



## **RESOLUCION DE LA HOJA DE EJERCICIOS 7**

**Escuela Profesional:** Ingeniería Informática y de sistemas

**Asignatura:** Algoritmos Avanzados

**Docente:** Ing. Hector Eduardo Ugarte Rojas

**Alumno:** Benjamin Alexander Hualverde Quispe

**Codigo:** 161367

**Semestre:** 2021\_I

**Cusco- Perú**

**2021**

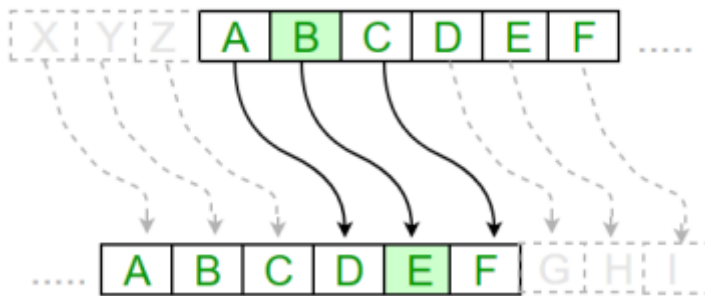
## HOJA DE EJERCICIOS 7 – ALGORITMOS AVANZADOS UNSAAC

### Ejercicio 1.

En el cifrado Cesar, la ofuscación puede ser representada usando aritmética modular, transformando primeramente la letra en números de acuerdo a su posición. En inglés A=0, B=1,..., Z=25. La ofuscación de una letra mediante una variante  $n$  puede ser descrita matemáticamente como:

Encriptación:  $E_n(x) = (x + n) \bmod 26$

Desencriptación:  $D_n(x) = (x - n) \bmod 26$



El siguiente código de Python implementa dicha encriptación:

```
def encriptar(texto,s):
    resultado = ""

    # Traversar texto
    for i in range(len(texto)):
        char = texto[i]

        # Encriptar caracteres mayuscula
        if (char.isupper()):
            resultado += chr((ord(char) + s-65) % 26 + 65)

        # Encriptar caracteres minuscula
        else:
            resultado += chr((ord(char) + s - 97) % 26 + 97)

    return resultado

#Ejemplo
texto = "LOS ALUMNOS DE ALGORITMOS AVANZADOS SON MUY RESPONSABLES"
s = 4
print("Texto          : " + texto)
print("Variante       : " + str(s))
print("Texto cifrado  : " + encriptar(texto,s))
```

#### A. Verifica el correcto funcionamiento de algoritmo anterior.

Ejecutando el código anterior, se obtiene el texto cifrado

```
>>>
= RESTART: C:\Users\Usuario\Desktop\Hoja Ejercicios 7\1.a. AlgoritmoCesar(Encrip
tar).py
Texto          : LOS ALUMNOS DE ALGORITMOS AVANZADOS SON MUY RESPONSABLES
Variante       : 4
Texto cifrado: PSWrEPYQrSWrHlrEPKSVMXQSWrEZErDEHSWrWSRrQYCrVIWTSrRWFEPiW
>>>
```

- B. De manera similar, implementa un método de descryptación (Codigo) que liste todas las posibles salidas para el texto

ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT  
MNKT

El método para descryptar es el siguiente:

```
def descryptar(texto,n):
    resultado=""
    #Traversar texto
    for i in range(len(texto)):
        char=texto[i]

        #Descryptar caracteres mayuscula
        if(char.isupper()):
            resultado+=chr((ord(char)-n-65)%26+65)

        #Descryptar caracteres minuscula
        else:
            resultado+=chr((ord(char)-n-97)%26+97)
    return resultado
```

Descryptamos el texto para cada variante de n(0-25):

#Programa Principal

#Ejercicio

#Descryptar el texto ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT MNKT

textCifrado="ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT MNKT"

```
for s in range(26):
    print("-----")
    print("Texto cifrado      : "+textCifrado)
    print("Variante             : "+str(s))
    print("Texto descifrado     : "+descryptar(textCifrado,s))
    print("-----")
```

Los resultados, son los siguientes:

```
= RESTART: C:\Users\Usuario\Desktop\Hoja Ejercicios 7\1.b. AlgoritmoCesar (Descryptar).py
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT MNKT
Variante           : 0
Texto descifrado   : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT MNKT
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT MNKT
Variante           : 1
Texto descifrado   : DSfJWSDARSUAGffVWdfHJGQWULGfXAFSdfWKfNALSDfHSJSfSHJGTSJfDSfSKAYfSLMJS
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGT MNKT
Variante           : 2
```

```
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 2
Texto descifrado   : CReIVRCZQRTZFEeUVCeGIFPVTkFeWZERCeVJeMZKRCeGRIReRGIFSRieCReRJZXERKLIR
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 3
Texto descifrado   : BQdHUQBYPQSYEDdTUBdFHEOUSJEdVYDQBdUIDLYJBdFQHQdQFHERQHdBQdQIYWDQJKHQ
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 4
Texto descifrado   : APcGTPAXOPRXDCcSTAcEGDNTRIDcUXCPAcThcKXIPAcEPGPcPEGDQPGcAPcPHXVCPIJGP
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 5
Texto descifrado   : ZObFSOZWNOQWCBBRSZbDFCMSQHcbTWBOZbSGbJWHOZbDOFobODFCPOfbZObOGWUBOHIFO
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 6
Texto descifrado   : YNaERNYVMNPVBAAQRyACEBLRPGBaSVANYaRfFaIVGNyACNENaNCeBONEaYNaNFVTANGHEN
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 7
Texto descifrado   : XMzDQMXULMOUAzZpQXzBDAKQOFaZRUZMXzQEzHUFMXzBMDMzMBDANMDzXMzMEUSZMFGDM
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 8
Texto descifrado   : WLyCPLWTKLNTZyYOPWyACZJPNEZyQTYLWyPDyGTELWyALCLyLACZMLCyWLyLDTRYLEFCL
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 9
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 9
Texto descifrado   : VKxBOKVSJKMSYXxNOVxZBYIOMDyXPSXKVxOCxFSDKVxZKBKxKZBYLKBxVKxKCSQXKDEBK
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 10
Texto descifrado   : UJwANJURIJLRXWwMNUwYAXHNLCxORWJUwNBwERCJUwYJAJwJYAXKJAwUJwJBRPWJCDAJ
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 11
Texto descifrado   : TivZMITQHlQWVvLMTvXZWMKKBWvNQVITvMAvDQBITvXIZivIXZWJizvTivIAQOVIBCZI
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 12
Texto descifrado   : SHuYLHSPGHJPVUuKLSuWYVFLJAVuMPUHSuLzuCPAHSuWHYHuHWYVIHYuSHuHZPNUHABYH
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 13
Texto descifrado   : RgtXKGROFGIOUTtJKRtVXUEKIZUtLOTGRtKYtBOZGRtVGXGtGVXUHGXtRGtGYOMTGZAXG
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 14
Texto descifrado   : QFsWJFQNEFHNTSsIJQsUWTDJHYTsKNSFQsJXsANYFQsUFWFsFUWTFWGsQFsFXNLSFYZWf
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 15
Texto descifrado   : PErVIEPMDEGMSRrHIPrTVSCIGXSrJMREPrIWZrZMXEPrTEVErETVSFEVrPErEWMKREXYVE
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEGIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 16
```

```

Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 16
Texto descifrado   : ODqUHDOLCDFLRQqGHOqSURBHFWRqILQDOqHVqYLWDOqSDUDqDSUREDUDqODqDVLJQDWXUD
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 17
Texto descifrado   : NCpTGCNKBCeKQPpFGNpRTOAGEVQpHKPCNPgUpXKVCNPgRCTCpCRTQDCTpNCpCUKIPCWVTC
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 18
Texto descifrado   : MBoSFBMJABDJPOoEFMOQSPZFDUPoGJOBMoFToWJUBMoQBSBoBQSPCBSoMBoBTJHOBUVSB
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 19
Texto descifrado   : LAnREALIZACIONnDeLnPROYECTOnFINALnESnVITALnPARAnAPROBARnLAnASIGNATURA
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 20
Texto descifrado   : KZmQDZKHYZBHNmCDKmOQNxDBSNmEHMZKmDRmUHSZKmOZOZmZOQNAZQmKZmZRHFMZSTQZ
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 21
Texto descifrado   : JYlPCYJGXyAGMLlBCJlNPMWCARmlDGLYJlCQlTGRYJlNYPYlYNPMZYPlJYlYQGElyRSPY
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 22
Texto descifrado   : IXkOBXIFWXZFLKkABIkMOLVBZQLkCFKXIkBPkSFQXIkMXOXkXMOLYXOkIXkXPFDKXQROX
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 23
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 23
Texto descifrado   : HWjNAWHEVWEKJjZAHjLNKUAYPKjBEJWHjAojREPWHjLWNWjWLNKXWnjHWjWOECJWPQNW
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 24
Texto descifrado   : GVimZVGDUVDJiIYZGiKMJTZXOJiADIVGiZNiQDOVGiKVMViVKMJWVMiGViVNBiVOPMV
-----
Texto cifrado      : ETgKXTEBSTVBHGgWXEgIKHRXVMHgYBGTEgXLgOBMTegITKTgTIKHUTKgETgTLBZGTMNKT
Variante          : 25
Texto descifrado   : FUhLYUFCTUWCiHhXYFhJLISYWNihZCHUFhYMHPCNUFhJULUUhJULIVLhFUhUMCAHUNOLU
-----
>>> |

```

### C. ¿Cuál es el mensaje que fue ofuscado?

En las salidas obtenidas se puede observar que hay un resultado cuyo resultado tiene sentido y es cuando la variante es 19, el mensaje ofuscado es:

**“ LA REALIZACION DEL PROYECTO FINAL ES VITAL PARA APROBAR LA ASIGNATURA ”**

### Ejercicio 2.

La escítala se puede representar como una matriz de caracteres donde el grosor o clave es el número de filas. El mensaje cifrado será la matriz leída por filas.



plaintext	E	L	P	E	R	R	O	L	A	D	R	A	P	O	C	O	
1	E		E				O			D			P		O		=EEO DPO
2	L			R				L			R			O			=LRLRO
3		P			R				A			A			C		=PRAAC
ciphertext	E	E	O	D	P	O	L	R	L	R	O	P	R	A	A	C	

**A. Cifra el texto plano: “TRES TRISTES TIGRES TRAGAN TIGRO” con clave 4, muestra la matriz y el texto cifrado.**

La matriz para cifrar el texto, es:

Texto Plano	T	R	E	S	T	R	I	S	T	E	S	T	I	G	R	E	S	T	R	A	G	A	N	T	R	I	G	O		
1	T				T				T				I				S				G				R				=	TTTISGR
2		R				R				E				G				T				A				I			=	RREGTAI
3			E				I			S					R				R					N			G		=	EISRRNG
4				S				S				T				E				A					T			O	=	SSTEATO
Texto Cifrado	T	T	T	I	S	G	R	R	E	G	T	A	I	E	I	S	R	R	N	G	S	S	T	E	A	T	O			

El texto cifrado del texto plano, es:

**“TTTISGRRRREGTAIEISRRNGSSTEATO”**

**B. Implementa en algún lenguaje de programación un método para encriptar de la manera explicada.**

El algoritmo para encriptar, es:

```
#Algoritmo para encriptar Texto Plano(Escitala)
def encriptarEscitala(texto,clave):
    #Eliminamos los espacios en blanco
    texto=texto.replace(" ","")
    #Dividimos la cadena en clave-partes
    if(len(texto)%clave==0):
        NroSubCadenas=len(texto)//clave
    else:#se rellena con espacios en blanco
        NroSubCadenas=(len(texto)//clave)+1
        texto=texto+(" "*(clave-(len(texto)%clave)))
    #Encriptamos
    TextoCifrado=""
    for i in range(clave):
        SubCadena=""
        for j in range(NroSubCadenas):
            SubCadena=SubCadena+texto[(j*clave):((j*clave)+clave)][i]

        TextoCifrado=TextoCifrado+SubCadena

    #Retornamos el texto cifrado eliminando los caracteres en blanco si en
    #Caso se le agregaron
    return TextoCifrado.replace(" ","")
```

A continuación algunos ejemplos:

```
#Ejemplo1:
print("-----")
textoPlano1="TRES TRISTES TIGRES TRAGAN TRIGO"
textCifrado1=encriptarEscitala(textoPlano1,4)
print("Texto Plano =",textoPlano1)
print("Texto Cifrado =",textCifrado1)
print("-----")

#Ejemplo2:
print("-----")
textoPlano2="TRES TRISTES TIGRES TRAGAN TRIGO EN El TRIGAL"
textCifrado2=encriptarEscitala(textoPlano2,4)
print("Texto Plano =",textoPlano2)
print("Texto Cifrado =",textCifrado2)
print("-----")
```

Los resultados son

```

===== RESTART: C:/Users/Usuario/Desktop/EscitalaCriptogra
-----
Texto Plano = TRES TRISTES TIGRES TRAGAN TRIGO
Texto Cifrado = TTTISGRRREGTAIEISRRNGSSTEATO
-----
-----
Texto Plano = TRES TRISTES TIGRES TRAGAN TRIGO EN EL TRIGAL
Texto Cifrado = TTTISGRE TARREGTAINRLEISRRNGEISSTEATOlg
-----

```

**C. Implementa en algún lenguaje de programación un método para descifrar de la manera explicada.**

El algoritmo para descifrar un texto cifrado, es:

```

#Algoritmo para descifrar un texto cifrado
def descifrarEscitala(texto, clave):
    if(len(texto)%clave==0):
        tamaño=len(texto)//clave

        TextoDescifrado=""
        for i in range(tamaño):
            SubCadena=""
            for j in range(clave):
                SubCadena=SubCadena+texto[(j*tamaño):(j*tamaño)+tamaño][i]
            TextoDescifrado=TextoDescifrado+SubCadena

    else:
        tamaño=(len(texto)//clave)+1
        Arreglo=[]
        #Agregamos un pedazo del texto a la matriz:
        for x in range(len(texto)%clave):
            Arreglo.append(texto[(x*tamaño):(x*tamaño)+tamaño])

        nuevoTexto=texto[(x+1)*tamaño::]
        #Agregamos el texto rstante a la matriz:
        for y in range(clave-(len(texto)%clave)):
            Arreglo.append(nuevoTexto[(y*(tamaño-1)):(y*(tamaño-1)+(tamaño-1))]+" ")

        #Desencirptamos el mensaje:
        TextoDescifrado=""
        for i in range(tamaño):
            SubCadena=""
            for j in range(clave):
                SubCadena=SubCadena+Arreglo[j][i]
            TextoDescifrado=TextoDescifrado+SubCadena

    return TextoDescifrado.replace(" ", "")

```

A continuación algunos ejemplos:



```

print("----- Cifrado -----")
textoPlano1="TRES TRISTES TIGRES TRAGAN TRIGO"
clave=4
textCifrado1=encriptarEscitara(textoPlano1,clave)
print("Texto Plano =",textoPlano1)
print("Clave=",clave)
print("Texto Cifrado =",textCifrado1)
print("-----")

print("----- Descifrado -----")
clave=4
textdesCifrado1=desencriptarEscitara(textCifrado1,clave)
print("Texto Cifrado =",textCifrado1)
print("Clave=",clave)
print("Texto DesCifrado =",textdesCifrado1)
print("-----")

#Ejemplo2:
print()
print("===== Ejemplo 2 =====")
print("----- Cifrado -----")
clave=4
textoPlano2="TRES TRISTES TIGRES TRAGAN TRIGO EN El TRIGAL"
textCifrado2=encriptarEscitara(textoPlano2,clave)
print("Texto Plano =",textoPlano2)
print("Clave=",clave)
print("Texto Cifrado =",textCifrado2)
print("-----")

print("----- Descifrado -----")
clave=4
textdesCifrado2=desencriptarEscitara(textCifrado2,clave)
print("Texto Cifrado =",textCifrado2)
print("Clave=",clave)
print("Texto DesCifrado =",textdesCifrado2)
print("-----")

```

Los resultados, son:

```

>>>
===== RESTART: C:\Users\Usuario\Desktop\EscitaraCriptografia.py =====
===== Ejemplo 1 =====
----- Cifrado -----
Texto Plano = TRES TRISTES TIGRES TRAGAN TRIGO
Clave= 4
Texto Cifrado = TTTISGRRREGTAIEISRRNGSSTEATO
-----
----- Descifrado -----
Texto Cifrado = TTTISGRRREGTAIEISRRNGSSTEATO
Clave= 4
Texto DesCifrado = TRESTRISTESTIGRESTRAGANTRIGO
-----

===== Ejemplo 2 =====
----- Cifrado -----
Texto Plano = TRES TRISTES TIGRES TRAGAN TRIGO EN El TRIGAL
Clave= 4
Texto Cifrado = TTTISGRE TARREGTAINRLEISRRNGEISSTEATOLG
-----
----- Descifrado -----
Texto Cifrado = TTTISGRE TARREGTAINRLEISRRNGEISSTEATOLG
Clave= 4
Texto DesCifrado = TRESTRISTESTIGRESTRAGANTRIGOENElTRIGAL
-----
>>>

```



### Ejercicio 3

Con esteganografía, podemos ocultar datos secretos en Imágenes, Video y Audio, de tal manera que al ver dicha media no parezca que se oculte algo. Consideremos una imagen de 4 x 3 pixeles.

Recordar que cada pixel esta representado por 3 bytes en la codificación RGB, Entonces nuestra imagen tendría 36 bytes:

[(27, 64, 164), (248, 244, 194), (174, 246, 250), (149, 95, 232),  
(188, 156, 169), (71, 167, 127), (132, 173, 97), (113, 69, 206),  
(255, 29, 213), (53, 153, 220), (246, 225, 229), (142, 82, 175)]

En la codificación ASCII cada carácter tiene tamaño de 1 byte = 8 bits. Por ejemplo, el mensaje "Hii" tiene como tamaño 3 bytes. El valor ASCII de H es 72, y su representación en 8 bits es: 01001000.

Podemos ocultar cada bit, en cada byte de nuestra imagen original. Cambiemos cada byte a par si se trata de un 0 y cambiemos a impar si se trata de un 1.

Entonces nuestros primeros 9 bytes quedarían:

(27, 64, 164), (248, 244, 194), (174, 246, 250).  
(26, 63, 164), (248, 243, 194), (174, 246, 250).

Notar que solo estamos usando los 8 primeros bytes, el 9 byte lo utilizaremos para indicar que todavía tenemos mas caracteres por codificar. Este será par si aun tenemos mas caracteres, caso contrario será impar.

Entonces continuando el mismo procedimiento con las "ii" restantes, cuyo valor ASCII es 105, tendríamos.

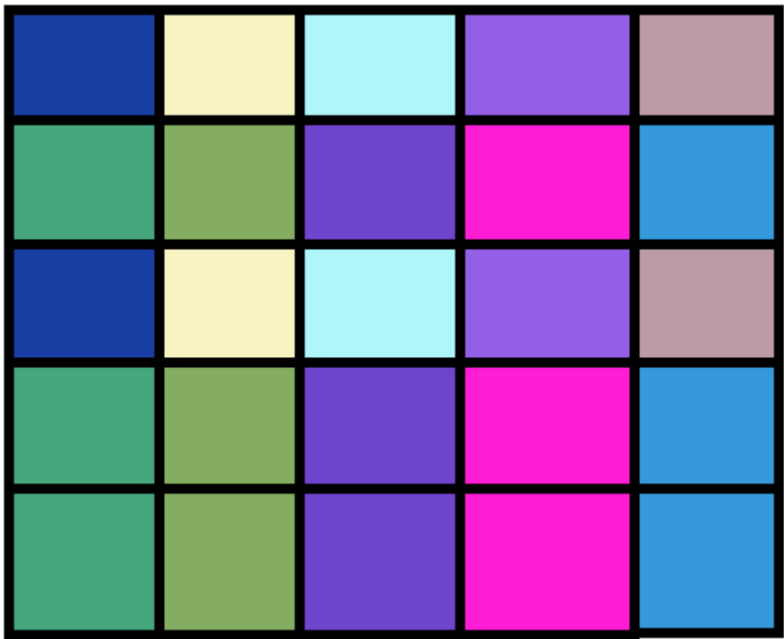
[(26, 63, 164), (248, 243, 194), (174, 246, 250), (148, 95, 231),  
(188, 155, 168), (70, 167, 126), (132, 173, 97), (112, 69, 206),  
(254, 29, 213), (53, 153, 220), (246, 225, 229), (142, 82, 175)]

Notar que para codificar "Hii" se utilizo 27 bytes, y que el ultimo byte es impar (213) indicando el final del ocultamiento.

**A. En la siguiente imagen de 5 x 5 pixeles, codifica el mensaje "UNSAAC" con el procedimiento antes descrito, muestra el resultado.**

[(26, 63, 164), (248, 243, 194), (174, 246, 250), (148, 95, 231), (188, 155, 168),  
(70, 167, 126), (132, 173, 97), (112, 69, 206), (254, 29, 213), (53, 153, 220),  
(26, 63, 164), (248, 243, 194), (174, 246, 250), (148, 95, 231), (188, 155, 168),  
(70, 167, 126), (132, 173, 97), (112, 69, 206), (254, 29, 213), (53, 153, 220),  
(70, 167, 126), (132, 173, 97), (112, 69, 206), (254, 29, 213), (53, 153, 220)]

Graficando el conjunto de pixeles 5x5, es:



Hallamos la representación de 8 bits de cada carácter:

Carácter	ASCII	Byte
U	85	0101 0101
N	78	0100 1110
S	83	0101 0011
A	65	0100 0001
A	65	0100 0001
C	67	0100 0011

Los nuevos bytes de la imagen 5x5 , ahora es:

**-Para la letra U:**

(26, 63, 164), (248, 243, 194), (174, 246, 250)

(26, 63, 164), (247, 242, 193), (174, 245, 250)

**-Para la letra N:**

(148, 95, 231), (188, 155, 168), (70, 167, 126)

(148, 95, 230), (188, 155, 167), (69, 166, 126)

**-Para la letra S:**

(132, 173, 97), (112, 69, 206), (254, 29, 213)

(132, 173, 96), (111, 68, 206), (253, 29, 212)

**-Para la letra A:**

(53, 153, 220), (26, 63, 164), (248, 243, 194)

(52, 153, 220), (26, 62, 164), (248, 243, 194)

**-Para la letra A:**

(174, 246, 250), (148, 95, 231), (188, 155, 168)

(174, 245, 250), (148, 94, 230), (188, 155, 168)

**-Para la letra C:**

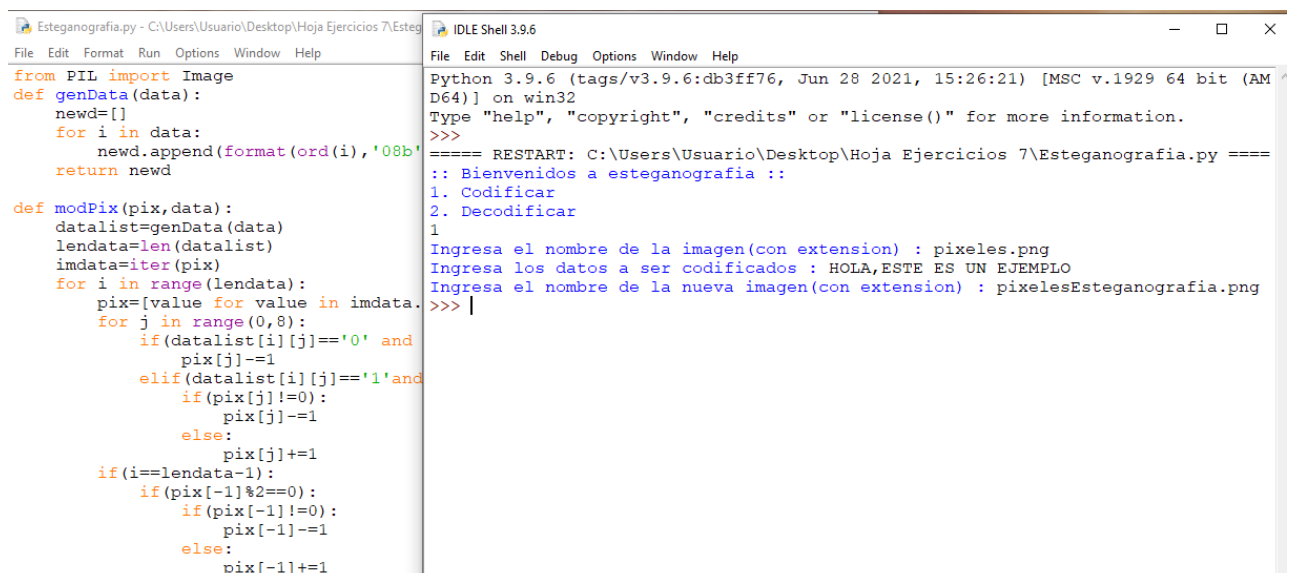
(70, 167, 126), (132, 173, 97), (112, 69, 206)

(70, 167, 126), (132, 172, 96), (111, 69, 205)

El conjunto de pixeles 5x5 resultaría:

[(26, 63, 164), (247, 242, 193), (174, 245, 250), (148, 95, 230), (188, 155, 167),  
(69, 166, 126), (132, 173, 96), (111, 68, 206), (253, 29, 212), (52, 153, 220),  
(26, 62, 164), (248, 243, 194), (174, 245, 250), (148, 94, 230), (188, 155, 168)  
(70, 167, 126), (132, 172, 96), (111, 69, 205), (254, 29, 213), (53, 153, 220),  
(70, 167, 126), (132, 173, 97), (112, 69, 206), (254, 29, 213), (53, 153, 220)]

**B. Ejecuta, entiende y comenta el siguiente código que implementa lo visto anteriormente.**



```
from PIL import Image
def genData(data):
    newd=[]
    for i in data:
        newd.append(format(ord(i), '08b'))
    return newd

def modPix(pix,data):
    datalist=genData(data)
    lendata=len(datalist)
    imdata=iter(pix)
    for i in range(lendata):
        pix=[value for value in imdata]
        for j in range(0,8):
            if(datalist[i][j]=='0' and
               pix[j]==1):
                pix[j]=0
            elif(datalist[i][j]=='1' and
                 pix[j]==0):
                pix[j]=1
            else:
                pix[j]=pix[j]
        if(i==lendata-1):
            if(pix[-1]>255):
                pix[-1]=255
            else:
                pix[-1]=pix[-1]
        else:
            pix[-1]=pix[-1]+1
```

Y como resultado, se obtiene:

1.a. AlgoritmoCesar(Encriptar)	17/08/2021 11:05	Python File	1 KB
1.b. AlgoritmoCesar(Desencriptar)	17/08/2021 11:30	Python File	2 KB
EscitalaCriptografia	22/08/2021 15:42	Python File	4 KB
Esteganografia	22/08/2021 13:25	Python File	3 KB
Informe Hoja Ejercicios 7	22/08/2021 15:50	Documento de Mi...	787 KB
pixeles	22/08/2021 12:56	Archivo PNG	14 KB
pixelesEsteganografia	22/08/2021 15:54	Archivo PNG	4 KB

Y para decodificar el texto que se almaceno en la imagen, el resultado que se obtiene, es:

```
==== RESTART: C:\Users\Usuario\Desktop\Hoja Ejercicios 7\Esteganografia.py ====
:: Bienvenidos a esteganografia ::
1. Codificar
2. Decodificar
2
Ingrese el nombre de la imagen(con extension): pixelesEsteganografia.png
Palabra decodificada : HOLA,ESTE ES UN EJEMPLO
>>> |
```