

```

1      .data
2  CTRL: .word32 0x10000      ; direcciones E/S, se colocan en memoria porque el MIPS
3  DATA: .word32 0x10008      ; no deja poner valores inmediatos mayores a 16 bits
4
5  Msj0: .ascii "Ingrese base (punto flotante): "
6  Msj1: .ascii "Ingrese exponente (entero sin signo): "
7  Msj2: .ascii "El resultado es: "
8  Ini:  .double 1.0
9
10
11     .text
12  lwu   $s0, CTRL($0)        ; constantes en todo el programa
13  lwu   $s1, DATA($0)
14  daddi $t0, $0, 6           ; orden: limpiar pantalla alfanumérica
15  daddi $t1, $0, 4           ; orden: imprimir texto
16  daddi $t2, $0, 8           ; orden: leer un numero
17  daddi $t3, $0, 3           ; orden: imprimir un flotante
18  daddi $t4, $0, Msj0        ; direccion del primer mensaje
19  daddi $t5, $0, Msj1        ; direccion del segundo mensaje
20  daddi $t6, $0, Msj2        ; direccion del ultimo mensaje
21  sd    $t0, 0($s0)          ; control: limpiar pantalla
22  sd    $t4, 0($s1)          ; data: mensaje ingresar base
23  sd    $t1, 0($s0)          ; control: mostrar en pantalla
24  sd    $t2, 0($s0)          ; control: leer base
25  l.d   f0, 0($s1)           ; data: traer base
26  sd    $t5, 0($s1)          ; data: mensaje ingresar exponente
27  sd    $t1, 0($s0)          ; control: mostrar en pantalla
28  sd    $t2, 0($s0)          ; control: leer exponente
29  ld     $a0, 0($s1)          ; data: traer exponente
30  jal    a_la_potencia        ; realizar operacion
31  sd    $t6, 0($s1)          ; data: mensaje final
32  sd    $t1, 0($s0)          ; control: mostrar en pantalla
33  s.d   f2, 0($s1)           ; data: resultado
34  sd    $t3, 0($s0)          ; control: mostrar un flotante
35  halt
36
37  ; recibo base por f0 y exponente por a0 (cantidad multiplicaciones)
38  ; devuelve el resultado por el registro f2
39  a_la_potencia: l.d    f2, Ini($0)      ; resultado inicial
40                beqz    $a0, terminar
41  loop_pot:      daddi    $a0, $a0, -1     ; decrementar restantes
42                mul.d    f2, f2, f0      ; multiplicar por base
43                bnez     $a0, loop_pot
44  terminar:      jr      $ra

```