

# Instrucciones y Registros en MIPS (resumen)

Luis Gil V31.748.026

Junio 2025

## Introducción

- 1 Para entender como funciona la comunicación de un computador con su hardware debemos dirigirnos a los lenguajes de bajo nivel; estos lenguajes son quienes se encuentran mas cerca de del computador, permitiendo un control directo y preciso del hardware, a continuación presento un resumen de las instrucciones y registros del lenguaje ensamblador MIPS, para así comprender el comportamiento de nuestro computador, según instrucciones que le demos.**

Operandos:

- 2 **Registros:** Consta de 32 registros, se usan para almacenar valores y realizar operaciones. Algunos registros comunes son: \$t0–\$t7, \$s0–\$s7, \$zero, \$v0–\$v1, \$a0–\$a3, \$sp (pila), \$ra (dirección de retorno). Ejemplo: \$s1 = 5.
- 3 **Palabras de memoria:** Direcciones como Memory[0], Memory[4], Memory[424967296] se pueden acceder a través de instrucciones de carga y almacenamiento.

## 4 Instrucciones:

### 4.1 Aritméticas

```
add \s1, \s2, \s3 → Esto suma: \s1 = \s2 + \s3
sub \s1, \s2, \s3 → Esto resta: \s1 = \s2 - \s3
addi \s1, \s2, 100 → Suma inmediata: \s1 = \s2 + 100
```

### 4.2 Transferencia de datos

```
lw \s1, 100(\s2) → Carga palabra de memoria: \s1 = Mem[\s2
+ 100]
sw \s1, 100(\s2) → Guarda palabra en memoria: Mem[\s2 + 100]
= \s1
lh \s1, 100(\s2) → Carga media palabra (16 bits, con signo)
lhu \s1, 100(\s2) → Carga media palabra sin signo
lb \s1, 100(\s2) → Carga byte (8 bits, con signo)
lbu \s1, 100(\s2) → Carga byte sin signo
lui \s1, 100 → Carga inmediato en parte alta: \s1 = 100 << 16
```

### 4.3 Lógica

```
and \s1, \s2, \s3 → \s1 = \s2 AND \s3
or \s1, \s2, \s3 → \s1 = \s2 OR \s3
nor \s1, \s2, \s3 → \s1 = NOT(\s2 OR \s3)
andi \s1, \s2, 10 → \s1 = \s2 AND 10
ori \s1, \s2, 10 → \s1 = \s2 OR 10
```

#### 4.4 Comparación

```
slt \s1, \s2, \s3 → Si \s2 < \s3, entonces \s1 = 1; si no,  
\s1 = 0  
slti \s1, \s2, 10 → Igual, pero con inmediato (constante):  
\s1 = 1 si \s2 < 10
```

#### 4.5 Saltos condicionales

```
beq \s1, \s2, etiqueta → Salta si \s1 == \s2  
bne \s1, \s2, etiqueta → Salta si \s1 != \s2
```

#### 4.6 Saltos incondicionales

```
j etiqueta → Salta directamente a una etiqueta  
jr \sra → Salta a la dirección contenida en \sra (útil para  
regresar de funciones)  
jal etiqueta → Salta a una subrutina y guarda la dirección de  
retorno en \sra
```