

Proyecto 2 "Calculadora Gráfica"

Andrés Urbano Guillermo Gerardo

30 de Mayo del 2020

1 Introducción

1.1 Descripción del problema

Representar de manera gráfica una calculadoras que tenga las funcionalidades de suma, resta, división, multiplicación y modulo.

Además de realizar las operaciones matemáticas de manera eficiente, utilizando el lenguaje ensamblador con la arquitectura x86.

2 Desarrollo del problema

2.1 Planteamiento y justificación del problema

Para poder realizar nuestro programa necesitamos primero conocer todas las instrucciones de nuestro lenguaje ensamblador, conocer como representa cada numero de una computadora para que nosotros podamos manipularlo de manera adecuada. Primero para realizar nuestro programa tenemos que saber primero como es se operan los números con sus operadores en lenguaje ensamblador, después de eso como poder capturar las entradas de nuestro usuario.

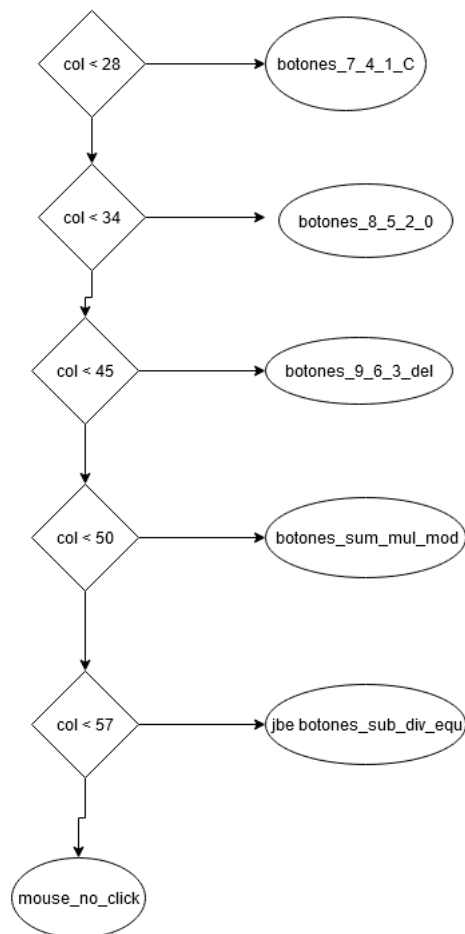
Después de eso saber como poder representar de manera gráfica una calculadora, saber como pintar un píxel en la pantalla, conocer las instrucciones e interrupciones que nos ayuden a graficar y poder representar los gráficos, además de eso, saber como poder leer las entradas del mouse y conocer toda la grilla de la pantalla.

2.2 Solución

Dado nuestro molde de calculadora, primero tuve que entender como se dibujaban los píxeles de la calculadora, respondiendo la pregunta de como se imprimía el numero, dado que el API del molde ya venia una función llamada IMPRIME BX, me ayude basándome en esa para poder imprimir los números, ahora en la primera parte teníamos métodos para detectar los botones del mouse y dado que en nuestro ya venia restringido la parte de donde no se ocupa los botones para que no se detectara el click y del mouse y fuera más óptimo, me concentre solamente en ese marco de píxeles, para hacer los condicioneles, medí

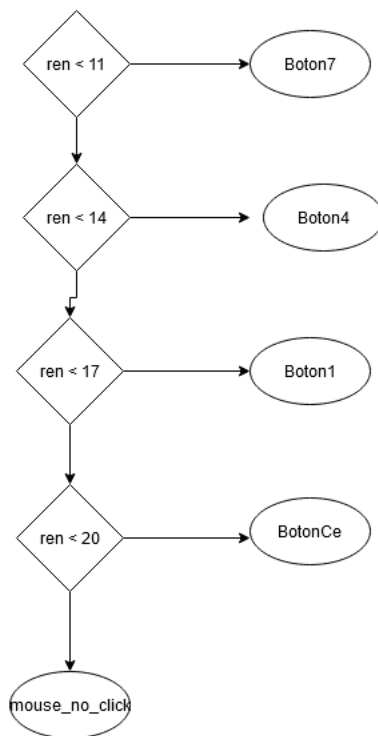
cada pixel para saber la ubicación correcta con respecto a los botones, poniendo condicioneles para poder decir que sí pasaba por tal píxel ya no correspondía con el botón sino con el margen, haciendo esto en cada botón pude detectar cada botón.

Comparamos cada columna para saber que si encuentra en esa, significará que habra ahi ciertos botones en los cuales haremos la comparacion con los renglones despues.



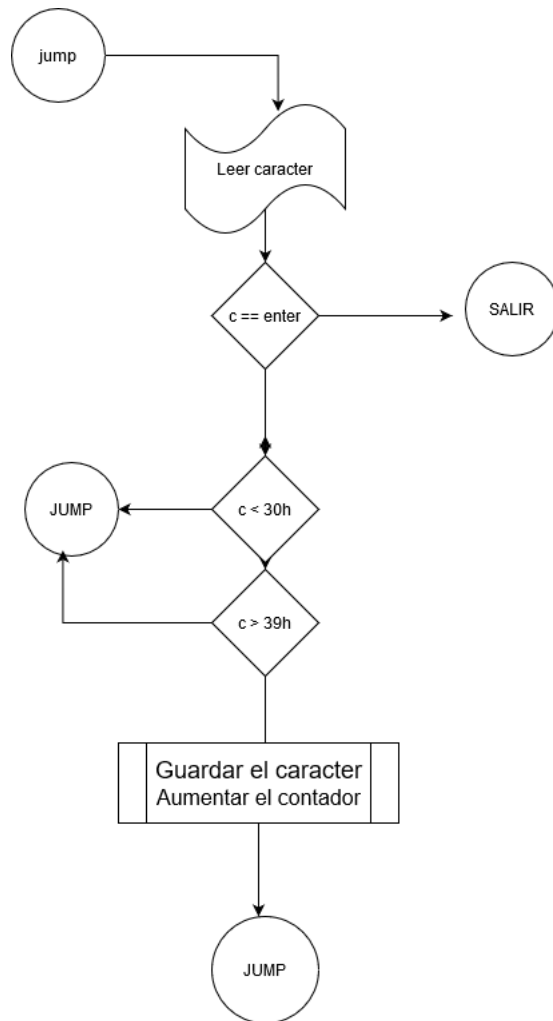
Una obtenidos los botones, me concentra ahora en la lógica de la calculadora, a cada botón le correspondía un valor para representar ese valor y con la función de imprimir bx, imprimí cada dígito de la calculadora, mostrándose su numero correspondiente.

Asi, para cada boton 8,5,2,9,6,3 y operandos +,-,/,%*
 Comparamos con cada renglon donde pertenece el mouse
 y verificamos si pertenece al boton



Una vez detectado los botones, cuando el usuario presiona un operador asignamos a nuestra variable operador un valor, cada operando le corresponde un numero para identificar que operando utiliza, para programar el segundo numero es igual de la misma manera que el numero num1 con los procedimientos de leer num1

Primero leímos las entradas del usuario, una de las restricciones que el usuario solo puede teclear 4 dígitos, entonces nosotros utilizaremos un contador para contar cada carácter que el usuario está tecleando, para la restricciones de que solo puede ir números, utilizaremos dos condiciones para acotar el rango de valores de nuestros caracteres que están representados con código ASCII, por lo tanto si queremos que usuario sólo puede escribir entre el rango de 0 a 9 tendremos que indicarle al programa que si su representación en código ASCII es menor 0 de su representación de código ASCII entonces que vuelva a leer el numero y para que no se ve en la pantalla utilizaremos la inserción de datos sin eco, y también tiene que cumplir con la condición de que si es mayor de 9 de su representación en código ASCII que vuelva a leer el número. Así podemos leer nuestros dos dígitos.

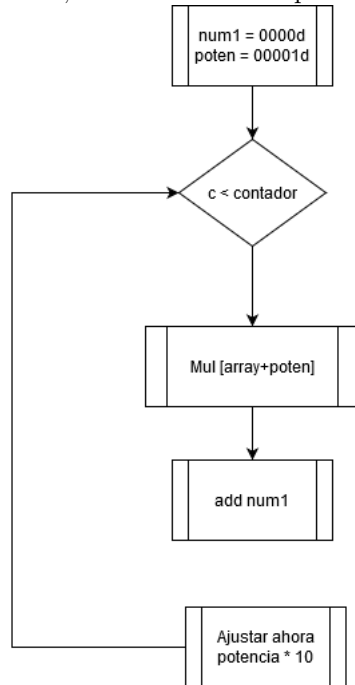


Una vez obtenidos nuestros dos dígitos podemos guardar nuestros números en un array que sería un lugar en la memoria con segmentos contiguos, ahí los almacenaremos y le restaremos 30h para convertir nuestros números ASCII en número decimales y es 30h porque la representación de nuestro carácter más bajo.

Ahora cuando el usuario presione el botón de igual, ahí es donde se realizara toda la lógica del programa, primero vamos a limpiar la pantalla donde se colocara el resultado por si el usuario quiere volver efectuar otra operación y borrar el resultado anterior, Después de eso tenemos que convertir nuestros dos número que están en formato ASCII en números que pueda decimales para que el computador pueda realizar las operaciones, para eso diseñamos un algoritmo que utiliza el fundamento del sistema de posicionamiento de un número.

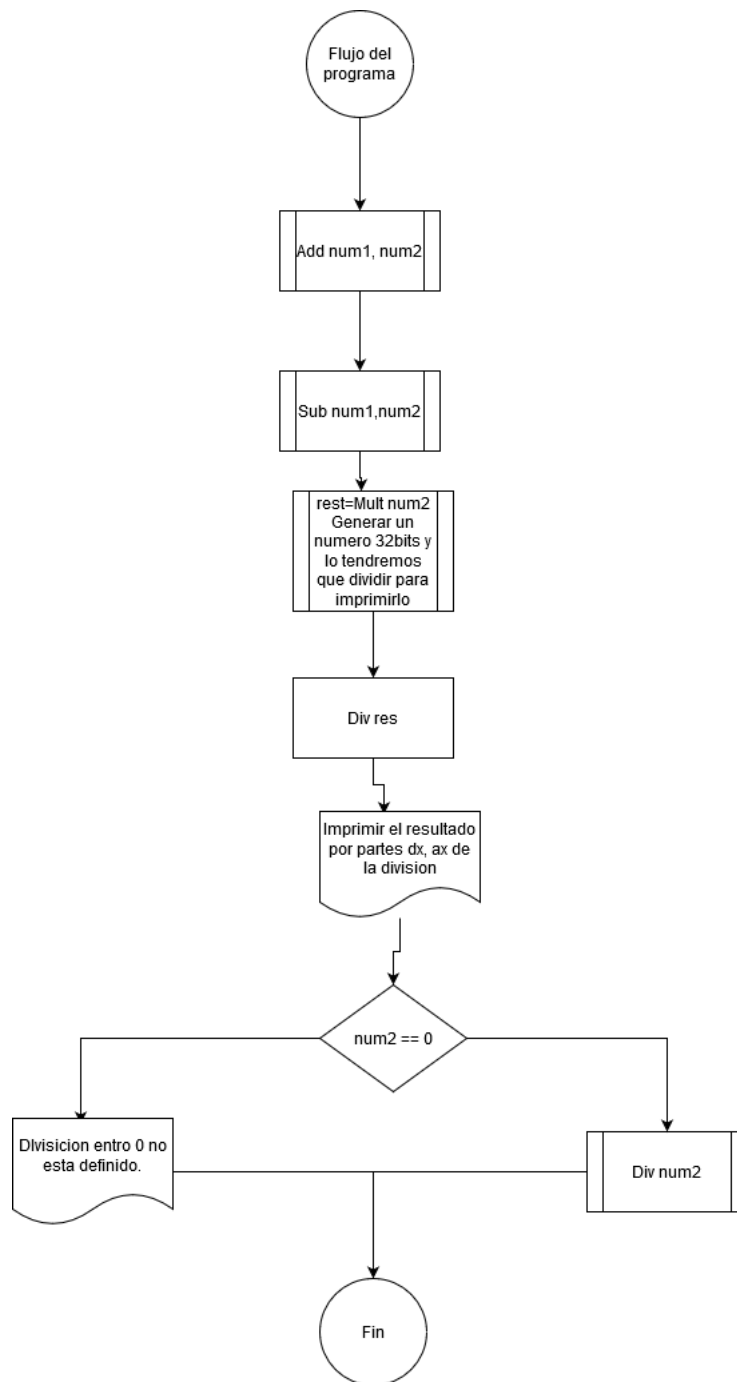
Ahora para darte una la representación final necesitamos multiplicar cada dígito por su respectivo sistema posicional para representar unidades, decenas,

centenas y milésimas, Para eso utilizamos un ciclo en cual, en cada posición de nuestro arreglo, será multiplicado por 10 elevado a la potencia de nuestro contador de nuestro ciclo. Después de obtener nuestros dos números de misma manera, ahora más fácil manipularlo ya que lo tenemos guardado en nuestra memoria, ahora solo nos ocuparemos con las operaciones.



Una obtenidos nuestros números realizamos nuestras operaciones comparando el valor del operando, para eso poniendo muchos saltos para ir comparando cada caso.

Para sumas solo guardamos nuestro primer numero en un registro para luego realizar el nemónico add, para la resta se realizar el mismo procedimiento.



Para en el caso de multiplicación el producto de nuestros dos dígitos será un valor mayor de 16 bits por lo tanto para poder imprimir ese valor tuvimos

que dividirlo entre 10 000 para poder separarlo en dos partes de 16 bits y poder imprimir cada parte, Para la división tenemos que considerar dos cosas si el divisor que tecleo el usuario es cero entonces imprimimos un letrero diciendo que la división de cero no está definida, y si no es cero hacemos la resta con respectivo nemónica, el cociente se guarda en ax y el residuo dx y lo imprimimos con nuestra función de impresión.

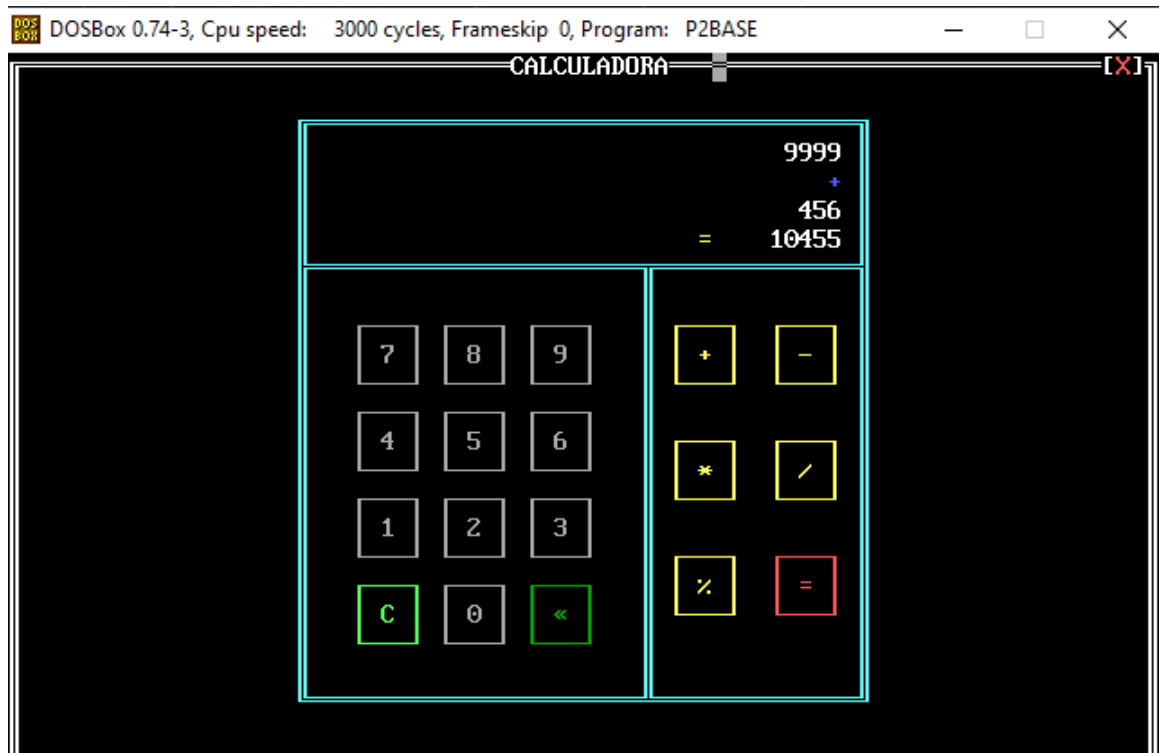
Para cada resulta solo tenemos que guardarlo en el registro BX para poder utilizar el procedimiento de imprimir el numero, poniendo la ubicación de renglón-columna para indicar la ubicación de la pantalla para ser impreso.

Para los botones de Ce que se ocupa de limpiar la pantalla solo fue necesario ocupar la función de limpiar la pantalla que viene implementada en el API del molde de la computadora, lo cual hacer limpiar cierta región de la pantalla.

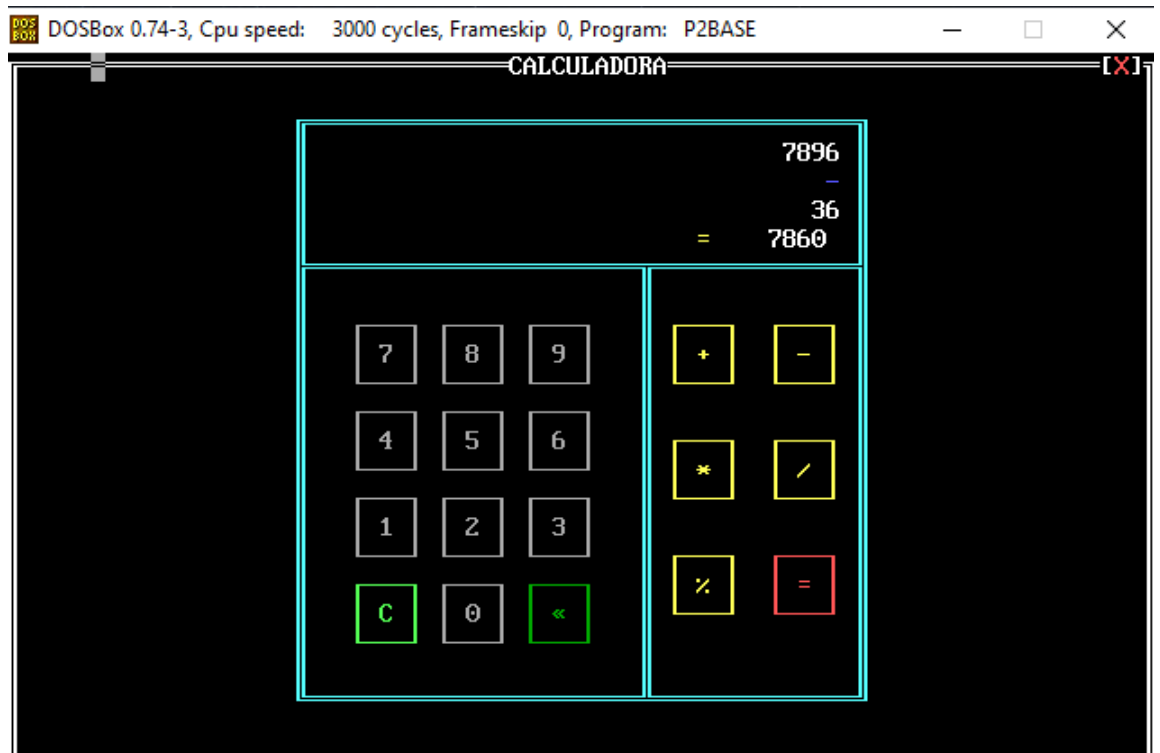
Para el botón de borrar un carácter, ese fue un poco mas complejo, fue el ultimo de implementarse, cuando es detectado lo que hacemos primero es identificar si solo hay un numero si es así, hacemos que el contador se inicialice en ceros y después imprimimos un espacio en vacío, en el caso contrario de que no haya un solo carácter, le restamos 1 al contador que va rastreando las posiciones de los números y después llamamos imprimir la función de imprimir BX, observaremos que se imprime correctamente los números y que no se borro el numero izquierdo de la ultima posición, debido a eso , debemos de imprimir un carácter vacío en esa posición porque al final ese numero no pertenece al numero, y eso surge porque la función imprimir numero lo que hace interiormente es volver a imprimir numero y coloco restando el valor del contador actual con una posición fija y debido esto siempre va sobra un numero, por eso imprimimos un carácter vacío para que no se vea,

3 Pruebas

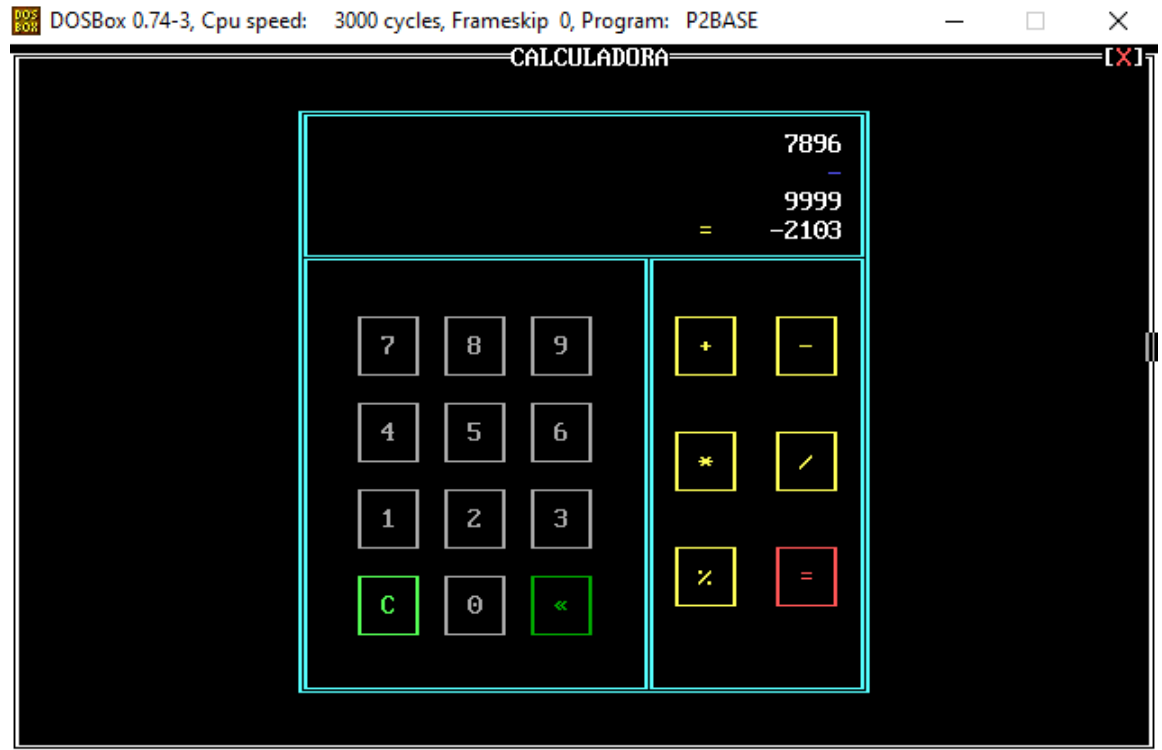
3.1 Suma



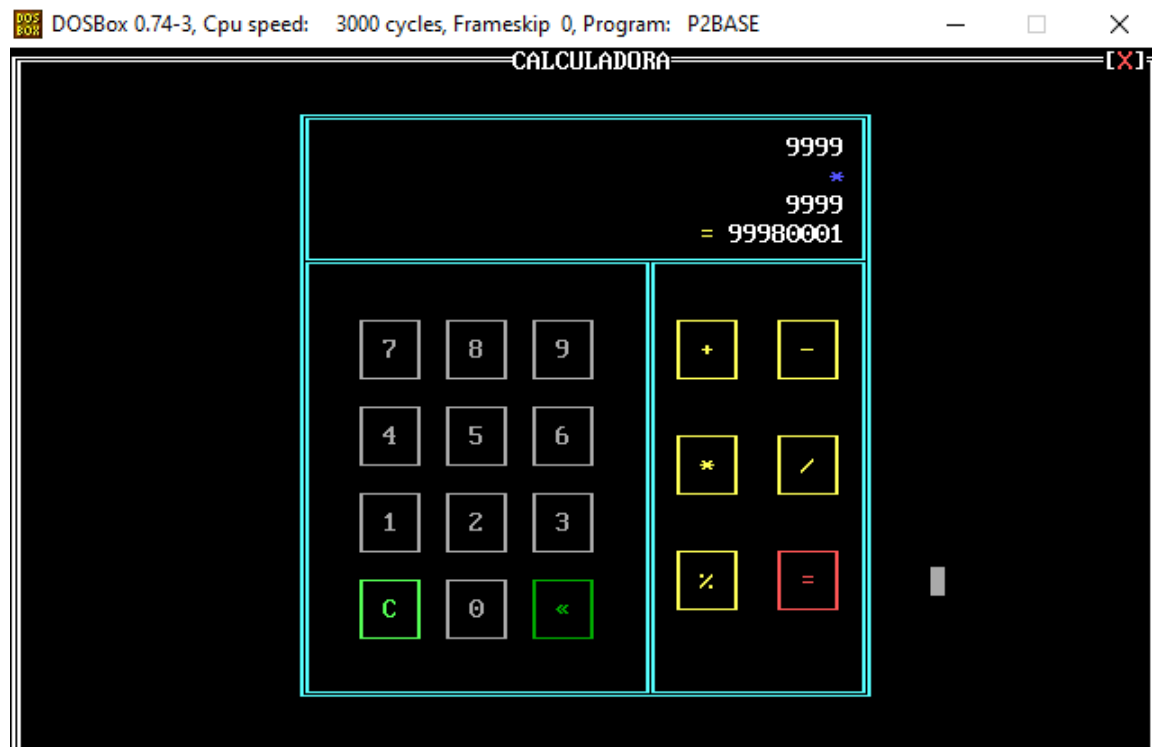
3.2 Resta



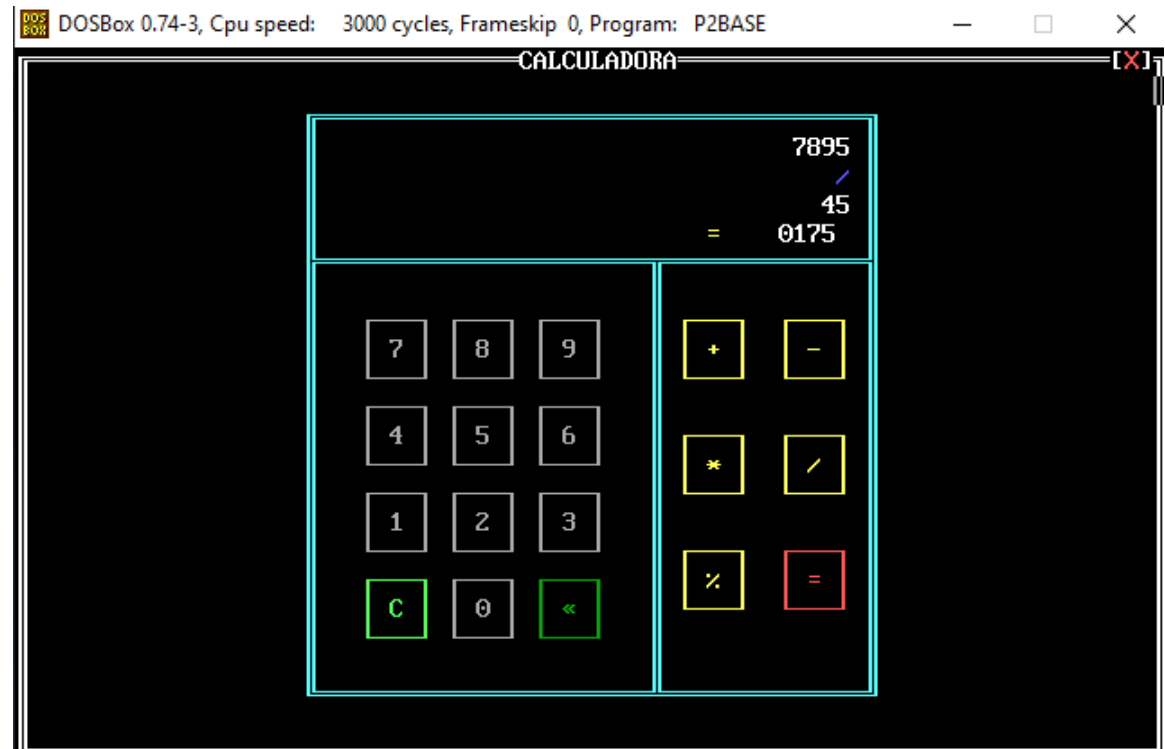
3.3 Resta con signo negativo



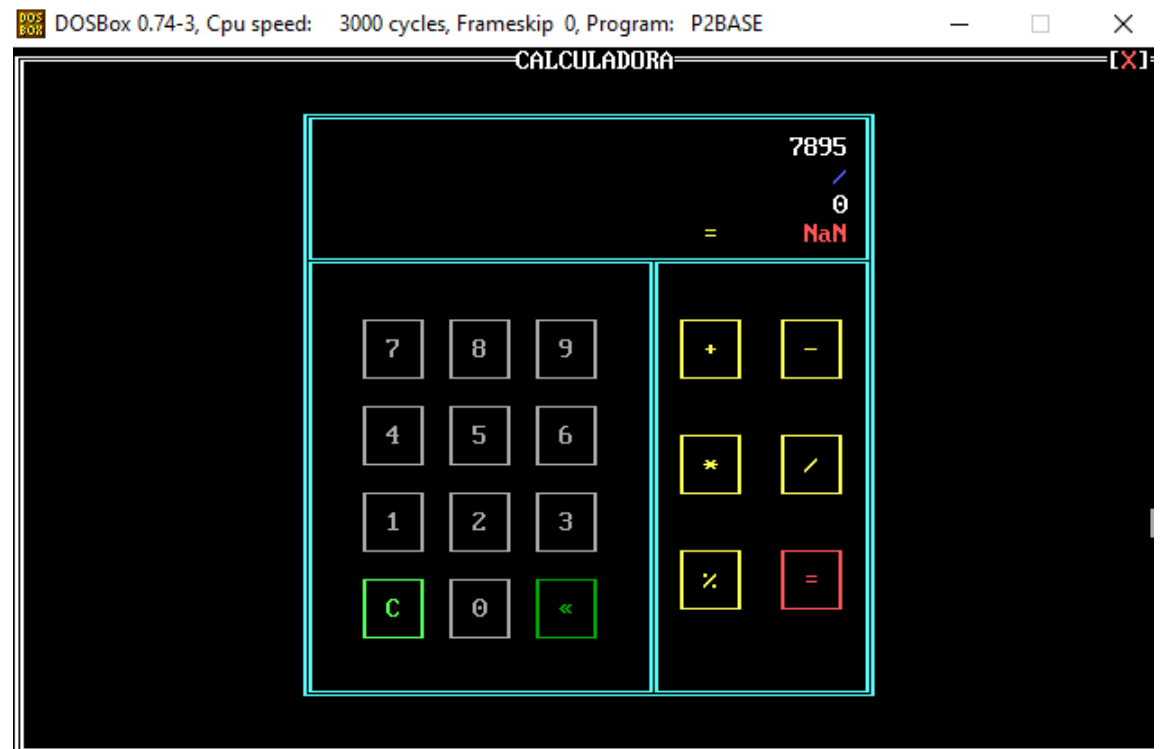
3.4 Multiplicación



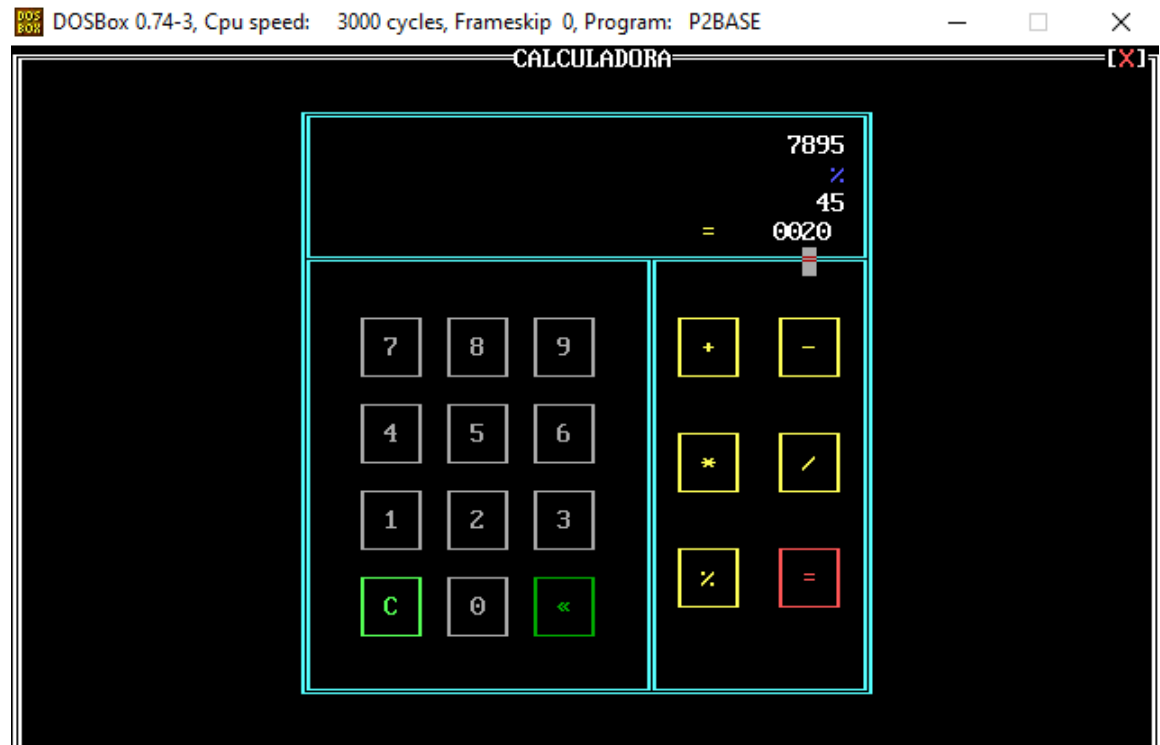
3.5 División



3.6 División entre cero



3.7 Residuo



4 Conclusión

Se pudo lograr el objetivo, se pudo construir de manera eficiente una calculadora con interfaz gráfica con cada uno de su funcionamiento, aprendimos a manejar correctamente las interrupciones, ver como funcionan los gráficos en lenguaje ensamblador, hacer interacción más a mena con el usuario, la dificultad que me encontré fue en el botón de retroceso, ya que tenía que analizar el funcionamiento de la impresión del numero, las operaciones no hubo ninguna dificultad debido que ya teníamos el proyecto anterior y sabia el funcionamiento de las operaciones.

Pudimos entender mejor las bases del lenguaje ensamblador, entendimos mejor la parte gráfica que con lleva programarla, el saber la ubicación de los píxeles respecto a la pantalla como también aprendimos entender mejor las interrupciones que contiene la computadora.