



La computación como herramienta de trabajo del profesional de Ingeniería

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M. I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No. de Práctica(s): 5

Integrante(s): Sánchez García Rocío

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista: 47

Semestre: 2021-1

Fecha de entrega: Sábado 7 de noviembre de 2020

Observaciones:

CALIFICACIÓN: _____

Práctica 5

Pseudocódigo

Objetivo

Elaborar 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.

Introducción

Una vez que un problema dado ha sido analizado (se obtiene el conjunto de datos de entrada y el conjunto de datos de salida esperado) y se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), se debe proceder a la etapa de codificación del algoritmo.

Para que la solución de un problema (algoritmo) pueda ser codificada, se debe generar una representación del mismo. 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).

Marco teórico

Propiedades del factorial de un número:

- Sólo está definido para todos los números enteros positivos.
- Es el producto de todos los enteros desde 1 hasta él.
Por ejemplo:

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

- Nos indica cuántas combinaciones diferentes podría tener un grupo de números.

Desarrollo

1. Realizar un pseudocódigo que determine el color del semáforo COVID en base a una muestra de 100 individuos de acuerdo con los siguientes criterios:

- Si hay más de 80 individuos con COVID el color del semáforo es rojo.
- Si hay de 51 a 80 individuos con COVID el color del semáforo es naranja.
- Si hay de 1 a 50 individuos con COVID el color del semáforo es amarillo.
- Si no hay individuos con COVID el color del semáforo es verde.

Para la elaboración del pseudocódigo es necesario escribir un algoritmo con todos los pasos para obtener el color del semáforo COVID.

Algoritmo:

- Definir la variable `muestras` como número entero.
- Leer la variable `muestras`.
- Mostrar un mensaje donde se indique que se debe ingresar la cantidad de personas con COVID.
- Hacer una estructura condicional anidada, donde la primera condición establece un rango de valores que van desde 80 hasta 100, si esta se cumple escribir un mensaje indicando que el color del semáforo es rojo.
- De no cumplirse agregar una nueva condición en donde rango sea de 51 a 80, si se cumple escribir un mensaje indicando que el color del semáforo es naranja.
- Si no se cumple agregar una nueva condición donde el rango establecido sea de 1 a 50, si esta se cumple escribir un mensaje indicando que el color del semáforo es amarillo.
- Por último, de no cumplirse la condición agregar una nueva donde si la variable `muestras = 0`, escribir un mensaje indicando que el color del semáforo es verde.
- Si ninguna de las condiciones anteriores se cumple, entonces escribir un mensaje indicando que la cantidad o valor proporcionado no es válido.

Pseudocódigo:

Definir muestras Como Entero

Escribir " En base a las muestras de 100 individuos, ingresa la cantidad de individuos con COVID "; Sin Saltar; leer muestras

Si muestras > 100 Entonces

Escribir " Valor no valido"

Repetir

Escribir " En base a las muestras de 100 individuos, ingresa la cantidad de individuos con COVID "; Sin Saltar; leer muestras

Hasta Que muestras < 100

SiNo

Si (muestras > 80) y(muestras = 100) Entonces

Escribir " El color del semáforo es rojo"

FinSi

Si (muestras <= 80) y (muestras >= 51) Entonces

Escribir " El color del semáforo es naranja "

FinSi

Si (muestras <= 50) y (muestras >= 1) Entonces

Escribir " El color del semáforo es amarillo "

FinSi

Si muestras = 0 Entonces

Escribir " El color del semáforo es verde"

FinSi

FinSi

2. Realizar un pseudocódigo que calcule el factorial de un número dado.

Ejemplo:

- $1! = 1$
- $2! = 2$
- $3! = 6$
- $4! = 24$

Algoritmo:

- Definir como enteros a las variables que se utilizaran, en este caso `num`, `factorial`, `i`
- Inicializar la variable `factorial` y la variable `i` con el valor de 1.
- Escribir un mensaje donde se indique que debe ingresarse un número.
- Leer la variable.
- Emplear una estructura repetitiva donde se ingresará una secuencia lógica, en este caso $i \leq \text{num}$.
- Mientras la condición se cumpla se realizará un conjunto de acciones, de lo contrario la iteración termina.
- A la variable `factorial` se le asigna la multiplicación $1 \cdot 2 \cdot 3 \cdot 4 \dots$, según sea el caso, hasta que `i` sea menor o igual a la variable `num`.
- La variable `i` se irá incrementando de 1 en 1.
- Dentro de esta estructura a la variable `factorial` se le asignara la variable `factorial` multiplicada por `i`.
- A la variable `i` se le asigna $i+1$.
- Fuera de la estructura se escribirá el mensaje donde se mostrará el resultado de la variable `factorial`.

Pseudocódigo:

```
Definir num, i, factorial Como Entero
    i<- 1;
    factorial<-1;
    Escribir " Ingresa un número "; sin saltar; Leer num
    Mientras i<= num Hacer
        factorial <- factorial*i;
        i<-i+1;
    FinMientras

    Escribir "El factorial es: ", factorial;
```

Conclusiones

- Los pseudocódigos representan los pasos a seguir para cumplir con una tarea y facilita la programación.
- Se trata de un código de mayor comprensión para las personas y el método que proporciona es similar al de programación.
- Existen diversas estructuras que facilitan la resolución de problemas como las condicionales, iterativas, control de flujo, etc.
- El lenguaje debe cumplir con reglas, de lo contrario no será posible la ejecución del mismo.

Bibliografía

Manual de prácticas del Laboratorio de Fundamentos de Programación, Facultad de ingeniería UNAM, recuperada el 4 de noviembre, en <http://lcp02.fi-b.unam.mx/>