

Cargo:	Docente		
Nombre:	M.Sc. David Fabián Cevallos Salas		
Asignatura:	Lógica de Programación		
Carrera:	Marketing Digital y Comercio Electrónico	Nivel:	Primer nivel
Estudiante:			

# ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

### RESOLUCIÓN DE PROBLEMAS ITERATIVOS COMPLEJOS.

# 1. Objetivos

- Practicar la resolución de problema mediante procesos iterativos complejos.
- Practicar la resolución de problemas básicos de programación: El estudiante resolverá cinco problemas sencillos de programación, enfocándose en operaciones simples con variables y lectura/escritura de datos por consola.
- Desarrollar habilidades de análisis y diseño de algoritmos: El estudiante deberá analizar el problema y escribir un pseudocódigo claro que lo resuelva.
- Fomentar el aprendizaje estructurado en la programación: El estudiante podrá comprobar el funcionamiento de sus algoritmos a través de pruebas y, si lo desea, implementarlos en un lenguaje de programación.

#### 2. Antecedentes/Escenario

La programación básica incluye operaciones simples, como la entrada de datos, operaciones aritméticas y la salida de resultados. Los ejercicios a resolver se centrarán en estos aspectos, sin la necesidad de estructuras complejas. El objetivo es familiarizar al estudiante con el proceso de descomponer un problema y representarlo mediante pseudocódigo antes de llegar a la implementación en código real.

### 3. Recursos necesarios

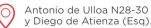
- Hoja en blanco o procesador de texto para escribir pseudocódigo, análisis y resultados.
- Lenguaje de programación (opcional), como Python, C++, Java o cualquier otro con el que el estudiante esté familiarizado, para la implementación opcional.
- Tiempo estimado: 2 a 3 horas para resolver los cinco problemas, escribir el pseudocódigo y hacer las pruebas.

### 5. Pasos por realizar

• Paso 1: Analizar el Problema











Lee y comprende el enunciado de cada ejercicio. Piensa qué datos necesitas recibir y qué operaciones realizarás con esos datos.

### Paso 2: Pseudocódigo

Escribe el pseudocódigo correspondiente para cada ejercicio, estructurando el algoritmo de manera clara y sencilla.

#### Paso 3: Prueba de Escrito

Realiza una prueba escrita de cómo se comportará el pseudocódigo con ciertos valores de entrada.

# Paso 4: Implementación (Opcional)

Si lo deseas, implementa el pseudocódigo en un lenguaje de programación.

### Paso 5: Revisión

Revisa tu trabajo para asegurarte de que las soluciones sean correctas y claras. Realiza las pruebas necesarias para validar el correcto funcionamiento de cada ejercicio.

# 6. Desarrollo

Presente los ejercicios con los siguientes elementos por cada uno:

- Enunciado
- Análisis
- Pseudocódigo
- Prueba de escrito
- (Opcional) Implementación en un lenguaje de programación

### Ejercicio 1

Utilizando un bucle, imprimir los primeros 10 números naturales (del 1 al 10).

### Ejercicio 2

Utilizando un bucle, calcular la suma de los primeros 5 números naturales (1 + 2 + 3 + 4)+ 5).

### Ejercicio 3

Utilizando un bucle, imprimir todos los números pares entre 1 y 20.

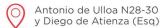
#### Ejercicio 4

Leer un número **n** y utilizar un bucle para imprimir la tabla de multiplicar de ese número (hasta 10).

Ejercicio 5











Utilizando un bucle, contar cuántos números impares existen entre 1 y un número n mayor a 1 leído desde teclado.







