

T3	Seguridad Informática		7B	E	#2
	#6	Fernando Escamilla Hernández			
	#15	Nancy Paredes Moreno			
	#18	Jahel Pérez Gutiérrez			
	#26	Arturo Serrano Valencia			
	#29	Paul Torres Rivera			
	#31	Sandro Iván Yllescas Lamas			

Métodos de encriptación escritos en Python 2.7.10

AUTHOR
Version 1
CREATEDATE

Tabla de contenidos

Table of contents

Indice de namespaces

Lista de 'namespaces'

Lista de los 'namespaces', con una breve descripción:

cifrados	4
-----------------------	---

Indice de archivos

Lista de archivos

Lista de todos los archivos con descripciones breves:

cifrados.py	10
--------------------------	----

Documentación de namespaces

Referencia del Namespace cifrados

Funciones

- def **main** ()
- def **abrirArchivo** ()
- def **cifrarASCII** ()
- def **grabartxt** (a)
- def **descifrarASCII** ()
- def **grabar2** (linea)
- def **cifrarEscitala** ()
- def **descifrarEscitala** ()
- def **cifrarMorse** ()
- def **verify** (string)
- def **descifrarMorse** ()
- def **cifrarCesar** ()
- def **descifrarCesar** ()

Variables

- string **archivo** = ""
- dictionary **CODE**
- dictionary **CODE2**
- string **abc** = 'abcdefghijklmnopqrstuvwxyz'

Descripción detallada

Variable global que contendra el archivo a trabajar

Documentación de las funciones

def **cifrados.abrirArchivo** ()

```
41 def abrirArchivo():
42     global archivo
43     archivo=raw_input("\n    Ingresa la ruta del archivo: ")
44
```

def **cifrados.cifrarASCII** ()

Se lee el archivo caracter a caracter y se va grabando con el metodo grabartxt

```
45 def cifrarASCII():
46     """Se lee el archivo caracter a caracter y se va grabando con el metodo grabartxt"""
47     archi=open (archivo,'r')
48     linea=archi.read(1)
49     while linea!="":
50         """la funcion ord() convierte a valores acssi"""
51         a = ord(linea)
52         grabartxt(a)
```

```

53     linea=archi.read(1)
54     archi.close()
55     print("\n Se ha terminado de cifrar, revisa el archivo cifradoASCII.txt")
56

```

def cifrados.cifrarCesar ()

```

231 def cifrarCesar():
232     archi=open(archivo,'r')
233     msg=archi.read()
234     c = msg.lower()
235     n = int(raw_input('Ingresa la clave numerica: '))
236
237     text_cifrado = ''
238     archi=open('cifradoCesar.txt','w')
239     for letra in c:
240         suma = abc.find(letra) + n
241         modulo = int(suma) % len(abc)
242         text_cifrado = text_cifrado + str(abc[modulo])
243     archi.write(text_cifrado)
244     archi.close()
245     print("\n Se ha terminado de cifrar, revisa el archivo cifradoCesar.txt");
246

```

def cifrados.cifrarEscitala ()

Se solicitan los valores al usuario

```

88 def cifrarEscitala():
89     """Se solicitan los valores al usuario"""
90     print(" Escriba el numero de largo:")
91     col=input()
92     print(" Escriba el numero de diametro:")
93     row=input()
94     """leemos el archivo"""
95     archi=open(archivo,'r')
96     text=archi.read()
97     """se guarda en una variable string"""
98     archi.close()
99     """se muestra el mensaje a cifrar"""
100    print("\n Mensaje a cifrar: "+text)
101
102    tam=len(text)
103
104    """hacemos el array que usaremos para guardar nuestro cifrado"""
105    cifrado=[]
106    for i in range(tam):
107        cifrado.append(0)
108
109    """hacemos el cifrado"""
110    cont=0
111    for i in range (tam):
112        x=(col*(i%row))+(i/row)
113        """metodo de ordenamiento para el cifrado"""
114        cifrado[cont]=text[x]
115        cont=cont+1
116    """cifrado es tipo list, lo pasamos a string para poder llenar el txt"""
117    cif=''.join(cifrado)
118    print("\nCifrado: "+cif)
119    archi=open('cifradoEscitala.txt','w')
120    archi.write(cif)
121    archi.close()
122

```

def cifrados.cifrarMorse ()

```

154 def cifrarMorse():
155     infile = open(archivo,'r')
156     msg = infile.read(200)
157     verify(msg)

```



```

158     print ('El fichero dice:'+ '\n' + msg + '\n')
159     print " "
160     print "encriptando informacion del fichero..."
161     print " "
162     archi=open('cifradoMorse.txt','w')
163
164     for char in msg:
165         if char == ' ':
166             print ' '*2,
167         else:
168             archi.write(CODE[char.upper()]+ " ")
169
170     archi.close()
171     print("\n Se ha terminado de cifrar, revisa el arhivo cifradoMorse.txt");
172

```

def cifrados.descifrarASCII ()

Aqui se descifra ascii

```

65 def descifrarASCII():
66     """ Aqui se descifra ascii"""
67     global archivo
68     archivo=raw_input("\n Ingrese la ruta del archivo a descifrar: ")
69     archi=open (archivo,'r')
70     linea=archi.read()
71     archi.close()
72     linea=linea.split(" ")
73     grabar2(linea)
74

```

def cifrados.descifrarCesar ()

```

247 def descifrarCesar():
248     global archivo
249     archivo=raw_input("\n Ingrese la ruta del archivo a descifrar: ")
250     cn = int(raw_input('Ingresa la clave numerica: '))
251     text_cifrado = ''
252     archi=open(archivo,'r')
253     msg=archi.read()
254     archi.close()
255     arch=open('mensajeCesar.txt','w')
256     for letra in msg:
257         suma = abc.find(letra) - cn
258         modulo = int(suma) % len(abc)
259         text_cifrado = text_cifrado + str(abc[modulo])
260     arch.write(text_cifrado)
261     arch.close()
262     print("\n Se ha terminado de descifrar, revisa el arhivo mensajeCesar.txt");
263

```

264 main()def cifrados.descifrarEscitala ()

```

123 def descifrarEscitala():
124     global archivo
125     archivo=raw_input("\n Ingrese la ruta del archivo a descifrar: ")
126     archi=open(archivo,'r')
127     text=archi.read()
128     print("\n Cifrado "+text)
129     texto=list(text)
130     print("\n Indique las columnas para descifrar:")
131     col=input()
132     print("\n Indique las filas para descifrar:")
133     row=input()
134     tam=col*row
135
136     """se crea el arreglo donde se escribira el mensaje"""
137     desCifrado=[]
138     for i in range(tam):
139         desCifrado.append(0)

```

```

140
141     cont=0
142     for i in range (tam):
143         x=(row*(i%col))+(i/col)
144         """metodo de ordenamiento para el descifrado"""
145         desCifrado[cont]=texto[x]
146         cont=cont+1
147     mensaje=''.join(desCifrado)
148     print("\n    Mensaje: "+mensaje)
149
150     archi=open('mensajeEscitara.txt','w')
151     archi.write(mensaje)
152     archi.close()
153

```

def cifrados.descifrarMorse ()

```

195 def descifrarMorse():
196     global archivo
197     archivo=raw input("\n    Ingresa la ruta del archivo a descifrar: ")
198     infile = open(archivo,'r')
199     msg = infile.read()
200     morse=msg.split(" ")
201     print ('El fichero dice:')
202     print(morse)
203     print "                                "
204     print "encriptando informacion del fichero..."
205     print "                                "
206     tam=len(morse)-1
207     archi=open('mensajeMorse.txt','w')
208     for i in range(tam):
209         archi.write(CODE2[morse[i]])
210     archi.close()
211     print("\n    Se ha terminado de descifrar, revisa el arhivo mensajeMorse.txt");
212

```

def cifrados.grabar2 (linea)

Aqui se graba el mensaje traducido

```

75 def grabar2(linea):
76     """ Aqui se graba el mensaje traducido"""
77     tam=len(linea)-1
78     archi=open ('mensajeASCII.txt','w')
79     for i in range(tam):
80         """se convierte a int porque viene de un string"""
81         a=int(linea[i])
82         """y de int convertimos a el valor letra, simobolo o numero"""
83         s=chr(a)
84         archi.write(s)
85     archi.close()
86     print("\n    Se ha terminado de descifrar, revisa el arhivo mensajeASCII.txt");
87

```

def cifrados.grabartxt (a)

Aqui se va escribiendo en el archivo txt creado

```

57 def grabartxt(a):
58     """Aqui se va escribiendo en el archivo txt creado"""
59     archi=open ('cifradoASCII.txt','a')
60     """convertimos a string con str() porque write solo escribe strings"""
61     b = str(a)
62     archi.write(b+" ")
63     archi.close()
64

```

def cifrados.main ()

Menu de opciones de todo el programa

```
4 def main():
5     """Menu de opciones de todo el programa"""
6     opc=0;
7     while opc!=10:
8         print("\nMenu\n")
9         print("1.- Ingresar un archivo .txt a descifrar o cifrar\n")
10        print("2.- Cifrado Morse      6.- Descifrar Morse\n")
11        print("3.- Cifrado ASCII      7.- Descifrar ASCII\n")
12        print("4.- Cifrado Cesar      8.- Descifrar Cesar\n")
13        print("5.- Cifrado Escitalla    9.- Descifrar Escitalla\n")
14        print("10.- Salir\n")
15        print("Opcion:")
16        opc=input()
17        if opc == 1:
18            abrirArchivo()
19        elif opc == 2:
20            cifrarMorse()
21        elif opc == 3:
22            cifrarASCII()
23        elif opc == 4:
24            cifrarCesar()
25        elif opc == 5:
26            cifrarEscitalla()
27        elif opc == 6:
28            descifrarMorse()
29        elif opc == 7:
30            descifrarASCII()
31        elif opc == 8:
32            descifrarCesar()
33        elif opc == 9:
34            descifrarEscitalla()
35        elif opc == 10:
36            print("\n      Adios!!!")
37        else:
38            opc=10;
39            print("\n      Adios!!!")
40
```

def cifrados.verify (string)

```
173 def verify(string):
174     keys = CODE.keys()
175     for char in string:
176         if char.upper() not in keys and char != ' ':
177             sys.exit('OJO con el caracter ' + char + ' no puede ser traducido a clave
178             morse en su lugar use espacio.')
179             raise BaseException(format(char))
179
```

Documentación de las variables

string cifrados.abc = 'abcdefghijklmnopqrstuvwxyz'

string cifrados.archivo = ""

dictionary cifrados.CODE

```
Valor inicial: 1 = {'A': '.-', 'B': '-...', 'C': '-.-.',
2               'D': '-..', 'E': '.', 'F': '..-.',
3               'G': '--.', 'H': '....', 'I': '.....',
4               'J': '-.-.-', 'K': '-.-', 'L': '-.-.',
5               'M': '--', 'N': '-.', 'O': '---',
```

```

6      'P': '.--.', 'Q': '-.-.-', 'R': '.--.',
7      'S': '...-', 'T': '-.-', 'U': '.--.-',
8      'V': '...--', 'W': '.--.-', 'X': '-.-.-',
9      'Y': '-.-.-', 'Z': '-.-.-',
10     '0': '-----', '1': '.-----', '2': '.-----',
11     '3': '---.-', '4': '----.-', '5': '-----',
12     '6': '-....', '7': '-....', '8': '-....',
13     '9': '-----',
14     }

```

dictionary cifrados.CODE2

```

Valor inicial: 1 = {'.-': 'A', '-...': 'B', '-.-.': 'C',
2      '-..': 'D', '.': 'E', '.--': 'F',
3      '-.-': 'G', '....': 'H', '.': 'I',
4      '.---': 'J', '-.-': 'K', '.--': 'L',
5      '-.': 'M', '-.': 'N', '---': 'O',
6      '.--': 'P', '-.-': 'Q', '.-': 'R',
7      '...': 'S', '-': 'T', '.--': 'U',
8      '.--': 'V', '.--': 'W', '-.-': 'X',
9      '-.-': 'Y', '-.-': 'Z',
10     '-----': '0', '-----': '1', '-----': '2',
11     '---.-': '3', '----.-': '4', '-----': '5',
12     '-....': '6', '-....': '7', '-....': '8',
13     '-----': '9'
14     }

```

Documentación de archivos

Referencia del Archivo cifrados.py

Namespaces

- `cifrados`

Funciones

- `def cifrados.main ()`
- `def cifrados.abrirArchivo ()`
- `def cifrados.cifrarASCII ()`
- `def cifrados.grabartxt (a)`
- `def cifrados.descifrarASCII ()`
- `def cifrados.grabar2 (linea)`
- `def cifrados.cifrarEscitola ()`
- `def cifrados.descifrarEscitola ()`
- `def cifrados.cifrarMorse ()`
- `def cifrados.verify (string)`
- `def cifrados.descifrarMorse ()`
- `def cifrados.cifrarCesar ()`
- `def cifrados.descifrarCesar ()`

Variables

- string `cifrados.archivo = ""`
- dictionary `cifrados.CODE`
- dictionary `cifrados.CODE2`
- string `cifrados.abc = 'abcdefghijklmnopqrstuvwxyz'`

Índice

INDEX