

# Tutoría 5

MODOS DE DIRECCIONAMIENTOS EN ENSAMBLADOR PARA MIPS

MODOS  
Direccionamiento



Datos o Instrucciones



DATOS

- ✓ Modo de direccionamiento inmediato.
- ✓ Modo de direccionamiento directo a registro.
- ✓ Modo de direccionamiento indirecto a registro con desplazamiento.

PROGRAMA  
INSTRUCCIONES

- ✓ Direccionamiento relativo PC + desplazamiento.
- ✓ Direccionamiento Pseudodirecto.



A diagram illustrating data flow. On the left is a solid brown rectangle labeled "DATOS". To its right is a brown arrow pointing left towards the rectangle. Further to the right is a list of three addressing modes, each preceded by a checkmark.

DATOS

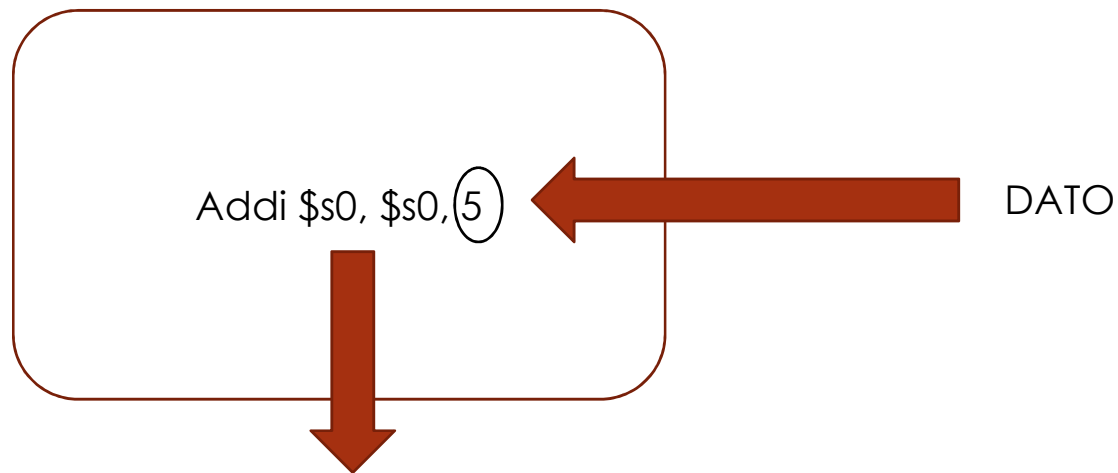
✓ Modo de direccionamiento inmediato.

✓ Modo de direccionamiento directo a registro.

✓ Modo de direccionamiento indirecto a registro con desplazamiento.

# Direcccionamiento Inmediato

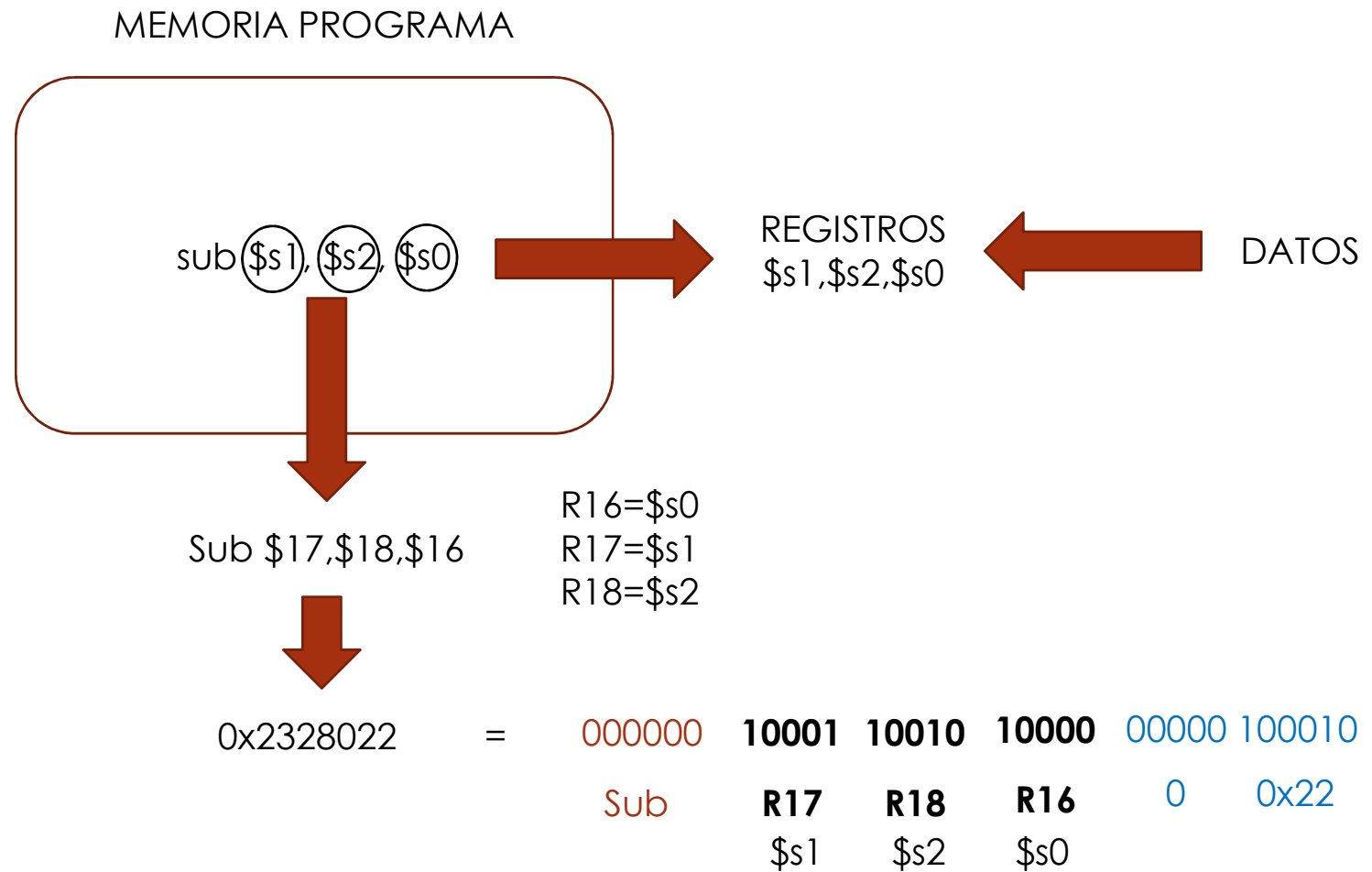
MEMORIA PROGRAMA



Addi \$16, \$16, 5      R16=\$s0

0x22100005 = 001000 10000 10000 0000000000000101  
Addi R16 R16 Valor inmediato 5

# Direccionamiento Directo a Registro



# INDIRECTO. BASE + DESPLAZAMIENTO

MEMORIA DE DATOS



VECTOR (ELEMENTOS WORD= 4 BYTES)



7, 2, 6, 8, 1, 9, 10, 3, 4, 5

User data segment [10000000]..[10040000]

[10000000]..[1000ffff]	00000000			
[10010000]	00000007	00000002	00000006	00000008
[10010010]	00000001	00000009	0000000a	00000003
[10010020]	00000004	00000005	00000000	00000000
[10010030]..[1003ffff]	00000000			

DIRECCIÓN BASE  
[ 10010000 ]

+

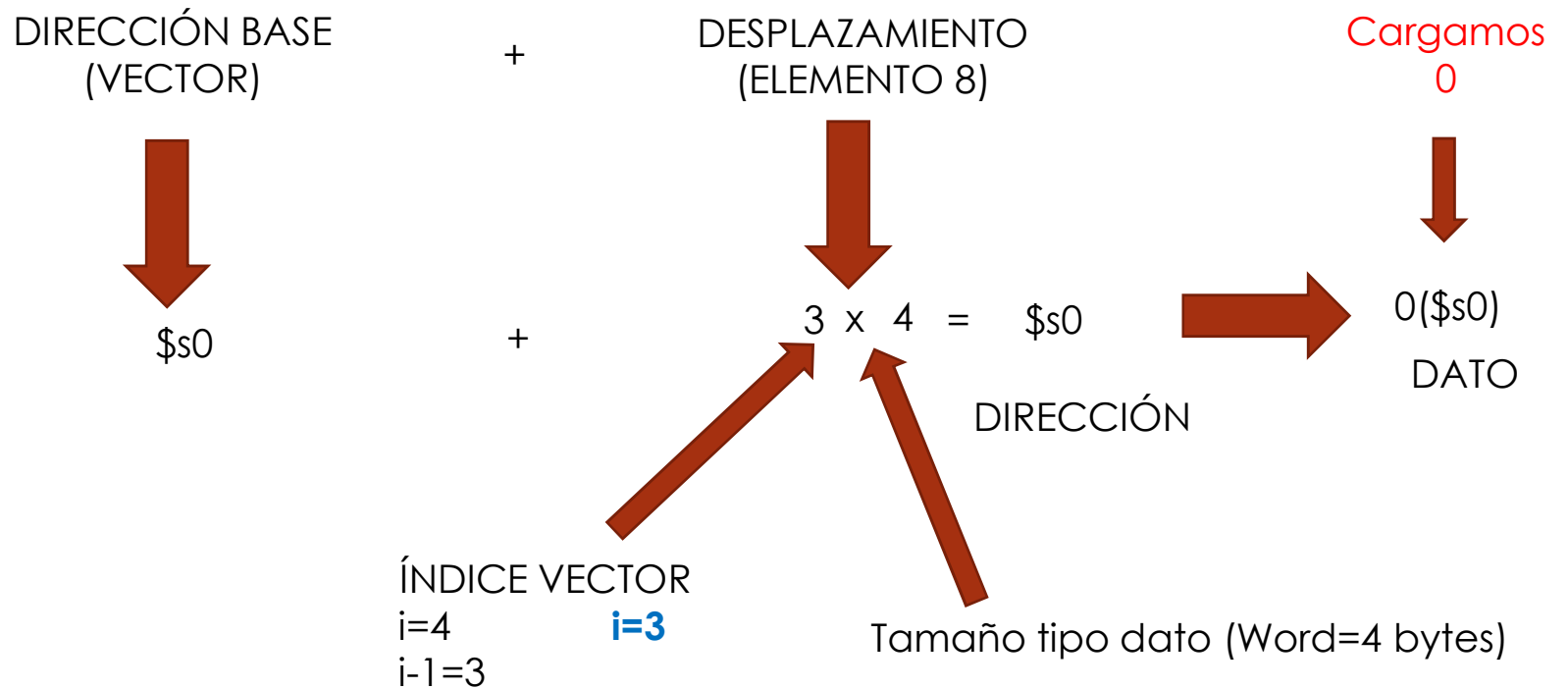
DESPLAZAMIENTO  
C

= [ 1 0 0 1 0 0 0 C ] C=12

VECTOR (ELEMENTOS **WORD= 4 BYTES**)

7 , 2 , 6 , 0 , 1 , 9 , 10 , 3 , 4 , 5 i=1,...,10

i= 0,..., 9







DATOS

- ✓ Modo de direccionamiento inmediato.
- ✓ Modo de direccionamiento directo a registro.
- ✓ Modo de direccionamiento indirecto a registro con desplazamiento.

PROGRAMA

- ✓ Direccionamiento relativo PC + desplazamiento.
- ✓ Direccionamiento Pseudodirecto.

PROGRAMA



- ✓ Direccionamiento relativo PC + desplazamiento.
- ✓ Direccionamiento Pseudodirecto.

# CONTADOR DE PROGRAMA + DESPLAZAMIENTO (salto incondicional **b**)

MEMORIA PROGRAMA

Instrucción 0 (Ocupa 4bytes)  
Instrucción 1 (Ocupa 4bytes)  
Instrucción 2 (Ocupa 4bytes)  
Instrucción 3 (Ocupa 4bytes)  
Instrucción 4 (Ocupa 4bytes)



**PC=PC+16**

PC= Contador Programa

Registro contiene dirección  
de la **INSTRUCCIÓN ACTUAL**  
que se está ejecutando (QtSpim)

MIPS contiene dirección  
de la **INSTRUCCIÓN SIGUIENTE**  
que se está ejecutando

PC

+

=

PC

Desplazamiento

(número de instrucciones x 4)=nº bytes= 16

# DIRECCIONAMIENTO PSEUDODIRECTO (salto incondicional *j*)

MEMORIA PROGRAMA

Instrucción 0 (Ocupa 4bytes)  
Instrucción 1 (Ocupa 4bytes)  
Instrucción 2 (Ocupa 4bytes)  
Instrucción 3 (Ocupa 4bytes)  
Instrucción 4 (Ocupa 4bytes)



**PC = Dirección Memoria  
Instrucción destino**

**Instrucción *j* =**

	6 bits	26 bits	
	Codificación	Etiqueta	0x 0810000d=
	Instrucción <i>j</i>		000010000010000000000000000000001101

Dirección Destino	4bits	26 bits	2 bits	
	más significativos PC	Etiqueta	ceros	
	0000	000001000000000000000000000000001101	00	= 0x00400034

# INDIRECTO. BASE + DESPLAZAMIENTO

## VECTORES

7, 2, 6, 8, 1, 9, 10, 3, 4, 5  $i=1, \dots, 10$  Elementos tipo Word  
 $i=0, \dots, 9$  (4 bytes)



$\text{Desplazamiento}(i) = (i-1) \times \text{Tamaño dato}$

$\text{Desplazamiento}(i) = i \times \text{Tamaño dato}$

$\text{Dirección}(i) = \text{Base} + (i-1) \times \text{Tamaño dato}$

$\text{Dirección}(i) = \text{Base} + i \times \text{Tamaño dato}$

# INDIRECTO. BASE + DESPLAZAMIENTO

## MATRICES

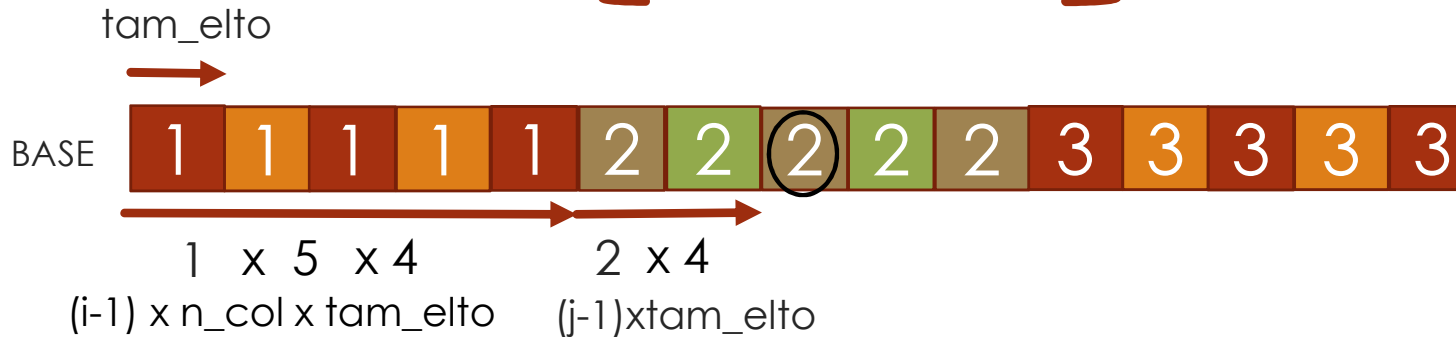
tam\_elto=4  
nfil=3  
ncol=5

→	1	1	1	1	1
→	2	2	2	2	2
→	3	3	3	3	3

$i=1,\dots,5$   
 $j=1,\dots,3$

$i=0,\dots,4$   
 $j=0,\dots,2$

Word  
Tamaño byte=4



Ejemplo:  $i=2$   $j=3$

$Desplazamiento(i,j) = ((i-1) \times ncol + (j-1)) \times tam\_elto$

$Desplazamiento(i,j) = (i \times ncol + j) \times tam\_elto$

Dirección  $(i,j) = Base + Desplazamiento(i,j)$

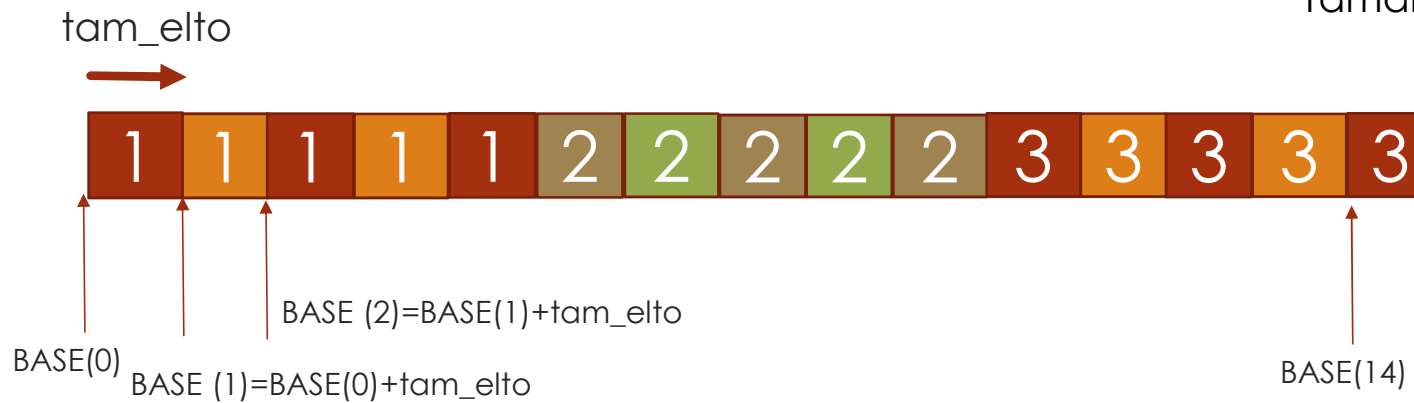
# INDIRECTO. BASE + DESPLAZAMIENTO MATRICES

tam\_elto=4  
nfil=3  
ncol=5

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3

$i=0,\dots,4$   
 $j=0,\dots,2$

Word  
Tamaño byte=4



Vector  $i=0,\dots,nfil \times ncol - 1 = 0,\dots,14$

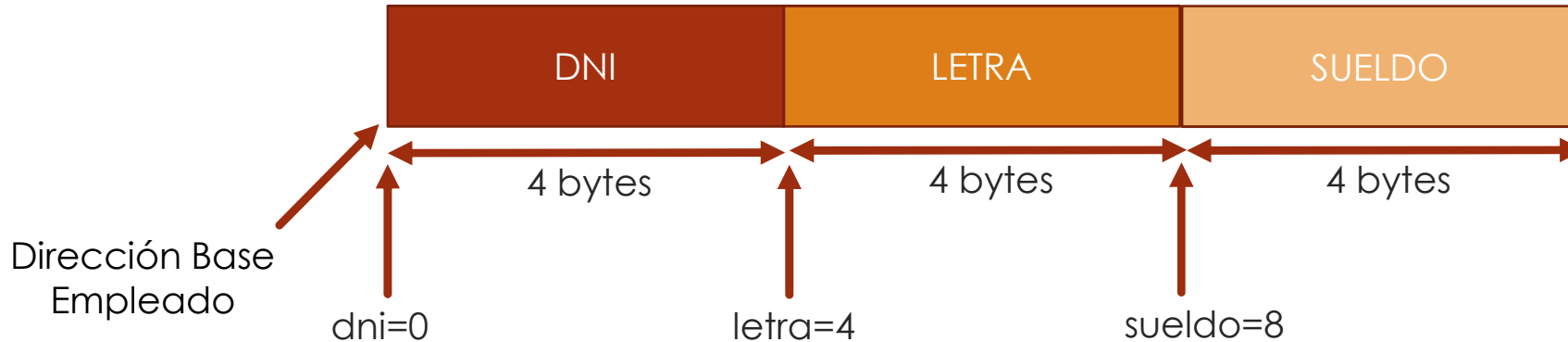
$BASE(i+1) = BASE(i) + tam\_elto$

Dirección (i) = Base(i)

# INDIRECTO. BASE + DESPLAZAMIENTO ESTRUCTURAS

tam\_dni=4  
tam\_letra=4  
tam\_sueldo=4  
tam\_empleado=12

EMPLEADO



`emp: .space tam_empleado`

← Reservar Espacio un Elemento de la Estructura Empleado. (nº bytes).

`Sueldo(Dirección Base Empleado)=sueldo($t1)`

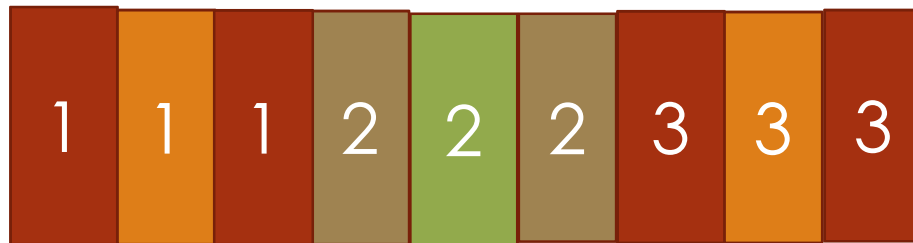
← Acceder un elemento de la estructura

`Dirección Memoria= Dirección Base Empleado + sueldo= $t1 + sueldo`



## SUMA ELEMENTOS DIAGONAL

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$



BASE(0)

BASE (1)=BASE(0)+(ncol+1) x tam\_elto

BASE (2)=BASE(1)+(ncol+1) x tam\_elto

Base(i+1)=Base(i)+(ncol+1) x tam\_elto