



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN



TAREA

P R E S E N T A

Alexis Hernández Zamudio

A P R O F E S O R

Jesús Hernández Cabrera

Gpo:1558

URL del repositorio:

https://github.com/AlextyrB/Analisis_Algoritmos/tree/main/Desifradior%20Cesar/Cesar

Descifrado Cesar

```
1  import java.io.*;
2  import java.nio.file.*;
3
4  public class CesarDecoder {
5      private static final String ALFABETO = "abcdefghijklmnñopqrstuvwxyz , ."; 5 usages
6
7  public static void main(String[] args) {
8      String nombreArchivo = "cesarr.txt";
9
10     try {
11         String mensajeCifrado = leerArchivo(nombreArchivo);
12         System.out.println("Mensaje cifrado: " + mensajeCifrado);
13         System.out.println("\nDESCIFRANDO POR FUERZA BRUTA\n");
14
15         String mejorResultado = "";
16         int mejorDesplazamiento = -1;
17
18         for (int desplazamiento = 0; desplazamiento < ALFABETO.length(); desplazamiento++) {
19             String mensajeDescifrado = descifrar(mensajeCifrado, desplazamiento);
20             System.out.println("Desplazamiento " + desplazamiento + ": " + mensajeDescifrado);
21
22             if (tieneSentido(mensajeDescifrado)) {
23                 mejorResultado = mensajeDescifrado;
24                 mejorDesplazamiento = desplazamiento;
25             }
26         }
27     }
28
29     if (mejorDesplazamiento != -1) {
30         System.out.println("\n" + "=" .repeat(60));
31         System.out.println("RESULTADO FINAL ENCONTRADO:");
32         System.out.println("Desplazamiento: " + mejorDesplazamiento);
33         System.out.println("Mensaje descifrado: " + mejorResultado);
34         System.out.println("=".repeat(60));
35     }
36
37     } catch (IOException e) {
38         System.err.println("Error al leer el archivo: " + e.getMessage());
39     }
40
41     private static String leerArchivo(String nombreArchivo) throws IOException { 1 usage
42         return new String(Files.readAllBytes(Paths.get(nombreArchivo)));
43     }
44
45     private static String descifrar(String mensaje, int desplazamiento) { 1 usage
46         StringBuilder resultado = new StringBuilder();
47
48         for (char c : mensaje.toCharArray()) {
49             int posicion = ALFABETO.indexOf(Character.toLowerCase(c));
50
51             if (posicion != -1) {
52                 int nuevaPosicion = (posicion - desplazamiento + ALFABETO.length()) % ALFABETO.length();
53                 resultado.append(ALFABETO.charAt(nuevaPosicion));
54             } else {
55                 resultado.append(c);
56             }
57         }
58     }
59 }
```

```
28
29     if (mejorDesplazamiento != -1) {
30         System.out.println("\n" + "=" .repeat(60));
31         System.out.println("RESULTADO FINAL ENCONTRADO:");
32         System.out.println("Desplazamiento: " + mejorDesplazamiento);
33         System.out.println("Mensaje descifrado: " + mejorResultado);
34         System.out.println("=".repeat(60));
35     }
36
37     } catch (IOException e) {
38         System.err.println("Error al leer el archivo: " + e.getMessage());
39     }
40
41     private static String leerArchivo(String nombreArchivo) throws IOException { 1 usage
42         return new String(Files.readAllBytes(Paths.get(nombreArchivo)));
43     }
44
45     private static String descifrar(String mensaje, int desplazamiento) { 1 usage
46         StringBuilder resultado = new StringBuilder();
47
48         for (char c : mensaje.toCharArray()) {
49             int posicion = ALFABETO.indexOf(Character.toLowerCase(c));
50
51             if (posicion != -1) {
52                 int nuevaPosicion = (posicion - desplazamiento + ALFABETO.length()) % ALFABETO.length();
53                 resultado.append(ALFABETO.charAt(nuevaPosicion));
54             } else {
55                 resultado.append(c);
56             }
57         }
58     }
59 }
```

```
58
59         return resultado.toString();
60     }
61
62     private static boolean tieneSentido(String mensaje) { 1 usage
63         String mensajeLower = mensaje.toLowerCase();
64
65         String[] palabrasComunes = {
66             "el ", "la ", "de ", "que ", "en ", "los ", "un ", "por ",
67             "con ", "para ", "una ", "es ", "del ", "las ", "y ",
68             "hola", "mundo", "mensaje", "texto", "cesar"
69         };
70
71         int coincidencias = 0;
72         for (String palabra : palabrasComunes) {
73             if (mensajeLower.contains(palabra)) {
74                 coincidencias++;
75                 if (coincidencias >= 2) {
76                     return true;
77                 }
78             }
79         }
80
81         return false;
82     }
83 }
```

DESCIFRANDO POR FUERZA BRUTA

RESULTADO DEL DESIFRADO

FUERZA BRUTA

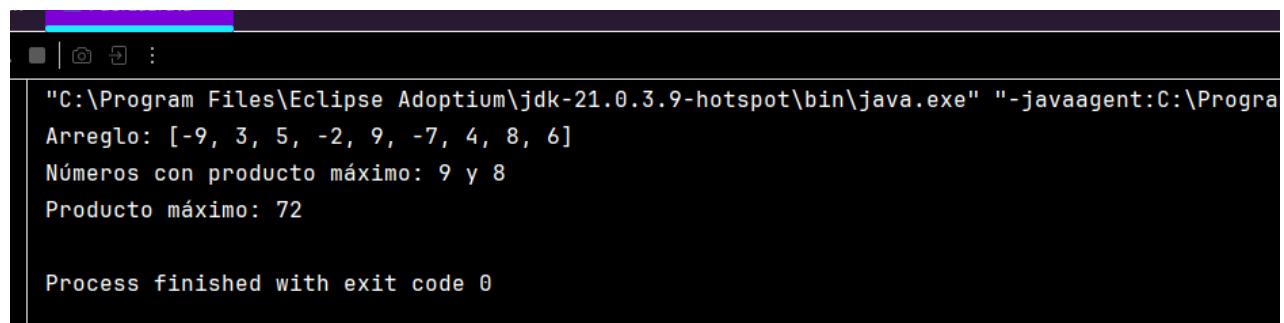
```
public class FuerzaBruta { new*
    public static void main(String[] args) { new*
        int[] A = {-9, 3, 5, -2, 9, -7, 4, 8, 6};

        int maxProducto = Integer.MIN_VALUE;
        int num1 = 0, num2 = 0;

        for (int i = 0; i < A.length; i++) {
            for (int j = i + 1; j < A.length; j++) {
                int producto = A[i] * A[j];
                if (producto > maxProducto) {
                    maxProducto = producto;
                    num1 = A[i];
                    num2 = A[j];
                }
            }
        }

        System.out.println("Arreglo: " + java.util.Arrays.toString(A));
        System.out.println("Números con producto máximo: " + num1 + " y " + num2);
        System.out.println("Producto máximo: " + maxProducto);
    }
}
```

RESULTADO DE FUERZA BRUTA



```
"C:\Program Files\Eclipse Adoptium\jdk-21.0.3.9-hotspot\bin\java.exe" "-javaagent:C:\Progra
Arreglo: [-9, 3, 5, -2, 9, -7, 4, 8, 6]
Números con producto máximo: 9 y 8
Producto máximo: 72

Process finished with exit code 0
```