**Actividad 10**

**CARRERA:** Ingeniería en Computación

**NOMBRE:** *Efrain Robles Pulido*

**CÓDIGO:** 221350095

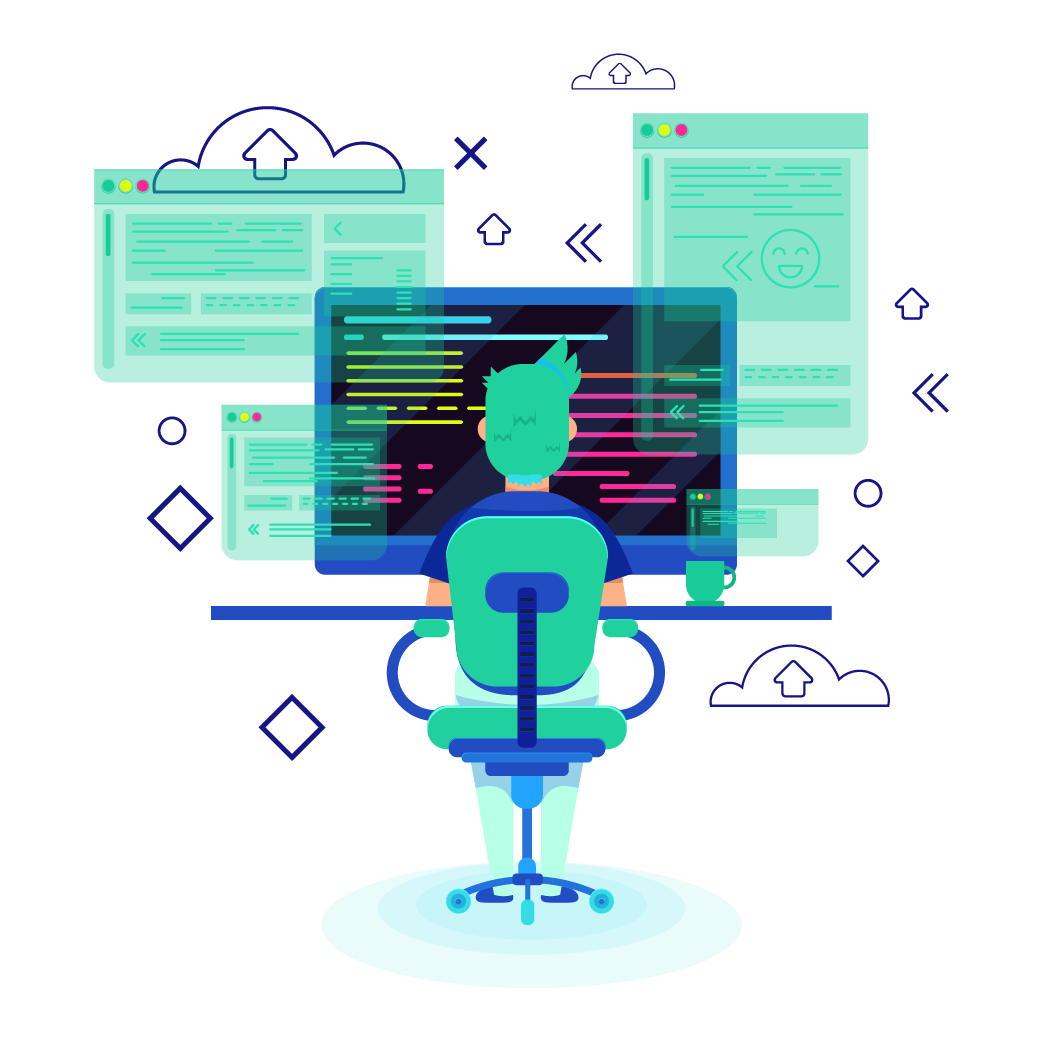
**MATERIA:** Seminario de Solución de Problemas de Traductores de Lenguajes I

**MAESTRA:** José Juan Meza Espinoza

**SECCIÓN:** D09 **CALENDARIO:** 2023A

**UNIVERSIDAD DE GUADALAJARA**



**e**

**Desarrollo**

* Dibujar un rectángulo.
* Posicionar con el mouse la coordenada 1 (x0, y0) y la segunda coordenada (x1, y1).
* Dibujarlo.

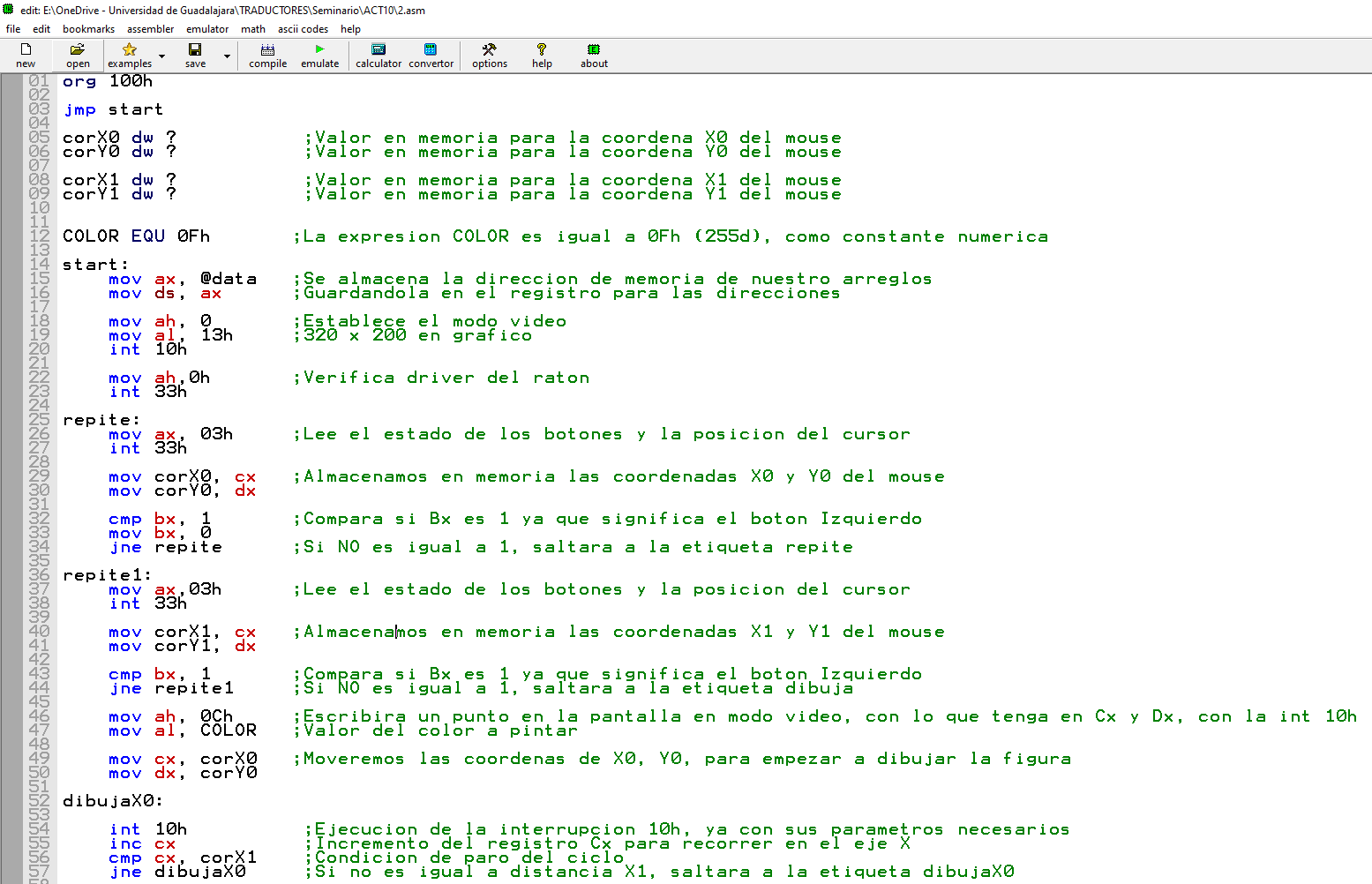
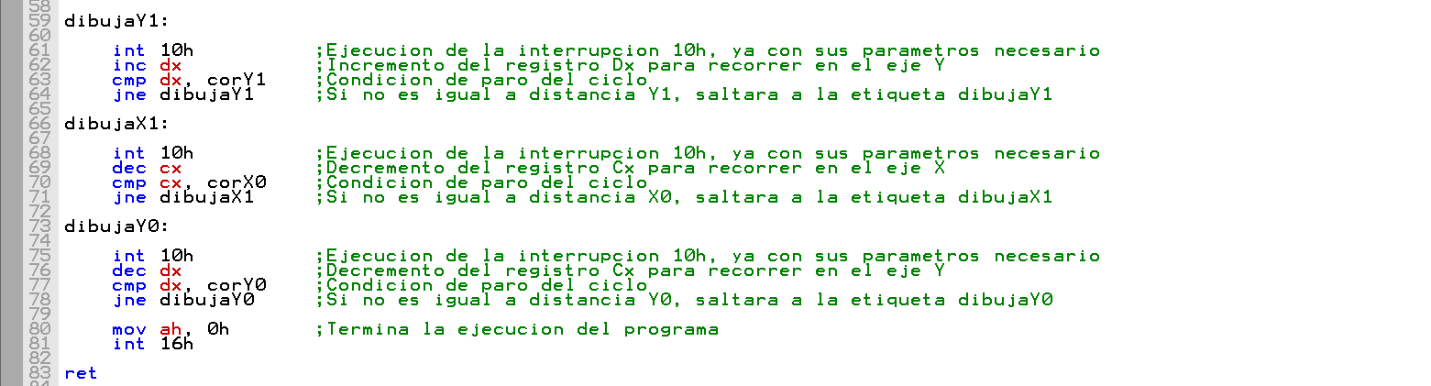
El código comenzará con el almacenamiento de la dirección de memoria de nuestras variables a utilizar, como las 2 coordenadas del mouse a utilizar y definiendo una constante numérica para determinar el color del píxel a pintar, después se establecerá el modo de video en Ah en 0 y su dimensión de la pantalla en Al en 13h mediante la interrupción 10h, después moveremos en Ah el valor de 0 para ejecutar la interrupción 33h para reiniciar y verificar el driver del mouse.

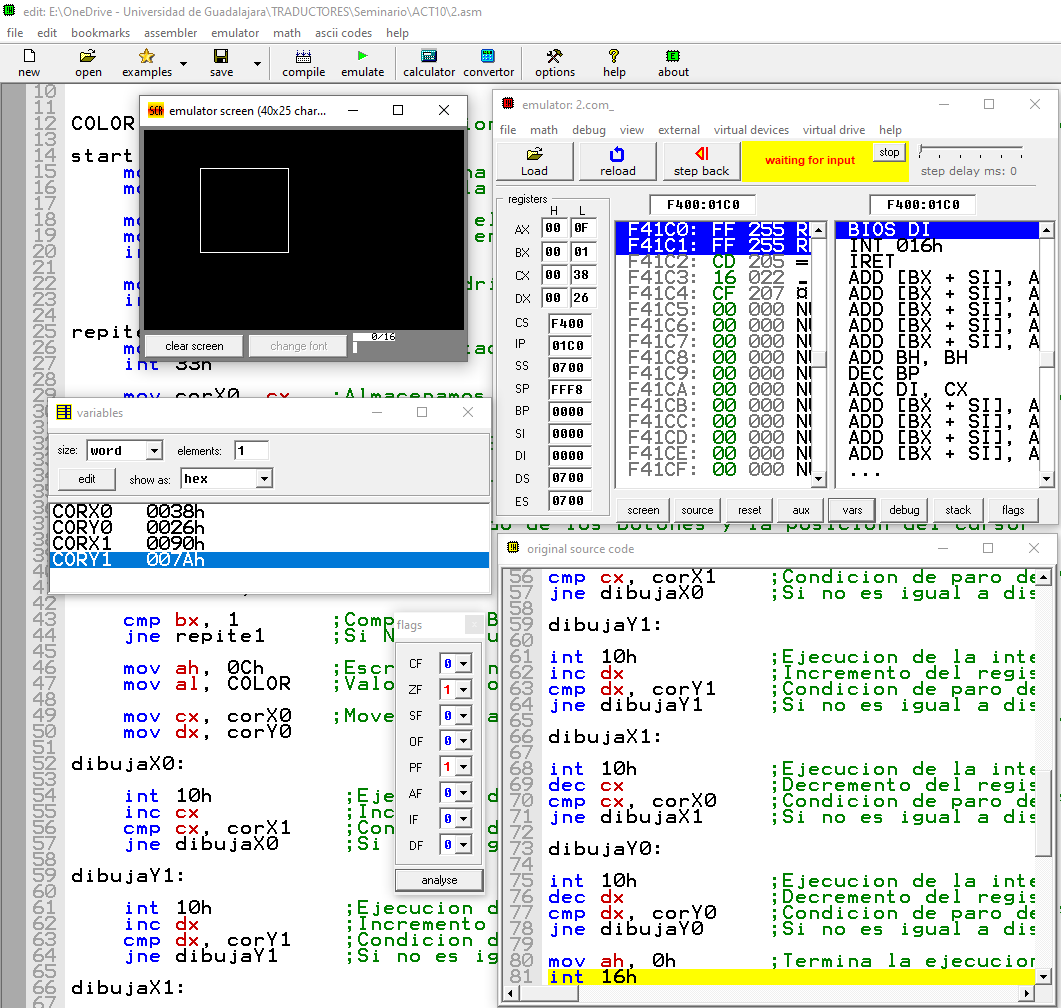
A continuación, se iniciará un ciclo en el que se estará moviendo el valor de 3h en Ax con interrupción 33h para leer el estado de los botones del mouse y su respectiva posición en la pantalla de video, en el que se estará almacenando el estado de los botones en el registro Bx, y la coordenada 0´s, X0 en Cx y Y0 en Dx, y a su vez se estará almacenando en memoria esas mismas coordenadas. También estaremos comparando si en Bx tenemos el valor 1 (botón izquierdo) para realizar el respectivo salto en “repite”, si no es igual a 1 se estará ciclando hasta apretar el botón del mouse.

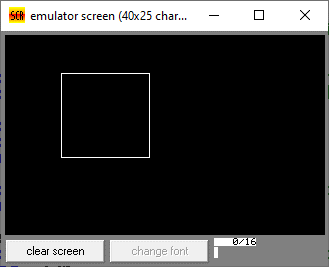
Una vez presionado el botón, pasaremos a otro ciclo en donde repetiremos el mismo procedimiento, pero ahora lo estaremos almacenando la coordenada 1´s, es sus respectivos ejes, X1 en Cx y Y1 en Dx, estará saltando en “repite1” para ciclar esta parte. A continuación, movemos 0Ch a Ah para configurar para escribir un punto en pantalla y moveremos el valor COLOR en Al para configurar el color de aquel punto a pintar, después moveremos las coordenadas X0, Y0 a Cx y Dx respectivamente para empezar a pintar este punto. En el que estaremos ciclando y incrementando en 1, el valor de la coordenada X0 (Cx) para irlo pintando continuamente con la condición de que no sea igual a la coordenada X1. Pasando a otro muy parecido en el que ahora es incrementara en 1 el valor de Dx (en el eje de las Ys) y se cambiara el valor de la condición con la coordenada Y1.

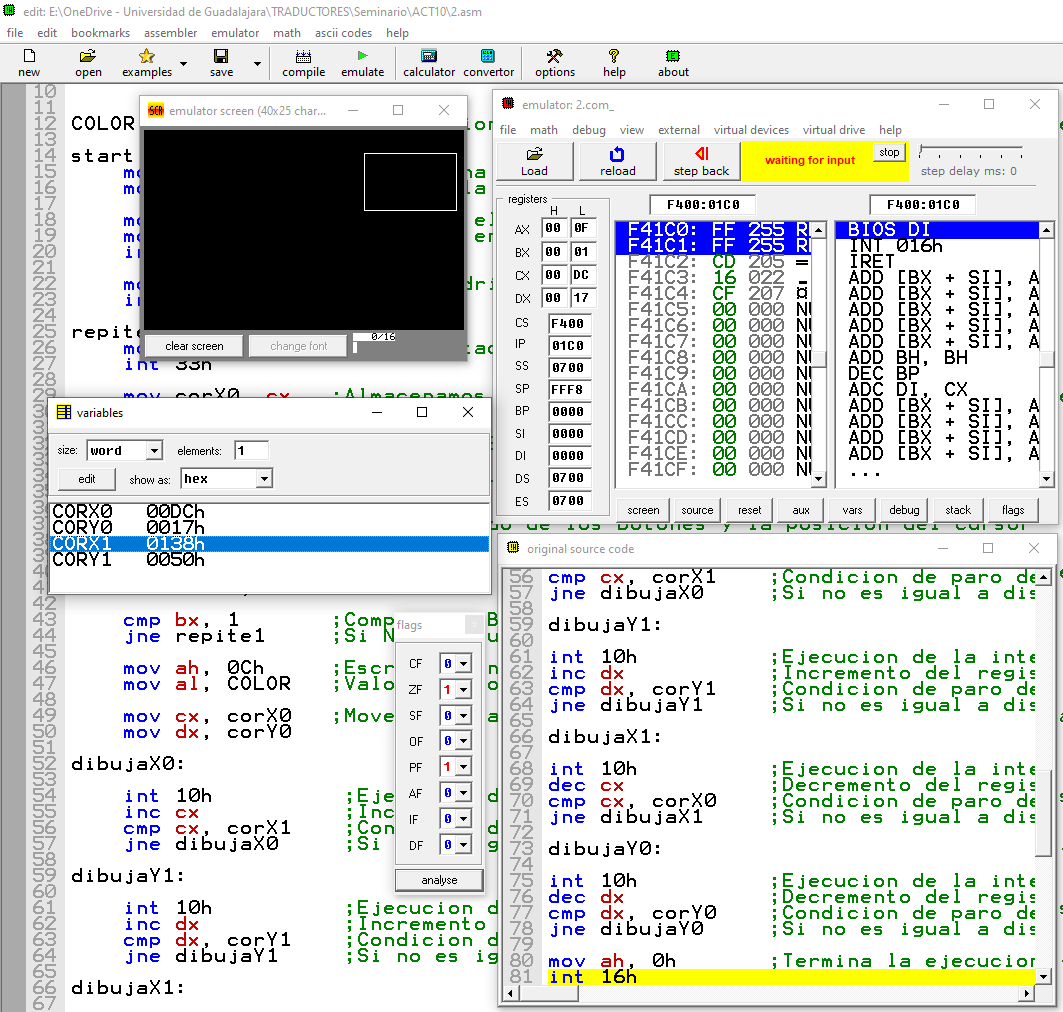
El siguiente ciclo es parecido a los anteriores, pero ahora se decrementará en 1 valor de Cx (eje de la Xs) hasta llegar a la coordenada de X0, y el siguiente ciclo con decrementara en 1 el valor en Dx(eje de la Ys) hasta llegar a la coordenada Y0.

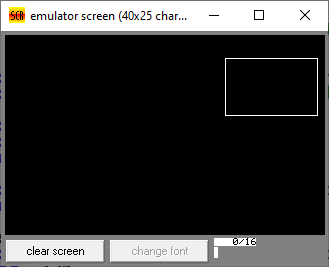
Por lo que cada ciclo tendrá su respectiva etiqueta para realizar sus ciclos individuales para formar el rectángulo. Finalmente terminara el código con la instrucción 0h de la interrupción 16h.

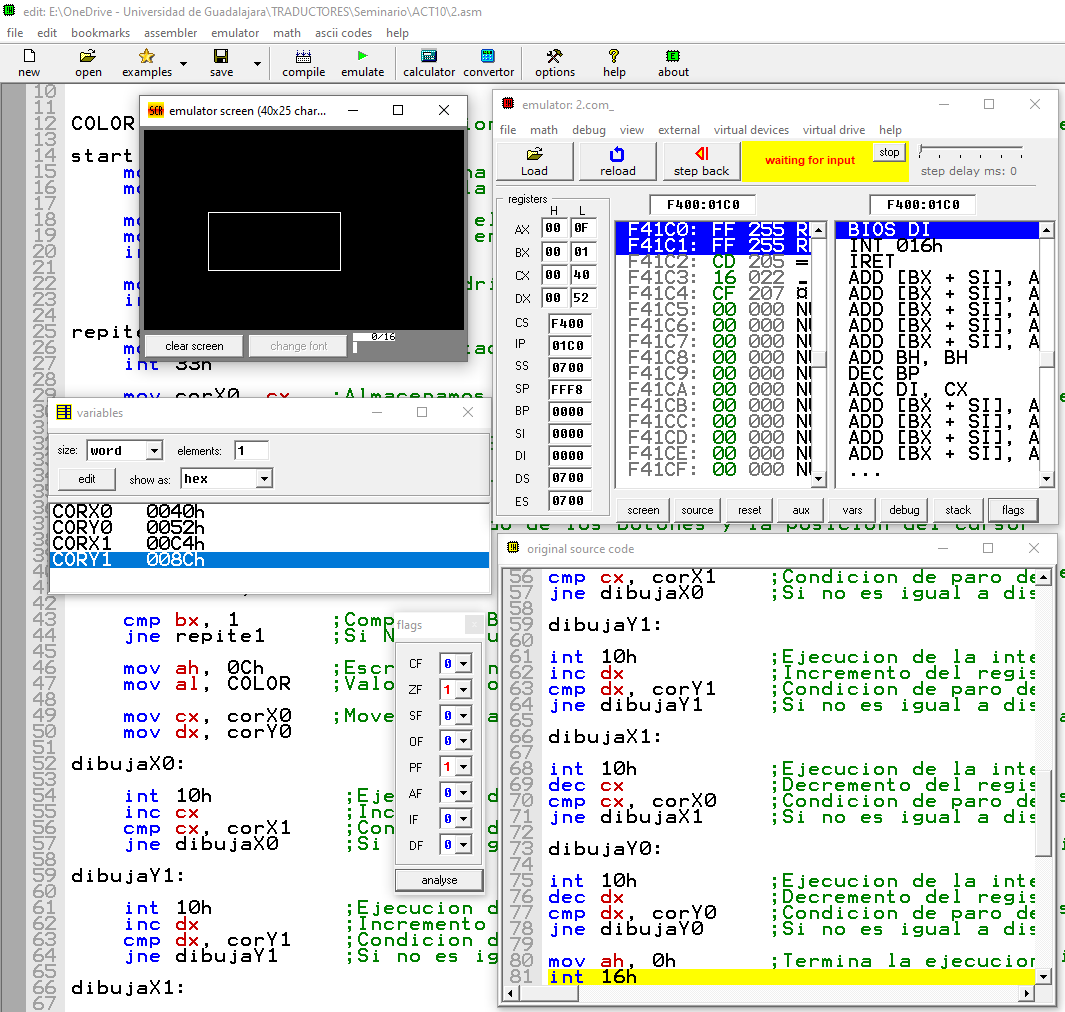


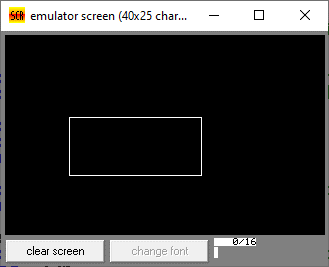


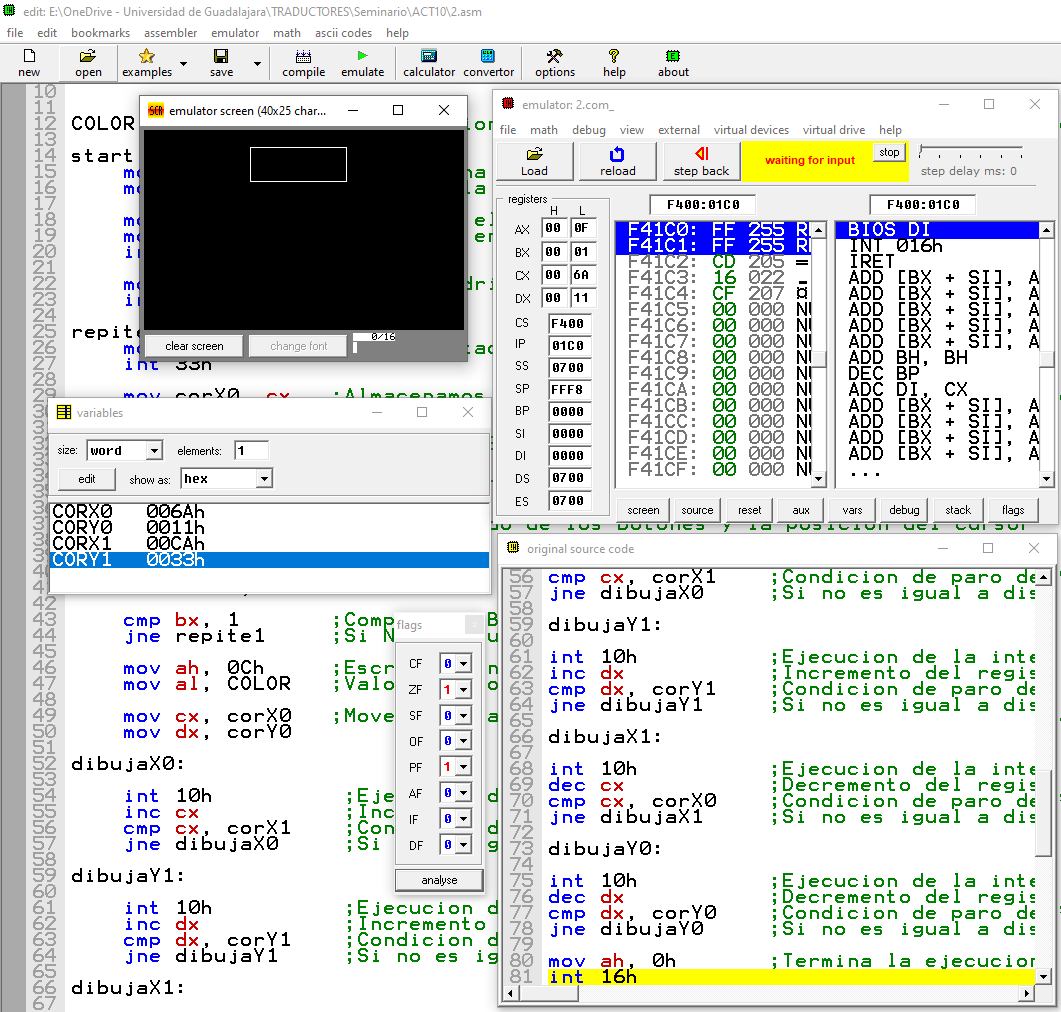


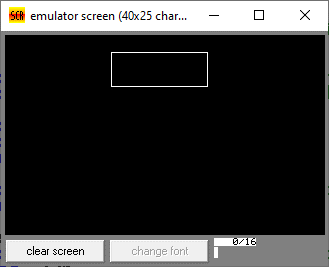












**Conclusión**

Al principio me costo trabajo como empezar a dibujar el rectángulo. Así que primero se empezó por realizar el código para almacenar las coordenadas de ambos puntos en la memoria en el que reutilice parte del código de la práctica anterior. Para después ir aumentando en uno en uno los puntos de cada eje respectivamente. Por lo que se iba recorriendo los ejes respectivamente para dibujar el píxel correspondiente, teniendo las coordenadas almacenadas como condiciones para parar o continuar dibujando. Entonces, una vez entendido la lógica se puedo realizar correctamente la práctica, así que considero que se podría hacer aún más eficiente el código, pero con este cumple su propósito.

**Bibliografía:**

Brey, B. B. (2006). Microprocesadores Intel - 7 Edición (7a). Pearson Publications Company.