

Maestro:

M.I. Marco Antonio Martínez Quintana

Grupo:

15

Fecha:

22/03/2021



PSEUDOCÓDIGO E IMPLEMENTACIÓN DEL CIFRADO CÉSAR

Opción 1

PSEUDOCÓDIGO – Lenguaje C++

INICIO

DECLARAR VARIABLES

void cifrar(char *textoNormal);

void descifrar(char *textoCifrado);

char abecedarioEnClaro[TAM_ABC] =

{'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'};

char abecedarioCifrado[TAM_ABC] =

{'D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','A','B','C'};

int contadorAbcedario, contadorPalabra, indice = 0

IMPRIMIR ¿Qué desea hacer?

LEER opción

SWITCH opción

CASO 1

IMPRIMIR cifrar mensaje

IMPRIMIR ingresar texto a cifrar

LEER texto

FOR contadorPalabra<textoNormal[contadorPalabra] AB

contadorPalabra++

FOR contadorAbcedario<TAM_ABC AB

contadorAbcedario++

IMPRIMIR abecedario cifrado

FIN FOR

FIN FOR

break

CASO 2

IMPRIMIR descifrar mensaje

```

    IMPRIMIR ingresar texto a descifrar
    LEER texto
    FOR contadorPalabra<textoCifrado[contadorPalabra] AB
        contadorPalabra++
        FOR contadorAbcedario<TAM_ABC AB
            contadorAbcedario++
            IMPRIMIR abecedario descifrado
        FIN FOR
    FIN FOR
    break
CASO 3
    IMPRIMIR salir
    break
CASO 4
    IMPRIMIR opción no válida
FIN

```

Opción 2

PSEUDOCÓDIGO – Python

INICIO

DECLARAR VARIABLES

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

//Menú

ESCRIBIR “Cifrado César”

ESCRIBIR "1. Cifrar mensaje"

ESCRIBIR "2. Descifrar mensaje"

ESCRIBIR "3. Salir"

LEER opción

SELECCIONAR (opción) EN

CASO 1

 FUNCIÓN cifrar (vacío) RET: vacío

 //Variables locales

 clave: ENTERO

 mensaje: CADENA

 mensajeCifrado []: CHARACTER //Arreglo dinámico

 LEER clave

 SI clave < 1 | clave >25 ENTONCES

 ESCRIBIR "El valor ingresado no es válido"

 FIN SI

 DE LO CONTRARIO

 FUNCIÓN alfabetos(clave: ENTERO) RET: vacío

 FUNCIÓN alfabetoCifrado[26]: CHARACTER

 PARA i EN alfabetoLatino //Cada elemento

 alfabetoCifrado[i] := alfabetoLatino[clave]

 clave := clave +1

 SI clave > 25 ENTONCES

 Clave := 0

 FIN PARA

 FIN FUNCIÓN

 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 FUNCIÓN

```

CASO 2

```

FUNCIÓN descifrar (vacío) RET: vacío
    //Variables locales
    clave: ENTERO
    mensaje: CADENA
    mensajeDescifrado [ ]: CHARACTER //Arreglo dinámico
    LEER clave
    SI clave < 1 | clave > 25 ENTONCES
        ESCRIBIR "El valor ingresado no es válido"
    FIN SI
    DE LO CONTRARIO
        FUNCIÓN alfabetos(clave: ENTERO) RET: vacío
            alfabetoCifrado[26]: CHARACTER
            PARA i EN alfabetoLatino //Cada elemento
                alfabetoCifrado[i] := alfabetoLatino[llave]
                clave := clave + 1
            SI clave > 25 ENTONCES
                clave := 0
            FIN PARA
        FIN FUNCIÓN
    FIN FUNCIÓN

```

```
        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 FUNCIÓN

CASO 3
POR DEFECTO
    ESCRIBIR "El valor ingresado no es válido"
FIN SELECCIONAR
MIENTRAS opción != 3
FIN
```

Opción 1

IMPLEMENTACIÓN – Lenguaje C++

Código Fuente

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<string.h>
using namespace std;

/*
 * Autor:Sánchez Estrada Angel Isaac
 * Nacionalidad: Mexicana
 * Fecha de elaboración: 19-03-2021
 * Ultima modificación: 20-03-2021
 * Sistema Operativo: Windows 10
 */

/*
 Programa que realiza la implementación del cifrado César
 Para cifrar y descifrar un mensaje
 */
//Definimos las variables y les asignamos un valor definido
#define TAM_PALABRA 20
#define TAM_ABC 26

//Declaración de Caracteres
char ao=162, aa=160, au=163, ai=161, ae=130, sp=168 ,aim=214;

//Declaración de matrices
char abecedarioEnClaro[TAM_ABC] =
{'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'};
char abecedarioCifrado[TAM_ABC] =
{'D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V',
'W',
'X','Y','Z','A','B','C'};

//Declaramos las funciones
void cifrar(char *textoEnClaro);
void descifrar(char *textoCifrado);
int main(){
```

```

short opcion = 0, contador;
char palabra[TAM_PALABRA];
while (1){
    //Menu del Programa
    cout<<"\n\tCIFRADO CESAR\n"<<endl;

    //Muestra los Abecedarios
    //Abecedario Normal
    for (contador=0 ; contador<26; contador++)
        cout<<*(abecedarioEnClaro+contador);
    cout<<"\n";
    //Abecedario Cifrado
    for (contador=0 ; contador<26; contador++)
        cout<<*(abecedarioCifrado+contador);

    //Opciones del menú
    cout<<"\n";
    cout<<"\nElegir una opcion\n";
    cout<<"1. Cifrar\n";
    cout<<"2. Descifrar\n";
    cout<<"3. Salir\n";
    cout<<">> ";
    cin>>opcion;
    switch(opcion){
        case 1:
            //Mensaje para que ingrese la palabra que quiere cifrar
            cout<<"\nIngresar la palabra a cifrar (en mayusculas)"<<endl;
            cout<<"======"<<endl;
            cin>>palabra;
            cifrar(palabra); //Recupera el codigo que se definio en void cifr
ar

            break;
        case 2:
            //Mensaje para que el usuario ingrese la palabra a descifrar
            cout<<"\nIngresar la palabra a descifrar (en mayusculas)"<<endl;
            cout<<"======"<<endl;
            cin>>palabra;
            descifrar(palabra); //Recupera el codigo que se definio en void d
escifar

            break;
        case 3:
            return 0;
        default:
            cout<<"Opción no valida."<<endl; //Mensaje que da por defecto si
no ingresa un numero que se asigne al menú

```



```

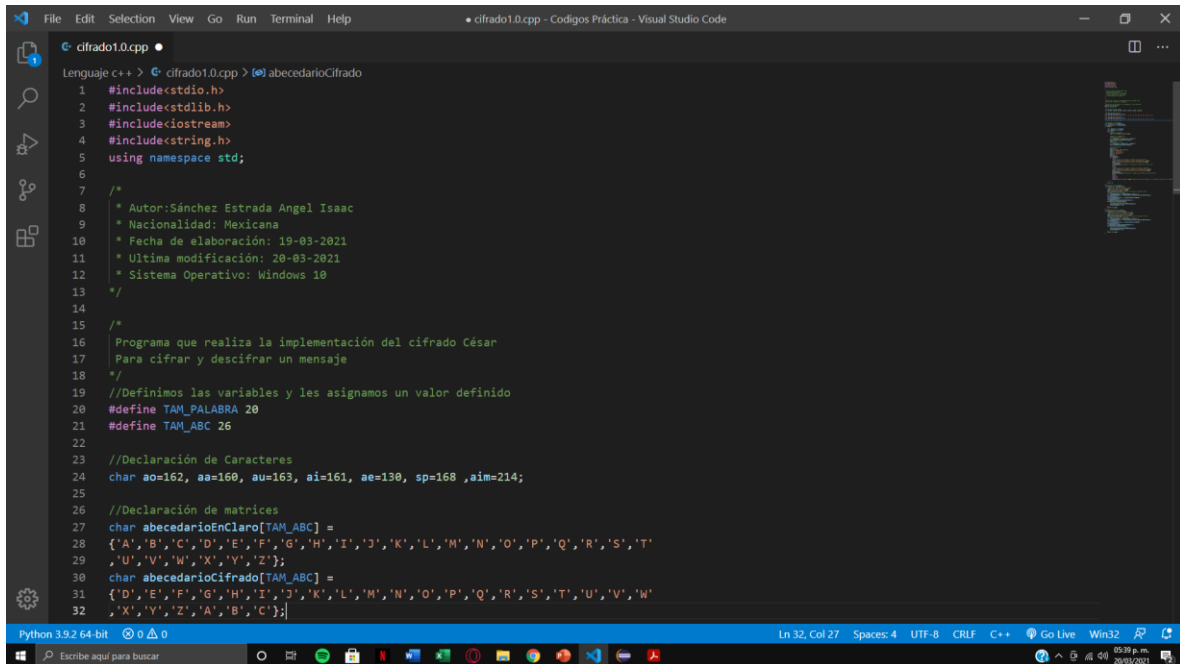
    }
}
return 0;
}

//Código para Cifrar mensaje
void cifrar(char *textoEnClaro){
    //Código que muestra el mensaje cifrado
    cout<<"\nEl texto cifrado es"<<endl;
    cout<<"======"<<endl;
    cout<<textoEnClaro<<endl; //Muestra en pantalla el mensaje normal
    //Código que cifra el mensaje
    int contadorAbcedario, contadorPalabra, indice = 0;
    for (contadorPalabra=0 ; contadorPalabra<textoEnClaro[contadorPalabra] ;
        contadorPalabra++)
        for (contadorAbcedario=0 ; contadorAbcedario<TAM_ABC ;
            contadorAbcedario++)
            if (abecedarioEnClaro[contadorAbcedario] ==
                textoEnClaro[contadorPalabra]){
                //Muestra el mensaje cifrado
                cout<<abecedarioCifrado[contadorAbcedario];
                contadorAbcedario = 26;
            }
        cout<<"\n"<<endl;
}

//Código para Descifrar mensaje
void descifrar(char *textoCifrado){
    //Código que muestra el mensaje descifrado
    cout<<"\nEl texto %s descifrado es "<<endl;
    cout<<"======"<<endl;
    cout<<textoCifrado<<endl; //Muestra en pantalla el mensaje cifrado
    //Código para descifrar el mensaje
    int contadorAbcedario, contadorPalabra, indice = 0;
    for (contadorPalabra=0 ; contadorPalabra<textoCifrado[contadorPalabra] ;
        contadorPalabra++)
        for (contadorAbcedario=0 ; contadorAbcedario<TAM_ABC ;
            contadorAbcedario++)
            if (abecedarioCifrado[contadorAbcedario] ==
                textoCifrado[contadorPalabra]){
                //Muestra el mensaje descifrado
                cout<<abecedarioEnClaro[contadorAbcedario];
                contadorAbcedario = 26;
            }
        cout<<"\n"<<endl;
}
}

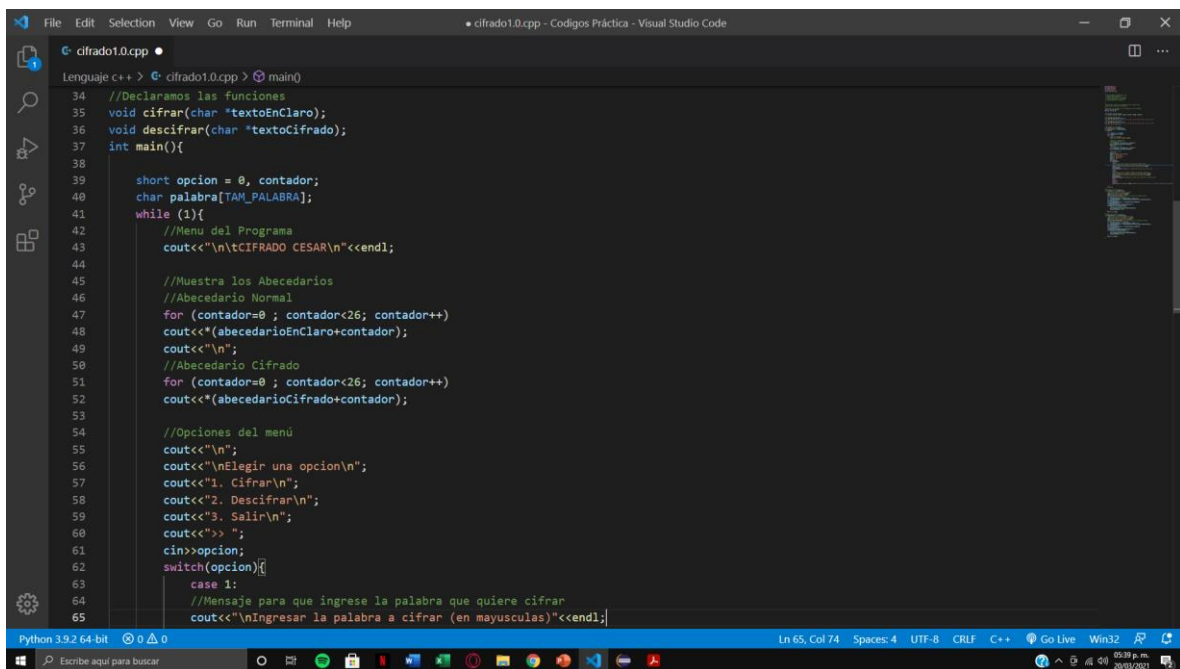
```

Código – Visual Studio Code



```
File Edit Selection View Go Run Terminal Help
• cifrado1.0.cpp - Codigos Práctica - Visual Studio Code

Lenguaje c++ > G cifrado1.0.cpp > (6) abecedarioCifrado
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<iostream>
4  #include<string.h>
5  using namespace std;
6
7  /*
8   * Autor:Sánchez Estrada Angel Isaac
9   * Nacionalidad: Mexicana
10  * Fecha de elaboración: 19-03-2021
11  * Última modificación: 20-03-2021
12  * Sistema Operativo: Windows 10
13  */
14
15  /*
16  Programa que realiza la implementación del cifrado César
17  Para cifrar y descifrar un mensaje
18  */
19  //Definimos las variables y les asignamos un valor definido
20  #define TAM_PALABRA 20
21  #define TAM_ABC 26
22
23  //Declaración de Caracteres
24  char ao=162, aa=160, au=163, ai=161, ae=130, sp=168 ,aim=214;
25
26  //Declaración de matrices
27  char abecedarioEnClaro[TAM_ABC] =
28  {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T',
29  'U','V','W','X','Y','Z'};
30  char abecedarioCifrado[TAM_ABC] =
31  {'D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W',
32  'X','Y','Z','A','B','C'};
33
Python 3.9.2 64-bit 0 0 0
Ln 32, Col 27 Spaces: 4 UTF-8 CRLF C++ Go Live Win32
Escribe aquí para buscar
```



```
File Edit Selection View Go Run Terminal Help
• cifrado1.0.cpp - Codigos Práctica - Visual Studio Code

Lenguaje c++ > G cifrado1.0.cpp > (6) main()
34  //Declaramos las funciones
35  void cifrar(char *textoEnClaro);
36  void descifrar(char *textoCifrado);
37  int main(){
38
39      short opcion = 0, contador;
40      char palabra[TAM_PALABRA];
41      while (1){
42          //Menu del Programa
43          cout<<"\n\tCIFRADO CESAR\n"<<endl;
44
45          //Muestra los Abecedarios
46          //Abecedario Normal
47          for (contador=0 ; contador<26; contador++)
48              cout<<*(abecedarioEnClaro+contador);
49          cout<<"\n";
50          //Abecedario Cifrado
51          for (contador=0 ; contador<26; contador++)
52              cout<<*(abecedarioCifrado+contador);
53
54          //Opciones del menú
55          cout<<"\n";
56          cout<<"\nElegir una opcion\n";
57          cout<<"1. Cifrar\n";
58          cout<<"2. Descifrar\n";
59          cout<<"3. Salir\n";
60          cout<<">>> ";
61          cin>>opcion;
62          switch(opcion){
63              case 1:
64                  //Mensaje para que ingrese la palabra que quiere cifrar
65                  cout<<"\nIngresar la palabra a cifrar (en mayusculas)"<<endl;
66
Python 3.9.2 64-bit 0 0 0
Ln 65, Col 74 Spaces: 4 UTF-8 CRLF C++ Go Live Win32
Escribe aquí para buscar
```

```
File Edit Selection View Go Run Terminal Help • cifrado1.0.cpp - Codigos Práctica - Visual Studio Code

Lenguaje c++ > G cifrado1.0.cpp > cifrar(char *)

66         cout<<"===== "<<endl;
67         cin>>palabra;
68         cifrar(palabra); //Recupera el codigo que se definio en void cifrar
69         break;
70     case 2:
71         //Mensaje para que el usuario ingrese la palabra a descifrar
72         cout<<"\nIngresa la palabra a descifrar (en mayusculas)"<<endl;
73         cout<<"===== "<<endl;
74         cin>>palabra;
75         descifrar(palabra); //Recupera el codigo que se definio en void descifrar
76         break;
77     case 3:
78         return 0;
79     default:
80         cout<<"Opción no valida."<<endl; //Mensaje que da por defecto si no ingresa un numero que se asigne al menú
81     }
82 }
83 return 0;
84 }
85 //Código para Cifrar mensaje
86 void cifrar(char "textoEnClaro"){
87     //Código que muestra el mensaje cifrado
88     cout<<"\nEl texto cifrado es"<<endl;
89     cout<<"===== "<<endl;
90     cout<<textoEnClaro<<endl; //Muestra en pantalla el mensaje normal
91     //Código que cifra el mensaje
92     int contadorAbecedario, contadorPalabra, indice = 0;
93     for (contadorPalabra=0 ; contadorPalabra<textoEnClaro[contadorPalabra] ;
94         contadorPalabra++)
95         for (contadorAbecedario=0 ; contadorAbecedario<TAM_ABC ;
96             contadorAbecedario++)
97             if (abecedarioEnClaro[contadorAbecedario] ==
```

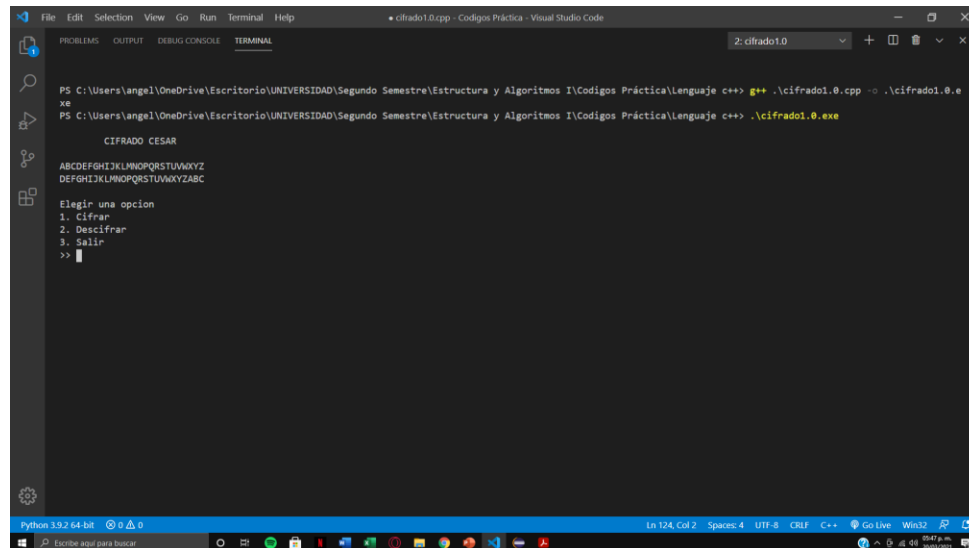
```
File Edit Selection View Go Run Terminal Help • cifrado1.0.cpp - Codigos Práctica - Visual Studio Code

Lenguaje c++ > G cifrado1.0.cpp > descifrar(char *)

95     for (contadorAbecedario=0 ; contadorAbecedario<TAM_ABC ;
96         contadorAbecedario++)
97         if (abecedarioEnClaro[contadorAbecedario] ==
98             textoEnClaro[contadorPalabra]){
99             //Muestra el mensaje cifrado
100             cout<<abecedarioCifrado[contadorAbecedario];
101             contadorAbecedario = 26;
102         }
103         cout<<"\n"<<endl;
104     }
105     //Código para Descifrar mensaje
106     void descifrar(char "textoCifrado"){
107         //Código que muestra el mensaje descifrado
108         cout<<"\nEl texto %s descifrado es "<<endl;
109         cout<<"===== "<<endl;
110         cout<<textoCifrado<<endl; //Muestra en pantalla el mensaje cifrado
111         //Código para descifrar el mensaje
112         int contadorAbecedario, contadorPalabra, indice = 0;
113         for (contadorPalabra=0 ; contadorPalabra<textoCifrado[contadorPalabra] ;
114             contadorPalabra++)
115         for (contadorAbecedario=0 ; contadorAbecedario<TAM_ABC ;
116             contadorAbecedario++)
117             if (abecedarioCifrado[contadorAbecedario] ==
118                 textoCifrado[contadorPalabra]){
119                 //Muestra el mensaje descifrado
120                 cout<<abecedarioEnClaro[contadorAbecedario];
121                 contadorAbecedario = 26;
122             }
123             cout<<"\n"<<endl;
124     }
```

Funcionamiento, compilación y ejecución – Lenguaje C++

Compilación y Ejecución



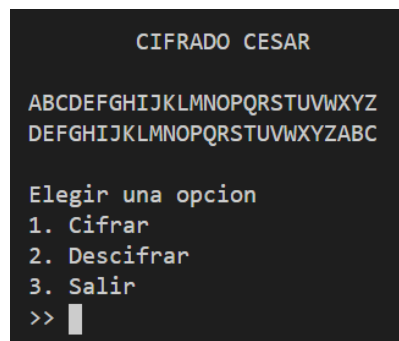
```
PS C:\Users\angel\OneDrive\Escritorio\UNIVERSIDAD\Segundo Semestre\Estructura y Algoritmos I\Codigos Práctica\Lenguaje c++> g++ .\cifrado1.0.cpp -o .\cifrado1.0.exe
PS C:\Users\angel\OneDrive\Escritorio\UNIVERSIDAD\Segundo Semestre\Estructura y Algoritmos I\Codigos Práctica\Lenguaje c++> .\cifrado1.0.exe

CIFRADO CESAR

ABCDEFGHIJKLMNPOQRSTUVWXYZ
DEFGHIJKLMNPOQRSTUVWXYZABC

Elegir una opcion
1. Cifrar
2. Descifrar
3. Salir
>> 
```

Menú principal



```
CIFRADO CESAR

ABCDEFGHIJKLMNPOQRSTUVWXYZ
DEFGHIJKLMNPOQRSTUVWXYZABC

Elegir una opcion
1. Cifrar
2. Descifrar
3. Salir
>> 
```

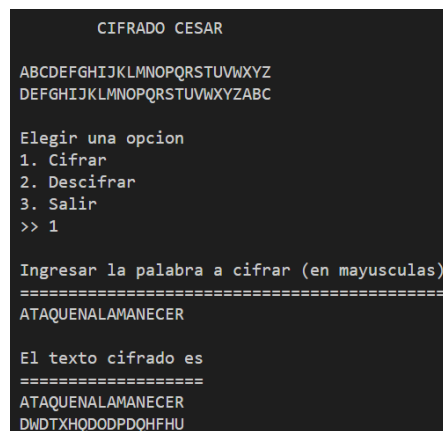
Opción 1 - Cifrado del mensaje

Mensaje:

ATAQUENALAMANECER

Cifrado:

DWDTXHQQDODPDQHFHU



```
CIFRADO CESAR

ABCDEFGHIJKLMNPOQRSTUVWXYZ
DEFGHIJKLMNPOQRSTUVWXYZABC

Elegir una opcion
1. Cifrar
2. Descifrar
3. Salir
>> 1

Ingresar la palabra a cifrar (en mayusculas)
=====
ATAQUENALAMANECER

El texto cifrado es
=====
ATAQUENALAMANECER
DWDTXHQQDODPDQHFHU
```

Opción 2 - Descifrado del mensaje

Cifrado:

DWDTXHQDODPDQHFHU

Descifrado:

ATAQUENALAMANECER

```
CIFRADO CESAR

ABCDEFGHIJKLMNPOQRSTUVWXYZ
DEFGHIJKLMNPOQRSTUVWXYZABC

Elegir una opcion
1. Cifrar
2. Descifrar
3. Salir
>> 2

Ingresar la palabra a descifrar (en mayusculas)
=====
DWDTXHQDODPDQHFHU

El texto %s descifrado es
=====
DWDTXHQDODPDQHFHU
ATAQUENALAMANECER
```

Opción 3 – Salir

```
CIFRADO CESAR

ABCDEFGHIJKLMNPOQRSTUVWXYZ
DEFGHIJKLMNPOQRSTUVWXYZABC

Elegir una opcion
1. Cifrar
2. Descifrar
3. Salir
>> 3
PS C:\Users\angel\OneDrive\Escritorio\UNIVERSIDAD\Segundo Semestre\Estructura
y Algoritmos I\Codigos Práctica\Lenguaje c++>
```

Repositorio GitHub – Lenguaje C++

<https://github.com/1an2l/Estructura-de-Datos-y-Algoritmos-I/tree/main/CÓDIGOS/CIFRADO%20CÉSAR/LENGUAJE%20C++>

Opción 2

IMPLEMENTACIÓN – Python

Código Fuente

```
# Autor:Sánchez Estrada Angel Isaac
# Nacionalidad: Mexicana
# Fecha de elaboración: 19-03-2021
```

```

# Ultima modificación: 20-03-2021
# Sistema Operativo: Windows 10

from os import system, name

#Declaracion de variables.
option = 0
#Declaración de matriz
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']

#Funcion que convierte una lista en una cadena de caracteres.
def listToString(list):
    cadena = ""
    return (cadena.join(list)) #Une cada elemento contenido en una lista en
una cadena de caracteres.

#Funcion que reordena el alfabeto de acuerdo con el valor de la variable ent
era clave.
def alfabetos(clave):
    global alfabetoCifrado #Se declara como una variable global.
    alfabetoCifrado = [] #Arreglo vacio que contendra el nuevo abecedario.

    for i in range(26):
        alfabetoCifrado.append(alfabetoLatino[clave]) #Añade un nuevo eleme
nto al arreglo alfabetoCifrado.
        clave = clave + 1

    if (clave > 25):
        clave = 0

    return None

#Código para cifrar un mensaje.
def cifrar():
    array2 = [] #Arreglo que contendra el mensaje cifrado.

    #Limita la entrada a solo numeros enteros.
    try:
        clave = int(input("Ingrese un número entero del 1 al 25: "))
    except ValueError: #Si el usuario ingresa otro valor que no sea un numer
o entero se ejecuta este bloque.
        print("\nIngresar un número entero")

```

```

        print("Presione ENTER para continuar")
        input() #Espera que el usuario presione ENTER o ingrese algun valor
para continuar.
        return None #Se termina anticipadamente la funcion y se regresa al m
enu principal.

    if (clave < 1 or clave > 25):
        print("\nEl valor ingresado no es válido")
        print("Presione ENTER para continuar")
        input() #Espera que el usuario presione ENTER o ingrese algun valor
para continuar.

    #Este bloque de codigo se ejecuta si el valor de clave esta entre 1 y 2
5.
    else:
        alfabetos(clave) #Pasa el valor de la variable clave a la funcion al
fabetos
        mensaje = str(input("Ingrese el mensaje que desea cifrar: "))
        array1 = list(mensaje.upper()) #Convierte la cadena de caracteres in
gresada en un arreglo de letras mayusculas.

        for i in range(len(array1)):
            j = 0

            #Busca la posicion que ocupa una letra en alfabetoLatino.
            while (array1[i] != alfabetoLatino[j]):
                j = j + 1

            array2.append(alfabetoCifrado[j]) #Añade un nuevo elemento al a
rreglo array2.

        mensajeCifrado = listToString(array2) #Convierte el arreglo array2 e
n una cadena de caracteres.
        print(mensajeCifrado)
        print("\nPresione ENTER para continuar")
        input() #Espera a que el usuario presione ENTER o ingrese algun valo
r.

    return None

#Código para descifrar un mensaje.
def descifrar():

    array2 = [] #Arreglo vacio que contendra el mensaje descifrado.

```

```

#Limita la entrada del usuario a solo numeros enteros.
try:
    clave = int(input("Ingrese un n\u00FAmero entero del 1 al 25: "))
except ValueError: #Si el usuario ingresa otro valor que no sea un numero
    o entero se ejecuta este bloque.
    print("\nIngresar un n\u00FAmero entero")
    print("Presione ENTER para continuar")
    input() #Espera que el usuario presione ENTER o ingrese algun valor
    para continuar.
    return None #Se termina anticipadamente la funcion y se regresa al m
    enu principal.

if (clave < 1 or clave > 25):
    print("\nEl valor ingresado no es v\u00E1lido")
    print("Presione ENTER para continuar\n")
    input() #Espera que el usuario presione ENTER o ingrese algun valor
    para continuar.

else:
    alfabetos(clave)
    mensaje = str(input("Ingrese el mensaje que desea descifrar: "))
    array1 = list(mensaje.upper()) #Convierte la cadena de caracteres in
    gresada en un arreglo de letras mayusculas.

    for i in range(len(array1)):
        j = 0

        #Busca la posicion que ocupa una letra en alfabetoCifrado.
        while (array1[i] != alfabetoCifrado[j]):
            j = j +1

        array2.append(alfabetoLatino[j]) #Añade un nuevo elemento al ar
        reglo array2.

    mensajeDescifrado = listToString(array2) #Convierte el arreglo array
    2 en una cadena de caracteres.
    print(mensajeDescifrado)
    print("\nPresione ENTER para continuar")
    input() #Espera que el usuario presione ENTER o ingrese algun valor
    para continuar.

    return None

```



```

#Ciclo principal
while option != 3:

    option = 0

    #Llamada a scripts para limpiar la terminal.
    if name == 'nt':
        _ = system('cls') #Script para Windows 10.

    else:
        _ = system('clear') #Script para MacOS y Linux.

    #Muestra el menu de opciones
    print("\n\tCifrado C\u00E9sar\n")
    print("1. Cifrar mensaje")
    print("2. Descifrar mensaje")
    print("3. Salir\n")

    #Limita la entrada del usuario a solo numeros enteros.
    try:
        option = int(input("Seleccione una opci\u00F3n: "))
    except ValueError: #Si el usuario ingresa otro valor que no sea un numero
        #o entero se ejecuta este bloque.
        print("\nIngresar un n\u00FAmero entero")

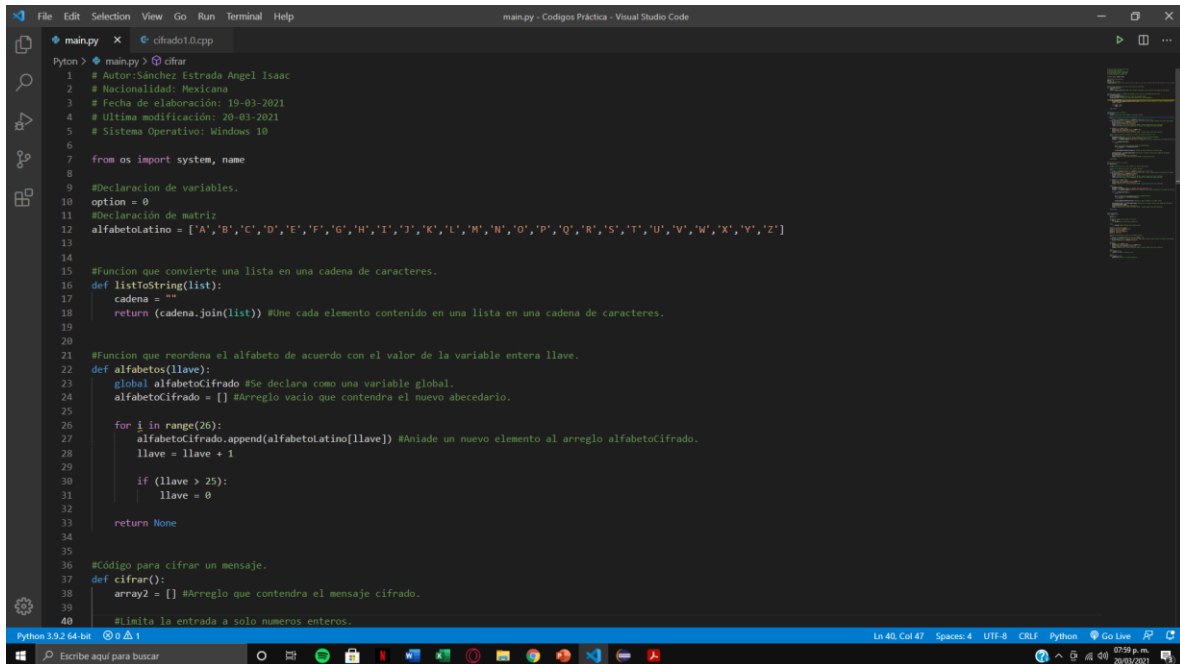
    #Default
    if (option < 1 or option > 3):
        print("\nEL valor ingresado no es v\u00E1lido")
        print("Presione ENTER para continuar")
        input() #Espera que el usuario presione ENTER o ingrese algun valor
        #para continuar.

    #Case 1
    elif (option == 1):
        cifrar() #Llama a la funcion cifrar

    #Case 2
    elif (option == 2):
        descifrar() #Llama a la funcion descifrar

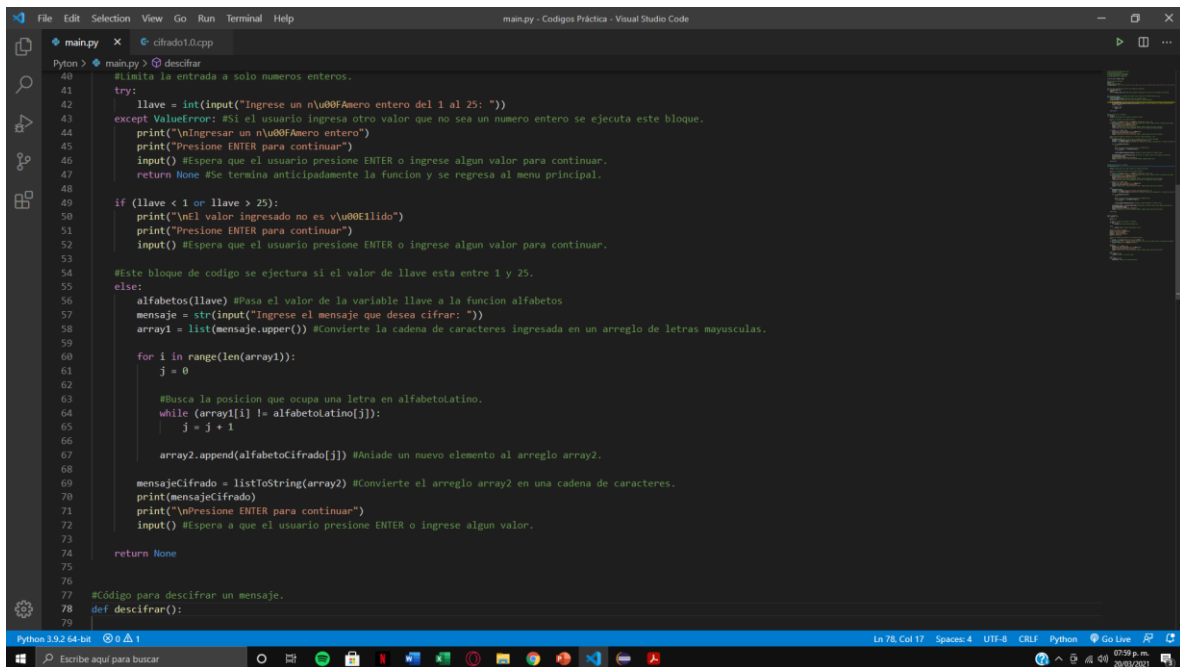
```

Código – Visual Studio Code



```
File Edit Selection View Go Run Terminal Help
main.py - Codigos Práctica - Visual Studio Code

main.py x cifrado1.0.cpp
Python > main.py > cifrar
1 # Autor: Sánchez Estrada Angel Isaac
2 # Nacionalidad: Mexicana
3 # Fecha de elaboración: 19-03-2021
4 # Última modificación: 20-03-2021
5 # Sistema Operativo: Windows 10
6
7 from os import system, name
8
9 #Declaración de variables.
10 option = 0
11 #Declaración de matriz
12 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']
13
14
15 #Función que convierte una lista en una cadena de caracteres.
16 def listToString(list):
17     cadena = ""
18     return (cadena.join(list)) #Une cada elemento contenido en una lista en una cadena de caracteres.
19
20
21 #Función que reordena el alfabeto de acuerdo con el valor de la variable llave.
22 def alfabetos(llave):
23     global alfabetoCifrado #Se declara como una variable global.
24     alfabetoCifrado = [] #Arreglo vacío que contendrá el nuevo abecedario.
25
26     for i in range(26):
27         alfabetoCifrado.append(alfabetoLatino[llave]) #Añade un nuevo elemento al arreglo alfabetoCifrado.
28         llave = llave + 1
29
30         if (llave > 25):
31             llave = 0
32
33     return None
34
35
36 #Código para cifrar un mensaje.
37 def cifrar():
38     array2 = [] #Arreglo que contendrá el mensaje cifrado.
39
40     #Limita la entrada a solo números enteros.
```



```
File Edit Selection View Go Run Terminal Help
main.py - Codigos Práctica - Visual Studio Code

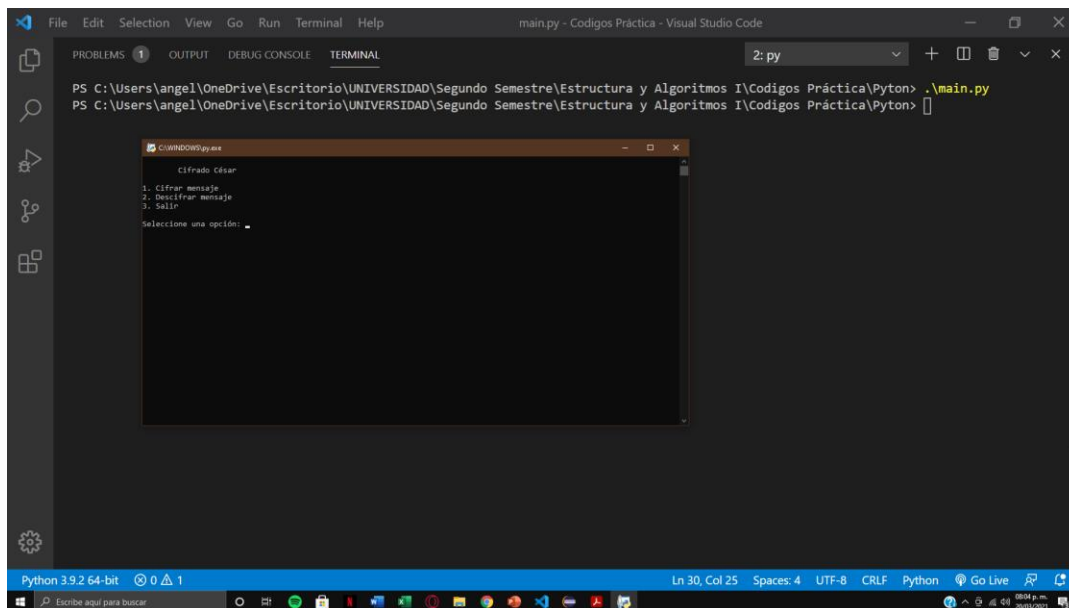
main.py x cifrado1.0.cpp
Python > main.py > descifrar
40 #Limita la entrada a solo números enteros.
41 try:
42     llave = int(input("Ingrese un número entero del 1 al 25: "))
43 except ValueError: #Si el usuario ingresa otro valor que no sea un número entero se ejecuta este bloque.
44     print("\nIngrese un número entero")
45     print("Presione ENTER para continuar")
46     input() #Espera que el usuario presione ENTER o ingrese algún valor para continuar.
47     return None #Se termina anticipadamente la función y se regresa al menú principal.
48
49 if (llave < 1 or llave > 25):
50     print("\nEl valor ingresado no es válido")
51     print("Presione ENTER para continuar")
52     input() #Espera que el usuario presione ENTER o ingrese algún valor para continuar.
53
54 #Este bloque de código se ejecuta si el valor de llave está entre 1 y 25.
55 else:
56     alfabetos(llave) #Pasa el valor de la variable llave a la función alfabetos
57     mensaje = str(input("Ingrese el mensaje que desea cifrar: "))
58     array1 = list(mensaje.upper()) #Convierte la cadena de caracteres ingresada en un arreglo de letras mayúsculas.
59
60     for i in range(len(array1)):
61         j = 0
62
63         #Busca la posición que ocupa una letra en alfabetoLatino.
64         while (array1[i] != alfabetoLatino[j]):
65             j = j + 1
66
67         array2.append(alfabetoCifrado[j]) #Añade un nuevo elemento al arreglo array2.
68
69     mensajeCifrado = listToString(array2) #Convierte el arreglo array2 en una cadena de caracteres.
70     print(mensajeCifrado)
71     print("\nPresione ENTER para continuar")
72     input() #Espera a que el usuario presione ENTER o ingrese algún valor.
73
74     return None
75
76
77 #Código para descifrar un mensaje.
78 def descifrar():
79
```

```
File Edit Selection View Go Run Terminal Help main.py - Codigos Práctica - Visual Studio Code
main.py X cifrado1.0.cpp
Python > main.py > ...
77 #Código para descifrar un mensaje.
78 def descifrar():
79
80     array2 = [] #Arreglo vacío que contendrá el mensaje descifrado.
81
82     #Limita la entrada del usuario a solo números enteros.
83     try:
84         llave = int(input("Ingrese un número entero del 1 al 25: "))
85     except ValueError: #Si el usuario ingresa otro valor que no sea un número entero se ejecuta este bloque.
86         print("\nIngrese un número entero")
87         print("Presione ENTER para continuar")
88         input() #Espera que el usuario presione ENTER o ingrese algún valor para continuar.
89         return None #Se termina anticipadamente la función y se regresa al menú principal.
90
91     if (llave < 1 or llave > 25):
92         print("\nEl valor ingresado no es válido")
93         print("Presione ENTER para continuar")
94         input() #Espera que el usuario presione ENTER o ingrese algún valor para continuar.
95
96     else:
97         alfabetos(llave)
98         mensaje = str(input("Ingrese el mensaje que desea descifrar: "))
99         array1 = list(mensaje.upper()) #Convierte la cadena de caracteres ingresada en un arreglo de letras mayúsculas.
100
101         for i in range(len(array1)):
102             j = 0
103
104             #Busca la posición que ocupa una letra en alfabetoCifrado.
105             while (array1[i] != alfabetoCifrado[j]):
106                 j = j + 1
107
108             array2.append(alfabetoLatino[j]) #Añade un nuevo elemento al arreglo array2.
109
110         mensajeDescifrado = listToString(array2) #Convierte el arreglo array2 en una cadena de caracteres.
111         print(mensajeDescifrado)
112         print("Presione ENTER para continuar")
113         input() #Espera que el usuario presione ENTER o ingrese algún valor para continuar.
114
115     return None
116
```

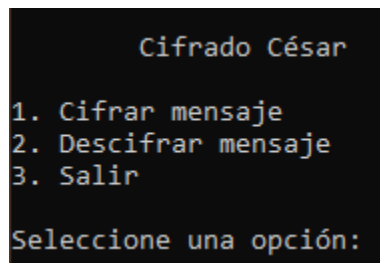
```
File Edit Selection View Go Run Terminal Help main.py - Codigos Práctica - Visual Studio Code
main.py X cifrado1.0.cpp
Python > main.py > ...
117
118 #Ciclo principal
119 while option != 3:
120
121     option = 0
122
123     #Llama a scripts para limpiar la terminal.
124     if name == 'nt':
125         _ = system('cls') #Script para Windows 10.
126
127     else:
128         _ = system('clear') #Script para MacOS y Linux.
129
130     #Muestra el menú de opciones
131     print("\n\tCifrado Caudésar\n")
132     print("1. Cifrar mensaje")
133     print("2. Descifrar mensaje")
134     print("3. Salir\n")
135
136     #Limita la entrada del usuario a solo números enteros.
137     try:
138         option = int(input("Seleccione una opción: "))
139     except ValueError: #Si el usuario ingresa otro valor que no sea un número entero se ejecuta este bloque.
140         print("\nIngrese un número entero")
141
142     #Default
143     if (option < 1 or option > 3):
144         print("\nEl valor ingresado no es válido")
145         print("Presione ENTER para continuar")
146         input() #Espera que el usuario presione ENTER o ingrese algún valor para continuar.
147
148     #Case 1
149     elif (option == 1):
150         cifrar() #Llama a la función cifrar
151
152     #Case 2
153     elif (option == 2):
154         descifrar() #Llama a la función descifrar
```

Funcionamiento, compilación y ejecución – Python

Compilación y Ejecución



Menú principal



Opción 1 - Cifrado del mensaje

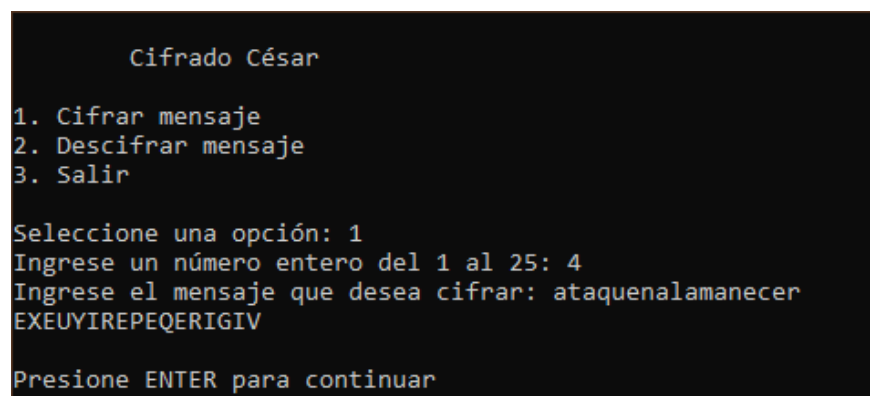
Puede colocar el numero que se desee mover el alfabeto para hacer el Cifrado

Mensaje:

ATAQUENALAMANECER

Cifrado:

EXEUYIREPEQERIGIV



Opción 2 - Descifrado del mensaje

Puede colocar el numero que se desee mover el alfabeto para hacer el Descifrado

Cifrado:

EXEUYIREPEQERIGIV

Descifrado:

ATAQUENALAMANECER

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 2
Ingrese un número entero del 1 al 25: 4
Ingrese el mensaje que desea descifrar: EXEUYIREPEQERIGIV
ATAQUENALAMANECER

Presione ENTER para continuar
```

Opción 3 – Salir

```
Cifrado César

1. Cifrar mensaje
2. Descifrar mensaje
3. Salir

Seleccione una opción: 3_
```

Sale del programa al poner 3

Repositorio GitHub – Python

<https://github.com/1an2l/Estructura-de-Datos-y-Algoritmos-I/tree/main/CÓDIGOS/CIFRADO%20CÉSAR/Python>

Bibliografías:

- Berrondo, R., Cabrera, N., Franco, G., Frederico, M., Mariani, F., & Rodríguez, L. (2015). Criptografía: una cuestión de códigos.

Referencias:

- Microsoft Corporation. (s.f.). Documentos de C++: inicio, tutoriales y referencias. Recuperado el 19 de marzo de 2021, de Microsoft Docs: <https://docs.microsoft.com/es-mx/cpp/cpp/?view=msvc-160>

- Solano, J. A. (25 de enero de 2019). Manual de prácticas del laboratorio de Estructuras de datos y algoritmos I. Recuperado el 19 de marzo de 2021, de Laboratorios Salas A y B: <http://lcp02.fi-b.unam.mx/>
- Víctor Gallego. (2015, 9 diciembre). Como usar la Ñ y acentos c++ [Vídeo]. YouTube. <https://www.youtube.com/watch?v=aW7c7r9ErCg>
- Barrios L., A. (s. f.). Links. Laboratorio de Arte Digital. Recuperado 20 de marzo de 2021, de <http://l-ad.blogspot.com/2016/09/links.html>
- Sanchez., M. (s. f.). Python. GitHub. Recuperado 20 de marzo de 2021, de <https://github.com/marco-sanchez-est/EDAI/blob/main/codigos/cifradoCesar/main.py>
- 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