

## Práctico 4: Estructuras repetitivas

### Objetivo:

Implementar ciclos para resolver problemas que requieran repetición de acciones.

### Resultados de aprendizaje:

1. Diseño y Desarrollo de Algoritmos Eficientes: El estudiante será capaz de diseñar y desarrollar algoritmos utilizando estructuras de control repetitivas (FOR, WHILE) para resolver problemas matemáticos y de lógica, aplicando correctamente operaciones matemáticas y cálculos.
2. Escritura Eficaz de Pseudocódigo y Documentación: El estudiante podrá escribir pseudocódigo de manera estructurada y clara, utilizando comentarios para explicar el funcionamiento de cada parte del algoritmo.
3. Interacción con el Usuario y Validación de Datos: El estudiante será capaz de diseñar programas que interactúen con el usuario, solicitando datos de entrada y proporcionando resultados claros y concisos.

### Actividades

- 1) Crea un programa que imprima en pantalla todos los números enteros desde 0 hasta 100 (incluyendo ambos extremos), en orden creciente, mostrando un número por línea.
- 2) Desarrolla un programa que solicite al usuario un número entero y determine la cantidad de dígitos que contiene.
- 3) Escribe un programa que sume todos los números enteros comprendidos entre dos valores dados por el usuario, excluyendo esos dos valores.
- 4) Elabora un programa que permita al usuario ingresar números enteros y los sume en secuencia. El programa debe detenerse y mostrar el total acumulado cuando el usuario ingrese un 0.
- 5) Crea un juego en el que el usuario deba adivinar un número aleatorio entre 0 y 9. Al final, el programa debe mostrar cuántos intentos fueron necesarios para acertar el número.
- 6) Desarrolla un programa que imprima en pantalla todos los números pares comprendidos entre 0 y 100, en orden decreciente.
- 7) Crea un programa que calcule la suma de todos los números comprendidos entre 0 y un número entero positivo indicado por el usuario.

8) Escribe un programa que permita al usuario ingresar 100 números enteros. Luego, el

programa debe indicar cuántos de estos números son pares, cuántos son impares, cuántos son negativos y cuántos son positivos. (Nota: para probar el programa puedes usar una cantidad menor, pero debe estar preparado para procesar 100 números con un solo cambio).

9) Elabora un programa que permita al usuario ingresar 100 números enteros y luego calcule la media de esos valores. (Nota: puedes probar el programa con una cantidad menor, pero debe poder procesar 100 números cambiando solo un valor).

10) Escribe un programa que invierta el orden de los dígitos de un número ingresado por el usuario. Ejemplo: si el usuario ingresa 547, el programa debe mostrar 745.