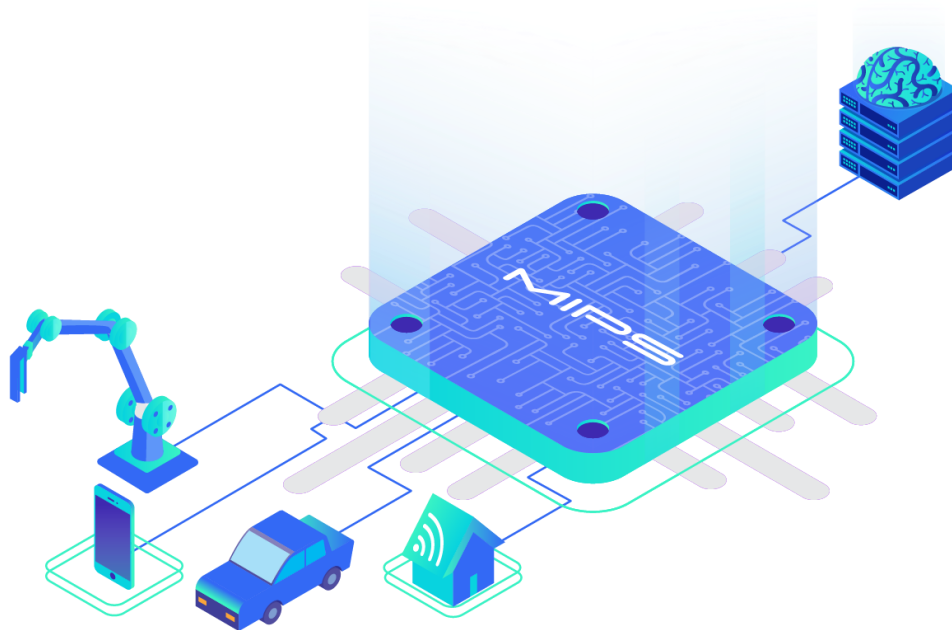


# Diseño e implementación de programas en lenguaje ensamblador de la arquitectura MIPS

A continuación se detallan los programas que han de implementarse en lenguaje ensamblador. El código de cada programa ha de incorporarse al portafolio como evidencia de aprendizaje.



## Normas de diseño de los programas

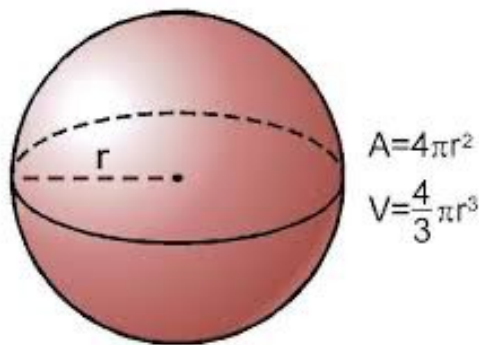
1. Priorizar la claridad del código frente a la eficiencia.
2. Usar comentarios con el fin de:
  - a. Identificar las partes del código.
  - b. Distinguir las llamadas al sistema.
  - c. Entender qué hace cada instrucción.
3. Formatear la salida de resultados según los ejemplos mostrados.
4. Comprobar el buen funcionamiento de los programas por medio de distintos valores de entrada.

# Programa #1 - Arquímedes

Arquímedes de Siracusa (ca. 287 a. n. e. - ca. 212 a. n. e.) está considerado uno de los científicos más grandes de la Antigüedad. Este físico, matemático, ingeniero, inventor y astrónomo griego demostró que el volumen y el área de la esfera son dos tercios de los del cilindro que la inscribe, incluyendo sus bases, lo cual se consideró el más grande de sus descubrimientos matemáticos. De acuerdo con su voluntad, encima de su tumba se colocaron una esfera y un cilindro, algo que pudo ver Cicerón unos cuantos años más tarde en la ciudad siciliana de Siracusa.



Implemente un programa que calcule la superficie y el área de una esfera. El radio de la esfera se introduce por teclado. Utilice aritmética real.



## Ejemplo de salida

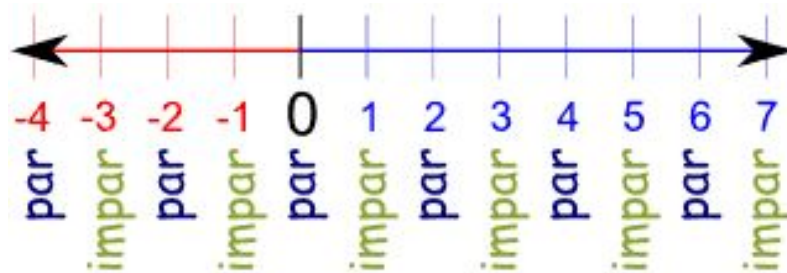
```
Programa #1
Cálculo del área y volumen de una esfera

Introduce el lado: ???
Área: ???
Volumen: ???

Fin del programa
```

## Programa #2 - Par o impar

Diseña un programa que indique si un número entero es par o impar. En matemáticas, un número entero es par si es divisible entre dos. Sin embargo, en el programa no está permitido utilizar la instrucción de división. El programa finaliza cuando el número introducido es cero.



### Ejemplo de salida

```
Programa #2
Identifica si un número es par o impar

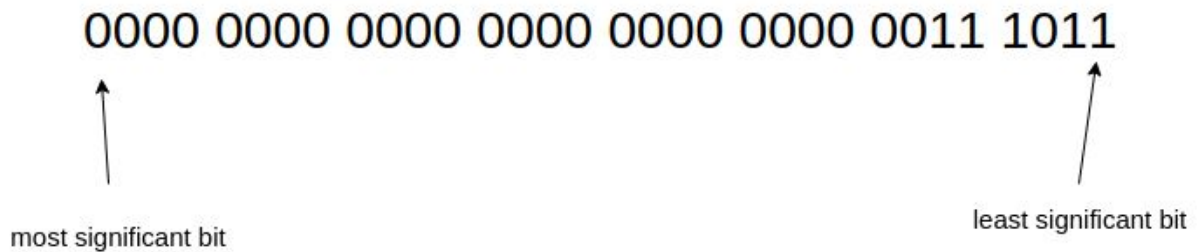
Introduce un número: 23
El número es impar
Introduce un número: 450
El número es par
Introduce un número: -97
El número es impar
Introduce un número: 0
El número es par

Fin del programa
```

## Programa #3 - Unos

Diseña un programa que, dado un número entero introducido desde el teclado, calcule el número de unos que hay en la palabra bits que lo representa. El programa finaliza cuando el número introducido es cero.

Por ejemplo, el número 73 se codifica en complemento a dos como 0x0000003B; por lo tanto, tiene 5 unos. La figura representa la codificación del número 73 en 32 bits.



### Ejemplo de salida

```
Programa #3
Calcula el número de unos de una palabra de bits

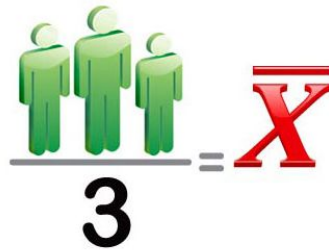
Introduce un número: 73
El número de unos es: 5
Introduce un número: -1
El número de unos es: 32
Introduce un número: 255
El número de unos es: 8
Introduce un número: 0

Fin del programa
```

## Programa #4 - Vector de enteros

Diseñe un programa que declare e inicialice en memoria un vector con 20 números enteros (se pueden elegir los valores que se quiera).

A continuación imprime los elementos del vector y calcula los siguientes valores: mínimo, máximo y suma; finalmente, también imprime la media aritmética pero se calcula utilizando aritmética real a fin de no perder precisión.



### Ejemplo de salida

```
Programa #4
Cálculos sobre un vector de enteros

Vector: -1 13 5 -5 9 4 -3 9 4 -7 24 2 15 6 -8 17 1 35 11 -2

Mínimo = -8
Máximo = 35
Suma    = 129
Media   = 6.45

Fin del programa
```

# Programa #5 - Fibonacci

Leonardo de Pisa (ca. 1170-1240), también conocido como Fibonacci, fue un matemático italiano. Difundió en Europa la utilidad práctica del sistema de numeración indoarábigo frente a la numeración romana, y fue el primer europeo en describir la sucesión numérica que lleva su nombre.



Diseñe un programa que imprima en pantalla la sucesión o función de Fibonacci, que se calcula como sigue:

$$\begin{aligned}a_1 &= 1 \\a_2 &= 1 \\a_3 &= 2 \\a_4 &= 3 \\a_5 &= 5 \\a_6 &= 8 \\a_7 &= 13 \\&\dots \\a_n &= a_{n-1} + a_{n-2}\end{aligned}$$



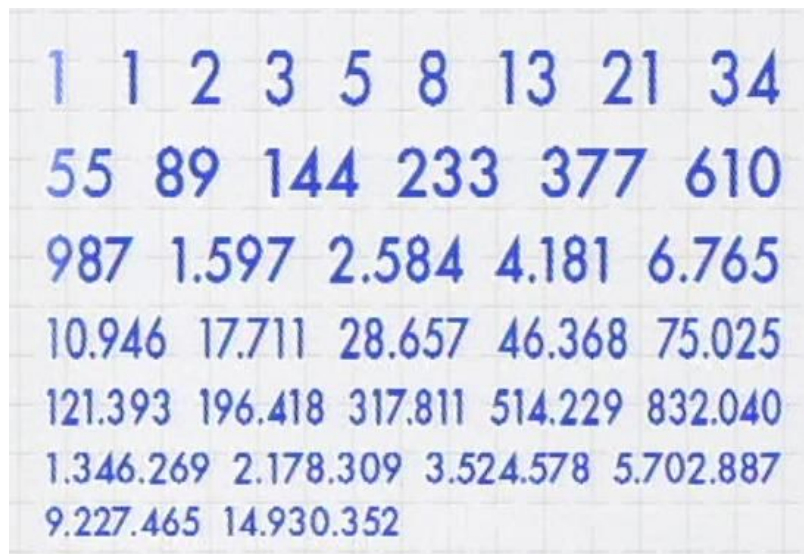
## Ejemplo de salida

```
Programa #5
Cálculo de la sucesión de Fibonacci

Introduce el término n: 25

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
6765 10946 17711 28657 46368 75025

Fin del programa
```



```
1 1 2 3 5 8 13 21 34
55 89 144 233 377 610
987 1.597 2.584 4.181 6.765
10.946 17.711 28.657 46.368 75.025
121.393 196.418 317.811 514.229 832.040
1.346.269 2.178.309 3.524.578 5.702.887
9.227.465 14.930.352
```

Dado que utilizamos aritmética entera es posible que se produzca un desbordamiento cuando se intente calcular el término  $n$ -ésimo, con  $n$  suficientemente grande. Determine haciendo uso de la reflexión y la paciencia, cuál es el término de la sucesión más grande que se puede calcular en aritmética entera. Nota importante: no olvide considerar, cuando se calcula el nuevo término de la sucesión, la diferencia entre las instrucciones `add` y `addu`, y tenga en cuenta que la llamada al sistema `print_int` interpreta el argumento como un número entero con signo codificado en complemento a dos. Así se las ponían a Fernando VII.