



# **DESARROLLO DE ALGORITMOS SENCILLOS**

**Recursividad**

**UCO**  
**ONLINE**

# Desarrollo de algoritmos

## Objetivos

Desarrollar pequeños programas en Python que utilicen funciones recursivas

## Temporización

40 minutos

## Enunciados

1. Escribe una solución recursiva que calcule el algoritmo de Euclides, usado para calcular el máximo común divisor de dos enteros. El algoritmo de Euclides se describe del siguiente modo:

$$\text{mcd}(x, y) = x, \text{ si } y = 0$$

$$\text{mcd}(x, y) = \text{mcd}(y, \text{mod}(x, y)) \text{ si } y > 0 \text{ y } x \geq y$$

2. Construye una función recursiva que calcule la división entera de dos números mediante el métodos de restas sucesivas. Implementa un pequeño programa para probarla.
3. Codifica una función recursiva que permita sumar los dígitos de un número. Implementa un programa para probarla.

### Ejemplo

*Entrada: 124*

*Salida →7*

4. Codifica una función recursiva que permita calcular el valor de  $\pi$  usando la serie de Leibniz:  $\pi = (4 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + 4/13 \dots)$

El programa recibirá como argumento el número “n” de términos a usar en la aproximación.

# Soluciones

## Ejercicio 1

```
# Programa que calcula el mcd mediante una función recursiva

import sys

def mcd(n, m):
    if m == 0:
        return n
    else:
        return mcd(m, n % m)

def main():
    if len(sys.argv) == 3:
        num1 = int(sys.argv[1])
        num2 = int(sys.argv[2])

        print(f'mcd({num1}, {num2})={mcd(num1, num2)}')
    else:
        print(f'Sintaxis incorrecta: {sys.argv[0]} <num1> <num2>')

if __name__ == '__main__':
    main()
```

## Ejercicio 2

```
# División entera de dos número mediante el método de restas sucesivas

import sys

def division(num1, num2):
    if num1 < num2:
        return 0
    else:
        return 1 + division(num1-num2, num2)
```

```
def main():
    if len(sys.argv) == 3:
        num1 = int(sys.argv[1])
        num2 = int(sys.argv[2])

        print(f'{num1}/{num2}={division(num1, num2)}')

    else:
        print(f'Sintaxis incorrecta: {sys.argv[0]} <num1>
<num2>')

if __name__ == '__main__':
    main()
```

### Ejercicio 3

```
# Programa que suma los digitos de un numero de manera
recursiva

import sys

def suma_digitos(numero):
    if numero == 0:
        return 0
    else:
        return numero % 10 + suma_digitos(numero//10)

def main():
    if len(sys.argv) == 2:
        numero = int(sys.argv[1])
        print(f'La suma de los digitos de {numero} es
{suma_digitos(numero)}')
    else:
        print(f'Sintaxis incorrecta: {sys.argv[0]} <numero>')

if __name__ == '__main__':
    main()
```

## Ejercicio 4

```
# Programa que calcula el número pi usando la serie de
Leibniz

import sys

def serie_Leibniz(termino):
    if termino == 1:
        return 4
    elif termino % 2 == 0:
        return -4/(2*(termino-1)+1) + serie_Leibniz(termino-
1)
    else:
        return 4/(2*(termino-1)+1) + serie_Leibniz(termino-1)

def main():
    if len(sys.argv) == 2:
        termino = int(sys.argv[1])
        print(f'El valor de pi con una precision de {termino}
terminos es \
{serie_Leibniz(termino)}')
    else:
        print(f'Sintaxis incorrecta: {sys.argv[0]}
<termino>')

if __name__ == '__main__':
    main()
```