



# ESTRUCTURA DE LOS COMPUTADORES

Práctica 4 – Programa en Ensamblador

Alejandro Bernabeu Calatayud

[abc4.onil4@gmail.com](mailto:abc4.onil4@gmail.com)

Grupo 40 I2ADE  
48788949-S

# Índice

Listado de archivos.....	2
Ejercicio 1. El número 80 .....	3
Ejercicio 2. Visualización continua.....	4
Ejercicio 3. Vector	
3.1. Media aritmética.....	6
3.2. Distancia euclídea.....	9
3.3. Factorial.....	10
Comentarios .....	11

## Archivos

*Ejercicio 1:*                    *numero80.asm*

*Ejercicio 2:*                    *bucle.asm*

*Ejercicio 3:*                    *Vector.asm*

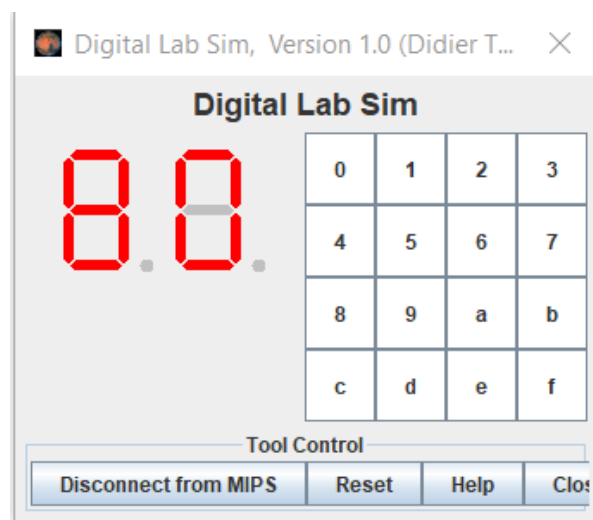
## Ejercicio 1 – El número 80

1. Modifica el programa para que aparezca el número 80 en los displays.

Código:

```
mips1.asm
1  #Ejemplo displaysdel periférico Digital LabSim
2  .text
3  main:
4  li $s0, 0xFFFF0010 # carga dirección base del displayderecho
5  li $s1, 0xFFFF0011 # carga dirección base del displayizquierdo
6  li $t1, 0x3F # El bit 0 activa el segmento A
7  sb $t1, 0($s0) # almacena en dirección del displayderecho el valor de $t1
8  li $t1, 0x7F # El punto decimal
9  sb $t1, 0($s1) # almacena en dirección del displayizquierdo el valor de $t1
10 li $v0, 10 # Fin programa
11 syscall
```

Digital Lab Sim:



## Ejercicio 2 – Visualización continua

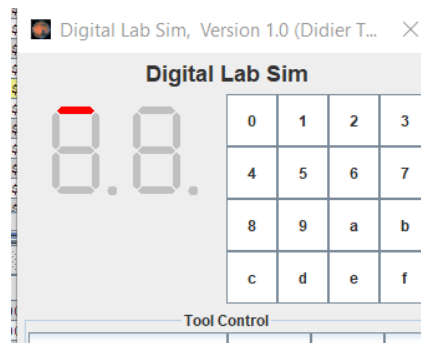
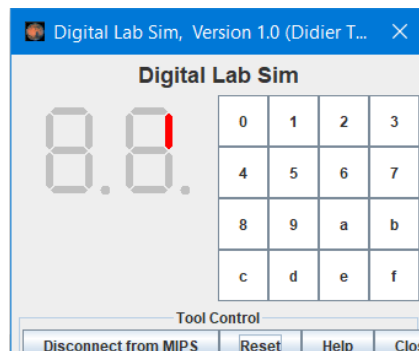
2. Haz que se visualice de forma consecutiva los siguientes segmentos:

A Display Derecho , A Display Izquierdo, F Display Izquierdo, E Display Izquierdo, D Display Izquierdo, D Display Derecho, C Display Derecho, B Display Derecho

Código:

```
1  #Ejemplo2 displays del periférico Digital LabsIM
2  .text
3  main:
4  li $s0, 0xFFFF0010 # carga dirección base del display
5  li $s1, 0xFFFF0011 # carga dirección base del display
6  li $t1, 0x01 # El segmento A derecho
7  sb $t1, 0($s0)
8  li $t1, 0x00 # El segmento A izquierdo a 0
9  sb $t1, 0($s0)
10 li $t1, 0x01 # El segmento A derecho
11 sb $t1, 0($s1)
12 li $t1, 0x20 # El segmento F Izquierdo
13 sb $t1, 0($s1)
14 li $t1, 0x10 # El segmento E Izquierdo
15 sb $t1, 0($s1)
16 li $t1, 0x08 # El segmento D Izquierdo
17 sb $t1, 0($s1)
18 li $t1, 0x00 # El segmento D Izquierdo a 0
19 sb $t1, 0($s1)
20 li $t1, 0x08 # El segmento D Derecho
21 sb $t1, 0($s0)
22 li $t1, 0x04 # El segmento C Derecho
23 sb $t1, 0($s0)
24 li $t1, 0x02 # El segmento B Derecho
25 sb $t1, 0($s0)
26 li $v0, 10 # Fin programa
27 syscall
```

Capturas:



2.1 Haz un bucle de la secuencia anterior.

```
1 # Ejemplo de programa para el display de 7 segmentos
2 .text
3 main:
4 li $s0, 0xFFFF0010 # carga dirección base del display
5 li $s1, 0xFFFF0011 # carga dirección base del display
6 Bucle:
7 li $t1, 0x01 # El segmento A derecho
8 sb $t1, 0($s0)
9 li $t1, 0x00 # El segmento A izquierdo a 0
10 sb $t1, 0($s0)
11 li $t1, 0x01 # El segmento A derecho
12 sb $t1, 0($s1)
13 li $t1, 0x20 # El segmento F Izquierdo
14 sb $t1, 0($s1)
15 li $t1, 0x10 # El segmento F Izquierdo
16 sb $t1, 0($s1)
17 li $t1, 0x08 # El segmento D Izquierdo
18 sb $t1, 0($s1)
19 li $t1, 0x00 # El segmento D Izquierdo a 0
20 sb $t1, 0($s1)
21 li $t1, 0x08 # El segmento D Derecho
22 sb $t1, 0($s0)
23 li $t1, 0x04 # El segmento C Derecho
24 sb $t1, 0($s0)
25 li $t1, 0x02 # El segmento B Derecho
26 sb $t1, 0($s0)
27 j Bucle
28 li $v0, 10 # Fin programa
29 syscall
```

¿Qué ocurre?

Que nos quedamos dentro de un bucle infinito porque no hay condición de parada.

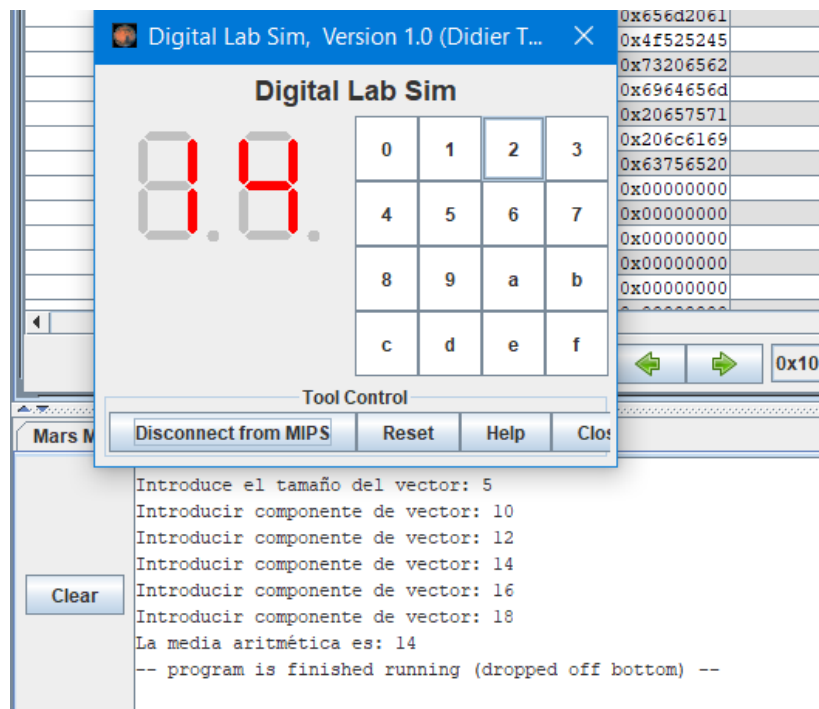
## Ejercicio 3 - Vector, distancia euclídea y factorial

3. Realizar un nuevo programa en lenguaje ensamblador en MIPS que pida el tamaño de un vector y a continuación pida introducir las componentes de dicho vector.

```
22  main:
23  # imprimir mensaje "Introduce el tamaño del vector: ..."
24  la $a0,msg01
25  li $v0,4
26  syscall
27  # leer tamaño del vector
28  li $v0,5
29  syscall
30  move $t0,$v0
31  # si cantidad de números a leer es cero, terminar.
32  beqz $t0,f01
33  # bucle de lectura y cálculo.
34  li $t1,0
35  li $t2,0
36  li $t3,0
37  li $t4,0
38  li $t7,99
39  b01:
40  # imprimir mensaje "Introducir número: "
41  la $a0,msg02
42  li $v0,4
43  syscall
44  # leer número
45  li $v0,5
46  syscall
```

- 3.2. Seguidamente, se debe mostrar por consola y por la TOOLS Digital LabSim **la media aritmética** de las componentes del vector.

```
47  # sumamos todos los factores // Media Aritmetica 01
48  add $t2,$t2,$v0
49  #bucle
50  addi $t1,$t1,1
51  blt $t1,$t0,b01
52  #divison de la suma de factores entre el numero de componentes // Media Aritmetica 02
53  div $t2,$t1
54  mflo $t2
55  #Comprobar que la media es correta y podmeos imprimirla
56  blt $t2,$zero ErrorNegativo
57  bge $t2,$t7 ErrorMayor99
58  # imprimir mensaje "La media aritmética es: ..."
59  la $a0,msg03
60  li $v0,4
61  syscall
62  # imprime el resultado por pantalla
63  move $a0,$t2
64  li $v0,1
65  syscall
```



```

70  #Comprobar si el numero es de dos digitos(mayor igual que 10)
71  bge $t2,$t9,TieneDosDigitos
72  blt $t2,$t9,TieneUnDigito
73
74  TieneUnDigito:
75  #-----UN DIGITO
76  #Ponemos un indice
77  li $t8,0
78  #Tiene un digito
79  beq $t2,$t8 MostrarZero
80  addi $t8,$t8,1
81  beq $t2,$t8 MostrarUno
82  addi $t8,$t8,1
83  beq $t2,$t8 MostrarDos
84  addi $t8,$t8,1
85  beq $t2,$t8 MostrarTres
86  addi $t8,$t8,1
87  beq $t2,$t8 MostrarCuatro
88  addi $t8,$t8,1
89  beq $t2,$t8 MostrarCinco
90  addi $t8,$t8,1
91  beq $t2,$t8 MostrarSeis
92  addi $t8,$t8,1
93  beq $t2,$t8 MostrarSiete
94  addi $t8,$t8,1
95  beq $t2,$t8 MostrarOcho
96  addi $t8,$t8,1
97  beq $t2,$t8 MostrarNueve
98
99  #Activamos las secciones correspondientes del Digital Lab Sim
100  MostrarZero:
101  li $t5,0x3F
102  j Imprimir
103  MostrarUno:
104  li $t5,0x6
105  j Imprimir

```



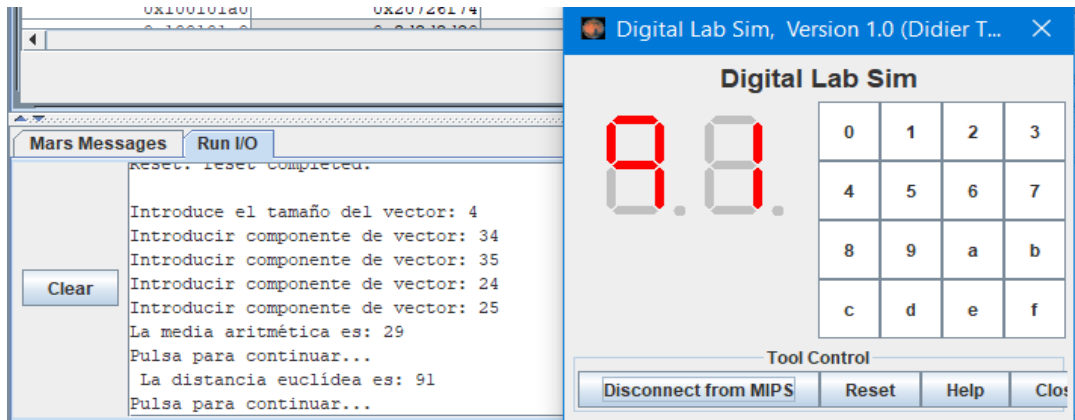
```

227
228 #Activamos las secciones correspondientes del Digital Lab Sim
229 MostrarZero3:
230 li $t5,0x3F
231 j Imprimir
232 MostrarUno3:
233 li $t5,0x6
234 j Imprimir
235 MostrarDos3:
236 li $t5,0x5B
237 j Imprimir
238 MostrarTres3:
239 li $t5,0x4F
240 j Imprimir
241 MostrarCuatro3:
242 li $t5,0x66
243 j Imprimir
244 MostrarCinco3:
245 li $t5,0x6D
246 j Imprimir
247 MostrarSeis3:
248 li $t5,0x7D
249 j Imprimir
250 MostrarSiete3:
251 li $t5,0x7
252 j Imprimir
253 MostrarOcho3:
254 li $t5,0x7F
255 j Imprimir
256 MostrarNueve3:
257 li $t5,0x67
258 j Imprimir

262 Imprimir:
263 # imprimir el resultado por LabSim
264 li $s0, 0xFFFF0010 # carga dirección base del display derecho
265 li $s1, 0xFFFF0011 # carga dirección base del display izquierdo
266 sb $t5, 0($s0)
267 j DistanciaEuclidea
268
269 # final del programa, la etiqueta f01 es por si el número de compoenntes del vector es 0
270 f01: li $v0,10
271 syscall
272
273 #ErrorMenor0
274 ErrorNegativo:
275 # imprimir mensaje
276 la $a0,msg05
277 li $v0,4
278 syscall
279 #Final
280 li $v0,10
281 syscall
282
283 #ErrorMayor99
284 ErrorMayor99:
285 # imprimir mensaje
286 la $a0,msg06
287 li $v0,4
288 syscall
289 #Final
290 li $v0,10
291 syscall
292
293
294 DistanciaEuclidea: #-----

```

- 3.2. Seguidamente, se debe mostrar por consola y por la TOOLS Digital LabSim la **distancia euclídea** respecto al origen de coordenadas (por la tools únicamente la parte entera y no la decimal).



En mi programa SOLO se trabaja con números enteros, por lo que la distancia euclídea i la media aritmética se truncan a un número entero.

```

310 DistanciaEuclídea: #----- DISTANCIA EUCLIDEA
311 # imprimir mensaje "Pulsa para continuar..."
312 la $a0,msg09
313 li $v0,4
314 syscall
315 # Esperar a que pulse
316 li $v0,12
317 syscall
318 #Anteriormente ya hemos elevado al cuadrado la suma de todos los componentes dle vector, que está almacenada en $t6
319 #Mostrar mensaje por pantalla "La distancia eucídea es ..."
320 la $a0,msg08
321 li $v0,4
322 syscall
323 #Mostrar resultado
324 move $a0,$t6
325 li $v0,1
326 syscall
327 #AVISO ifDistancia euclídea mayor de 99
328 move $t2,$t6
329 li $s3,100
330 blt $t2,$s3 ImprimirEuclídea
331 # imprimir mensaje "La distancia no se puede mostrarr por Lab Sim...."
332 la $a0,msg12
333 li $v0,4
334 syscall
335 j Factorial
336
337 ImprimirEuclídea:#-----
338
339 #Comprobar si el numero es de dos digitos(mayor igual que 10)
340 li $s3,10
341 bge $t2,$s3,TieneDosDigitos_
342 blt $t2,$s3,TieneUnDigito_

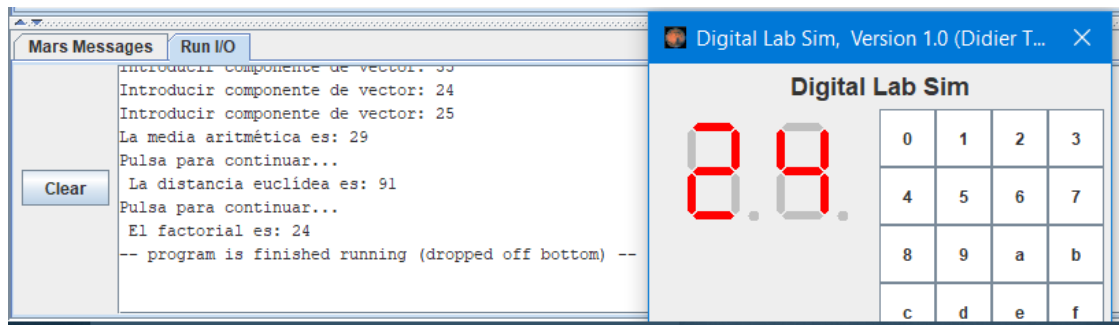
```

Código repetido

### 3.3. Finalmente, mostrar **el factorial** del número del tamaño del vector también por consola y por la tools.

```
297 Factorial: #-----
298 # Indice
299 li $s3,0
300 # Guardamos la multiplicacion en una variable
301 li $s4,1
302 InicioBucle02:
303 addi $s3,$s3,1
304 mult $s3,$s4
305 mflo $s4
306 blt $s3,$t0 InicioBucle02
307 # imprimir mensaje
308 la $a0,msg07
309 li $v0,4
310 syscall
311 # imprime el resultado por pantalla
312 move $a0,$s4
313 li $v0,1
314 syscall
315
316
317
```

```
580 li $t8,4
581 #if
582 bgt $t0,$t8 Error_NoLabSim
583 #Imprime el resultado por Lab Sim
584 #Ponemos un indice
585 li $t1,1
586 li $t2,2
587 li $t3,6
588 li $t4,24
589 # Comparamos
590 beq $s4,$t1 Mostrar_1
591 beq $s4,$t2 Mostrar_2
592 beq $s4,$t3 Mostrar_6
593 beq $s4,$t4 Mostrar_24
594 #Activamos las secciones correspondientes del Digital Lab Sim
595 Mostrar_1:
596 li $t4,0x00
597 li $t5,0x6
598 j Imprimir_1
599 Mostrar_2:
600 li $t4,0x00
601 li $t5,0x5B
602 j Imprimir_1
603 Mostrar_6:
604 li $t4,0x00
605 li $t5,0x7D
606 j Imprimir_1
607 Mostrar_24:
608 li $t4,0x5B
609 li $t5,0x66
610 j Imprimir_1
```



## #Comentarios sobre la práctica

Lo más difícil ha sido mostrar un número por el Digital Lab sim, y mucho más mostrar un número de dos cifras.

No he trabajado con decimales ya que para su expresión por la tools volvería a duplicar mucho código.

Seguramente el código se podría mejorar para no tener que repetir tantas instrucciones que son iguales y lo hacen muy extenso y complicado.

Así y todo, he intentado comentar cada paso para que quede claro y poder aclararme yo también.

He intentado incluir soluciones y avisos a la mayoría de errores que se podrían cometer por parte del usuario, pero aun así puede que me deje alguno.

Gracias a esto es posible que el programa funcione bien aunque el vector sea 5 o más, pero sin mostrar en Lab Sim.

El factorial es solamente la elevación al cuadrado de la suma de todos los componentes del vector.