



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Karina García Morales

Asignatura: Fundamentos de programación

Grupo: 20

No. de práctica(s): Practica 5

Integrante(s): Francisco Javier Gómez Mendoza

No. de lista o brigada: 22

Semestre: 2023-1

Fecha de entrega: 25 de octubre del 2022

Observaciones:

CALIFICACIÓN: _____

Pseudocódigo

Objetivo:

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

Introducción

Ya una vez razonado el problema y se obtienen los datos requeridos que son los de entrada y salida, y se a diseñado el algoritmo que resuelva el problema de manera eficiente, se debe pasar a la siguiente. Estaba que es codificación del algoritmo.

Para que se pueda codificar nuestro algoritmo se debe generar una répresentacion de este una representación algorítmica elemental es el pseudocodigo.

El pseudocodigo es la representación escrita del algoritmo, en palabras mas entendibles son los pasos a seguir de manera de texto, posee la sintaxis para realizar la representación del algoritmo.

Sintaxis de pseudocodigo

El lenguaje pseudocódigo tiene diversas reglas semánticas y sintácticas

1. **Alcance del programa:** cualquier pseudocodigo va contar con dos etiquetas, INICIO y FIN, dentro de ellas van las instrucciones.
2. **Palabras reservadas con mayúsculas:** Las palabras propias del pseudocódigo deben de ser escritas en mayúsculas.
3. **Sangría o tabulación:** 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 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.
6. **Operadores aritméticos:** Se tiene la posibilidad de utilizar operadores aritméticos y lógicos.
7. **Notación de camello.** Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

Estructuras de control de flujo

Permiten la ejecución condicional y la representación de un conjunto de instrucciones, existen 3 tipos, 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.

Ejemplo

```
INICIO
    x : REAL
    x := 5.8
    x := x * 2
FIN
```

Estructuras de control condicionales.

Permiten evaluar una expresión lógica, esta puede ser falsa o verdadera, dependiendo del resultado, se realiza uno o otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes o se realiza una acción o otra.

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

```
SI condición ENTONCES
    [Acciones]
FIN SI
```

Estructuras de control iterativas o repetitivas.

Las estructuras de control de flujo iterativas o repetitivas, permiten realizar una serie de instrucciones mientras se cumpla la expresión lógica. Existen 2 tipos MIENTRAS y HACER- MIENTRAS.

MIENTRAS. Primero valida la condición y si ésta es verdadera y así pueda realizar las instrucciones de la estructura y regresa a validar la condición, hasta que es falsa rompe el ciclo. El fin de la estructura lo determina la etiqueta FIN MIENTRAS.

Ejemplo

```
INICIO
    valorInicial,valorFinal:ENTERO
    valorInicial:=0
    valorFinal:=3
    MIENTRAS valorInicial < valorFinal
        ESCRIBIR valorInicial
        valorInicial := valorInicial + 1
    FIN MIENTRAS
FIN
```

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.

Ejemplo

```
INICIO
    valorInicial,valorFinal:ENTERO
    valorInicial:=0
    valorFinal:=3
    HACER
        ESCRIBIR valorInicial
        valorInicial := valorInicial + 1
    MIENTRAS valorInicial < valorFinal
FIN
```

TRABJO EN CLASE

1. En clase resolvimos el problema de recaudar datos con registro, en mi caso pedía información escolar sobre alumnos, estos datos que junte son datos de mis compañeros de clase.

```

alumno: REG
numerodecuenta: ENTERO
nombre: CADENA
facultad: CADENA
FIN REG

alumno01: REGalumno {
    alumno01.numerodecuenta: = "320302149"
    alumno01.nombre: = "Angel Santiago"
    alumno01.facultad: = "Ingeniería"
}

alumno02: REGalumno {
    alumno02.numerodecuenta: = "320330308"
    alumno02.nombre: = "Diego Martinez"
    alumno02.facultad: = "Ingeniería"
}

alumno03: REGalumno {
    alumno03.numerodecuenta: = "320131329"
    alumno03.nombre: = "Ivan Avarez"
    alumno03.facultad: = "Ingeniería"
}
    
```

2. Segundo trabajo en clase.

Problema: Solicitar al usuario un valor de 1 al 10 entero, si el usuario ingresa un valor de 1 a 5 imprimir "aprobado", si ingresa un valor de 6 al 10 imprimir "aprobado".

```

INICIO
    NUMERO: ENTERO
    VARIABLES: ENTERO
                CARACTER

    ESCRIBIR "Selecciona una opción:
    1. Imprimir números de 1 a 5
    2. Preguntar al usuario si es o no aprobado
    3. Imprimir suma de números de 1 a 10"

    LEER VARIABLE
    SELECCIONAR (VARIABLE)
    CASO 1 NUMERO = 1
        MIENTRAS (NUM <= 5) ENTONCES
            ESCRIBIR num
            num++
        FIN MIENTRAS
    CAS 2: VAL B: ENTERO
        ESCRIBIR "¿quieres al cine o a comer?:"
        1. "al cine"
        2. "a comer"

        LEER B
        VAL B = 1 ENTONCES
            ESCRIBIR "Tu vas al cine"
        O DE LO CONTRARIO
            ESCRIBIR "Tu vas a comer"
        FIN SI
    CAS 3 NUMERO = 0, VAL C = 0
        MIENTRAS (NUM <= 10) ENTONCES
            num++
            VAL C = VAL C + NUM
        FIN MIENTRAS
        ESCRIBIR "El total de la suma es " VAL C
    FIN SELECCIONAR
FIN
    
```

TAREA

1.- Algoritmo que valide la calificación que ingrese el usuario, esta debe estar entre 5 y 10. Indicar si ha aprobado con calificación mayor a 6; un letrero "Aprobado, felicidades", en caso de no aprobar "Reprobado, tienes una nueva oportunidad"

```
1. INICIO
    Calificación: ENTERO
    ESCRIBIR "ingrese su calificación"
    LEER Calificación
    SI Calificación > 4 && Calificación < 11 Entonces
        - SI Calificación > 6
            ESCRIBIR "Aprobado, felicidades"
        - SI Calificación <= 6
            ESCRIBIR "Reprobado, tienes una nueva oportunidad"
    FIN SI
FIN
```

2.- Del ejercicio de clase (el menú) modifica con el ciclo hacer mientras, en donde valide, si el usuario no elige las opciones 1 ó 2 para caracteres deben ser 'a', 'b' o 'c') imprimir un letrero de "error"(es el default) y volver a mostrarle el menú..

```
2. INICIO
    Variable: ENTERO
    ESCRIBIR "opcion:
    1. comer
    2. cine"
    LEER Variable
    HACER
        - SI Variable
            ESCRIBIR "1 al comer"
        - SI Variable = 2
            ESCRIBIR "1 al cine"
    FIN SI
    MIENTRAS Variable = 11 Variable = 2
    ESCRIBIR "ERROR"
FIN
```


3.- Algoritmo que muestre la numeración del 1 al 1000, emplea la estructura MIENTRAS, HACER MIENTRAS Y PARA

```
3.
INICIO
    numero: ENTERO
    numero := 0
    total: ENTERO
    total := 0
    MIENTRAS numero < 1001 ENTONCES
        HACER
            numero ++
            total := total + numero
        MIENTRAS numero <= 1000
    FIN MIENTRAS
    ESCRIBIR "El resultado es: " total.
FIN
```

4.- Solicitar al usuario que ingrese la variable, si ingresa una variable diferente a la letra 'a' o 'A', volver a solicitarla, en caso de contrario, imprimir la letra 'a' o 'A' que ingresó el usuario.

```
4.
INICIO
    var: ENTERO
    ESCRIBIR "ingrese 'a' o 'A' si es caso"
    LEER var
    SI var == a || var == A
        ENTONCES
            - SI var == a
                ESCRIBIR "a"
            - SI var == A
                ESCRIBIR "A"
        DE LO CONTRARIO
            ESCRIBIR "Vuelva a ingresar su variable"
    FIN SI
FIN
```

Conclusiones

1. Esta ovación la preactiva se torno más difícil, tal vez por que la clase fue en línea y quedaron lagunas de cosas, pero tuve que pedir ayuda de mis compañeros para resolver dudas y un poco de ayuda.
2. El pseudocódigo es un poco complicado para mi, pues la notación de como acomodar los datos se me hace aún difícil
3. El apoyo de los digamma de flujo es in fluyente, me gustaría usarlos en los siguientes ejercicios pero esta vez me queda como lección esto.
4. También empezamos a ver como funciona el peso del código en cómo hacer la notación de sintaxis Y así para la siguiente etapa que es la programación sea mucho más fácil y no tener problemas en el proceso.
5. En lo personal es la practica que menos me a gustado pero no estuvo tan mal, creo que de todo se aprende y fue el caso. Pero me gusta que todo se mas didáctico.

Bibliografía

Laboratorio Salas A y B. (2022, 21 febrero). Manual de prácticas del Laboratorio de Fundamentos de programación. Recuperado 28 de agosto de 2022, de <http://lcp02.fi-b.unam.mx/>

Liga de github

[https://github.com/JavierMend/practica1_fdp/blob/e057ebb63f415fa39e11545dfa3ee77425a03e12/Practica 1.pdf](https://github.com/JavierMend/practica1_fdp/blob/e057ebb63f415fa39e11545dfa3ee77425a03e12/Practica%201.pdf)