



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

~ Facultad de Ingeniería ~

## Estructura de Datos y Algoritmos I

Actividad Asíncrona 4 viernes

“Cifrado Cesar”

Nombre: Raudales Palma Leonardo de Jesús

Fecha: 22/03/2021



## PSEUDOCÓDIGO “CIFRADO CESAR”

INICIO

\*\* Función que permite codificar/decodificar un texto en cifrado Cesar

FUNC Cesar (mensaje: carácter, posición: entero, abecedario: arreglo)

RET: Mensaje codificado/descodificado

op: ENTERO

op = 1

SELECCIONAR (op) EN:

CASO 1 ->

ESCRIBIR “Codificar mensaje.”

CASO 2 ->

ESCRIBIR “Descodificar mensaje.”

CASO 3 ->

ESCRIBIR “Salir.”

DEFECTO ->

ESCRIBIR “Opción no válida.”

FIN SELECCIONAR

FUNC codificar: (mensaje: carácter, posición: entero, abecedario: arreglo)

RET: Mensaje cifrado

mensaje: = carácter

posición: = entero

abecedario: = arreglo

nuevo mensaje: = (mensaje + posición) en abecedario

RET nuevo mensaje

MIENTRAS “mensaje” PERTENEZCA A “abecedario”

ESCRIBIR mensaje + posición

FIN MIENTRAS

ESCRIBIR RET

FIN FUNC

FUNC descodificar: (mensaje: carácter, posición: entero, abecedario: arreglo)

RET: Mensaje descifrado

mensaje: = carácter

posición: = entero

abecedario: = arreglo

nuevo mensaje: = (mensaje - posición) en abecedario

RET nuevo mensaje

MIENTRAS "mensaje" PERTENEZCA A "abecedario"

    ESCRIBIR mensaje - posición

FIN MIENTRAS

ESCRIBIR RET

FIN FUNC

FIN

// >>> nuevo mensaje

## CÓDIGO “CIFRADO CESAR” EN JAVA

```
Lenguaje Java - Cifrado Cesar/src/Cesar.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

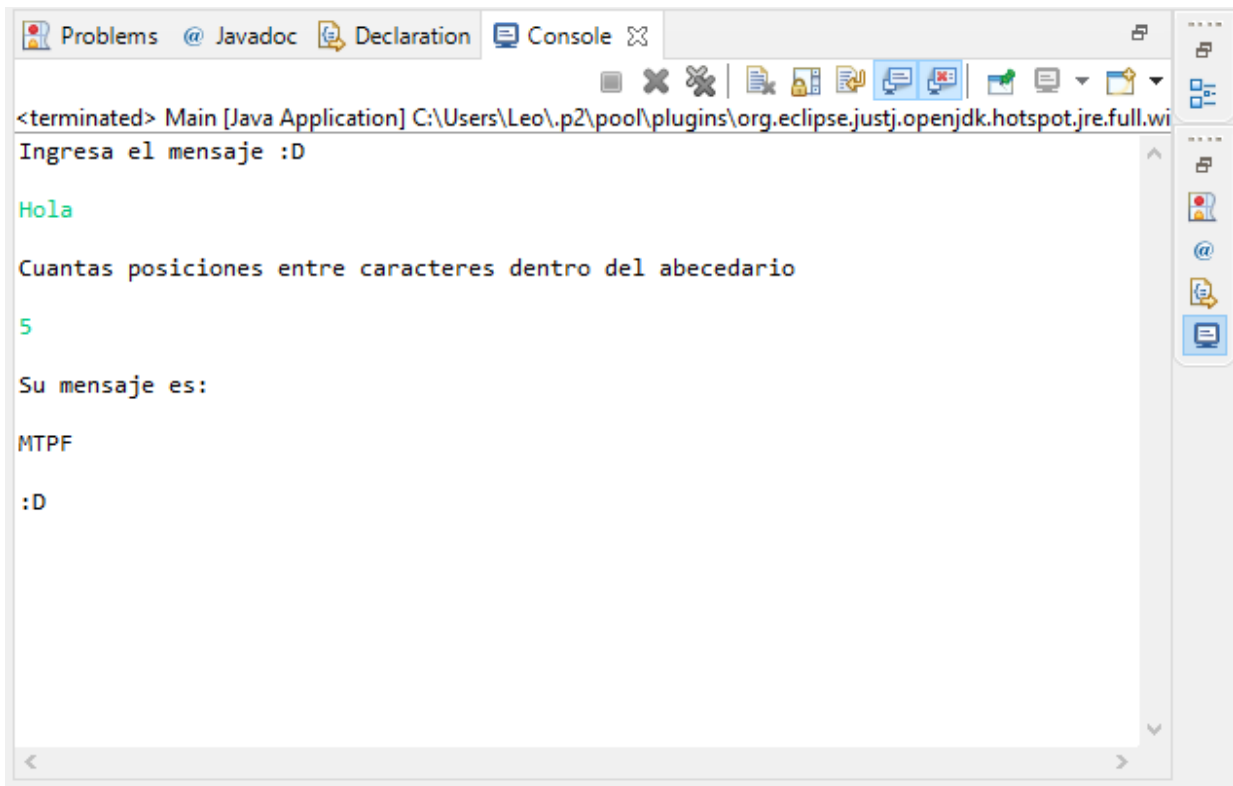
Cesar.java Main.java
1 // Importando la libreria para escanear texto desde el teclado.
2 import java.util.Scanner;
3
4 // Función del cifrado
5 public class Cesar
6 {
7
8     // Declarando las variables como el mensaje, el número de espacios y el arreglo abecedario.
9     private String mensaje;
10    private int llave;
11    private static char[] alfabeto = {'A','B','C','D','E',
12    'F','G','H','I','J','K','L','M','N','Ñ','O','P','Q','R','S',
13    'T','U','V','W','X','Y','Z'};
14
15    // Generando funciones para poder acceder a las variables.
16    public Cesar()
17    {
18        super();
19    }
20
21    public Cesar(String mensaje, int llave)
22    {
23        super();
24        this.mensaje=mensaje;
25        this.llave=llave;
26    }
27
28    public String getMensaje() {
29        return mensaje;
30    }
31
32    public void setMensaje(String mensaje) {
33        this.mensaje = mensaje;
34    }
35
36    public int getLlave() {
37        return llave;
38    }
39
40    public void setLlave(int llave) {
41        this.llave = llave;
42    }
43
44    public static char[] getAlfabeto() {
45        return alfabeto;
46    }
47
48    public static void setAlfabeto(char[] alfabeto) {
49        Cesar.alfabeto = alfabeto;
50    }
51
52    public void leertexto()
53    {
54        // Instrucción para que detecte que oprimas una tecla
55        Scanner sc = new Scanner(System.in);
56
57        // Solicitar el mensaje.
58        System.out.println("Ingresa el mensaje :D\n");
59        setMensaje(sc.nextLine().toUpperCase());
```

File Edit Source Refactor Navigate Search Project Run Window Help

```
60
61 // Solicitar el número de lugares en el abecedario para sustituir caracteres.
62 System.out.println("\nCuantas posiciones entre caracteres dentro del abecedario\n");
63 setLlave(sc.nextInt());
64
65 sc.close();
66
67
68 // Función que ejecuta toda la acción de codificar
69 public char[] codificar()
70 {
71     char[] mensajecod = new char[getMensaje().length()];
72
73     for (int i=0; i<getMensaje().length(); i++)
74     {
75         if (getMensaje().charAt(i)==' ');
76         {
77             mensajecod[i]=getMensaje().charAt(i);
78         }
79
80         {
81             for(int j = 0; j<getAlfabeto().length; j++)
82             {
83                 if (getMensaje().charAt(i)==getAlfabeto()[j])
84                 {
85                     if (getLlave()<0)
86                     {
87                         mensajecod [i]=getAlfabeto()[((j+getLlave()+getAlfabeto().length)%getAlfabeto().length)];
88                         j=getAlfabeto().length;
89                     }
90                     else
91                     {
92                         mensajecod [i]=getAlfabeto()[((j+getLlave())%getAlfabeto().length)];
93                         j=getAlfabeto().length;
94                     }
95                 }
96                 else
97                 {
98                     mensajecod[i]=getMensaje().charAt(i);
99                 }
100             }
101         }
102     }
103
104     return mensajecod;
105
106 }
107
108 // Función para descifrar el mensaje.
109 public char[] decodificar()
110 {
111     char[] mensajecod = new char[getMensaje().length()];
112
113     for (int i=0; i<getMensaje().length(); i++)
114     {
115         if (getMensaje().charAt(i)==' ');
116         {
117             mensajecod[i]=getMensaje().charAt(i);
118
119         }
120
121         {
122             for(int j = 0; j<getAlfabeto().length; j++)
123             {
124                 if (getMensaje().charAt(i)==getAlfabeto()[j])
125                 {
126                     if (j<getLlave())
127                     {
128                         mensajecod [i]=getAlfabeto()[((j-getLlave()+getAlfabeto().length)%getAlfabeto().length)];
129                         j=getAlfabeto().length;
130                     }
131                     else
132                     {
133                         mensajecod [i]=getAlfabeto()[((j-getLlave())%getAlfabeto().length)];
134                         j=getAlfabeto().length;
135                     }
136                 }
137                 else
138                 {
139                     mensajecod[i]=getMensaje().charAt(i);
140                 }
141             }
142         }
143     }
144
145     return mensajecod;
146
147 }
148
149 }
150
```

```
1 // Función principal para ejecutar el cifrado cesar en java.
2 public class Main
3 {
4
5     // Ejecución del cifrado del texto dentro de la función principal.
6     public static void main(String[] args)
7     {
8         Cesar c= new Cesar();
9         c.leertexto();
10        System.out.println("\nSu mensaje es:\n");
11        System.out.println(c.codificar());
12        System.out.println("\n:D");
13    }
14
15    // Ejecución del descifrado del texto dentro de la función principal
16    public static void main(String[] args)
17    {
18        Cesar c= new Cesar();
19        c.leertexto();
20        System.out.println("\nSu mensaje es:\n");
21        System.out.println(c.decodificar());
22        System.out.println("\n:D");
23    }
24 }
25 }
```

## COMPILACIÓN "CIFRADO CESAR" EN JAVA



```
<terminated> Main [Java Application] C:\Users\Leo\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
Ingresa el mensaje :D

Hola

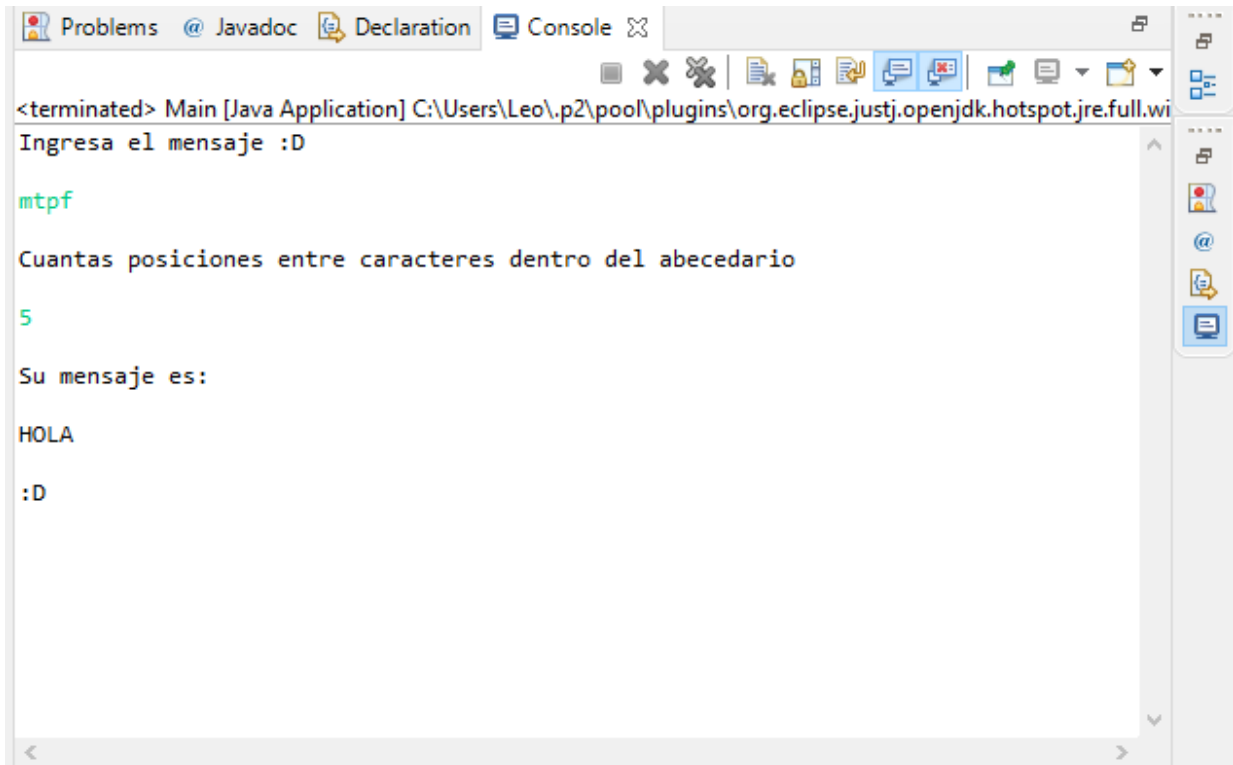
Cuantas posiciones entre caracteres dentro del abecedario

5

Su mensaje es:

MTPF

:D
```



```
<terminated> Main [Java Application] C:\Users\Leo\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
Ingresa el mensaje :D

mtpf

Cuantas posiciones entre caracteres dentro del abecedario

5

Su mensaje es:

HOLA

:D
```