GUIÓN DE LA ACTIVIDAD AEV3:

Título

Depuración. Enigma: descifra el mensaje.

Objetivos

- Entender las ventajas del uso de las herramientas de depuración que ofrecen algunos editores de código y los entornos de desarrollo (IDE).
- Conocer las utilidades propias de la depuración, para poder controlar y gestionar el flujo de ejecución de una aplicación.
- Utilizar el procedimiento de visualización de valores de variables en tiempo de ejecución, para detectar posibles anomalías o errores lógicos.
- Dedicar tiempo a practicar depuración para adquirir destreza.

Temporalización

Se estima una dedicación de **3 horas**. Teniendo en cuenta que habrá que revisar los recursos facilitados en el curso en Florida Oberta para poder de realizar la actividad.

Proceso de desarrollo

1. Leer, analizar y entender cada paso propuesto.
2. Realizar las capturas de pantalla que sean necesarias para argumentar que se han realizado cada uno de los pasos propuestos.
3. Generar un documento PDF con las capturas y las explicaciones de texto que consideres convenientes.
4. Entregar el documento PDF, debidamente identificado, que incluya cada enunciado con la respuesta correspondiente, a través de Florida Oberta.

Evaluación

La actividad consiste realizar una serie de pasos. Cada paso se valorará en función de su dificultad y su importancia respecto a los objetivos de la actividad. En total los pasos sumarán 10 puntos. Cada error, carencia o imprecisión implica un descuento de entre 0,25 puntos, si se trata de una cuestión leve y hasta 8 puntos, si la cuestión es grave o muy grave.

Recursos

Puestos a disposición del alumno en el curso correspondiente del campus virtual Florida Oberta.

Detalle de la actividad

Lee la introducción y sigue los pasos propuestos. Recuerda que el objetivo principal de la actividad es depurar código:

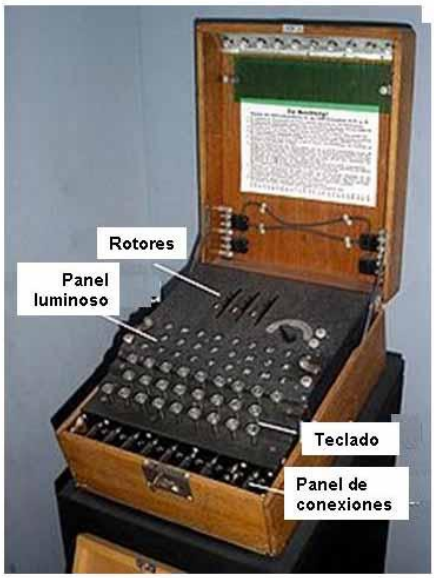
Introducción:

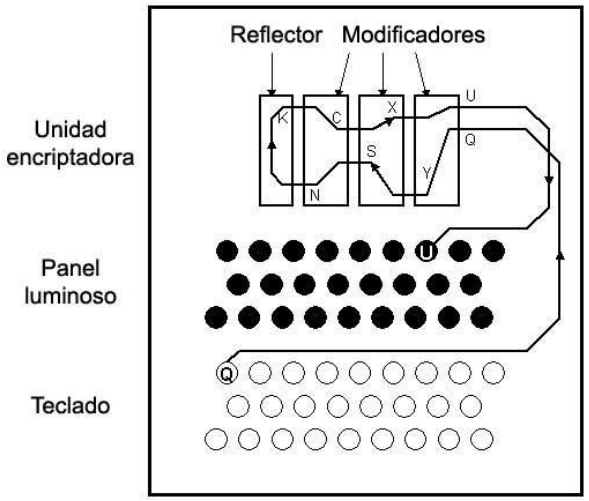
Se considera que la **2ª guerra mundial** se inició en el año **1939**, con la invasión de **Polonia** por parte del ejército nazi alemán. Este ejército, puso en jaque a casi toda Europa y a una buena parte del mundo, debido a su potencial **bélico y tecnológico**.

Entre otros muchos artefactos, el ejército nazi comenzó a utilizar una máquina para realizar **comunicaciones de forma cifrada**. Dicha máquina funcionaba mediante un mecanismo **electromecánico de rotores**, conectado a un **teclado** y a un **panel de luces** con las letras del alfabeto, que permitía cifrar y descifrar mensajes. Esta máquina se denominó **Enigma**.

El corazón de la máquina Enigma era mecánico y constaba de **varios rotores** conectados entre sí. **Cada rotor** es un disco circular plano con **26 contactos eléctricos** en cada cara, uno por cada **letra del alfabeto**. Dentro de la máquina había, en la mayoría de las versiones, **3 ranuras** para alojar sendos rotores en el momento de ponerla en funcionamiento, de entre 5 rotores diferentes que disponía el equipamiento complementario de la máquina.

Cuando se pulsaba una tecla en el teclado, por ejemplo, la letra “Q”, la corriente eléctrica procedente de la batería se dirigía hasta el contacto correspondiente a la letra “Q” del primer rotor. La corriente atravesaba el cableado interno del primer rotor y se situaba, por ejemplo, en el contacto correspondiente a la letra “Y” en el lado contrario. De ese modo, la corriente seguía su camino a través de todos los rotores, hasta que llegaba a iluminarse en el panel la letra correspondiente al cifrado de la pulsación de teclado, por ejemplo, la letra “U”.





La complejidad de los cifrados no dejó de evolucionar. El cifrado de un mensaje variaba en función de cómo se establecía la **configuración inicial**:

- Los **3 rotores seleccionados** para ser instalados o insertados en las 3 ranuras, de entre los 5 rotores distintos disponibles.
- El **orden** de instalación de los 3 rotores.
- La **posición inicial** de cada uno de los rotores. Disponían de 26 posiciones diferentes.
- Las **conexiones** entre pares de letras de los diferentes rotores.

Una aproximación al cálculo de combinaciones de estados posibles de la máquina Enigma, sería la siguiente:

- Suponiendo que se han seleccionado 3 rotores, los posibles órdenes de colocación en las 3 hendiduras serían:

  3 * 2 * 1 = 6 colocaciones.

- Como cada rotor tiene 26 letras o posiciones, habiendo 3 rotores y 6 posibles colocaciones, el número de posiciones total de los rotores sería:

  26 * 26 * 26 = 17.576

  17.576 * 6 = 105.456 posiciones

- Posteriormente se definían las conexiones entre cada rotor a nivel de pares de letras (máximo 13 pares, 26 letras):

  $n! /((n-2m)!\ m!\ 2^m)$ → Siendo "n" el número de letras por rotor

  → Siendo "m" el número de pares conectados

  $26! / (0!\ 13!\ 2^{13})$ = **7.905.853.580.625 estados** diferentes de la máquina

  **Curiosamente se obtiene el mayor número de combinaciones diferentes posibles cuando se configuran 11 pares de letras:

  $26! / (4!\ 11!\ 2^{11})$ = 205.552.193.096.250 estados diferentes de la máquina

El **funcionamiento** de las versiones más comunes de la máquina Enigma era **simétrico**, en el sentido de que el proceso de descifrado era análogo al proceso de cifrado. Para obtener el mensaje original solo había que introducir las letras del mensaje cifrado en la máquina, y ésta devolvía, una a una, las letras del mensaje original, siempre y cuando la configuración inicial de la máquina fuera idéntica a la utilizada para cifrar la información.

Este artefacto, resultó crucial para el avance del ejército nazi, que parecía imparable. Sin embargo, **la resistencia polaca interceptó una** máquina **Enigma**, enviada desde Berlín a Varsovia. Aunque no era una versión militar, permitió que un grupo de matemáticos polacos iniciaran los trabajos de investigación para "romper" la seguridad de Enigma y **desentrañar su funcionamiento**.

****Curiosidad:** durante la guerra civil española, el bando franquista contó con una docena de máquinas Enigma, vendidas por los nazis. Aunque fueron versiones obsoletas, dado que los nazis no se fiaban… En el museo histórico militar de Sevilla podemos encontrar una de ellas actualmente.

Se formaron varios **grupos internacionales de expertos en matemática y criptografía** de la época, entre ellos había especialistas españoles exiliados.
También entre estos grupos de expertos, destacó un matemático, criptógrafo, científico y filósofo británico, quizá **uno de los primeros informáticos de la historia**. Su nombre era **Alan Turing** y es considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna. Entre otras muchas cosas, es conocido porque:

- Desarrolló la conocida como **prueba de Turing**, un criterio según el cual puede juzgarse la inteligencia de una máquina, si sus respuestas en la prueba son indistinguibles de las de un ser humano. Utilizada en inteligencia artificial. Se usa para averiguar si quien nos responde a una serie de cuestiones es una persona o una máquina.

- Empezó a escribir un **programa de ajedrez** para un ordenador que aún no existía.
- Diseñó la **máquina de Turing**.
- Fue uno de los precursores del término **algoritmo** y su significado moderno.
- …

No obstante, uno de sus logros más importantes fue **liderar el equipo que consiguió descifrar Enigma**, consiguiendo así debilitar al ejército nazi y propiciando la **victoria de los aliados**, acortando la duración del conflicto bélico, se calcula que en **2 años**.

La **carrera** profesional de Turing se vio **truncada** cuando **lo procesaron por su homosexualidad**. Se le imputaron los cargos de «indecencia grave y perversión sexual» (los actos de homosexualidad eran ilegales en el Reino Unido en esa época), los mismos que a Oscar Wilde más de 50 años antes. **Falleció por envenenamiento con cianuro**, aparentemente tras **morder una manzana** envenenada que no llegó a ingerir completamente, en un contexto que se estimó oficialmente como **suicidio**.

****Curiosidad:** Una leyenda urbana asegura que el **logo de Apple** (mordisco de la manzana) rinde **homenaje a Alan Turing** y su suicidio comiendo una manzana envenenada con cianuro. Incluso, el arco iris en el logo sería un homenaje a la homosexualidad de Turing. Sin embargo, estas **suposiciones** fueron **desmentidas** por Rob Janoff, creador del logo de Apple. De hecho, los colores ni siquiera se muestran en el mismo orden que en la bandera arco iris, dado que ésta fue diseñada dos años más tarde de la creación de dicha imagen.

Vamos a imaginar que **formamos parte de un equipo de criptografía de la 2ª Guerra Mundial** y **debemos descifrar un mensaje interceptado al enemigo** para averiguar dónde y cuándo tiene previsto atacar. De ese modo, nuestros ejércitos podrán anticiparse, desplegando sus tropas y preparándose para eliminar los activos bélicos del oponente.

Sigue los pasos:

1. **Preparación:** descarga el código fuente 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#). **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:

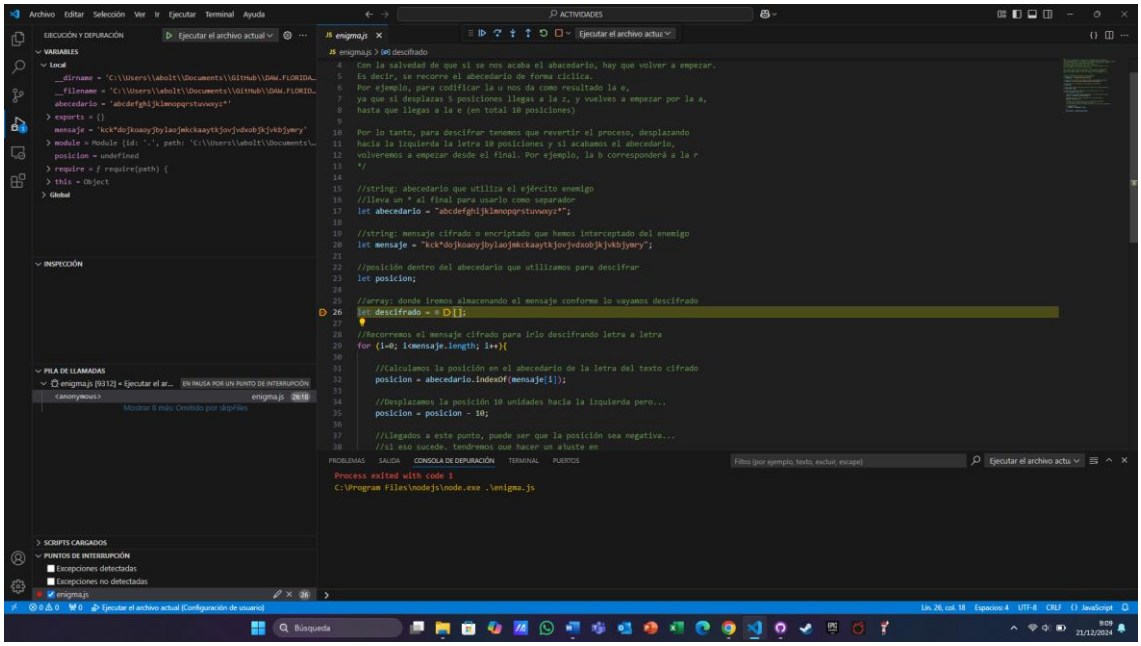
```
/*

 if (............){
  ......................................;
 }

*/
```

   Te facilito el esqueleto de la estructura de control comentada. Sólo tienes que sustituir las líneas de puntos por el código correspondiente, es decir, la **condición** del if y la **sentencia** del cuerpo del if**,** y quitar las marcas de comentarios. Ésta es la única modificación que podrás hacerle al código fuente que te facilito. **(2 puntos)**

2. **Descifrado**: una vez dispongas del código completo y correcto, tendrás que **depurar** para poder ir descifrando el mensaje letra a letra. Indica cuál es el **mensaje descifrado**. Durante la depuración, debes capturar un **mínimo de 8 pantallazos** (y un máximo de 12), correspondientes al proceso de descifrado, que argumenten **como estás realizando el proceso de depuración** y, además, debes ir proporcionando una **explicación de todo el proceso**. **(8 puntos)**

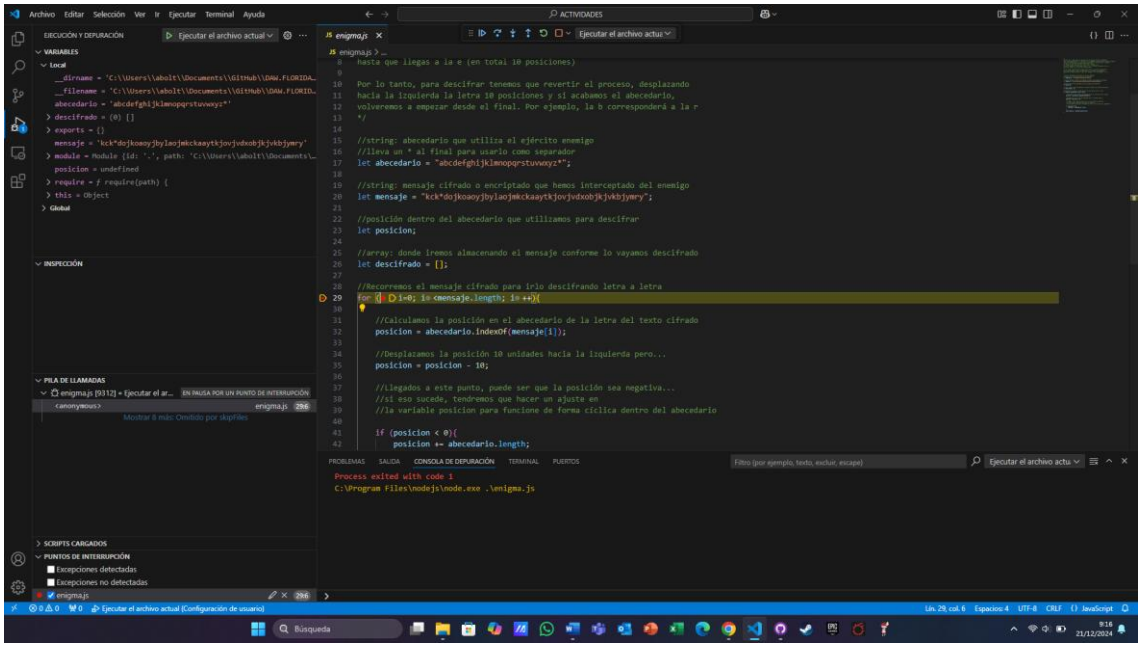
Mensaje cifrado: **ataque aereo sobre catarroja el lunes a las ocho**

Inicio programa



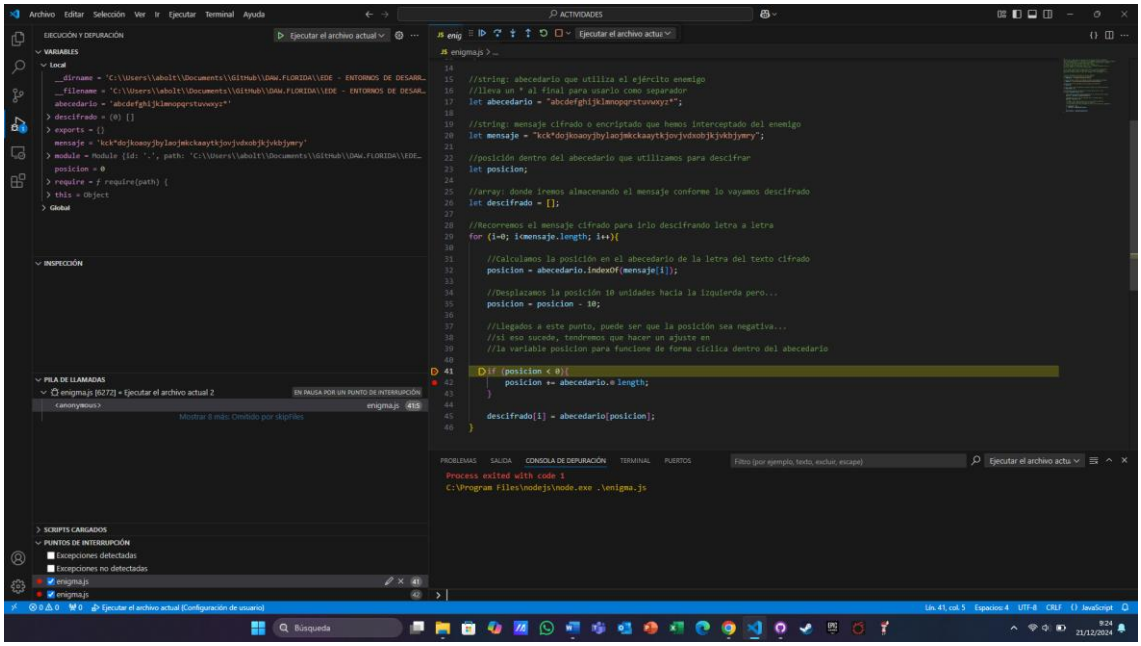
Empieza el programa, comprobamos las variables inicializadas, en este caso, abecedario, mensaje y descifrado. Punto interrupción en línea 26, donde comienza a almacenarse en un array el mensaje descifrado.

Primera iteración del bucle



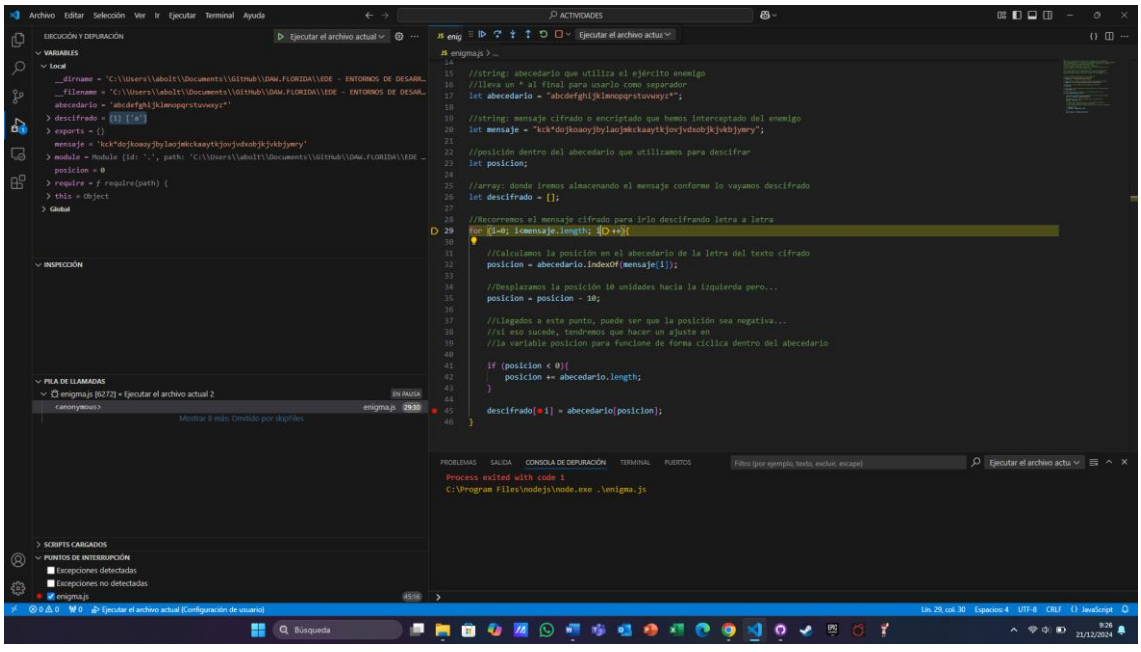
Primera iteración del bucle, comprobamos la posición en la que se encuentra el programa, en la posición inicial.

Ajuste de posición negativa



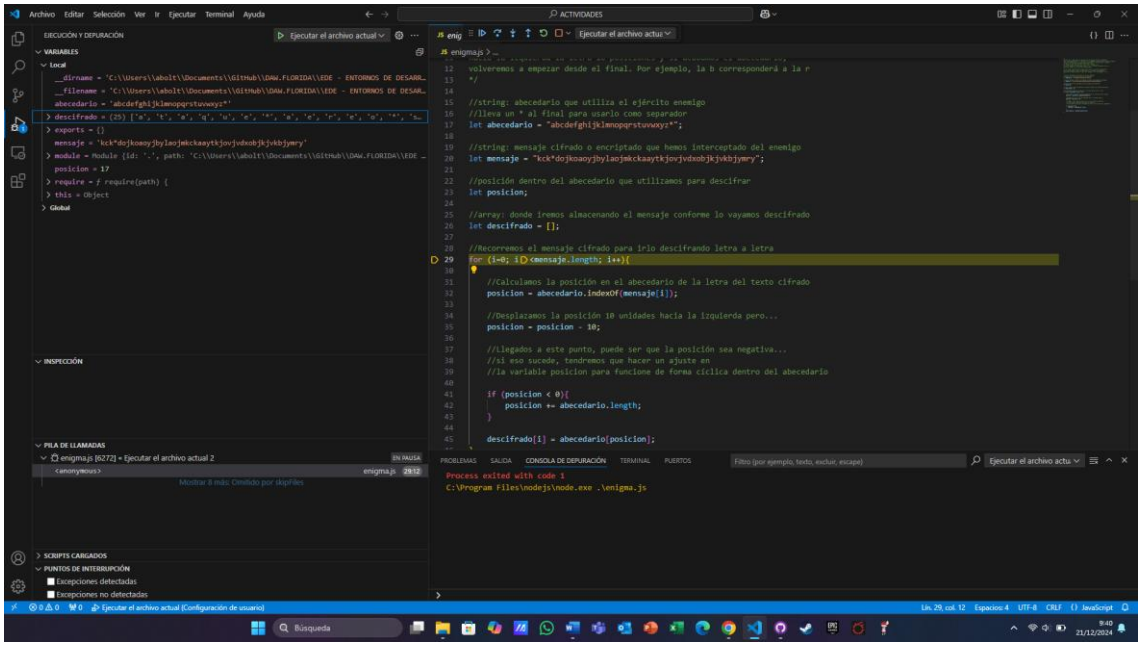
Aquí se muestra que el código funciona correctamente para manejar los casos cíclicos en el abecedario.

Almacenamiento de la letra descifrada



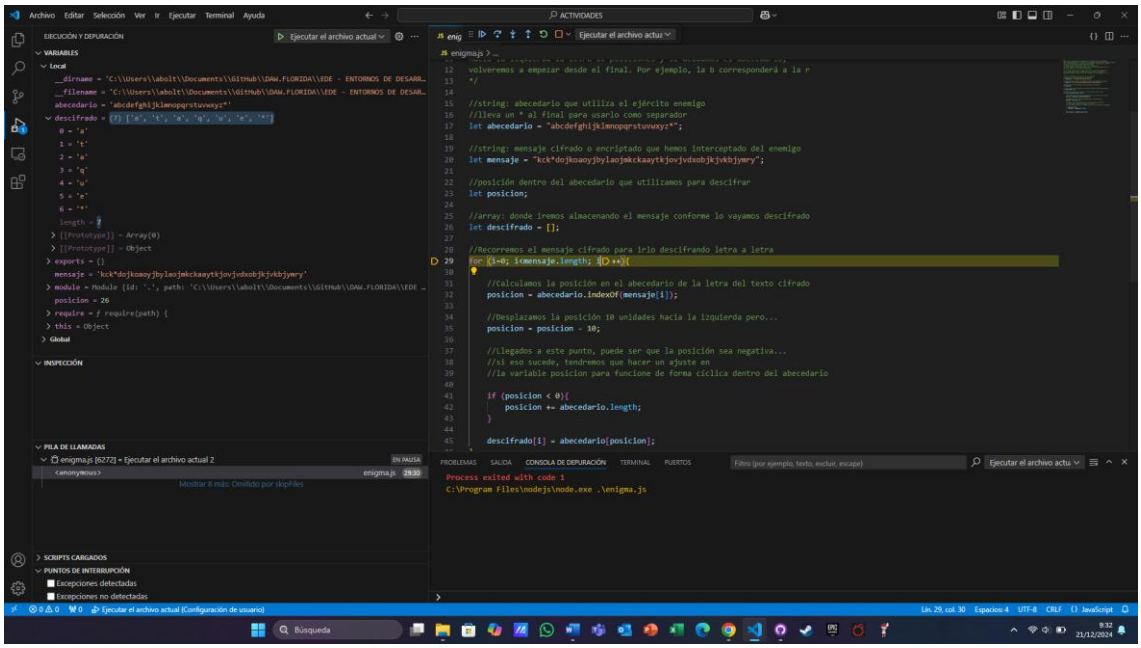
Aquí vemos como ha descifrado y almacenado la primera letra

Iteración a mitad del mensaje



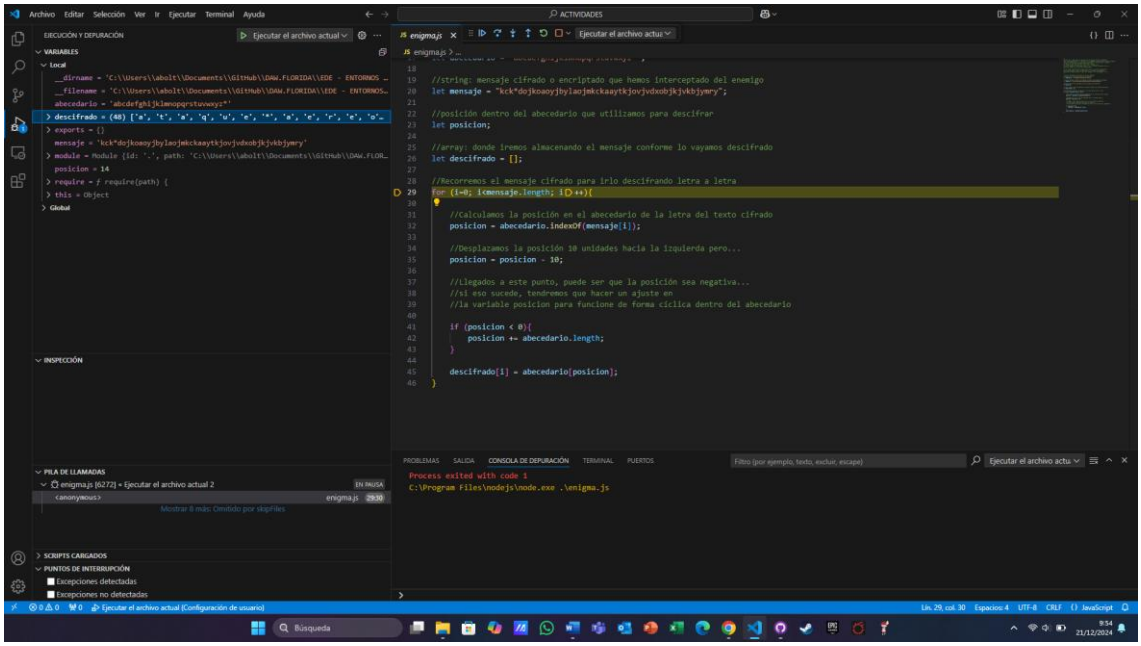
Comprobamos que el programa funciona correctamente, a mitad del mensaje aproximadamente, vemos que va descifrando el mensaje.

Detectar carácter separador



Aquí comprobamos que el programa detecta el carácter separador, como es “*”.

Ultima iteración del bucle



Después de esta iteración se acaba el programa, dando por finalizado el descifrado del mensaje.