



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor(a): Ing. Karina García Morales

Asignatura: Fundamentos de Programación (L)

Grupo: 22

No. de práctica(s): Práctica 5: Pseudocódigo

Integrante(s): González Márquez José Luis

No. de lista o brigada: 20

Semestre: 2026-1

Fecha de entrega: Lunes 6 de octubre de 2025

Observaciones:

CALIFICACIÓN: _____

Objetivo:

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

1.-Conceptos:**Algoritmo:**

Conjunto ordenado y finito de pasos que permiten resolver un problema o realizar una tarea.

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)

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.

Palabras reservadas con mayúsculas:

Todas las palabras propias del pseudocódigo deben de ser escritas en mayúsculas.

Sangría o tabulación:

El pseudocódigo debe tener diversas alineaciones para que el código sea más fácil de entender y depurar

Lectura / escritura:

Para indicar lectura de datos se utiliza la etiqueta LEER. Para indicar escritura de datos se utiliza la etiqueta ESCRIBIR

Declaración de variables:

La declaración de variables la definen un identificador (nombre), seguido de dos puntos, seguido del tipo de dato

Tipos de datos que se pueden utilizar:

- ENTERO -> valor entero positivo y/o negativo
- REAL -> valor con punto flotante y signo
- BOOLEANO -> valor de dos estados: verdadero o falso
- CHARACTER -> valor tipo carácter
- CADENA -> cadena de caracteres

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 (<>).

Notación de camello:

Se usa 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)

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.

La estructura HACER-MIENTRAS:

Primero realiza las instrucciones descritas en la estructura y después válida la expresión lógica.

2.-Palabras reservadas de las tres estructuras de control de flujo

Estructura Secuencial:

- INICIO
- FIN
- LEER, ESCRIBIR

Estructura Condicional (selectiva):

- SI ... ENTONCES
- DE LO CONTRARIO
- FIN SI

Estructura Iterativa (cíclica o repetitiva):

- MIENTRAS ... HACER
- FIN MIENTRAS
- También puede usarse: REPETIR ... HASTA QUE

3.- Ejercicios de clase:

a) ¿Qué hace la siguiente secuencia?

$i \leftarrow 1$

MIENTRAS $i \leq 3$ HACER

 ESCRIBIR i

$i \leftarrow i + 1$

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.

b) 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 mient

INICIO

HACER

ESCRIBIR "Ingrese una calificación entre 5 y 10"

Leer Calificación

MIENTRAS Calificación < 5 o calificación > 10

Escribir "La calificación es" Calificación

FIN

Iteración	Entrada (Calificación)	Condición(<5 o >10)	Salida
1	9	Verdadero	La calificación es 9
2	8	Verdadero	La calificación es 8
3	4	Falso	-

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

La eficiencia se calcula con la fórmula:

$$\text{Eficiencia} = (\text{Producción real} / \text{Producción esperada}) \times 100$$

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

ESCRIBIR "Ingrese la producción real"

LEER Producción real

ESCRIBIR "Ingrese la producción esperada"

LEER Producción esperada

Eficiencia := (Producción real / Producción esperada) * 100

ESCRIBIR "La eficiencia es" Eficiencia "%"

Si

Eficiencia > 90 **Entonces**

ESCRIBIR "Eficiencia alta"

SINO

Si

Eficiencia > 70 y Eficiencia <= 90 **Entonces**

ESCRIBIR "Eficiencia media"

SINO

ESCRIBIR "Eficiencia baja"

FIN SI

FIN SI

FIN

A	B	C	D	E
Iteración	Producción Real	Producción esperada	Eficiencia	Mensaje mostrado en pantalla
1	59	100	59%	La eficiencia es 59% Eficiencia baja
2	74	100	74%	La eficiencia es 74% Eficiencia media
3	93	100	93%	La eficiencia es 93% Eficiencia alta

1. Programa: Solicita un número entero positivo y cuenta desde 1 hasta el número dado
(Usando ciclo HACER-MIENTRAS y ciclo MIENTRAS)

HACER-MIENTRAS:

INICIO

Var 1: ENTERO

Cont: ENTERO

ESCRIBIR "Ingrese un número entero positivo:"

LEER numero

contador := 1

HACER

ESCRIBIR contador

 contador := contador + 1

MIENTRAS contador <= numero

FIN

Iteraciòn	Entrada	Salida
1	4	4
2	6	6
3	1	1

MIENTRAS:

INICIO

numero, contador COMO ENTERO

ESCRIBIR "Ingrese un número entero positivo:"

LEER numero

contador := 1

MIENTRAS contador <= numero **HACER**

ESCRIBIR contador

 contador := contador + 1

FINMIENTRAS

FIN

Iteraciòn	Entrada	Salida
1	2	2
2	4	4
3	-4	-

2. Programa: Menú con validación de opciones usando HACER-MIENTRAS

INICIO

opción : ENTERO

HACER

```
ESCRIBIR "Menú de programas:";  
ESCRIBIR "1. Programa de clase 1";  
ESCRIBIR "2. Programa de clase 2";  
ESCRIBIR "3. Programa de tarea";  
ESCRIBIR "4. Salir";  
ESCRIBIR "Por favor seleccione una opción:";  
LEER opcion
```

```
SI opción= 1 ENTONCES;  
    ESCRIBIR "Ejecutando programa de clase 1";
```

SINO

```
    SI opción= 2 ENTONCES;  
        ESCRIBIR "Ejecutando programa de clase 2";
```

SINO

```
        SI opción= 3 ENTONCES;  
            ESCRIBIR "Ejecutando programa de tarea";
```

SINO;

```
            SI opción= 4 ENTONCES ;  
                ESCRIBIR "Saliendo del programa...";
```

SINO

```
    ESCRIBIR "ERROR: Opción inválida. Intente de nuevo.";
```

FIN SI

FIN SI

FIN SI

```
MIENTRAS opcion <> 4;
```

FIN

Iteración	Opción	Salida
1	2	Ejecutando programa de clase 2
2	4	Salir
3	1	Ejecutando programa de clase 1

3. Preguntas sobre el pseudocódigo de control de productos defectuosos

INICIO

nProductos, i, codProducto, cantStock, cantDefectuosos, totalDefectuosos Como Entero

porcentajeDefectos, porcentajeTotal Como Real

productoConMasDefectos, maxDefectos Como Entero

totalDefectuosos \leftarrow 0

productoConMasDefectos \leftarrow 0

maxDefectos \leftarrow -1

ESCRIBIR "Ingrese la cantidad de productos a registrar:"

LEER nProductos

i \leftarrow 1

MIENTRAS i \leq 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 \leftarrow (cantDefectuosos / cantStock) * 100

 ESCRIBIR "Porcentaje de defectos del producto ", codProducto, ": ", porcentajeDefectos, "%"

 totalDefectuosos \leftarrow totalDefectuosos + cantDefectuosos

 Si cantDefectuosos > maxDefectos Entonces

 maxDefectos \leftarrow cantDefectuosos

 productoConMasDefectos \leftarrow codProducto

 FINSI

 i \leftarrow i + 1

FINMIENTRAS

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

$\text{porcentajeTotal} \leftarrow (\text{totalDefectuosos} / \text{cantStock}) * 100$

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

Preguntas:

1-¿Cuál es el propósito principal de este algoritmo?

Tiene como propósito registrar información de productos en inventario, calcular el porcentaje de defectos por producto y en total, y detectar cuál producto tiene más defectos.

2-¿Qué datos se deben ingresar para cada producto durante la ejecución del algoritmo?

Para cada producto tenemos que ingresar:, se ingresan:Código del producto,Cantidad en stock y Cantidad de piezas defectuosas.

3-¿Qué significa el valor almacenado en la variable maxDefectos?

Son la mayor cantidad de piezas defectuosas encontrada hasta el momento en un solo producto.

4-¿Por qué se utiliza una variable de tipo real para porcentajeDefectos y porcentajeTotal?

Porque los porcentajes pueden dar como resultado valores decimales (ejemplo: 7.5%), y no siempre son enteros. Así que necesitamos la variable real para expresar ese valor con precisión.

5-¿Qué estructura de control se utiliza para procesar los productos? ¿Por qué es adecuada en este caso?

Se usa la estructura **MIENTRAS**, porque se conoce desde el principio cuántos productos se deben procesar (nProductos), y se repite el proceso mientras el contador i no exceda esa cantidad.

6-¿Qué sucede cuando se encuentra un producto con más piezas defectuosas que el producto anterior con más defectos?

Se tiene que reemplazar el valor de la variable maxDefectos y productoConMasDefectos con la información del nuevo producto que tiene más defectos, reemplazando al anterior.

7-¿Cómo se calcula el porcentaje de defectos para cada producto?

Usando la fórmula que está escrita en el pseudocódigo:

porcentajeDefectos ← (cantDefectuosos / cantStock) * 100

Segun yo ya reemplazamos la flecha por :=

8-¿Cómo modificarías el algoritmo para que, además de mostrar el producto con más defectos, también muestre el producto con menos defectos?

Yo agregaria otra variable, modificando la que ya tenemos pero ahora con mínimos defectos el producto con menos defectos: minDefectos y productoConMenosDefectos. Y en cada una verificar que se cumpla.

9-¿Qué pasaría si en algún producto la cantidad en stock fuera 0? ¿Cómo podrías evitar un error en el cálculo?

No se podría realizar porque la división entre 0 nos va a dar error, así que para evitarlo pondría una verificación para que el usuario no pueda ingresar ese valor y tenga que poner otro valor.

10-¿Qué ventajas tiene calcular el porcentaje total de defectos respecto al inventario total?

Eso nos va a permitir tener una mejor calidad del inventario, evaluar si la cantidad de defectos es riesgosa, y facilitar la toma de decisiones para mejorar los procesos de control de calidad.

Conclusión:

Como conclusión debo decir que sinceramente me costó bastante trabajo porque la verdad todavía no terminaba de entender algunas cositas como para poder hacer las otras, pero me sirvió mucho para fortalecer en lo que estaba fallando y también para poder realizar la tarea. Sin duda alguna mi favorita de las tres que nos dejó la maestra fue la tres, ya que se trataba mucho de analizar, pensar y poner en práctica lo que habíamos aprendido. Creo que esta práctica se sintió más larga que las otras, pero también me ayudó para poder repasar, aprender y tener más claro algunas cosas del tema que todavía no las tenía muy claras.

Fuentes:

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. 52-72. Recuperado el 29 de septiembre de 2025 de <http://lcp02.fi-b.unam.mx/>