



**Universidad Central del Ecuador**  
**Criptografía**  
**Ing. Giovanni Moncayo**

**Integrantes:**

Coloma Pozo Dillan Renato  
Coyago Reinoso Henry Israel  
Estrella Caicedo Juan Carlos  
Llumiquinga Molina Daniel Bernardo  
Ortega Salinas Jimmy Fabricio

**Tarea:**

1. Algoritmo que escriba **todas las permutaciones posibles de una palabra de longitud n SIN espacios (Anagrama)**. La palabra se ingresa al iniciar el algoritmo. El algoritmo debe mostrar el número total de permutaciones y las 10 primeras ordenadas alfabéticamente.

**Código Fuente:**

```
package algoritmos;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class AlgoritmoPermutacion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese una palabra: ");
        String palabra = scanner.nextLine();

        ArrayList<String> permutaciones = permutarPalabra(palabra);
        Collections.sort(permutaciones);

        System.out.println("Número total de permutaciones: " +
            permutaciones.size());
        System.out.println("Las 10 primeras permutaciones ordenadas
            alfabéticamente:");

        for (int i = 0; i < 10 && i < permutaciones.size(); i++) {
            System.out.println(permutaciones.get(i));
        }

        public static ArrayList<String> permutarPalabra(String palabra) {
            ArrayList<String> permutaciones = new ArrayList<>();
            permutarRecursoivo("", palabra, permutaciones);
            return permutaciones;
        }

        private static void permutarRecursoivo(String prefijo, String
            palabra, ArrayList<String> permutaciones) {
            if (palabra.length() == 0) {
                permutaciones.add(prefijo);
            } else {
                for (int i = 0; i < palabra.length(); i++) {
                    permutarRecursoivo(prefijo + palabra.charAt(i),
```

```

palabra.substring(0, i) + palabra.substring(i + 1), permutaciones);
        }
    }
}

```

### Resultado de ejecución:

```

> Task :AlgoritmoPermutacion.main()
Ingrese una palabra: alfabeto
Número total de permutaciones: 40320
Las 10 primeras permutaciones ordenadas alfabéticamente:
aabeflot
aabeflot
aabeflto
aabeflto
aabefolt
aabefolt
aabefotl
aabefotl
aabeftlo
aabeftlo

```

2. Algoritmo que realice el cifrado de un mensaje por **permutación de filas, teniendo como clave n filas**. Tanto n como el texto del mensaje se ingresan al iniciar el algoritmo. El algoritmo debe controlar que el número de caracteres del mensaje (sin espacios), sea menor o igual que n x n. Imprima la matriz de cifrado, el mensaje original y el mensaje cifrado. Si en la matriz de cifrado sobran espacios para almacenar los caracteres del mensaje original, estos deben llenarse con "\*".

### Código Fuente:

```

package algoritmos;
import java.util.Scanner;
public class AlgoritmoPermutacionFilas {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el número de filas (n): ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consumir el salto de línea restante

        System.out.print("Ingrese el mensaje: ");
        String mensaje = scanner.nextLine().replaceAll("\\s+", ""); //
        Eliminar espacios en blanco

        if (mensaje.length() > n * n) {
            System.out.println("El mensaje es demasiado largo para la
matriz de cifrado.");
            return;
        }
    }
}

```

```

        char[][] matrizCifrado = new char[n][n];

        // Llenar la matriz de cifrado con el mensaje de forma
        recursiva
        llenarMatrizCifrado(matrizCifrado, mensaje, 0, 0);

        // Imprimir la matriz de cifrado
        System.out.println("Matriz de cifrado:");
        imprimirMatriz(matrizCifrado);

        // Imprimir el mensaje original
        System.out.println("Mensaje original: " + mensaje);

        // Imprimir el mensaje cifrado
        String mensajeCifrado = generarMensajeCifrado(matrizCifrado,
n);
        System.out.println("Mensaje cifrado: " + mensajeCifrado);
    }

    private static void llenarMatrizCifrado(char[][] matrizCifrado,
String mensaje, int fila, int columna) {
        if (columna == matrizCifrado[0].length) {
            fila++;
            columna = 0;
        }

        if (fila == matrizCifrado.length || mensaje.isEmpty()) {
            // Llenar los espacios restantes con '*'
            llenarConCaracter(matrizCifrado, fila, columna, '*');
            return;
        }

        matrizCifrado[fila][columna] = mensaje.charAt(0);
        llenarMatrizCifrado(matrizCifrado, mensaje.substring(1), fila,
columna + 1);
    }

    private static String generarMensajeCifrado(char[][]
matrizCifrado, int n) {
        StringBuilder mensajeCifrado = new StringBuilder();
        for (int columna = 0; columna < n; columna++) {
            for (int fila = 0; fila < n; fila++) {
                if (matrizCifrado[fila][columna] != '*') {
mensajeCifrado.append(matrizCifrado[fila][columna]);
                }
            }
        }
        return mensajeCifrado.toString();
    }

    private static void imprimirMatriz(char[][] matriz) {
        for (char[] fila : matriz) {
            for (char elemento : fila) {
                System.out.print(elemento + " ");
            }
            System.out.println();
        }
    }

    private static void llenarConCaracter(char[][] matriz, int

```

```

filaInicio, int columnaInicio, char caracter) {
    for (int fila = filaInicio; fila < matriz.length; fila++) {
        for (int columna = (fila == filaInicio) ? columnaInicio :
0; columna < matriz[0].length; columna++) {
            matriz[fila][columna] = caracter;
        }
    }
}
}
}

```

### Resultado de ejecución:

```

06:22:36 a. m.: Executing ':AlgoritmoPermutacionFilas.main()'...

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :AlgoritmoPermutacionFilas.main()
Ingrese el número de filas (n): 9
Ingrese el mensaje: hola Andres queria saber como estas
Matriz de cifrado:
h o l a A n d r e
s q u e r i a s a
b e r c o m o e s
t a s * * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Mensaje original: holaAndresquieriasabercomoestas
Mensaje cifrado: hsbtoqealursaecAronimdaorseeas

```

**3. Algoritmo que realice el cifrado de un mensaje por **permutación de columnas**, teniendo como clave **n columnas**.** Tanto n como el texto del mensaje se ingresan al iniciar el algoritmo. El algoritmo debe controlar que el número de caracteres del mensaje (sin espacios), sea menor o igual que  $n \times n$ . Imprima la matriz de cifrado, el mensaje original y el mensaje cifrado. Si en la matriz de cifrado sobran espacios para almacenar los caracteres del mensaje original, estos deben llenarse con "\*".

### Código Fuente:

```

class Ejercicio {

    public static void main(String[] args) {

```

```
String mensaje = "holamundo";

int n = 5;

if (mensaje.length() > n * n) {

    System.out.println("El número de caracteres del mensaje (sin
espacios) debe ser menor o igual que n x n.");

    return;

}

char[][] matriz = new char[n][n];

int index = 0;

// Llenar la matriz con el mensaje
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (index < mensaje.length()) {
            matriz[i][j] = mensaje.charAt(index);
            index++;
        } else {
            matriz[i][j] = '*';
        }
    }
}

// Imprimir la matriz de cifrado
System.out.println("Matriz de cifrado:");

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
```

```

        System.out.print(matriz[i][j] + " ");

    }

    System.out.println();

}

// Imprimir el mensaje original

System.out.println("Mensaje original: " + mensaje);

// Imprimir el mensaje cifrado

System.out.print("Mensaje cifrado: ");

for (int j = 0; j < n; j++) {

    for (int i = 0; i < n; i++) {

        System.out.print(matriz[i][j]);

    }

}

}

}

```

### Resultado de ejecución:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Matriz de cifrado:
h o l a m
u n d o *
* * * * *
* * * * *
* * * * *

Mensaje original: holamundo
Mensaje cifrado: hu***on***ld***ao***m***
PS C:\Users\fabri\OneDrive\Documentos\Proyectos>

```

4. Algoritmo que realice el cifrado de una cadena de caracteres mediante un **método de sustitución Monoalfabético de desplazamiento n caracteres a la derecha**. Tanto la palabra como el valor de n se ingresan al iniciar el algoritmo. El algoritmo debe mostrar el alfabeto original, el alfabeto cifrado, la cadena de caracteres ingresada y su resultado.

#### Código Fuente:

```
def cifrado_monoalfabetico(texto, desplazamiento):  
  
    abecedario =  
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'  
  
    resultado = ''  
  
    for caracter in texto:  
  
        if caracter in abecedario:  
  
            indice = (abecedario.index(caracter) + desplazamiento) %  
len(abecedario)  
  
            resultado += abecedario[indice]  
  
        else:  
  
            resultado += caracter  
  
    return resultado  
  
palabra = input("Ingrese la palabra a cifrar: ")  
  
desplazamiento = int(input("Ingrese el valor de desplazamiento: "))  
  
print("abecedariooriginal:",  
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ')  
  
print("abecedario cifrado:",  
cifrado_monoalfabetico('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNÑOPQR  
STUVWXYZ', desplazamiento))  
  
print("Texto original:", palabra)  
  
print("Texto cifrado:", cifrado_monoalfabetico(palabra,  
desplazamiento))
```

#### Resultado de ejecución:

```
Ingrese la palabra a cifrar: Saludos
Ingrese el valor de desplazamiento: 2
abecedario original: abcdefghijklmñopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
abecedario cifrado: cdefghijklmñopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZab
Texto original: Saludos
Texto cifrado: Ucnwfqu
```

5. Algoritmo que realice el cifrado de una cadena de caracteres mediante un **método de sustitución Polialfabético de Vigenère**. La cadena se ingresa al iniciar el algoritmo. El algoritmo debe mostrar la cadena de caracteres ingresada, la clave de cifrado y la cadena de caracteres cifrada.

#### Código Fuente:

```
public static String cifrarVigenere(String mensaje, String clave) {

    StringBuilder mensajeCifrado = new StringBuilder();

    String alfabeto = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    clave = clave.toUpperCase();

    int claveIndex = 0;

    for (char caracter : mensaje.toCharArray()) {

        if (Character.isLetter(caracter)) {

            char caracterUpper = Character.toUpperCase(caracter);

            int fila = alfabeto.indexOf(caracterUpper);

            int
                                columna
                                =
alfabeto.indexOf(clave.charAt(claveIndex));

            char caracterCifrado = alfabeto.charAt((fila + columna)
% 26);

            mensajeCifrado.append(caracterCifrado);

            claveIndex = (claveIndex + 1) % clave.length();

        } else {
```



```

        mensajeCifrado.append(caracter);

    }

}

return mensajeCifrado.toString();

}

public static String descifrarVigenere(String mensajeCifrado, String
clave) {

    StringBuilder mensajeDescifrado = new StringBuilder();

    String alfabeto = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    clave = clave.toUpperCase();

    int claveIndex = 0;

    for (char caracter : mensajeCifrado.toCharArray()) {

        if (Character.isLetter(caracter)) {

            char caracterUpper = Character.toUpperCase(caracter);

            int fila = alfabeto.indexOf(caracterUpper);

            int
                columna
                =
alfabeto.indexOf(clave.charAt(claveIndex));

            char
                caracterDescifrado
                =
alfabeto.charAt((fila
columna + 26) % 26);

            mensajeDescifrado.append(caracterDescifrado);

            claveIndex = (claveIndex + 1) % clave.length();

        } else {

            mensajeDescifrado.append(caracter);

```

```

    }

    }

    return mensajeDescifrado.toString();
}

public static void main(String[] args) {

    String mensaje = "CRIPTOGRAFIA";

    String clave = "OCTAVO";

    System.out.println("Cadena de Caracteres Ingresada: "+ mensaje);

    System.out.println("Clave: "+clave);

    String mensajeCifrado = cifrarVigenere(mensaje, clave);

    System.out.println("Mensaje cifrado: " + mensajeCifrado);

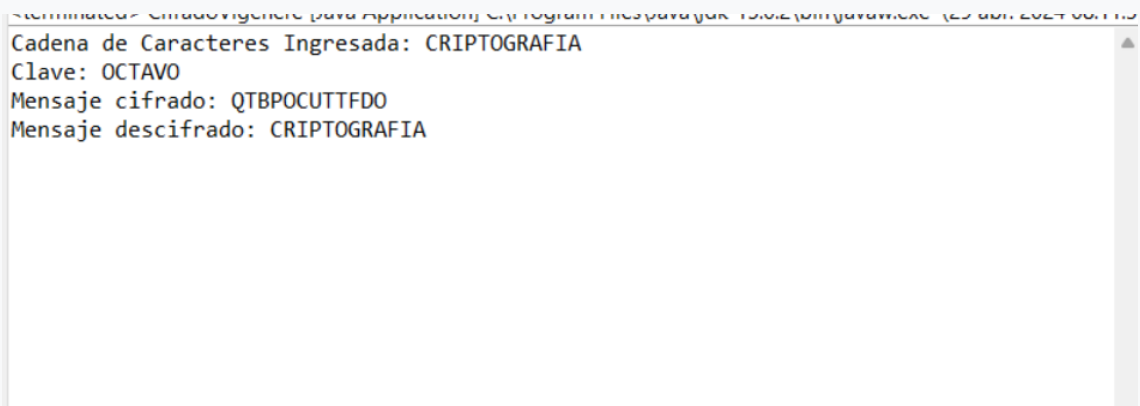
    String mensajeDescifrado = descifrarVigenere(mensajeCifrado,
clave);

    System.out.println("Mensaje descifrado: " + mensajeDescifrado);

}

```

### Resultado de ejecución:



Terminated: CifradoVigenere para Application, Cif program files para Java 1.8.0\_201 para Windows (23 de Oct. 2021 00:11:12)  
 Cadena de Caracteres Ingresada: CRIPTOGRAFIA  
 Clave: OCTAVO  
 Mensaje cifrado: QTBPOCUTTFDO  
 Mensaje descifrado: CRIPTOGRAFIA

6. Algoritmo que realice el cifrado de una cadena de caracteres utilizando la siguiente **tabla de cifrado**:

*	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

**Código Fuente:**

**Clase: Literal6 (Main)**

```
public class Literal6 {  
    public static void main(String[] args) {  
  
        Scanner scan=new Scanner(System.in);  
        System.out.print("Ingrese la cadena a encriptar: ");  
        String cadena = scan.nextLine();  
  
        System.out.println();  
  
        char[][] tablaCifrado = new char[6][6]; //Inicializo la tabla  
        de encriptado  
  
        tablaCifrado[0][0]='*';  
        tablaCifrado[0][1]='A';  
        tablaCifrado[0][2]='S';  
        tablaCifrado[0][3]='D';  
        tablaCifrado[0][4]='F';  
        tablaCifrado[0][5]='G';  
  
        tablaCifrado[1][0]='Q';
```

```
tablaCifrado[1][1]='a';  
tablaCifrado[1][2]='b';  
tablaCifrado[1][3]='c';  
tablaCifrado[1][4]='d';  
tablaCifrado[1][5]='e';
```

```
tablaCifrado[2][0]='W';  
tablaCifrado[2][1]='f';  
tablaCifrado[2][2]='g';  
tablaCifrado[2][3]='h';  
tablaCifrado[2][4]='i';  
tablaCifrado[2][5]='j';
```

```
tablaCifrado[3][0]='E';  
tablaCifrado[3][1]='k';  
tablaCifrado[3][2]='l';  
tablaCifrado[3][3]='m';  
tablaCifrado[3][4]='n';  
tablaCifrado[3][5]='o';
```

```
tablaCifrado[4][0]='R';  
tablaCifrado[4][1]='p';  
tablaCifrado[4][2]='q';  
tablaCifrado[4][3]='r';  
tablaCifrado[4][4]='s';  
tablaCifrado[4][5]='t';
```

```
tablaCifrado[5][0]='T';  
tablaCifrado[5][1]='u';  
tablaCifrado[5][2]='v';  
tablaCifrado[5][3]='x';  
tablaCifrado[5][4]='y';  
tablaCifrado[5][5]='z';
```

```

        Polibio cifrado=new Polibio(tablaCifrado);
        String cadenaCifrada=cifrado.cifrar(cadena); //llamado al
método

        System.out.println("Tabla de cifrado: ");
        for(int i = 0; i < 6; i++) { //Impresión de la tabla
            for(int j = 0; j < 6; j++) {
                System.out.print(cifrado.getTablaCifrado()[i][j] + "
");
            }
            System.out.println();
        }

        System.out.println("Mensaje Original: " + cadena);
        System.out.println("Mensaje Cifrado: " + cadenaCifrada);

    }
}

```

### Clase: Polibio

```

public class Polibio {

    private final char[][] tablaCifrado;

    public Polibio(char[][] tablaCifrado){ //ingresa la tabla
        this.tablaCifrado=tablaCifrado;
    }

    public String cifrar(String cadena){
        StringBuilder cadenaCifrada = new StringBuilder();
        for (char c : cadena.toCharArray()) { //for each para recorrer
la cadena a encriptar
            boolean encontrado = false;
            for (int i = 0; i < tablaCifrado.length; i++) {

```

```

        for (int j = 0; j < tablaCifrado[i].length; j++) {
            if (tablaCifrado[i][j] == c) {
                //Entonces si el elemento de la tabla coincide
con el carácter,
                // yo agrego al StringBuilder las coordenadas
de la tabla.

cadenaCifrada.append(tablaCifrado[i][0]).append(tablaCifrado[0][j]);

                encontrado = true;
                break;
            }
        }
        if (encontrado) {
            break;
        }
    }
    if (!encontrado) {
        //si no se llega a encontrar agrego **
        cadenaCifrada.append("**");
    }
}

return cadenaCifrada.toString();
}

public char[][] getTablaCifrado() {
    return tablaCifrado;
}
}

```

**Resultado de ejecución:**

```
> Task :Literal6.main()
Ingrese la cadena a encriptar: hola centralinos

Tabla de cifrado:
* 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
Mensaje Original: hola centralinos
Mensaje Cifrado: WDEGESQA**QDQGEFRGRDQAESWFEFEGRF
```