



RISC

MIPS



## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

**MIPS-32**

Código de operación	Registro1	Registro 2	Registro 3
Registro de Instrucción (IR)			

Cada segmento en que se encuentra dividida la cadena de bits del **registro de instrucción** recibe el nombre de **Campo**.

El número de campos, el número de bits de cada campo y la función que desempeña cada campo de bits recibe el nombre de **formato de instrucción**.



**RISC**

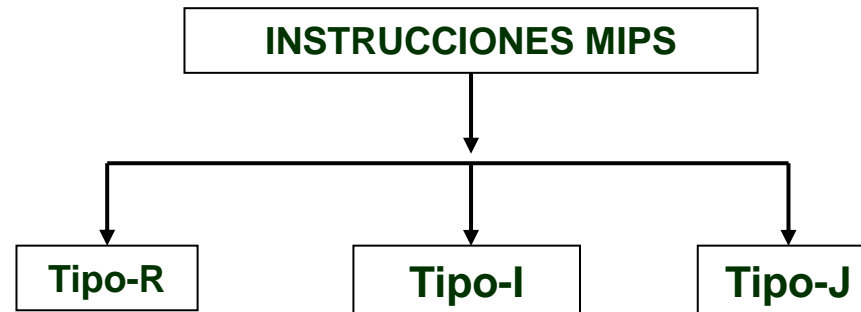
**MIPS**

## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

### Instrucciones MIPS

**MIPS-32**

Se clasifican en tres categorías





RISC

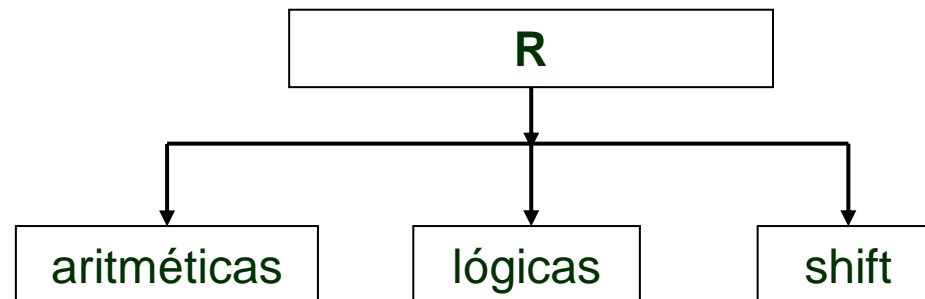
MIPS

## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

**MIPS-32**

### Instrucciones Tipo-R

El formato tipo **R** es utilizado en operaciones aritméticas, lógicas y de desplazamiento





RISC

MIPS

CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo-R

(Definición de los campos)

MIPS-32

opcode	rs	rt	rd	shamt	funct
Definición de campos en una instrucción MIPS					

En instrucciones Tipo-R, los campos “opcode” y “function” tienen funciones complementarias

“**opcode**” contiene el código binario asociado a una familia de operaciones:

“**function**” contiene el código binario asociado a una operación particular, de la familia especificada



RISC

MIPS

CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo-R

(Definición de los campos)

MIPS-32

opcode	rs	rt	rd	shamt	funct
Definición de campos en una instrucción MIPS					

Instrucciones Tipo-R  
(algunos casos)

opcode	función	Nom Inst	function	tipo
000000	Shift left logical	sll	000000	shift
000000	Shift right logical	srl	000010	shift
000000	Shift right arithm	sra	000011	shift
000000	Shift left logical variable	sllv	000100	shift
000000	Shift right logical variable	srlv	000110	shift
000000	Shift right arithm variable	srav	000111	shift



RISC

MIPS

CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo-R

(Definición de los campos)

MIPS-32

opcode	rs	rt	rd	shamt	funct
Definición de campos en una instrucción MIPS					

Instrucciones Tipo-R  
(algunos casos)

opcode	función	Nom.Inst	function	tipo
000000	multiply	mult	011000	arithmetic
000000	Unsignes multiply	multu	011001	arithmetic
000000	Divide with overflow	div	011010	arithmetic
000000	Divide without overflow	divu	011011	arithmetic
000000	addition with overflow	add	100000	arithmetic
000000	addition without overflow	addu	100001	arithmetic



RISC

MIPS

CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo-R

(Definición de los campos)

MIPS-32

opcode	rs	rt	rd	shamt	funct
Definición de campos en una instrucción MIPS					

opcode	función	Nom.Inst	function	tipo
000000	and	and	100100	logical
000000	or	or	100101	logical
000000	xor	xor	100110	logical
000000	nor	nor	100111	logical

Instrucciones Tipo-R  
(algunos casos)



RISC

MIPS

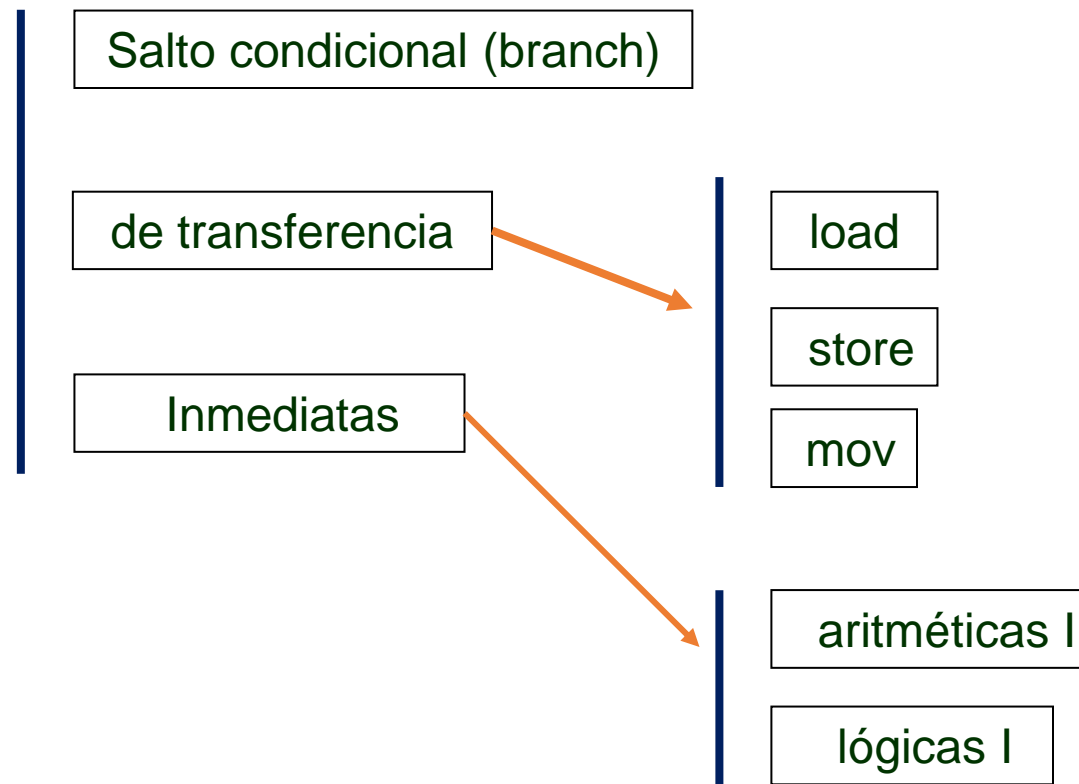
## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

### Instrucciones Tipo-I

(Definición de los campos)

MIPS-32

### Instrucciones Tipo-I







RISC

MIPS

CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo-I

(Definición de los campos)

MIPS-32

opcode	rs	rt	imm
6 bits	5 bits	5 bits	16 bits
Definición de los campos para instrucciones tipo inmediatas			

opcode	función	Nom.Inst	imm
001000	Addition immediate (with overflow)	addi	constant
001001	Addition immediate (without overflow)	addiu	constant
0x0C	AND immediate	andi	constant
0x0D	OR immediate	ori	constant

Instrucciones Tipo-I  
(algunos casos)



RISC

MIPS

CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo - J

(Definición de los campos)

MIPS-32

opcode	tarjet
6 bits	26 bits
Definición de los campos para instrucciones tipo inmediatas	

opcode	función	Nom.Inst	tarjet
0000010	Jump to target address	J target	address
0x03	jump and link	Jal target	address

Instrucciones Tipo-J

(unconditional branches)

J target: Unconditionally jump to the instruction at target

jal target: Unconditionally jump to the instruction at target. Save the address of the next instruction in register \$ra: registro 31



**RISC**

**MIPS**



## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

### Instrucciones Tipo-R

**MIPS-32**

Pseudoinstrucciones: Extienden un conjunto de instrucciones disponibles para el programador. No pueden codificarse en lenguaje máquina ya que no forman parte del repertorio de instrucciones de la arquitectura en cuestión

Son instrucciones proporcionadas por el lenguaje ensamblador para facilitar un poco la vida del programador

El programa ensamblador se encargará de traducir automáticamente cada pseudoinstrucción por aquella instrucción máquina o secuencia de instrucciones máquina que realicen la operación correspondiente



RISC

MIPS



CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

Instrucciones Tipo-R

MIPS-32

Pseudoinstrucciones:

move \$s0, \$zero	pseudoinstruction
mul rdest, rsrc1, src2	pseudoinstruction
abs rdest, rsrc	pseudoinstruction
div rdest, rsrc1, src2	pseudoinstruction
divu rdest, rsrc1, src2	pseudoinstruction
mulo rdest, rsrc1, src2	pseudoinstruction
la rdest, address	pseudoinstruction



**RISC**

**MIPS**

## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

### Instrucciones Tipo-R

**MIPS-32**

Pseudoinstrucciones:

blt rsrc1, rsrc2, label

pseudoinstruction

Branch on less than

bltu rsrc1, rsrc2, label

pseudoinstruction

Branch on less than unsigned

bnez rsrc, label

pseudoinstruction

Branch on not equal zero



# MIPS

## Instrucciones Tipo-R

# MIPS-32

mult rs, rt

pseudoinstruction

Multiply

## Type- R

mult rs, rt

0	rs	rt	0	0x18
---	----	----	---	------

Multiplica los registros rs y rt. Deja la parte baja de la palabra del producto en el registro “lo” y la parte alta de la palabra del producto en el registro “hi”.



## RISC

# MIPS



## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

## Instrucciones Tipo-R

# MIPS-32

## Pseudoinstrucciones:

multu rs, rt

pseudoinstruction

## Unsigned multiply

multu rs, rt

0	rs	rt	0	0x19
---	----	----	---	------

Multiply registers `rs` and `rt`. Leave the low-order word of the product in register `lo` and the high-order word in register `hi`.

La unidad de multiplicar y dividir produce su resultado en dos registros adicionales, hi y lo. Estas instrucciones mueven valores hacia y desde estos registros. Las pseudoinstrucciones de multiplicar, dividir y restar que hacen que esta unidad parezca operar en los registros generales mueven el resultado después de que finaliza el cálculo.



**RISC**

**MIPS**



## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

### Instrucciones Tipo-R

**MIPS-32**

Instrucciones:

Una instrucción en ensamblador se corresponde con una única instrucción máquina

Ocupa 32 bits en el MIPS 32

Ejemplo:

```
addi $t1, $t0, 4
```





**RISC**

**MIPS**



## CONJUNTO DE INSTRUCCIONES (SET DE INSTRUCCIONES): CLASIFICACION

### Instrucciones Tipo-R

**MIPS-32**

Pseudoinstrucciones:

Una pseudoinstrucción se puede utilizar en un programa en ensamblador pero no se corresponde con ninguna instrucción máquina

Ejemplo de pseudoinstrucciones:

```
li $v0, 4  
move $t1, $t2
```

En el proceso de ensamblado se sustituyen por la secuencia de instrucciones máquina que realizan la misma funcionalidad. Las dos pseudoinstrucciones del párrafo anterior se sustituyen por instrucciones, como se indica:

```
ori $v0, $0, 4    sustituye a: li $v0, 4  
addu $t1, $0, $t2 sustituye a: move $t1, $t2
```