

# Resultados De Ejecución

---

**Tema:** Técnicas de Cifrado y Permutación

## 1. Introducción

Este documento contiene la ejecución de los 6 algoritmos que describen diferentes técnicas para la Encriptación.

## 2. Permutaciones de Caracteres (Anagramas)

Lenguaje: Python

Este algoritmo genera todas las permutaciones posibles de una palabra ingresada por el usuario. La palabra ingresada será HOLA.

```
Algoritmo1.py > ...
1  from itertools import permutations
2
   Tabnine | Edit | Test | Explain | Document
3  def generar_permutaciones(palabra):
4      palabra = palabra.strip()
5      permutaciones = sorted(set([''.join(p) for p in permutations(palabra)]))
6      total = len(permutaciones)
7
8      print(f"Palabra ingresada: {palabra}")
9
PROBLEMAS  SALIDA  TERMINAL  PUERTOS  CONSOLA DE DEPURACIÓN  Pyth

Primeras 10 permutaciones ordenadas alfabéticamente:
Halo
Hao1
Hlao
Hloa
Hoal
Hola
aHlo
aHol
alHo
aloH
```

### 3. Cifrado por Permutación de Filas

Lenguaje: Python

Este algoritmo organiza los caracteres en una matriz por filas y luego lee la matriz en orden de filas.

La palabra que vamos a ingresar es: BIENVENIDOACLASE, la cual encaja perfectamente en una matriz 4x4

```
1 def cifrado_permutacion_filas():
2     n = int(input("Ingrese el número de filas (clave): "))
3     mensaje = input("Ingrese el mensaje a cifrar: ").replace(" ", "")
4
5     longitud_maxima = n * n
6
7     if len(mensaje) > longitud_maxima:
8         print(f"El mensaje es demasiado largo. Máximo permitido: {longitud_maxima} caractere
9         return
10
```

#### 4. Cifrado por Permutación de Columnas

Lenguaje: Python y Java

Este algoritmo organiza los caracteres en una matriz por filas y luego lee la matriz en orden de columnas.

```

1 def cifrado_columnas(mensaje, n):
2     mensaje = mensaje.replace(" ", "")
3     longitud = len(mensaje)
4     tamaño = n * n
5
6     if longitud > tamaño:
7         print("El mensaje es muy largo para la matriz.")
8         return
9
10

```

PROBLEMAS

SALIDA

TERMINAL

PUERTOS

CONSOLA DE DEPURACIÓN

```

Ingrese el mensaje: DECRYPTOGRAFIA
Ingrese el número de columnas: 4
Matriz de cifrado:
D E C R
I P T O
G R A F
I A * *
Mensaje original: DECRYPTOGRAFIA**
Mensaje cifrado: DTGTEPRACTA*ROF*

```

## 5. Cifrado por Sustitución Monoalfabética (César)

Este algoritmo desplaza cada letra del alfabeto un número fijo de posiciones.

```
Algoritmo4.py > cifrado_desplazamiento
2 def cifrado_desplazamiento(texto, n):
3     alfabeto = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
4     texto = texto.upper()
5
6     alfabeto_cifrado = alfabeto[n % 26:] + alfabeto[:n % 26]
7     sustitucion = {original: cifrada for original, cifrada in zip(alfabeto, alfabeto_cifrado)}
8
9     resultado = ''
10    for letra in texto:
11        if letra in sustitucion:
12            resultado += sustitucion[letra]
13
PROBLEMAS SALIDA TERMINAL PUERTOS CONSOLA DE DEPURACIÓN
```

```
sers/Andres/Desktop/Entregable/Algoritmo4.py
Ingrese la cadena a cifrar: CIFRADODEUNMENSAJE
Ingrese el valor de desplazamiento: 3
Alfabeto original: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alfabeto cifrado : DEFGHIJKLMNOPQRSTUVWXYZABC
Texto original   : CIFRADODEUNMENSAJE
Texto cifrado    : FLIUDGRGHXQPHQVDMH
```

## 6. Cifrado por Sustitución Polialfabética (Vigenère)

Lenguaje: Python

Este algoritmo cifra cada letra del mensaje con una letra distinta, de acuerdo a una clave que se repite.

Palabra: ALGORITMO

Clave: JADEJADE (La clave tiene que tener la misma extensión que la palabra)

```
Algoritmo5.py > cifrado_vigenere
1 def cifrado_vigenere(mensaje, clave):
2     alfabeto = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
3     mensaje = mensaje.upper().replace(" ", "")
4     clave = clave.upper().replace(" ", "")
5
6     clave_repetida = (clave * (len(mensaje) // len(clave) + 1))[:len(mensaje)]
7
8     cifrado = ''
9     for i in range(len(mensaje)):
10         if mensaje[i] in alfabeto:
11             m = alfabeto.index(mensaje[i])
            c = alfabeto.index(clave_repetida[i])
            n = (m + c) % len(alfabeto)
            cifrado += alfabeto[n]
        else:
            cifrado += mensaje[i]
    return cifrado

PROBLEMAS SALIDA TERMINAL PUERTOS CONSOLA DE DEPURACIÓN Python
```

Ingrese el mensaje a cifrar: ALGORITMO  
Ingrese la clave: JADEJADEJ  
Mensaje original : ALGORITMO  
Clave repetida : JADEJADEJ  
Mensaje cifrado : JLJSAIWQX

## 7. Cifrado con Tabla de Coordenadas

Lenguaje: Python

Este algoritmo reemplaza cada letra del mensaje por su posición (fila, columna) en una tabla predefinida.

Palabra: CRIPTOGRUPOCUATRO

```
Algoritmo6.py > crear_matriz
1 def crear_matriz():
2     filas = ['Q', 'W', 'E', 'R', 'T']
3     columnas = ['A', 'S', 'D', 'F', 'G']
4     letras = [chr(i) for i in range(ord('a'), ord('z') + 1) if chr(i) != 'w']
5
6     matriz = {}
7     index = 0
8     for f in filas:
9         for c in columnas:
10             if index < len(letras):
11                 matriz[f+c] = letras[index]
12                 index += 1
13
14     return matriz

PROBLEMAS SALIDA TERMINAL PUERTOS CONSOLA DE DEPURACIÓN Python
```

sers/Andres/Desktop/Entregable/Algoritmo6.py

	A	S	D	F	G
Q	a	b	c	d	e
W	f	g	h	i	j
E	k	l	m	n	o
R	p	q	r	s	t
T	u	v	x	y	z

Ingrese el mensaje a cifrar: CRIPTOGRUPOCUATRO

Mensaje original : CRIPTOGRUPOCUATRO  
Mensaje cifrado : QDRDWFRARGEGWSRDTARAEGQDTAQARGRDEG