

Alumno: _____

Alumno: _____

Grupo (horario): _____

Sesión II

Introducción al lenguaje ensamblador

El objetivo de esta sesión es introducir el lenguaje ensamblador del MIPS y el simulador que utilizaremos para la realización de las prácticas.

Al finalizar esta sesión de prácticas se conocerá la arquitectura del procesador monociclo simulado y su representación gráfica, en concreto: el segmento de datos, el segmento de texto y los registros de proposito general.

Carga y ejecuta el siguiente código:

```
# programa primero.s
# segmento de texto

.text
.globl main

main:
# cargar los operandos
    la $t0,numA
    la $t1,numB
    lw $t2,0($t0)
    lw $t3,0($t1)

# operar
    add $a0,$t2,$t3

# almacenar el resultado
    sw $a0,0($t0)

# finalizar
    addi $v0,$0,10
    syscall

# segmento de datos
.data
numA: .word 3
numB: .word 6
```

Contesta a las siguientes cuestiones:

- Explica lo que hace el código
- Utilizando la siguiente tabla indica en la segunda columna el valor de los registros implicados en la instrucción en curso después de que esta se haya ejecutado. En la tercera columna explica brevemente lo que hace la instrucción.

Código	Contenido de los registros	Finalidad de la instrucción
<code>lw \$t2,0(\$t0)</code>		
<code>add \$a0,\$t2,\$t3</code>		

- ¿Cuál es la finalidad de la instrucción `syscall`?
- ¿Cuántos ciclos tarda en ejecutarse el programa anterior?
- ¿Qué queda almacenado en el registro `$a0` al final de la ejecución? Indícalo en hexadecimal, binario y decimal
- ¿Cuál es la dirección en memoria de la instrucción `add`?
- ¿Cuál es la dirección en memoria del número 6 (`numB`)?
- Si quisiera imprimir este número por pantalla, ¿qué instrucciones necesitaría añadir y dónde? Hazlo y pruébalo.