

Proyecto1 "Calculadora"

Andrés Urbano Guillermo Gerardo

April 2020

1 Introducción

1.1 Descripción del problema

Nuestra tarea será realizar una calculadora que nos permita a nosotros realizar operaciones matemáticas de manera eficiente, utilizando el lenguaje ensamblador con la arquitectura x86.

1.2 Planteamiento del problema

Realizar un programa que nos ayude realizar operaciones matemáticas sumas, restas, divisiones y multiplicaciones.

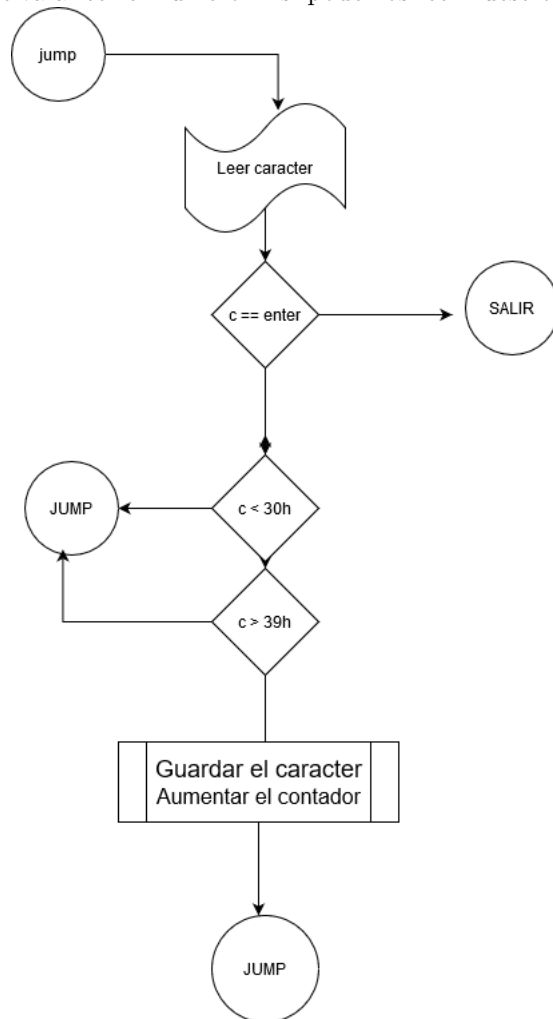
2 Desarrollo

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.

2.1 Solución

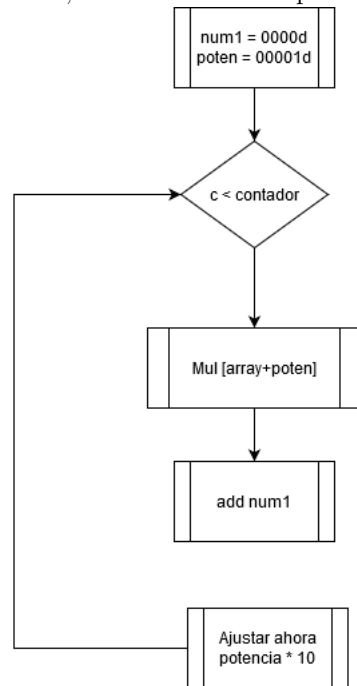
Para eso nosotros haremos varios procedimientos que nos ayuden a simplificar el problema y varias etiquetas para poder movernos en el programa según sea nuestras condiciones que definamos. Primero leímos las entradas del usuario, una de las restricciones que el usuario solo puede teclear 4 dígitos, entonces 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 número 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.

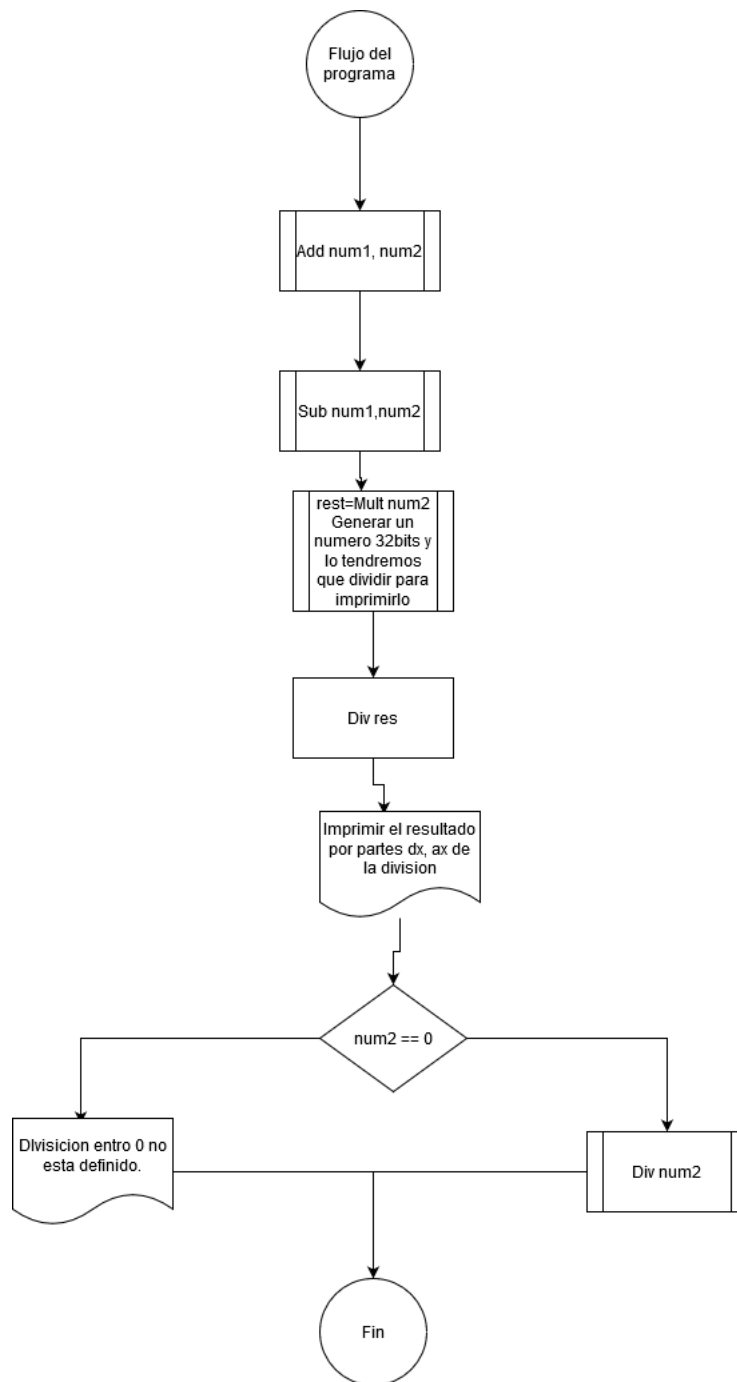


Una vez obtenidos nuestros dos dígitos podemos guardar nuestros números en un array que sería un lugar en la memoria con segmentos contiguos, ahí los almacenaremos y le restaremos 30h para convertir nuestros números ASCII en número decimales y es 30h porque la representación de nuestro carácter más bajo.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



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



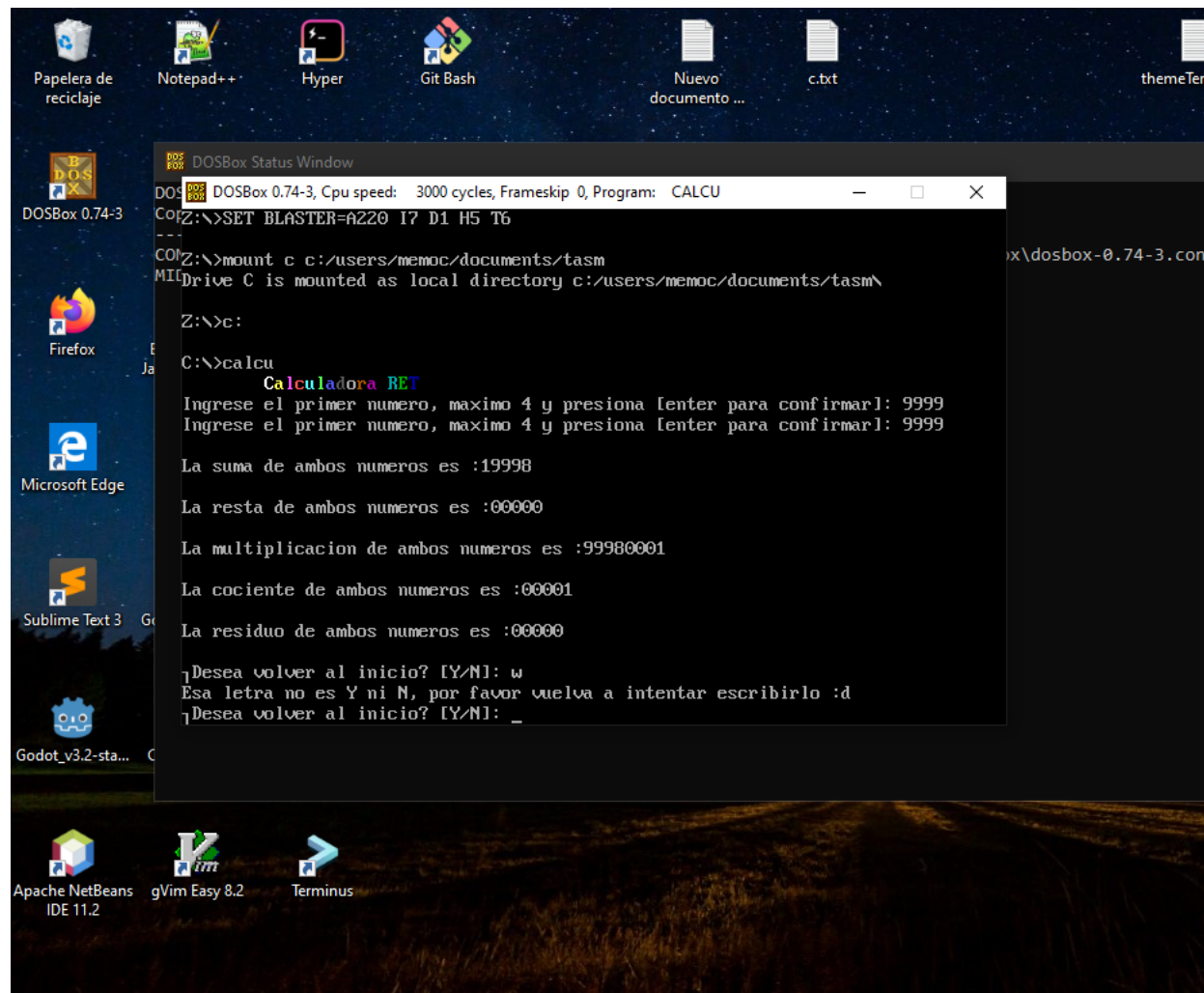
Para en el caso de multiplicación el producto de nuestros dos dígitos será un valor mayor de 16 bits por lo tanto para poder imprimir ese valor tuvimos

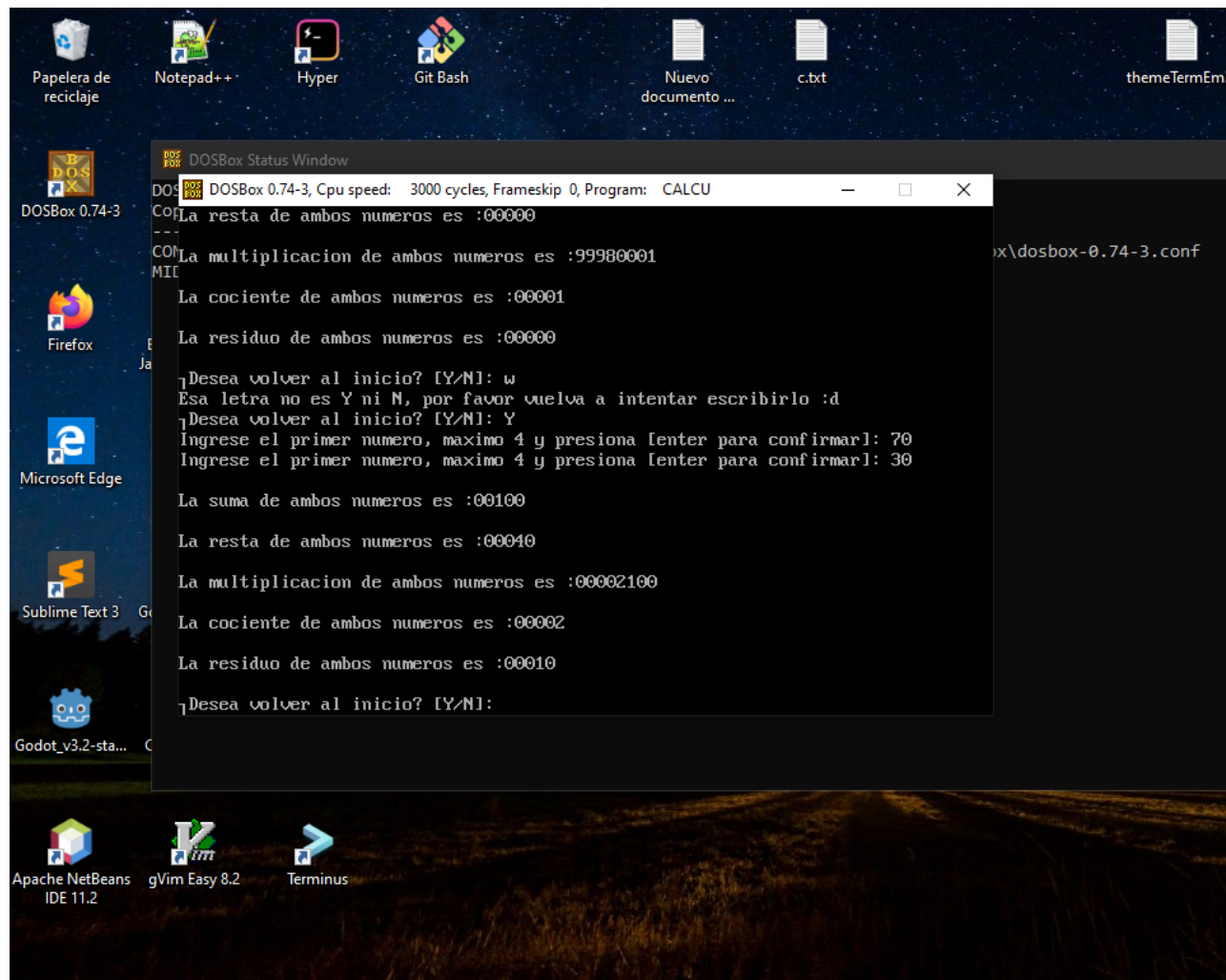
que dividirlo entre 10 000 para poder separarlo en dos partes de 16 bits y poder imprimir cada parte, Para la división tenemos que considerar dos cosas si el divisor que tecleo el usuario es cero entonces imprimimos un letrero diciendo que la división de cero no está definida, y si no es cero hacemos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

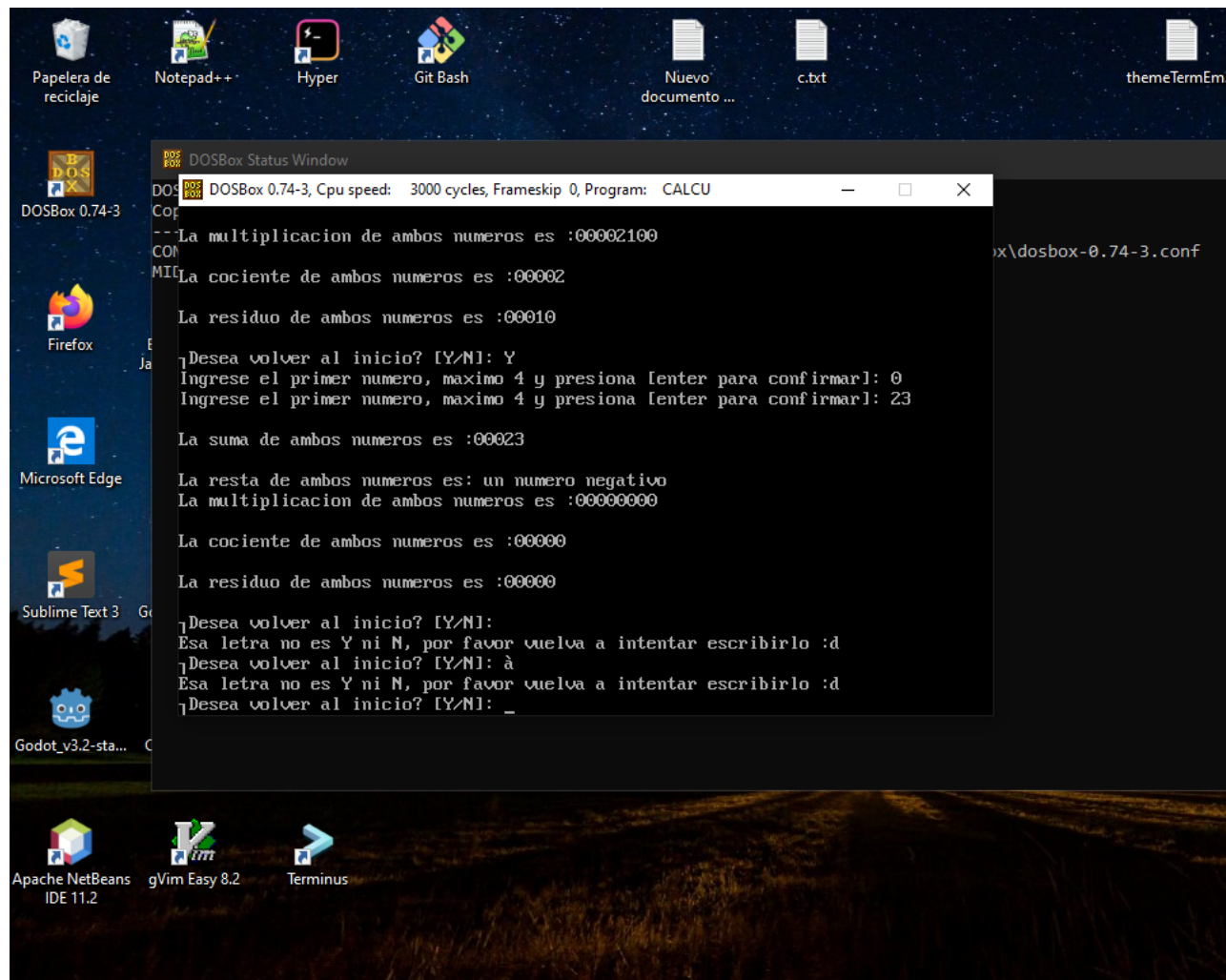
Para finalizar el programa le indicamos al usuario si quiere continuar, para eso leemos un carácter del teclado y hacemos la comparación si es Y saltamos al inicio del programa para volver a reiniciar, y si es N no saltamos y se dejamos que termine la ejecución correctamente.

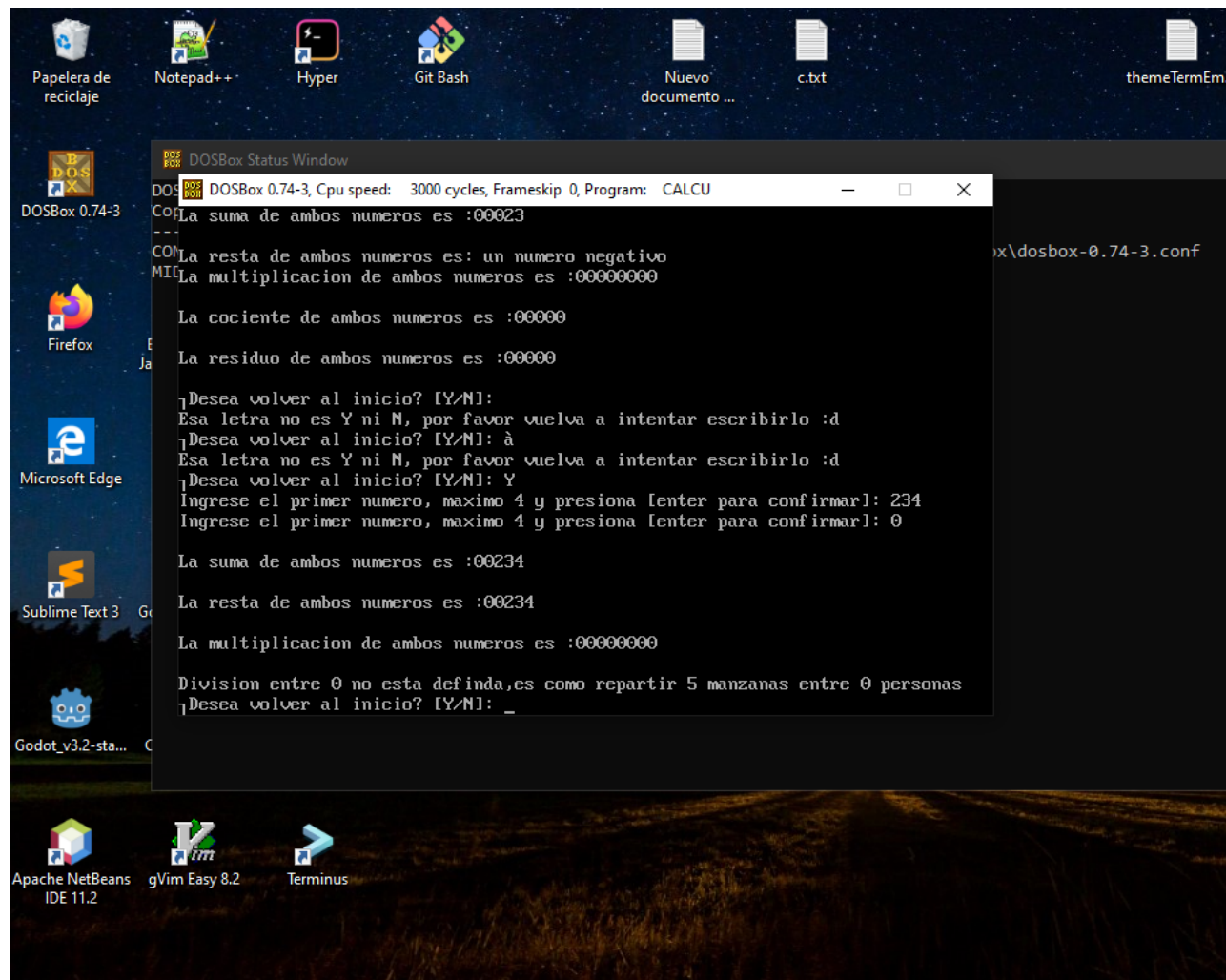
Prueba de Escritorio

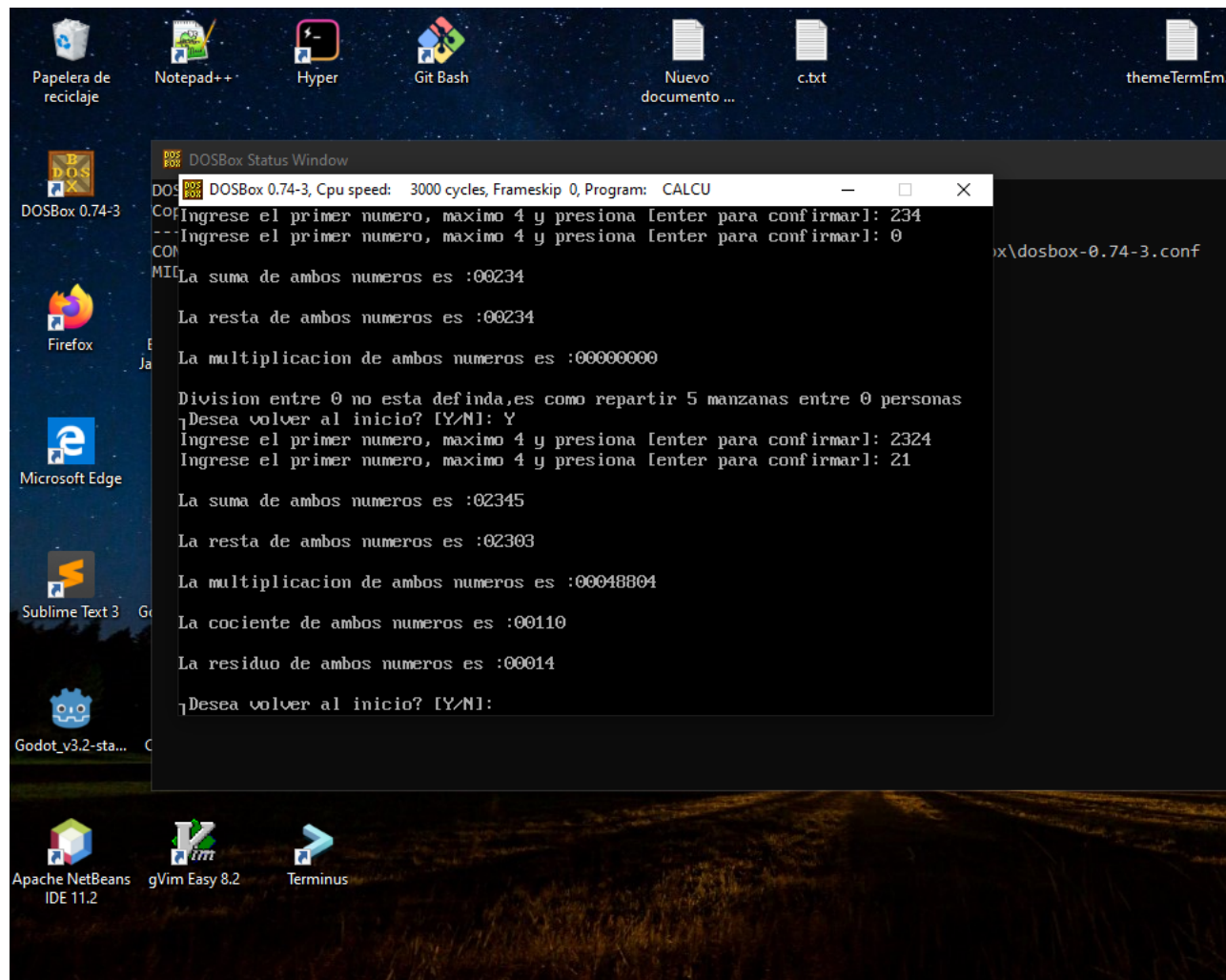
NUM1	NUM2	SUMA	RESTA	DIV	MUL
9999d	9999d	4E1Eh	0	1h	5F5 92E1h
70d	30d	64h	1E	834h	2











3 Conclusión

Pudimos cumplir nuestro objetivo de realizar todas nuestras operaciones en un programa en lenguaje ensamblador correctamente, algunos inconvenientes fueron en el proceso de la multiplicación debido a que teníamos registros de 16 bits y al multiplicarlos generaba un numero de 32 bits y no sabíamos como imprimirlo, al final lo pudimos resolver dividiéndolo entre 10 000 para dividirlo a la mitad y poder imprimir cada parte con nuestra función de imprimir caracteres, otra dificultad fue la de capturar los datos del usuario y guardarlo en la memoria con una variable, ya que cada carácter se guardaba como una representación de código ASCII y no como un numero entonces tuvimos que buscar una manera para convertirlo en un número y después que cada numero tuviera

su propio peso de unidad, decenas, centenas y milésimas. Todo fue un éxito pudimos aprender a usar más el lenguaje ensamblador, a saber como operar con operaciones aritméticas, aprendimos también a utilizar mas los saltos y ciclos que tiene ensamblador como también a leer y pintar en la pantalla, una de las aplicaciones que se me ocurren es que como ya sabes operar con entrada , salidas, condicionales y ciclos ya podemos ejecutar cualquier programa ya que cualquier programa solo se base esas 4 instrucciones como diría el teorema de la programación estructurada.