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

LABORATORIO: IMPLEMENTACIÓN DE SENTENCIAS DE CONTROL PARA SOLUCIÓN DE CASOS PRÁCTICOS.

1. Objetivos

- **Aprender a implementar** sentencias de control condicionales para resolución de problemas.
- **Reconocer la importancia** de la sentencia de control if e if-else.
- **Aprender la estructura del pseudocódigo**: El estudiante desarrollará la habilidad de escribir algoritmos de manera estructurada usando pseudocódigo, sin necesidad de un lenguaje de programación específico.
- **Resolver problemas mediante pseudocódigo**: El estudiante será capaz de representar soluciones a problemas mediante algoritmos de pseudocódigo, utilizando estructuras de control, variables y operaciones básicas.
- **Fomentar el pensamiento lógico y secuencial**: El estudiante aplicará su lógica para organizar y estructurar soluciones a problemas de programación.

2. Antecedentes/Escenario

El pseudocódigo es una forma de describir algoritmos utilizando un lenguaje natural que no está vinculado a un lenguaje de programación específico. Se utiliza para planificar la solución de problemas antes de escribir código real. Un pseudocódigo debe describir claramente la lógica del algoritmo, como si se estuviera dando instrucciones a una persona, utilizando estructuras como secuencias, decisiones (condicionales) y repeticiones (bucles).

El objetivo de esta actividad es que el estudiante sea capaz de representar soluciones a problemas sencillos de programación mediante pseudocódigo. El pseudocódigo debe ser claro y lógico, utilizando las estructuras adecuadas para que cualquier persona pueda entenderlo. La tarea consiste en resolver problemas de programación básicos con pseudocódigo y, al hacerlo, demostrar cómo se puede pensar en la solución antes de escribir código.

3. Recursos necesarios

- **Planteamiento de problemas**: Se necesitan problemas de programación para aplicar el pseudocódigo.
- **Herramientas para escribir el pseudocódigo**: Hoja en blanco, procesador de texto o cualquier herramienta de notas.
- **Tiempo estimado**: 1 a 2 horas para practicar la escritura de pseudocódigo y ejemplos.
- **4. Planteamiento del problema**

5. Pasos por realizar

5.1. Paso 1: Comprensión del Problema

- **Acción**: Antes de escribir el pseudocódigo, lee cuidadosamente el problema de programación. Asegúrate de entender qué se requiere como salida, qué entrada necesitas y cómo debe funcionar el proceso.
- **Consejo**: Si el problema es complejo, trata de dividirlo en pasos más pequeños.

5.2. Paso 2: Identificación de las Estructuras

- **Acción**: Identifica las estructuras que utilizarás en el pseudocódigo, como:
  - **Secuencia**: Paso a paso, de arriba hacia abajo.
  - **Condicionales**: Estructuras como `Si... entonces... sino...` para decisiones.
  - **Bucles**: `Mientras` o `Para` para repetir acciones.
- **Consejo**: No te olvides de usar palabras simples y claras en tu pseudocódigo.

5.3. Paso 3: Escribir el Pseudocódigo

- **Acción**: Escribe el pseudocódigo que resuelve el problema, organizando las instrucciones paso a paso. Recuerda:
  - Usar un lenguaje claro y sencillo.
  - Evitar detalles innecesarios del lenguaje de programación.
  - Ser específico con las operaciones que deben realizarse.
- **Consejo**: Si es un problema complejo, divide el pseudocódigo en partes más pequeñas.

5.4. Paso 4: Ejemplo de Pseudocódigo

- **Acción**: A continuación, se presentan algunos ejemplos sencillos de pseudocódigo para resolver problemas básicos de programación:

Pseudocódigo para calcular la suma de dos números:

```
Inicio
   Leer número1
   Leer número2
   suma ← número1 + número2
   Imprimir suma
Fin
```

6. Desarrollo

Presente varios ejercicios con los siguientes elementos por cada uno:

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