**Actividad 8**

**CARRERA:** Ingeniería en Computación

**NOMBRE:** *Efrain Robles Pulido*

**CÓDIGO:** 221350095

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

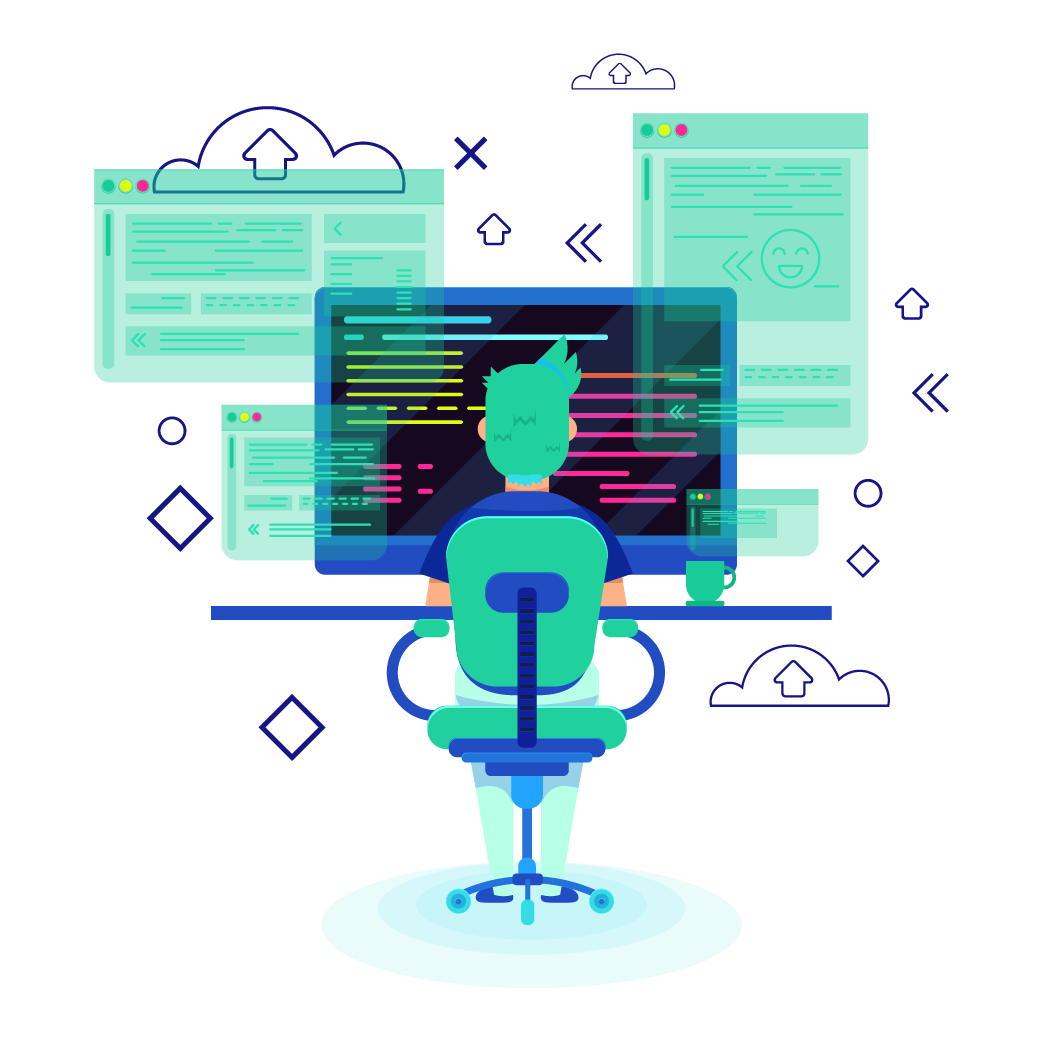
**MAESTRA:** José Juan Meza Espinoza

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

**UNIVERSIDAD DE GUADALAJARA**



**e**

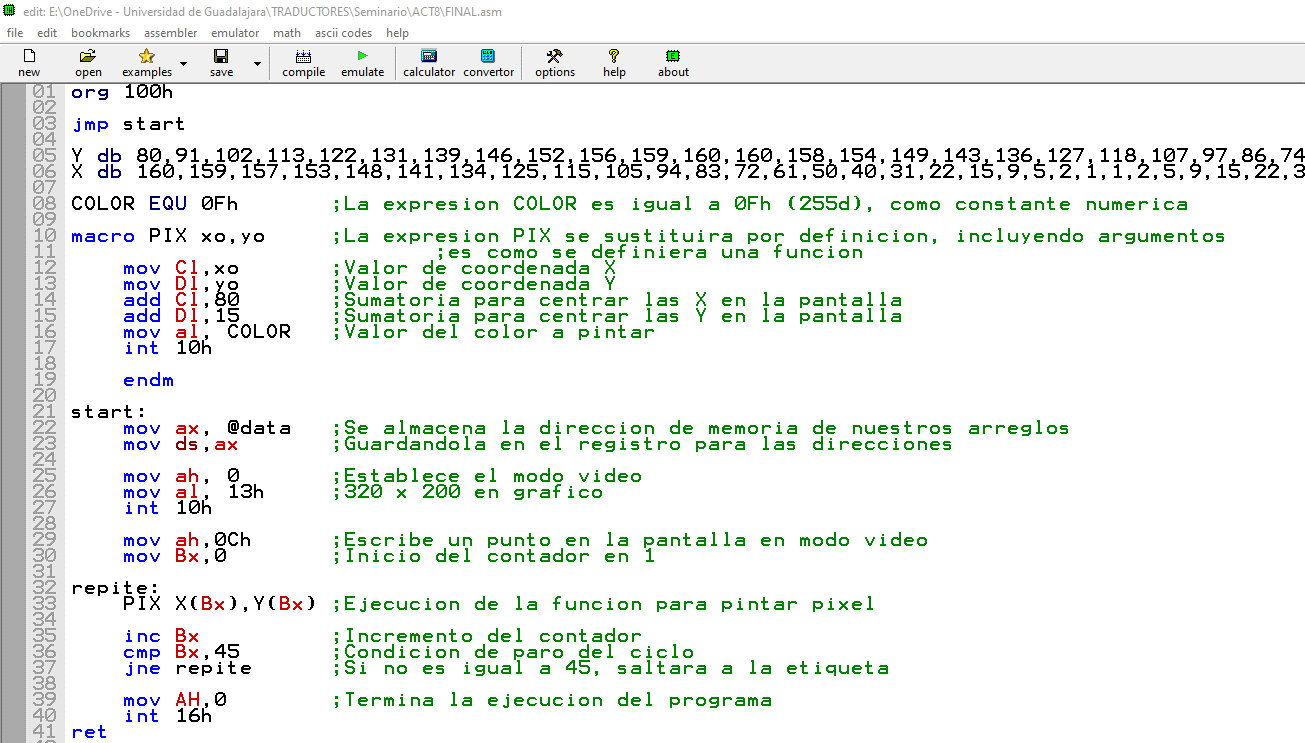


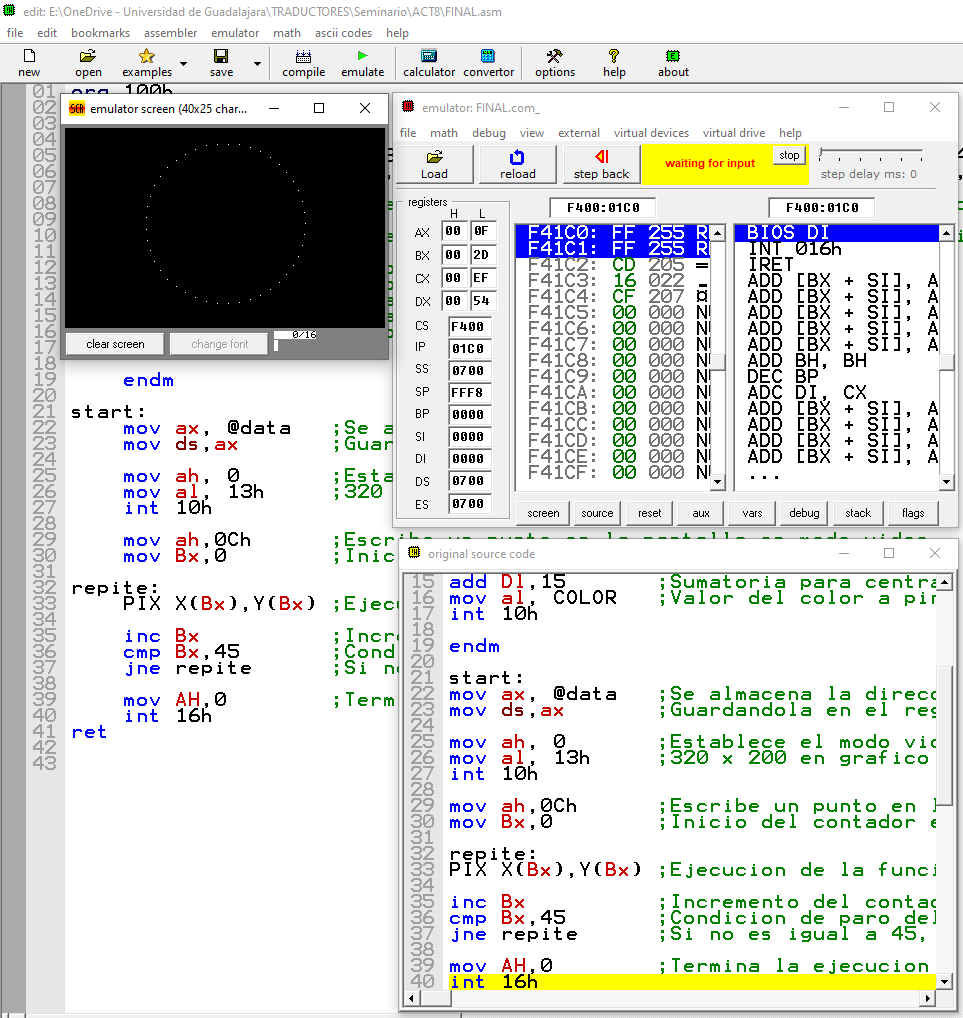
**Desarrollo**

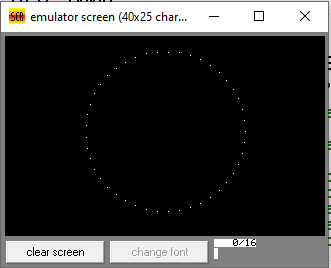
Obtendrás los datos de una función paramétrica de un círculo usando Excel, está la usaras para declarar un arreglo de números, que corresponden a los valores de n de X y Y

* Usando todos los modos de direccionamiento, y la interrupción 10h para modo gráfico.
* Y dibujaras la circunferencia

Primero se tuvo que utilizar un documento de Excel para generar los valores a utilizar en nuestros arreglos X y Y del código, en el que se utilizó una función para generar una figura con forma de circulo. El código comenzara con el almacenamiento de la dirección de memoria de nuestro arreglo, para luego empezar estableciendo el modo de video en Ah en 0 y su dimensión de la pantalla en Al en 13h mediante la interrupción 10h, después se moverá el valor a 0Ch en Ah para el resto del código, en el que se configura para que se escriba un punto en la pantalla. Además de que se iniciara un contador en 0 para acceder a nuestros arreglos en memoria. Después, se sustituye instrucciones por el nombre de PIX (utilizándolo como una función), para que se escriba un punto en una coordenada especifica mediante el direccionamiento hacia los arreglos en memoria de X y Y, mediante Cx para la coordenada “x” y Dl para la coordenada “y”, en el que cada coordenada será un valor respectivo a sus arreglos de acuerdo a nuestro contador y en Al se le asigna el color con el que se escribirá el punto (0Fh para el blanco), además de que se les sumo en la coordenada X y Y un valor en específico (X+80,Y+15) para que quedara centrado nuestra figura. Se estará incrementando nuestro contador por cada vez que se escriba un punto en pantalla hasta llegar a los 45, para finalizar con el código con la instrucción 0h de la interrupción 16h.







**Conclusión**

Esta actividad me pareció muy entretenida y un poco difícil de entender al principio debido a las interrupciones que teníamos que usar para entrar al modo video del EMU. Además de que al principio me salían algunos errores debido a la falta de memoria, pero esto era debido a que estaba almacenando grandes cantidades de datos para un valor que se puede utilizar en uno más pequeño, con el dw (2 bytes) del arreglo lo pase a db (1 byte) para que sea más eficiente el almacenamiento de los datos y se asignó cada valor de los arreglos s sus respectivos registros para pintar el píxel en pantalla bien. Así como determinar la cantidad correcta de cada elemento de los arreglos para que se dibujara correctamente el circulo, además de que si me daba 0 en alguna posición de los arreglos tenía que ponerlo en 1 para evitar fallas a la hora de pintar el píxel debido a las coordenadas que utiliza nuestra pantalla de video. Así como sumarle un poco a cada coordenada un valor en específico para que se pudiera centrar adecuadamente en pantalla.

**Bibliografía:**

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