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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | García Morales Karina |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 22 |
| *No. de práctica(s):* | 5 |
| *Integrante(s):* | Hernández Baltazar Erik |
| *No. de lista o brigada:* | 22 |
| *Semestre:* | 2025 – 1 |
| *Fecha de entrega:* | Martes 17 de Septiembre de 2024 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Guía práctica de estudio 05: Pseudocódigo

**Objetivo:**

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

**Actividades:**

• Elaborar un pseudocódigo que represente la solución algorítmica de un problema en el cual requiera el uso de la estructura de control de flujo condicional.

• A través de un pseudocódigo, representar la solución algorítmica de un problema en el cual requiera el uso de la estructura de control iterativa.

**Desarrollo:**

Algo importante que debemos conocer son las reglas de generación del pseudocódigo, por lo que la maestra nos pidió agregar esta información al desarrollo.

El pseudocódigo sigue un conjunto de reglas tanto semánticas como sintácticas. A continuación, explicaré algunas de las más importantes:

1. Estructura del programa: Es importante saber que todo pseudocódigo debe comenzar con la palabra clave INICIO y finalizar con FIN. Todas las instrucciones y operaciones que conforman el algoritmo deben de estar contenidas entre estas dos etiquetas, ya que se asegura una correcta delimitación del programa.
2. Uso de palabras reservadas en mayúsculas: Las palabras clave o reservadas que forman parte del lenguaje pseudocódigo deben de escribirse completamente en mayúsculas. Esto incluye los siguientes comandos INICIO, FIN, LEER, ESCRIBIR, ENTONCES, DE LO CONTRARIO, SELECCIONAR, MIENTRAS, HACER-MIENTRAS, PARA; lo que permite diferenciarlos claramente de otras partes del código.
3. Sangría o tabulación: Es fundamental utilizar sangrías o tabulaciones para organizar el código de manera jerárquica. Esto no solo mejora la legibilidad del pseudocódigo, también facilita su análisis y depuración, nos permite visualizar claramente las estructuras anidadas y las secuencias de instrucciones.
4. Lectura y escritura de datos: Para realizar operaciones de entrada y salida en el pseudocódigo, se requiere utilizar comandos específicos. La palabra clave LEER se emplea para recibir datos, mientras que ESCRIBIR se utiliza para mostrar o registrar datos. Estas palabras clave nos ayudan a identificar rápidamente las partes del código donde se manejan

entradas y salidas de información.

Como parte del desarrollo de la práctica la maestra nos pidió hacer un cuadro en donde debíamos de poner la estructura, su respectivo diagrama de flujo y su pseudocódigo. A continuación adjunto mi recuadro.

|  |  |  |
| --- | --- | --- |
| **ESTRUCTURA** | **DIAGRAMA** | **PSEUDOCÓDIGO** |
| Secuencial | Interfaz de usuario gráfica  Descripción generada automáticamente  Imagen que contiene Interfaz de usuario gráfica  Descripción generada automáticamente | ESCRIBIR/IMPRIMIR  LEER |
| Condicional SIMPLE | Diagrama  Descripción generada automáticamente | SI condición ENTONCES  [Acciones]  FIN SI |
| Condicional COMPUESTA | Diagrama  Descripción generada automáticamente | SI expresión lógica  ENTONCES  [Acciones ENTONCES]  DE LO CONTRARIO  [Acciones DE LO CONTRARIO]  FIN DEL SI |
| Condicional MÚLTIPLE | Diagrama  Descripción generada automáticamente | SELECCIONAR (variable) EN  CASO valor1 ->  [Acciones]  CASO valor2 ->  [Acciones]  CASO valor3 ->  [Acciones]  DEFECTO ->  [Acciones]  FIN SELECCIONAR |
| Iterativa MIENTRAS | Diagrama  Descripción generada automáticamente | MIENTRAS condición ENTONCES  [Acciones]  FIN MIENTRAS |
| Iterativa HACER – MIENTRAS | Diagrama  Descripción generada automáticamente | HACER  [Acciones]  MIENTRAS condición |
| Iterativa PARA |  | PARA (valor inicial, condición, incremento/decremento)  [Instrucciones]  FIN PARA |

Ahora explicaré como es que funcionan para mi todas las estructuras de control de flujo que vimos en estas últimas clases con la maestra.

1. Estructura de control secuencial: Es la más sencilla de entender para la mayoría ya que cada paso se realiza en el orden exacto que se escribe, sin desviaciones. Utilizar esta estructura es adecuado cuando no se requieren cambios según condiciones o repeticiones.
2. Estructura de control condicional SIMPLE: Esta estructura básicamente lo que hace es evaluar si una condición es verdadera o falsa y de acuerdo con el resultado, ejecuta una acción. Si la condición es verdadera, se ejecuta un bloque de código y, si es falsa se salta ese bloque y continúa con el resto del programa. Utilizar esta estructura es muy útil cuando solo quieres que ciertas instrucciones solo se ejecuten bajo ciertas condiciones específicas.
3. Estructura de control condicional COMPUESTA: Esta estructura tiene una gran diferencia a la anterior ya que evalúa una condición, pero también ofrece una alternativa, esto en caso de que la condición se falsa. Esta estructura es práctica cuando queremos hacer algo, sin importar si la condición es verdadera o falsa.
4. Estructura de control condicional MÚLTIPLE: Esta estructura nos permite tomar decisiones entre más de dos opciones, toma en cuenta varias condiciones y con ello ejecuta el código correspondiente con la primera condición que sea verdadera. En los ejercicios en clase he podido ocupar esta estructura cuando tengo varias posibilidades y necesito que solo una de ellas se ejecute.
5. Estructura de control iterativa MIENTRAS: Esta estructura se utiliza cuando necesitamos que una acción se repita viarias veces, mientras se cumpla una condición. La idea es que, mientras la condición sea verdadera, el bloque de código se siga ejecutando, cuando la condición deja de ser verdadera, se detiene el ciclo y se continúa con las siguientes instrucciones. En clase hemos utilizado esta estructura cuando nos sabemos cuántas veces necesitaremos repetir una acción pero, sabemos bajo que condición debemos seguir haciéndola.
6. Estructura de control iterativa HACER – MIENTRAS: Esta estructura se utiliza para repetir un bloque de código mientras se cumple una condición. Lo primero que hace el programa es ejecutar lo que está en la estructura HACER, después el programa verifica la condición especificada en MIENTRAS, por último, si la condición es verdadera, el programa repite el bloque de código y si la condición es falsa, el programa termina el bucle y se sigue con el código que viene después. Un punto importante es que la estructura de HACER – MIENTRAS siempre se ejecuta al menos una vez, porque la condición se evalúa después de la ejecución del bloque.
7. Estructura de control iterativa PARA: La gran diferencia al anterior es que la estructura iterativa PARA se usa para repetir un bloque de código un número específico de veces. Lo primero que se hace es establecer una variable inicial que se usa para controlar el número de repeticiones después, en la condición es la que se evalúa antes de cada iteración, si la condición es verdadera el bucle va a empezar a ejecutarse, cuando se ejecuta el bloque de código se actualiza la variable y aquí es donde incrementa o disminuye su valor. Como último paso si es verdadero el bloque se ejecuta nuevamente y si no el bucle termina y el programa continúa con el código que sigue.

**Actividades en clase:**

Durante la práctica de laboratorio la maestra nos enseñó a como hacer registros y como parte del desarrollo de la práctica cada uno hizo su registro de tema libre, en mi caso elegí hacer mi registro sobre series. Anexaré a continuación mi registro elaborado en clase.

Series REGISTRO

genero: CARACTER

capitulos: ENTERO

duración: REAL

FIN REGISTRO

serie1: REG series → variable llamada serie1 de tipo registro

INICIO

// Asignación de valores a los campos de registro

serie1.genero := ”Terror”

serie1.capitulos := 8

serie1.duracion := 54:00

// Mostrar los valores

IMPRIMIR “genero:”, serie1.genero

IMPRIMIR “capitulos:”, serie1.capitulos

IMPRIMIR “duración:”, serie1.duracion

FIN

serie2: REG series → variable llamada serie2 de tipo registro

INICIO

// Asignación de valores a los campos de registro

serie2.genero := ”Romance”

serie2.capitulos := 8

serie2.duracion := 48:00

// Mostrar los valores

IMPRIMIR “genero:”, serie2.genero

IMPRIMIR “capitulos:”, serie2.capitulos

IMPRIMIR “duración:”, serie2.duracion

FIN

serie3: REG series → variable llamada serie3 de tipo registro

INICIO

// Asignación de valores a los campos de registro

serie3.genero := ”Ficcion”

serie3.capitulos := 22

serie3.duracion := 35:00

// Mostrar los valores

IMPRIMIR “genero:”, serie3.genero

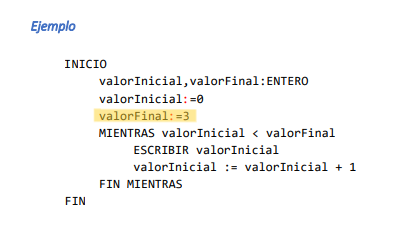
IMPRIMIR “capitulos:”, serie3.capitulos

IMPRIMIR “duración:”, serie3.duracion

FIN

Como se puede observar registré 3 series con sus respectivos pasos, para realizar mis registros me ayudé en la presentación de la maestra “PSEUDOCÓDIGO ESTRUCTURA ITERATIVA” en esa presentación la maestra nos dio un ejemplo de cómo hacer registros.

La siguiente actividad en clase fue modificar un ejemplo que venía en la práctica de laboratorio, a continuación adjuntó el ejemplo; la actividad consistía en pedir al usuario que ingrese el valor final.



En la imagen de la arriba subraye el paso del pseudocódigo que debo de modificar.

Modificación realizada por mi:

INICIO

valorInicial, valorFinal: ENTERO

valorInicial:=0

ESCRIBIR “Ingrese el valor Final:”

LEER valorFinal

MIENTRAS valorInicial < valorFinal

ESCRIBIR valorInicial

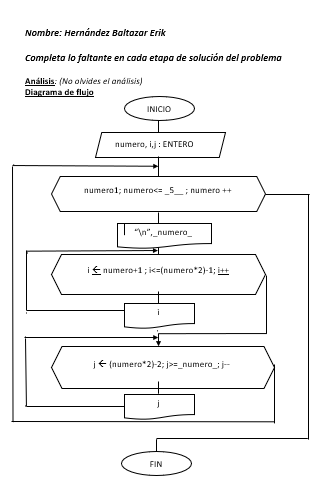
valorInicial := valorInicial + 1

FIN MIENTRAS

FIN

Como se puede ver no hubo una gran modificación, solo agregué dos líneas en donde la primera le pide al usuario ingresar el valor final y en la segunda línea el pseudocódigo se encarga de leer el valor ingresado para así poder ejecutar el programa.

La siguiente actividad fue completar un ejercicio donde nos dan el diagrama de flujo y el pseudocódigo, a continuación anexaré mi ejercicio.



Tabla

Descripción generada automáticamente

:

**Actividad de tarea:**

Para repasar los temas vistos en las clases la maestra nos dejó varias tareas sobre pseudocódigo.

1. Desarrolla un algoritmo en pseudocódigo que permita calcular el promedio de las notas de un estudiante. El algoritmo debe permitir al usuario ingresar las notas de 5 asignaturas y calcular el promedio. Si el promedio es mayor o igual a 6, se mostrará un letrero de aprobado con la calificación obtenida, en caso contrario le mostrará un letrero de Reprobado, intentarlo de nuevo.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Análisis:

PROBLEMA: Calcular el promedio de las notas de un estudiante

DATOS DE ENTRADA: Cinco notas (Nota1, Nota2, Nota3, Nota4, Nota5)

DATOS DE SALIDA: Promedio de las cinco notas del estudiante.

Pruebas de escritorio:

|  |  |  |
| --- | --- | --- |
| **Calificaciones** | **Proceso** | **Salida** |
| Nota1: 8  Nota2: 7  Nota3: 9  Nota4: 6  Nota5: 10 | (8 + 7 + 9 + 6 + 10) / 5 | Aprobado con la calificación: 8 |
| Nota1: 5  Nota2: 7  Nota3: 9  Nota4: 6  Nota5: 6 | (5 + 7 + 9 + 6 + 6) / 5 | Aprobado con la calificación: 6.6 |
| Nota1: 5  Nota2: 5  Nota3: 5  Nota4: 4  Nota5: 5 | (5 + 5 + 5 + 4 + 5) / 5 | Reprobado, intente de nuevo |
| Nota1: 10  Nota2: 10  Nota3: 10  Nota4: 10  Nota5: 9 | (10 + 10 + 10 + 10 + 9) / 5 | Aprobado con la calificación: 9.8 |

1. Genera el pseudocódigo del siguiente ejercicio que trabajaste en la práctica 4.

Solicitar al usuario los centímetros cúbicos para el pago de agua y debe ser mayor a cero, en caso contrario se le debe volver a pedir el valor, los rangos son: De 1 a 500 centímetros cúbicos pagará 5.45 por centímetro cúbico , de 501 a 750 centímetros cúbicos pagará 7.49 por centímetro cúbico y mayor a 750 centímetros cúbicos pagará 10.3 por centímetro cúbico, además si excede los 200 pesos se le aplicará un descuento del %10.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Análisis:

PROBLEMA: Calcular el costo del consumo de agua de un usuario con base en los cm cúbicos utilizados y además determinar si se le puede aplicar un descuento.

DATOS DE ENTRADA: Pago del agua mayor a 0.

DATOS DE SALIDA: Costo total a pagar por el consumo de agua.

Pruebas de escritorio:

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Proceso** | **Salida** |
| 300 | 300 \* 5.45  1635 \* .10 | 1471.5 con descuento |
| 700 | 700 \* 7.49  5243 \* .10 | 4718.7 con descuento |
| 950 | 950 \* 10.3  9785 \* .10 | 8806.5 con descuento |
| 0 | ----------------- | ----------------- |

La siguiente actividad fue:

1. Pseudocódigo que muestre el ejercicio de la hoja utilizando la estructura MIENTRAS en lugar de PARA.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Prueba de escritorio**:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| numero | Iteración de i | Iteración de j | Salida de i | Salida de j | Salida final |
| 2 | i= 3 | j=2 | 3 | 2 | 232 |
| 3 | i= 4 hasta i= 5 | j= 4 | 4, 5 | 4 | 3454 |
| 4 | i= 5 hasta i= 7 | j= 6 hasta j= 4 | 5, 6, 7 | 6, 5, 4 | 4567654 |

**Conclusión:**

Esta práctica estuvo interesante pero también me resultó un poco laboriosa, lo que más me gustó fue que empezamos a trabajar con pseudocódigo y gracias a la manera de explicar de la maestra no fue tan difícil, en el laboratorio sentí que el tiempo pasó muy rápido porque estábamos concentrados en hacer nuestros registros. Lo que más se me dificultó fue justamente en aprender a estructurar mi pseudocódigo, pero mientras siga practicando todo me saldrá mejor.

**Fuentes de consulta:**

* García, Edgar & Solano, Jorge. (2022, febrero 21). Guía práctica de estudio 05: Pseudocódigo. Manual de prácticas del Laboratorio de Fundamentos de programación. Facultad de Ingeniería. pp. 76 – 87. Recuperado el 17 de septiembre de 2024 de <http://lcp02.fi-b.unam.mx/>
* Universidad Nacional Autónoma de México. (s. f.). Curso: Karina García Morales - Fundamentos de programación. TuAulaVirtual. Recuperado 14 de septiembre de 2024, de <https://tuaulavirtual.educatic.unam.mx/course/view.php?id=4821>