

```

1  .data
2  CONTROL:    .word32 0x10000
3  DATA:      .word32 0x10008
4
5  .text
6  lwu    $s0, DATA($zero)          ; Cargo dir DATA en $s0
7  lwu    $s1, CONTROL($zero)        ; Cargo dir CONTROL en $s1
8  daddi  $t0, $zero, 6               ; Orden: limpiar pantalla alfanumerica
9  sd     $t0, 0($s1)                 ; Lo mando a CONTROL (se limpia...)
10
11 loop_main:  jal    ingreso          ; Leo un caracter valido
12            dadd   $a0, $zero, $v0   ; Primer argumento
13            jal    ingreso          ; Leo otro caracter valido
14            dadd   $a1, $zero, $v0   ; Segundo argumento
15            jal    resultado        ; Mostrar suma
16            j      loop_main
17            halt
18
19 ; RECIBO DIRECCIONES E/S POR REGISTROS $s0 y $s1
20 ; DEVUELVO UN CARACTER VALIDO POR REGISTRO $v0
21 ingreso:   daddi  $t0, $zero, 9      ; Orden: leer un caracter
22 reintentar: sd     $t0, 0($s1)        ; Lo mando a CONTROL
23            ld     $v0, 0($s0)        ; Posible retorno: caracter leído
24            slti   $t1, $v0, 0x30     ; Comparo con '0'
25            bnez   $t1, reintentar    ; Reintentar si es menor a '0'
26            slti   $t1, $v0, 0x3A     ; Comparo con siguiente a '9' (NO 0x40!!)
27            beqz   $t1, reintentar    ; Reintentar si es mayor a '9'
28            jr     $ra
29
30 ; RECIBO DIRECCIONES E/S POR REGISTROS $s0 y $s1
31 ; RECIBO DOS CARACTERES DIGITOS POR REGISTROS $a0 y $a1
32 resultado: dadd   $t0, $a0, $a1      ; Sumar caracteres
33            daddi  $t0, $t0, -0x60    ; Castear a entero
34            daddi  $t1, $zero, 1      ; Orden: imprimir unsigned int
35            sd     $t0, 0($s0)        ; Mando resultado a DATA
36            sd     $t1, 0($s1)        ; Mando orden a CONTROL (se imprime...)
37            jr     $ra

```