
 GOBIERNO DEL ESTADO DE MÉXICO	<h1 style="text-align: center;">RETO</h1> <p style="text-align: center;">Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales</p>	 TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

5

NOMBRE DE LA PRÁCTICA	SUMA DE DOS NÚMEROS CON DOS DÍGITOS			No.	UNIDAD 1
ASIGNATURA:	LENGUAJE INTERFAZ	CARRER A:	ISIC	PLAN:	ISIC-2010-204

Nombre: Jesús Navarrete Martínez

Grupo: 3501

Objetivo: Desplegar la suma de dos números con dos dígitos y su resultado sea también de dos dígitos también.

Elabora un programa en ensamblador que realice la suma de dos números con dos dígitos y el resultado también sea de dos dígitos. Explica lo que realiza cada renglón.

Escribe las instrucciones y captura de pantalla que demuestre que el programa si corrió:

### Código Fuente Escrito

```

1  .model small
2  .stack
3  .data
4      num1 db 0
5      num2 db 0
6      resultado db 0
7      mensaje db 10, 13, "Primer numero (dos digitos): $"
8      mensaje2 db 10, 13, "Segundo numero (dos digitos): $"
9      mensaje3 db 10, 13, "La suma es: $"
10     nueva_linea db 10, 13, "$"
11
12 .code
13 main proc
14     mov ax, @data
15     mov ds, ax
16
17     ; Solicitar primer n?mero de dos d?gitos
18     mov ah, 09h
19     lea dx, mensaje
20     int 21h
21
22     ; Leer el primer d?gito del primer n?mero
23     mov ah, 01h
24     int 21h
25     sub al, 30h      ; Convertir car?cter a n?mero
26     mov bl, al       ; Guardar el primer d?gito en bl
27
28     ; Leer el segundo d?gito del primer n?mero
29     mov ah, 01h
30     int 21h
31     sub al, 30h      ; Convertir car?cter a n?mero
32     mov bh, al       ; Guardar el segundo d?gito en bh
33
34     ; Calcular num1 = (primer_digito * 10) + segundo_digito
35     mov al, bl
36     mov ah, 0
37     mov cl, 10
38     mul cl           ; Multiplicar por 10
39     add al, bh       ; Sumar el segundo d?gito
40     mov num1, al     ; Guardar el n?mero final en num1
41
42     ; Solicitar segundo n?mero de dos d?gitos
43     mov ah, 09h
44     lea dx, mensaje2
45     int 21h
46
47     ; Leer el primer d?gito del segundo n?mero
48     mov ah, 01h
49     int 21h
50     sub al, 30h      ; Convertir car?cter a n?mero
51     mov bl, al       ; Guardar el primer d?gito en bl
52
53     ; Leer el segundo d?gito del segundo n?mero
54     mov ah, 01h
55     int 21h
56     sub al, 30h      ; Convertir car?cter a n?mero

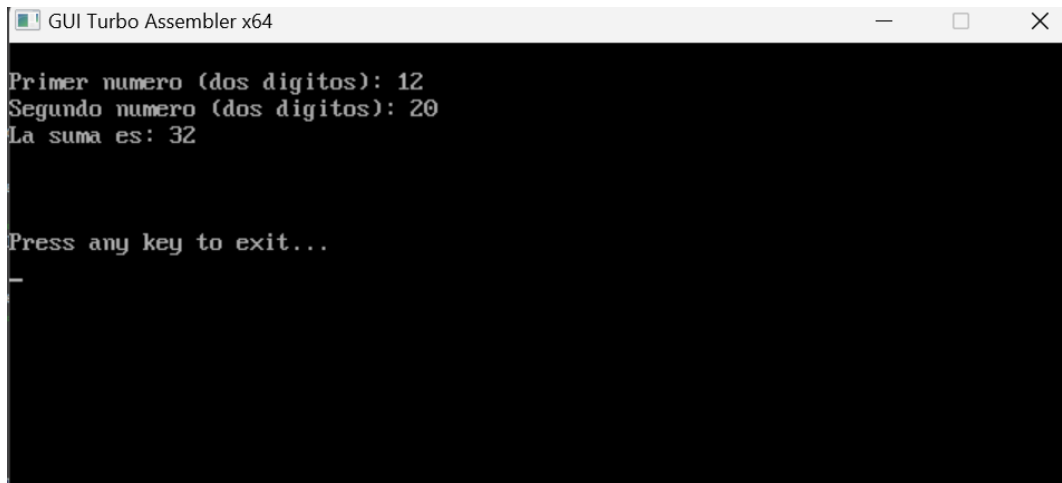
```

```

57      mov bh, al          ; Guardar el segundo d?gito en bh
58
59      ; Calcular num2 = (primer_digito * 10) + segundo_digito
60      mov al, bl
61      mov ah, 0
62      mul cl              ; Multiplicar por 10
63      add al, bh          ; Sumar el segundo d?gito
64      mov num2, al        ; Guardar el n?mero final en num2
65      ; Sumar num1 y num2
66      mov al, num1
67      add al, num2
68      mov resultado, al   ; Guardar la suma en resultado
69      ; Mostrar el mensaje de la suma
70      mov ah, 09h
71      lea dx, mensaje3
72      int 21h
73
74      ; Preparar para mostrar el resultado
75      mov al, resultado    ; Cargar resultado en AL
76      xor ah, ah          ; Limpiar AH para que AX tenga solo el resultado
77      cmp al, 0           ; Verificar si el resultado es cero
78      je mostrar_cero
79      ; Convertir a caracteres ASCII
80      mov bx, 10          ; Divisor
81      xor cx, cx          ; Limpiar CX para contar los d?gitos
82      convertir:
83      xor dx, dx          ; Limpiar DX antes de dividir
84      div bx              ; Dividir AX entre 10
85      push dx             ; Guardar el residuo (d?gito)
86      inc cx              ; Incrementar el contador de d?gitos
87      test ax, ax         ; Verificar si AX es cero
88      jnz convertir      ; Repetir hasta que AX sea cero
89      ; Mostrar los d?gitos en orden
90      mostrar_digitos:
91      pop dx              ; Obtener el d?gito
92      add dl, 30h         ; Convertir a ASCII
93      mov ah, 02h
94      int 21h            ; Mostrar d?gito
95      loop mostrar_digitos ; Repetir para todos los d?gitos
96      jmp fin
97
98      ; Mostrar "0"
99      mov dl, "0"
100     mov ah, 02h
101     int 21h
102
103     fin:
104     ; Nueva l?nea
105     mov ah, 09h
106     lea dx, nueva_linea
107     int 21h
108
109     ; Terminar el programa
110     mov ax, 4C00h
111     int 21h
112
113     main endp
114     end main

```

## Ejecución y compilación del código





```

GUI Turbo Assembler x64

Primer numero (dos digitos): 12
Segundo numero (dos digitos): 20
La suma es: 32

Press any key to exit...

```

 GOBIERNO DEL ESTADO DE MÉXICO	<h1>RETO</h1> Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales	 TESI TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

## CONCLUSIONES:

En esta práctica de programación en ensamblador, se ha implementado una solución para sumar dos números de dos dígitos ingresados por el usuario. El programa utiliza técnicas de manipulación de registros acumuladores y de datos para convertir caracteres ASCII a valores numéricos, realizar la operación de suma, y luego convertir el resultado nuevamente a su forma legible antes de mostrarlo en pantalla.

A lo largo de esta práctica, se destaca la importancia del manejo detallado de los registros en ensamblador, como el uso de AL, BL, y BH para realizar cálculos aritméticos y conservar los valores intermedios. Asimismo, se refuerza la capacidad del lenguaje ensamblador para manejar operaciones de varios dígitos mediante la descomposición de los números en decenas y unidades. Una característica notable de esta práctica es la gestión eficiente del despliegue del resultado, donde se convierte y muestra correctamente cada dígito del resultado, incluso cuando este tiene más de un dígito. Este ejercicio proporciona una excelente base para comprender cómo el ensamblador permite un control detallado sobre los datos y el hardware, algo que no es visible en lenguajes de alto nivel.