

```
1: .data
2: number: .word 5          # Definimos un número que vamos a cuadrar
3: message: .asciiz "El cuadrado es: " # Mensaje para imprimir antes del resultado
4:
5: .text
6: .globl main
7:
8: main:
9:     lw $a0, number        # Cargamos el valor de 'number' en $a0
10:    jal square             # Llamamos a la función 'square'
11:
12:    # Guardamos el resultado en un registro temporal
13:    move $t0, $v0          # Mover el resultado a $t0 para preservarlo
14:
15:    # Imprimir el mensaje "El cuadrado es: "
16:    li $v0, 4              # Syscall para imprimir cadena
17:    la $a0, message        # Cargar la dirección del mensaje en $a0
18:    syscall                # Llamamos a la syscall
19:
20:    # Imprimir el resultado (el cuadrado calculado)
21:    li $v0, 1              # Syscall para imprimir entero
22:    move $a0, $t0          # Mover el resultado almacenado en $t0 a $a0
23:    syscall                # Llamamos a la syscall
24:
25:    # Finalizar el programa
26:    li $v0, 10             # Syscall para salir
27:    syscall                # Llamamos a la syscall para terminar el programa
28:
29:    # Función: square
30:    # Propósito: Calcular el cuadrado de un número
31:    # Argumento: $a0 (el número a cuadrar)
32:    # Retorno: $v0 (el cuadrado del número)
33: square:
34:     mul $v0, $a0, $a0      # Multiplica $a0 por sí mismo y almacena el resultado en $v
35:     jr $ra                # Retorna a la dirección almacenada en $ra
36:
```