



Florida

Universitatària

CICLO

1º Desarrollo de Aplicaciones Multiplataforma

ACTIVIDAD

Descifrar la Máquina Enigma (AEV3)

Realizado por: Héctor Sánchez Luque

Módulo formativo: Entornos de Desarrollo

En Catarroja, a 14/01/2022

1. **Preparación:** descarga el programa adjunto a este documento en el curso de Florida Oberta. Puedes elegir entre realizar los pasos en Visual Studio Code (JS) o Visual Studio (C#), ambos depuradores funcionan de forma similar. **Lo primero** que tendrás que hacer es **revisar y entender el código para completarlo**, puesto que le falta una estructura de control que posibilita que funcione correctamente:

```
/*  
  
if (.....){  
.....;  
}  
  
*/
```

Te facilito la estructura de control básica comentada. Sólo tienes que quitar las marcas de comentarios y sustituir las líneas de puntos por el código correspondiente, es decir, la **condición** del if y la **sentencia**. Ésta es la única modificación que podrás hacerle al código. Si te resulta útil, puedes depurar para deducir el código que falta. **(2 puntos)**

```
if (posicion < 0){  
    posicion = posicion * (-1);  
    posicion = 27 - posicion;  
}
```

He introducido esta condicional, cuya función es, primero comprobar que el valor de posición sea menor de 0. Si se da el caso, primero multiplicaría el valor de posición por -1 para hacerlo positivo (dado que si no lo hacía me daba “undefined” en el depurador) y después restar el número total de letras del abecedario proporcionado al valor (ahora positivo) de posición, haciendo que funcione de forma cíclica.

2. **Descifrado:** una vez dispongas del código correcto, tendrás que **depurar** para poder ir descifrando el mensaje letra a letra. Indica cuál es el **mensaje descifrado**. Durante la depuración, debes capturar 8 pantallazos, correspondientes por ejemplo al descifrado de 8 letras, que argumenten como estás realizando el proceso de depuración. (8 puntos)

0: 'a'	7: 'a'	13: 's'	19: 'c'	29: 'e'	38: 'a'	44: 'o'
1: 't'	8: 'e'	14: 'o'	20: 'a'	30: 'l'	39: '*'	45: 'c'
2: 'a'	9: 'r'	15: 'b'	21: 't'	31: '*'	40: 'l'	46: 'h'
3: 'q'	10: 'e'	16: 'r'	22: 'a'	32: 'l'	41: 'a'	47: 'o'
4: 'u'	11: 'o'	17: 'e'	23: 'r'	33: 'u'	42: 's'	
5: 'e'	12: '*'	18: '*'	24: 'r'	34: 'n'	43: '*'	
6: '*'			25: 'o'	35: 'e'		
			26: 'j'	36: 's'		
			27: 'a'	37: '*'		
			28: '*'			

The screenshot shows a JavaScript IDE with a file named 'JS enigmas.js'. The code defines an alphabet string 'abecedario' and a ciphertext 'mensaje'. It then iterates through the ciphertext, calculating the position of each letter in the alphabet and adjusting it by -10 to decrypt it. The 'WATCH' panel on the left shows the state of the 'descifrado' array after the first 8 iterations, which matches the first row of the table above.

```

19 //String: mensaje cifrado o encriptado que hemos interceptado del enemigo
20 let mensaje = "kck*dojkoaoyjbylaojmkckaaytkjovjvdxobjkjbkbjymry";
21
22 //posición dentro del abecedario que utilizamos para descifrar
23 let posicion;
24
25 //array: donde iremos almacenando el mensaje conforme lo vayamos descifrando
26 let descifrado = [];
27
28 //Recorremos el mensaje cifrado para irlo descifrando letra a letra
29 for (let i=0; i<mensaje.length; i++){
30
31     //Calculamos la posición en el abecedario de la letra del texto cifrado
32     posicion = abecedario.indexOf(mensaje[i]);
33
34     //Desplazamos la posición 10 unidades hacia la izquierda pero...
35     posicion = posicion - 10;
36
37     //Llegados a este punto, puede ser que la posición sea negativa...
38     //si eso sucede, tendremos que hacer un ajuste en
39     //la variable posicion para funcione de forma cíclica dentro del abecedario
40
41     if (posicion < 0){
42         posicion = posicion * (-1);
43         posicion = 27 - posicion;
44     }
45
46     descifrado[i] = abecedario[posicion];
47 }

```

VARIABLES

- Local
 - abecedario: 'abcdefghijklmnopqrstuvwxyz'
 - descifrado: (48) ['a', 't', 'a', 'q', 'u', ...]
 - exports: {}
 - mensaje: 'kck*dojkoaoyjbylaojmkckaaytkjovjvdxobjkjbkbjymry'
 - module: Module {id: '.', path: 'c:\\Users\\...\\Programas DAM > Primer Año DAM > Entornos > AEV3(1) > JS enigmas.js > ...'
 - posicion: 14
 - require: f require(path) {\\r\\n return...
 - __dirname: 'c:\\Users\\Usuario\\Desktop\\P...'
 - __filename: 'c:\\Users\\Usuario\\Desktop\\...'
 - this: Object
 - Global

WATCH

- descifrado: (48) ['a', 't', 'a', 'q', 'u', ...]
 - 0: 'a'
 - 1: 't'
 - 2: 'a'
 - 3: 'q'
 - 4: 'u'
 - 5: 'e'
 - 6: '*'
 - 7: 'a'
 - 8: 'e'

Para el proceso de depuración simplemente he puesto un breakpoint al principio del for y he puesto en "watch" la variable descifrado para ver el valor que daba en cada vuelta del bucle.