

Estrutura geral da instrução MIPS (32 bits)

Todas as instruções têm 32 bits, mas são organizadas de formas diferentes dependendo do tipo: R, I ou J.

Tipo R (Register)

Usado para operações entre registradores (ex: `add`, `sub`, `and`, `or`, `slt`, etc.)

| Campo | Tamanho | Posição nos bits |
|---------------------|---------|------------------|
| <code>opcode</code> | 6 bits | 31 – 26 |
| <code>rs</code> | 5 bits | 25 – 21 |
| <code>rt</code> | 5 bits | 20 – 16 |
| <code>rd</code> | 5 bits | 15 – 11 |
| <code>shamt</code> | 5 bits | 10 – 6 |
| <code>funct</code> | 6 bits | 5 – 0 |

Campos:

- **opcode (31–26):** sempre 0 para instruções tipo R
 - **rs (25–21):** registrador fonte 1
 - **rt (20–16):** registrador fonte 2
 - **rd (15–11):** registrador destino (onde o resultado vai)
 - **shamt (10–6):** quantidade de deslocamento (shift), usado em instruções como `sll`, `srl`
 - **funct (5–0):** código da operação (ex: `100000` para `add`, `100010` para `sub`)
-

Tipo I (Immediate)

Usado para instruções que envolvem constantes, carregamento de dados, branches, etc. (ex: `addi`, `lw`, `sw`, `beq`, `bne`, etc.)

| Campo | Tamanho | Posição nos bits |
|--------|---------|------------------|
| opcode | 6 bits | 31 – 26 |
| rs | 5 bits | 25 – 21 |
| rt | 5 bits | 20 – 16 |
| rd | 5 bits | 15 – 11 |
| shamt | 5 bits | 10 – 6 |
| funct | 6 bits | 5 – 0 |

Campos:

- **opcode (31–26):** código da operação (ex: `001000` para `addi`)
- **rs (25–21):** registrador fonte
- **rt (20–16):** registrador destino
- **immediate (15–0):** valor imediato (constante ou offset)

Exemplo: `addi $t0, $t1, 5` → Soma 5 ao valor de `$t1` e guarda em `$t0`

Tipo J (Jump)

Usado para instruções de desvio absoluto (ex: `j`, `jal`)

| Campo | Tamanho | Posição nos bits |
|---------|---------|------------------|
| opcode | 6 bits | 31 – 26 |
| address | 26 bits | 25 – 0 |

Campos:

- **opcode (31–26):** código da operação (`000010` para `j`, `000011` para `jal`)
- **address (25–0):** endereço de destino do salto (junta com os 4 bits mais altos do PC)

Resumo rápido por tipo

| Tipo | Bits | Campo | Função |
|------|-------|-----------|--|
| R | 31–26 | opcode | Tipo de operação (sempre <code>000000</code>) |
| | 25–21 | rs | Registrador fonte 1 |
| | 20–16 | rt | Registrador fonte 2 |
| | 15–11 | rd | Registrador destino |
| | 10–6 | shamt | Shift amount (quantidade de deslocamento) |
| | 5–0 | funct | Função/Operação específica |
| I | 31–26 | opcode | Código da operação |
| | 25–21 | rs | Registrador base |
| | 20–16 | rt | Registrador destino |
| | 15–0 | immediate | Valor imediato ou offset |
| J | 31–26 | opcode | Código da operação |
| | 25–0 | address | Endereço de salto |

Se quiser, posso te mostrar exemplos binários decodificando instruções reais (tipo transformar binário em instrução MIPS ou vice-versa). Quer?