

# Actividad 3

**Christian Geovany Muñoz Rodríguez**

**Ingeniería en computación**

**Código: 221350605**

**Traductores de lenguajes I – D04 (Martes y  
Jueves de 11 a 1)**

**Maestro: José Juan Meza Espinosa**

**Universidad de Guadalajara**

**Centro Universitario de Ciencias Exactas e  
Ingenierías**

**27 de febrero del 2023**



## Código:

```
.model small

.stack 100

.data

    unid db 0

    decc db 0

    n1 dw 0

    n2 dw 0

    res dw 0

    num1 db 0

    num2 db 0

    num3 db 0

    num4 db 0

    msg db 10, 13, 'Ingresa un numero de dos digitos:', '$'

    msg2 db 10, 13, 'Ingresa otro numero de dos digitos:', '$'

    msg3 db 10, 13, 'El resultado es:', '$'

.code

    MOV AX, data
```

MOV ds, AX

lea DX, msg

MOV AH, 09H

INT 21H

MOV AH, 1

INT 21H

MOV decc, AL

SUB decc, 30H

MOV AH, 1

INT 21H

MOV unid, AL

SUB unid, 30H

MOV BL, 10

MOV AL, decc

MUL BL

MOV decc, AL

MOV AX, 0

MOV AL, unid

ADD AL, decc

MOV n1, AX

lea DX, msg2

MOV AH, 09H

INT 21H

MOV AH, 1

INT 21h

MOV decc, AL

SUB decc, 30H

MOV AH, 1

INT 21H

MOV unid, AL

SUB unid, 30H

MOV BL, 10

MOV AL, decc

MUL BL

MOV decc, AL

MOV AX, 0

MOV AL, unid

ADD AL, decc

MOV n2, AX

```
MOV AX, n1  
ADD AX, n2  
MOV res, AX
```

```
lea DX, msg3  
MOV AH, 09H  
INT 21H
```

```
MOV AX, res  
aam
```

```
MOV AL, AH  
aam
```

```
MOV AL, AH  
ADD AX, 30H  
MOV DL, AL  
MOV AH, 2  
INT 21H
```

```
MOV AX, res  
aam  
MOV AL,AH  
aam
```

```
ADD AX, 30H
```

```

        MOV DL, AL

        MOV AH, 2

        INT 21H

        MOV AX, res

        aam

        ADD AX, 30h

        MOV dl, AL

        MOV AH, 2

        int 21h

end

```

## Desarrollo

El objetivo de este programa es realizar una operación básica de suma de dos números de dos dígitos cada uno, permitiendo obtener un resultado de hasta tres dígitos.

Para ello, en el código se declaran las variables necesarias para almacenar los valores de entrada (n1 y n2), el resultado de la operación (res), así como variables auxiliares para manejar las cifras decimales de los números ingresados (decc y unid).

```

unid db 0
decc db 0

n1 dw 0
n2 dw 0

res dw 0

```

Además, se definen mensajes de texto que serán mostrados al usuario para solicitar los números de entrada y para mostrar el resultado de la operación.

```
msg db 10, 13, 'Ingresa un numero de dos digitos:', '$', '$'
msg2 db 10, 13, 'Ingresa otro numero de dos digitos:', '$', '$'
msg3 db 10, 13, 'El resultado es:', '$', '$'
```

El programa solicita al usuario que ingrese los dos números de dos dígitos cada uno, y los almacena en las variables correspondientes. Luego, suma los dos números y almacena el resultado en la variable res.

```
MOV AH, 1
INT 21H
MOV unid, AL
SUB unid, 30H

MOV BL, 10
MOV AL, decc
MUL BL
MOV decc, AL

MOV AX, 0
MOV AL, unid
ADD AL, decc
MOV n1, AX

lea DX, msg2
MOV AH, 09H
INT 21H

MOV AH, 1
INT 21H
MOV decc, AL
SUB decc, 30H

MOV AH, 1
INT 21H
MOV unid, AL
SUB unid, 30H

MOV BL, 10
MOV AL, decc
MUL BL
MOV decc, AL

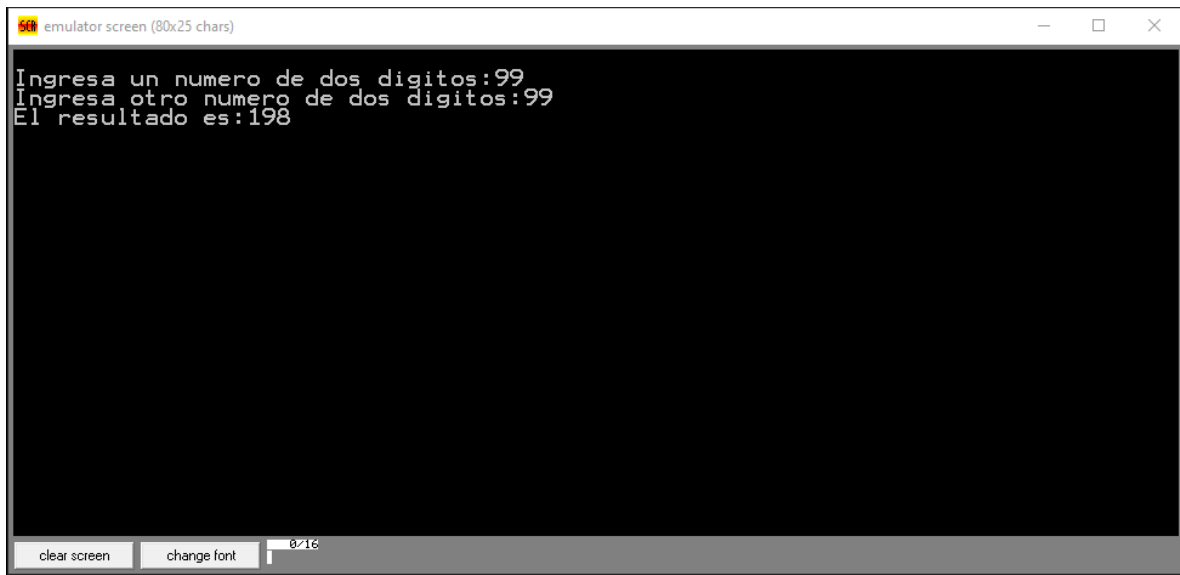
MOV AX, 0
MOV AL, unid
ADD AL, decc
MOV n2, AX

MOV AX, n1
ADD AX, n2
MOV res, AX

lea DX, msg3
MOV AH, 09H
INT 21H

MOV AX, res
aam
```

Para mostrar el resultado en la pantalla, se divide el número en tres cifras decimales (centenas, decenas y unidades), se convierte cada una a su representación ASCII correspondiente, y se imprime en la pantalla utilizando la interrupción 21h del BIOS.



```
emulator screen (80x25 chars)
Ingresa un numero de dos digitos:99
Ingresa otro numero de dos digitos:99
El resultado es:198
clear screen change font 0/16
```

## Conclusiones

En resumen, este programa es un excelente ejemplo de cómo se puede escribir un programa de ensamblador simple pero útil para realizar una operación básica como la suma. El código es fácil de entender y seguir, lo que lo hace accesible incluso para aquellos que no tienen experiencia previa en programación. Además, este programa es una excelente herramienta para aquellos que deseen aprender a programar en ensamblador o para aquellos que deseen entender cómo funcionan los programas de bajo nivel. En general, este programa es una valiosa contribución al campo de la programación de bajo nivel y una excelente herramienta de enseñanza para principiantes en programación.

## Bibliografía:

Brey, B. B. (2006). *Microprocesadores Intel : 8086/8088, 80186/80188, 80286, 80386 y 80486, Pentium, procesador Pentium Pro, Pentium II, Pentium III y Pentium 4: arquitectura, programación e interfaces*. Pearson Educación.