|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorios de docencia |

Laboratorio de Computación

Salas A y B

*Profesor(a):* Ing. Karina García Morales

*Asignatura:* Fundamentos de Programación

*Grupo:* 22

*No de Práctica(s):* Práctica 5

*Integrante(s):*  Edilberto Vicente Martínez

*No. de lista o brigada: 49*

*Semestre:* 2026-1

*Fecha de entrega:* 23 de septiembre de 2025.

*Observaciones:*

# 

# CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Práctica 05: Pseudocódigo**

Objetivo:

El alumno elaborará pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

Desarrollo:

Para que la solución de un problema (algoritmo) pueda ser codificada, se debe generar una representación de éste. Una representación algorítmica elemental es el pseudocódigo. Un pseudocódigo es la representación escrita de un algoritmo, es decir, muestra en forma de texto los pasos a seguir para solucionar un problema. El pseudocódigo posee una sintaxis propia para poder realizar la representación del algoritmo (solución de un problema).

**Sintaxis de pseudocódigo**

El lenguaje pseudocódigo tiene diversas reglas semánticas y sintácticas. A continuación,

se describen las más importantes:

1. Alcance del programa: Todo pseudocódigo está limitado por las etiquetas de

INICIO y FIN. Dentro de estas etiquetas se deben escribir todas las instrucciones

del algoritmo.

2. Palabras reservadas con mayúsculas: Todas las palabras propias del pseudocódigo

deben de ser escritas en mayúsculas.

3. Sangría o tabulación: El pseudocódigo debe tener diversas alineaciones para que

el código sea más fácil de entender y depurar.

4. Lectura / escritura: Para indicar lectura de datos se utiliza la etiqueta LEER. Para

indicar escritura de datos se utiliza la etiqueta ESCRIBIR.

5. Declaración de variables: La declaración de variables la definen un identificador

(nombre), seguido de dos puntos, seguido del tipo de dato, es decir:

<nombreVariable>:<tipoDeDato>

Los tipos de datos que se pueden utilizar son:

ENTERO -> valor entero positivo y/o negativo

REAL -> valor con punto flotante y signo

BOOLEANO -> valor de dos estados: verdadero o falso

CARACTER -> valor tipo carácter

CADENA -> cadena de caracteres

Es posible declarar más de una variable de un mismo tipo de dato utilizando

arreglos, indicando la cantidad de variables que se requieren, su sintaxis es la

siguiente:

<nombreVariable>[cantidad]:<tipoDeDato>

Existe un tipo de dato compuesto, es decir, que puede contener uno o más tipos de

datos simples diferentes. Este tipo de dato se conoce como registro o estructura y su

sintaxis es la siguiente:

<nombreRegistro>:REG

<nombreVariable\_1>:<tipoDeDato>

...

<nombreVariable\_N>:<tipoDeDato>

FIN REG

Para crear una variable tipo registro se debe indicar el nombre del registro y el

nombre de la variable. Para acceder a los datos del registro se hace uso del operador

punto (.)

Es posible crear variables constantes con la palabra reservada CONST, la cual indica que un identificador no cambia su valor durante todo el pseudocódigo. Las constantes (por convención) se escriben con mayúsculas y se deben inicializar al momento de declararse.

6. **Operadores aritméticos**: Se tiene la posibilidad de utilizar operadores aritméticos y

lógicos:

Operadores aritméticos: suma (+), resta (-), multiplicación (\*), división (/), división

entera (div) esto ya es en programación hay que omitirla, módulo (mod),

exponenciación (^), asignación (:=).

Operadores lógicos: igualdad (=), Y-lógica o AND (&), O-lógica u OR (|), negación

o NOT (!), relaciones de orden (<, >, <=, >=) y diferente (<>).

La tabla de verdad de los operadores lógicos Y (AND), O (OR) y Negación (NOT) se

describe en la Tabla 1.

**7. Notación de camello.** Para nombrar variables y nombres de funciones se debe hacer

uso de la notación de camello.

En la notación de camello (llamada así porque parecen las jorobas de un camello) los

nombres de cada palabra empiezan con mayúscula y el resto se escribe con

minúsculas. Existen dos tipos de notaciones de camello: lower camel case que en la cual la primera letra de la variable inicia con minúscula y upper camel case en la cual todas las palabras inician con mayúscula. No se usan puntos ni guiones para separar las palabras (a excepción de las constantes que utilizan guiones bajos). Además, para saber el tipo de variable se recomienda utilizar un prefijo.

Estructuras de control de flujo

Las estructuras de control de flujo permiten la ejecución condicional y la repetición de

un conjunto de instrucciones. Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

**Estructura de control secuencial**

Las estructuras de control secuenciales son las sentencias o declaraciones que se realizan

una a continuación de otra en el orden en el que están escritas.

**Estructuras de control condicionales (o selectivas)**

Las estructuras de control condicionales permiten evaluar una expresión lógica (condición que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes (o se realiza una acción o se realiza la otra)

La estructura de control de flujo más simple es la estructura condicional SI, su sintaxis

es la siguiente:

Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se realizan las instrucciones del bloque [Acciones]. Si no se cumple la condición, se continúa con el flujo del pseudocódigo descartando las [Acciones].

Además de la estructura de control condicional simple SI, existe la estructura condicional completa nombrada SI-DE LO CONTRARIO. Se evalúa la expresión lógica y si se cumple (si es verdadera) se realizan las instrucciones del bloque SI [Acciones ENTONCES]. Si no se cumple la expresión lógica (si es falsa) se realizan las instrucciones del bloque DE LO CONTRARIO [Acciones DE LO CONTRARIO]. Al final de la estructura condicional, el flujo continúa con las estructuras que le sigan.

Otra estructura de control condicional es SELECCIONAR-CASO, la cual valida el valor de la variable que está entre paréntesis y comprueba si es igual al valor que está definido en cada caso. Si la variable no tiene el valor de ningún caso se va a la instrucción por defecto (DEFECTO).

**Estructuras de control iterativas o repetitivas**

Las estructuras de control de flujo iterativas o repetitivas (también llamadas cíclicas) permiten realizar una serie de instrucciones mientras se cumpla la expresión lógica. Existen dos tipos de expresiones cíclicas MIENTRAS y HACER- MIENTRAS.

La estructura MIENTRAS (WHILE en inglés) primero valida la condición y si ésta es verdadera procede a realizar el bloque de instrucciones de la estructura [Acciones] y regresa a validar la condición, esto lo realiza mientras la condición sea verdadera; cuando la condición es Falsa (no se cumpla) se rompe el ciclo y continúa el flujo normal del pseudocódigo.

El final de la estructura lo determina la etiqueta FIN MIENTRAS. La estructura HACER-MIENTRAS primero realiza las instrucciones descritas en la estructura y después valida la expresión lógica.

Si la condición se cumple vuelve a realizar las instrucciones de la estructura, de lo contrario rompe el ciclo y sigue el flujo del pseudocódigo. Esta estructura asegura que, por lo menos, se realiza una vez el bloque de la estructura, ya que primero realiza las instrucciones del bloque y después pregunta por la condición.

**Ejercicios de clase**

1. **¿Qué hace la siguiente secuencia?**

i ← 1   
MIENTRAS i <= 3 HACER   
    ESCRIBIR i   
    i ← i + 1   
FIN MIENTRAS 

         A) Imprime 1, 2, 3 y luego termina.  
         B) No imprime nada.  
         C) Imprime 1, 2 y luego entra en ciclo infinito.  
         D) Imprime 3 veces el número 1.

1. **Genera el pseudocódigo de un problema que solicite una calificación entre 5 y 10, mostrar en pantalla, en caso contrario, volver a pedir la calificación (Estructura hacer mientras)**

INICIO

DEFINIR varCal: ENTERO

HACER

ESCRIBIR “Introduce una calificación entre 5 y 10”.

LEER varCal

MIENTRAS varCal< 5 && varCal >10

ESCRIBIR varCal

FIN

**c) Genera el pseudocódigo para el cálculo de eficiencia de una línea de producción**

La eficiencia se calcula con la fórmula:  
Eficiencia = (Producción real / Producción esperada) × 100

Escribe un pseudocódigo que lea ambos valores y determine si  
la eficiencia es:

* Mayor  
  al 90% → "Eficiencia alta"
* Entre  
  70% y 90% → "Eficiencia media"
* Menor al 70% → "Eficiencia baja"

INICIO

DEFINIR produccionReal, produccionEsperada, eficiencia como REAL

ESCRIBIR "Ingrese la producción real:"

LEER produccionReal

ESCRIBIR "Ingrese la producción esperada:"

LEER produccionEsperada

eficiencia ←

(produccionReal/produccionEsperada) \* 100

ESCRIBIR "La eficiencia es: ", eficiencia, "%"

SI eficiencia > 90 Entonces

ESCRIBIR "Eficiencia alta"

SINO

SI eficiencia >= 70 Entonces

ESCRIBIR "Eficiencia media"

SINO

ESCRIBIR "Eficiencia baja"

FINSI

FIN

Tarea: Genera el pseudocódigo  
  
1.- Solicita un número entero positivo y cuenta desde 1 hasta el número dado (Estructura hacer mientras y mientras).

INICIO

numeroLimite, contador Como Entero

ESCRIBIR “Ingrese un número entero positivo (N):”

LEER numeroLimite

MIENTRAS numeroLimite <= 0 HACER

ESCRIBIR “Error: El número debe ser positivo. Ingrese otro número:”

LEER numeroLimite

FIN MIENTRAS

Inicialización del contador

Contador ← 1

ESCRIBIR “Iniciar conteo desde 1 hasta “, numeroLimite, “:”

MIENTRAS contador <= numeroLimite HACER

Mostrar el valor actual del contador

ESCRIBIR contador

Incrementar el contador en 1

Contador ← contador + 1

FIN MIENTRAS

ESCRIBIR “Conteo finalizado.”

FIN

2.- Genera un menú de los 3 programas (2 de clase y el de tarea), agregar el ciclo hacer mientras, en donde valide, si el usuario no elige las opciones 1, 2 o 3 imprimir un letrero de "error"(es el default) y volver a mostrarle el menú, hasta que seleccione la opción 4 como salir.

INICIO

DEFINIR programa1, programa2, reg tenis, opción COMO ENTERO

HACER

IMPRIMIR “Menú de Programas”

IMPRIMIR “1. Programa1”

IMPRIMIR “2. Programa2”

IMPRIMIR “3. Reg tenis”

IMPRIMIR “4. Salir”

IMPRIMIR “--------------------------“

IMPRIMIR “Por favor, ingrese el número de su opción:”

LEER opcion

SEGÚN opcion

HACER

OPCIÓN 1:

IMPRIMIR “Ha seleccionado ‘programa1’. Ejecutando…”

PAUSA

OPCIÓN 2:

IMPRIMIR “Ha seleccionado ‘programa2’. Ejecutando…”

PAUSA

OPCIÓN 3:

IMPRIMIR “Ha seleccionado ‘reg tenis’. Ejecutando...”

PAUSA

OPCIÓN 4:

IMPRIMIR “Ha seleccionado ‘Salir’. Terminando el programa…”

DEFECTO:

// Letrero de error para opciones no válidas (1, 2, 3, o 4)

IMPRIMIR “¡ERROR! Opción no válida. Por favor, ingrese 1, 2, 3 o 4.”

PAUSA // Breve pausa para que el usuario lea el mensaje de error

FIN SEGÚN

MIENTRAS (opcion ¡= 4)

FIN

3.Contesta las preguntas del siguiente pseudocódigo:  
  
INICIO  
  
    nProductos, i, codProducto, cantStock, cantDefectuosos, totalDefectuosos Como Entero  
    porcentajeDefectos, porcentajeTotal Como Real  
    productoConMasDefectos, maxDefectos Como Entero  
  
    totalDefectuosos ← 0  
    productoConMasDefectos ← 0  
    maxDefectos ← -1  
  
    ESCRIBIR "Ingrese la cantidad de productos a registrar:"  
    LEER nProductos  
  
    i ← 1  
    MIENTRAS i <= nProductos Hacer  
        ESCRIBIR "Producto ", i  
        ESCRIBIR "Ingrese código del producto:"  
        LEER codProducto  
  
        ESCRIBIR "Ingrese cantidad en stock:"  
        LEER cantStock  
  
        ESCRIBIR "Ingrese cantidad de piezas defectuosas:"  
        LEER cantDefectuosos  
  
        porcentajeDefectos ← (cantDefectuosos / cantStock) \* 100  
  
        ESCRIBIR "Porcentaje de defectos del producto ", codProducto, ": ", porcentajeDefectos,"%"  
  
        totalDefectuosos ← totalDefectuosos + cantDefectuosos  
  
        SI cantDefectuosos > maxDefectos Entonces  
            maxDefectos ← cantDefectuosos  
            productoConMasDefectos ← codProducto  
        FIN SI  
  
        i ← i + 1  
    FIN MIENTRAS  
  
    ESCRIBIR "Total de piezas defectuosas en inventario: ", totalDefectuosos  
  
    ESCRIBIR "Calcular porcentaje total de defectos respecto al inventario total"  
    ESCRIBIR "Ingrese cantidad total de piezas en inventario:"  
    LEER cantStock  
  
    porcentajeTotal ← (totalDefectuosos / cantStock) \* 100  
  
    ESCRIBIR "Porcentaje total de defectos: ", porcentajeTotal, "%"  
  
    SI porcentajeTotal > 5 ENTONCES  
        ESCRIBIR "ALERTA: El porcentaje total de defectos supera el 5%. Se recomienda revisar el inventario."  
    SINO  
        ESCRIBIR "El porcentaje total de defectos está dentro del rango aceptable."  
    FINSI  
  
FIN  
  
¿Cuál es el propósito principal de este algoritmo?  
Crear un algoritmo que determine la cantidad y el porcentaje de objetos defectuosos dentro de un rango de n productos.   
  
  
¿Qué datos se deben ingresar para cada producto durante la ejecución del algoritmo?  
 **codProducto** (Código del producto): Un identificador único.

**cantStock**(Cantidad en stock): La cantidad total de piezas de ese producto.

**cantDefectuosos**(Cantidad de piezas defectuosas)

¿Qué significa el valor almacenado en la variable maxDefectos?  
Es la máxima cantidad de piezas defectuosas que hay en el inventario.  
  
¿Por qué se utiliza una variable de tipo real para porcentajeDefectos y porcentajeTotal?  
Porque el porcentaje puede salir en valores decimales.  
  
  
¿Qué estructura de control se utiliza para procesar los productos? ¿Por qué es adecuada en este caso?  
Estructura mientras, en este caso nos sirve porque la acción las veces que sea necesaria hasta que el usuario indique alguna otra cosa.  
  
¿Qué sucede cuando se encuentra un producto con más piezas defectuosas que el producto anterior con más defectos?  
  
  
  
¿Cómo se calcula el porcentaje de defectos para cada producto?

El porcentaje de defectos según el algoritmo se calcula de la siguiente manera:

PorcentajedeDefectos ← (

no pueden Defectuosos / no pueden Stock )×100  
  
¿Cómo modificarías el algoritmo para que, además de mostrar el producto con más defectos, también muestre el producto con menos defectos?

1. **Declarar variables**

productoConMenosDefectoscomo **Entero** .

minDefectoscomo **Entero** .

1. **Inicializar variables**
2. **Lógica de comparación dentro del bucle ej:** Estructura MIENTRAS.  
     
     
   ¿Qué pasaría si en algún producto la cantidad en stock fuera 0? ¿Cómo podrías evitar un error en el cálculo?  
   Se produciría un error en el algoritmo, ya que se llegaría a una división de dos ceros, para poder solucionarlo podríamos utilizar una condicional SI para verificar si el stock es mayor a cero, y si no escribir “error”.  
     
   ¿Qué ventajas tiene calcular el porcentaje total de defectos respecto al inventario total?  
   Sirve para poder ver la calidad de los productos que queremos tener en el inventario, lo que sirve para hacer una evaluación a los procesos de fabricación de dicho producto.

Tarea

Tenis: REG

colorTenis: CADENA

numeroTenis: REAL

marcaTenis: CADENA

FIN REG

Tenis1: REG Tenis

Tenis2: REG Tenis

Tenis3: REG Tenis

Tenis1.color: = “Azul”

Tenis1.numero: = “26”

Tenis1.marca: = “Nike”

Tenis2.color: = “Negro”

Tenis2.numero: = “27.5”

Tenis2.marca: = “Puma”

Tenis3.color: = “Rojo”

Tenis3.numero: = “24”

Tenis3.marca: = “Adidas”

Conclusión

En esta práctica aprendimos sobre las reglas que tenemos que seguir al escribir un pseudocódigo, el cómo se diferencia el pseudocódigo con el código que se utiliza en los diagramas de flujo, también se vio como se trabaja este diagrama en sus distintos tipos de estructuras (secuencial, condicional e iterativa). Creo que es muy importante el conocer este tipo de herramientas porque nos acerca más a la programación que realiza dentro de nuestros dispositivos electrónicos y gracias a ello, podemos darnos cuenta la gran importancia que tiene dentro de nuestra vida cotidiana.

Bibliografía

\*Facultad de Ingeniería. (2025). Manual de prácticas del laboratorio de Fundamentos de

Programación. Laboratorio de computación. Salas A y B. Universidad Nacional Autónoma de

México. pp. 36-51. Recuperado el 6 de octubre de 2025 de <http://lcp02.fi-b.unam.mx/>