



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de programación

Grupo: 20

No. de práctica(s): 05

Integrante(s): Juan Manuel Cuellar Orbezo

No. de lista o brigada: 07

Semestre: 2024 - 2

Fecha de entrega: 12 Marzo 2024

Observaciones:

CALIFICACIÓN: _____

Práctica 05 - Pseudocódigo

Objetivo:

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

Desarrollo

El pseudocódigo y sus estructuras básicas

El pseudocódigo es como una receta para programar. En lugar de usar palabras complicadas de un lenguaje de programación, se utilizan palabras comunes y frases sencillas para describir los pasos que debe seguir un programa.

El pseudocódigo, al ser una herramienta intuitiva y comprensible, facilita el aprendizaje de la programación y optimiza el desarrollo de software. Permite a cualquier persona entender el programa, facilita la planificación y organización de los pasos, y ayuda a detectar y corregir errores con mayor facilidad.

A continuación una tabla con tipos de estructuras usadas comúnmente para pseudocódigo y las palabras reservadas más comúnmente asociadas a las mismas; también se incluye un ejemplo del uso de cada una.

Estructura	Palabras reservadas	Uso
Secuencial Las instrucciones se ejecutan una tras otra en orden descendente.	LEER	LEER Num
	ESCRIBIR	ESCRIBIR "Hello world"
	ESPERAR	ESPERAR (10)
Condicional Las instrucciones se ejecutan de forma selectiva según una condición.	SI	SI (Fecha > 5) ESCRIBIR "Todavía no puedes acceder" FIN SI
	SINO	SI (Mes = 4) ESCRIBIR "Abril" SINO ESPERAR (100) FIN SI
	SELECCIONAR	SELECCIONAR (Dia) CASO 1 -> ESCRIBIR "Lun" CASO 2 -> ESCRIBIR "Mar" ... CASO 7 -> ESCRIBIR "Dom"

		FIN SELECCIONAR
Iterativa Las instrucciones se repiten continuamente hasta que se cumple una condición.	HACER MIENTRAS	HACER ESCRIBIR "Pierde" MIENTRAS (Moneda = 0)
	PARA	PARA (t=0, t>60, t++) ESCRIBIR t "segundos" FIN PARA
	MIENTRAS	MIENTRAS (Age < 18) ESCRIBIR "Menor" FIN MIENTRAS

Variables register

Las variables "register" son como los bolsillos de un pantalón; son muy rápidas para acceder, pero no se pueden guardar muchas cosas en ellas. Se usan variables "register" para que la computadora guarde en un compartimento especial del procesador, lo que puede hacer que un programa sea más rápido.

Sin embargo, no siempre es posible usar variables "register". A veces, la computadora no tiene espacio o no puede guardar el tipo de información que se necesita.

A continuación se adjunta un ejemplo usando una variable register

```

INCIO
  Calif:REG
  Nom:CADENA
  Proj:REAL
  Exam:REAL
  FIN REG

  Aguilar.REG Calif
  Aguilar.Nom:="Armando"
  Aguilar.Proj:=8.6
  Aguilar.Exam:=9.1

  ...

  Velazquez.REG Calif
  Velazquez.Nom:="Valeria"
  Velazquez.Proj:=9.7
  Velazquez.Exam:=7.4
FIN
  
```

Ejercicio A - Calculadora de sueldo

PROBLEMA: Solicita al usuario su salario diario y muestra en pantalla su sueldo mensual.

RESTRICCIONES: Mes = 4 semanas. & Números enteros

DATOS ENTRADA: Sal (Salario diario)

DATOS SALIDA: Sue (Sueldo Mensual)

```
INICIO
    Sue:REAL
    Sal:REAL
    ESCRIBIR "Hola! Esta aplicación calculará tu salario mensual."
    ESPERAR (10)
    ESCRIBIR "Introduce tu sueldo diario"
    LEER Sue
    Sal:= Sue * (5*4)
    ESCRIBIR "Tu salario mensual es de $" Sal
FIN
```

Ejercicio B - Comparador de cifras

PROBLEMA: Solicita dos números enteros al usuario, calcular cuál es el mayor de los dos.

RESTRICCIONES: Números enteros

DATOS DE ENTRADA: a (Número 1), b (Número 2)

DATOS DE SALIDA: String con respuesta.

```
INICIO
    a,b:ENTERO

    ESCRIBIR "Elige un valor para el primer número"
    LEER a
    ESCRIBIR "Elige un valor para el segundo número"
    LEER b

    SI a > b
        ENTONCES
            ESCRIBIR "a es mayor"

        DE LO CONTRARIO
            ESCRIBIR "b es mayor"

    FIN DEL SI
FIN
```

Ejercicio C - Adivina mi número

PROBLEMA: Solicita un número entero del uno al 10 al usuario, si el número no cumple el criterio debe de cambiarlo. Después pedirle que adivine números hasta que le atine a la cifra elegida por el programa.

RESTRICCIONES: Números enteros

DATOS DE ENTRADA: n (Número adivinado)

DATOS DE SALIDA: String con respuesta.

```
INICIO
    n: ENTERO
    m: ENTERO

    ESCRIBIR "Elige un número entre 1 y 10"
    LEER n

    m:=4

    SI n > 10

        ESCRIBIR "Entre 1 y 10 por favor"

    DE LO CONTRARIO

        MIENTRAS n != m

            ESCRIBIR "Elige un número entre 1 y 10"
            LEER n

        DE LO CONTRARIO

            ESCRIBIR "¡Le atinaste!"

        FIN MIENTRAS

    FIN SI
FIN
```

Ejercicio D - Menú general

PROBLEMA: Crear un menú con los programas anteriores. Después de ejecutar los ejemplos anteriores preguntarle al usuario si desea hacer otro cálculo hasta que diga que no.

RESTRICCIONES: Números enteros (Sólo en el menú)

DATOS DE ENTRADA: yn(char), menu(char) (Sólo en el menú)

DATOS DE SALIDA: String con respuesta.

```

INICIO
    yn,menu:CHAR
    yn:Y

    MIENTRAS yn = Y

        ESCRIBIR "Hola! Bienvenido a nuestro menú, elige una opción:"
        ESCRIBIR "A - Calculadora de sueldo / B - Comparador de cifras /
        C - Adivina mi número"

        LEER menu

        SI menu = A | B | C

            ENTONCES
                SELECCIONAR menu

                    CASO A
                        Algoritmo A - Calculadora de sueldo
                        ...
                        % Vea ejemplo A %

                    CASO B
                        Algoritmo B - Comparador de cifras
                        ...
                        % Vea ejemplo B %

                    CASO C
                        Algoritmo C - Adivina mi número
                        ...
                        % Vea ejemplo C %

                FIN SELECCIONAR

            DE LO CONTRARIO
                ESCRIBIR "Lo siento, tu opción no está en este menú"
                ESPERAR (10)

            FIN SI

        ESCRIBIR "Gracias por usar este algoritmo. ¿Deseas hacer otra
        operación? Y/N?"
        LEER yn

    DE LO CONTRARIO

        ESCRIBIR "Hasta luego!"

```

```
FIN MIENTRAS
```

```
FIN
```

Tarea

Algoritmo 1

Programa que valide la calificación que ingrese el usuario; esta debe estar entre 5 y 10. Indicar si ha aprobado o reprobado.

```
INICIO
    calif:ENTERO

    ESCRIBIR "Introduce tu calificación entre 1 y 10"
    LEER calif

    SI calif < 0 | calif > 10

        ESCRIBIR "Entre 1 y 10 por favor"

    DE LO CONTRARIO
        SI calif >= 6

            ESCRIBIR "Felicidades! Has aprobado la materia"

        DE LO CONTRARIO

            ESCRIBIR "Has reprobado la materia, lo siento."

        FIN SI
    FIN SI
FIN
```

Algoritmo 2

Pseudocódigo que muestre la numeración del 1 al 1000

```
INICIO

    cont:ENTERO
    cont:= 1

    ESCRIBIR "Contemos del 1 al 1000 usando WHILE"
```

```
HACER MIENTRAS cont <= 1000

    ESCRIBIR cont
    cont:= cont + 1

FIN MIENTRAS

ESCRIBIR "Ta-da! Los números del 1 al 1000 usando WHILE"
ESPERAR 15
cont:= 1

ESCRIBIR "Contemos del 1 al 1000 usando DO WHILE"

HACER

    ESCRIBIR cont
    cont:= cont + 1

MIENTRAS cont <= 1000

    ESCRIBIR "Ta-da! Los números del 1 al 1000 usando DO WHILE"
    ESPERAR 15
    cont:= 1

    ESCRIBIR "Contemos del 1 al 1000 usando FOR"

    PARA (cont:=1, cont:=1000, cont++)

        ESCRIBIR cont

    FIN PARA

    ESCRIBIR "Ta-da! Los números del 1 al 1000 usando FOR"
    ESPERAR 15
    cont:= 1

FIN
```

Algoritmo 3

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


```

INICIO
    var:CHAR

    ESCRIBIR "Escribe la letra A o me enojo"
    LEER var

    MIENTRAS var != 'a' | var != 'A'

        ESCRIBIR "Te dije que la letra A! >:("

    DE LO CONTRARIO

        ESCRIBIR "Muchas gracias :D que amable"

    FIN MIENTRAS
FIN

```

Algoritmo 4

Un menú con los 3 programas indicados

```

INICIO
    yn,menu:CHAR
    yn:Y

    MIENTRAS yn = Y

        ESCRIBIR "Hola! Bienvenido a nuestro menú, elige una opción:"
        ESCRIBIR "A - Pase la materia? / B - Contador a mil / C - Programa enojón"

        LEER menu

        SI menu = A | B | C

            ENTONCES
                SELECCIONAR menu

                    CASO A
                        Algoritmo 1 - Pase la materia?
                        ...
                        % Vea ejercicio 1 %

                    CASO B
                        Algoritmo B - Contador a mil
                        ...

```

```

                                % Vea ejercicio 2 %
                                CASO C
                                Algoritmo C - Programa enojón
                                ...
                                % Vea ejercicio 3 %

                                FIN SELECCIONAR
                                DE LO CONTRARIO
                                    ESCRIBIR "Lo siento, tu opción no está en este menú"
                                    ESPERAR (10)
                                FIN SI

                                ESCRIBIR "Gracias por usar este algoritmo. ¿Deseas hacer otra
                                operación? Y/N?"
                                LEER yn

                                DE LO CONTRARIO

                                    ESCRIBIR "Hasta luego!"

                                FIN MIENTRAS

                                FIN

```

Pruebas de escritorio

Algoritmo 1

Iteración	calif	Salida
1	8	"Felicidades! Has aprobado la materia"
2	2	"Has reprobado la materia, lo siento."
3	12	"Entre 1 y 10 por favor"

Algoritmo 2

N/A - El usuario no introduce ningún dato

Algoritmo 3

Iteración	var	Salida
1	a	"Muchas gracias :D que amable"
2	A	"Muchas gracias :D que amable"

3	J	"Te dije que la letra A! >:{"
---	---	-------------------------------

GitHub

A continuación, la liga al repositorio de GitHub - https://github.com/CuellarJM/practica5_fdp/

Conclusión

El pseudocódigo, como lenguaje universal de programación nos permite comprender mejor los algoritmos, descomponer problemas complejos y fortalecer el pensamiento computacional.

En definitiva, el pseudocódigo es una base sólida para el desarrollo de software y la resolución de problemas en general, abriendo las puertas a un sinfín de posibilidades en el mundo digital. Dominarlo nos convierte en mejores pensadores, comunicadores y solucionadores de problemas.

Bibliografía

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Edu

GitHub, Inc. (2024). Hello world. GitHub Docs.
<https://docs.github.com/en/get-started/start-your-journey/hello-world>

Solano, J. A., García, E. E., & Montañó, L. S. (2022, February 21). Manual de prácticas del laboratorio de Fundamentos de programación. CDMX; Facultad de Ingeniería.