|  |
| --- |
| Laboratorio de Computación  Salas A y B |
| |  |  | | --- | --- | | Profesor: | Claudia Rodriguez Espino | | Asignatura: | Fundamentos de Programación | | Grupo: | 1104 | | No de Práctica(s): | 1 | | Integrante(s): | Valencia Mancera Erick Samuel | |  |  | |  |  | |  |  | | No. de Equipo de cómputo empleado: | #52 | | Semestre: | 1ª | | Fecha de entrega: | 18 de agosto del 2018 | | Observaciones: |  | |  |  | |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Practica #5: Pseudocódigo**

**Objetivo:** Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

**Desarrollo**

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).

El lenguaje pseudocódigo tiene diversas reglas semánticas y sintácticas. A continuación, se describen las más importantes: 1. 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 programa.

2. Palabras reservadas con mayúsculas: Todas las palabras propias del pseudocódigo deben de ser escritas en mayúsculas.

3. Sangría o tabulación: El pseudocódigo 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 se utiliza la etiqueta ESCRIBIR. La lectura de datos se realiza, por defecto, desde el teclado, que es la entrada estándar del sistema. La escritura de datos se realiza, por defecto, en la pantalla, que es la salida estándar del sistema.

Ejemplo

**ESCRIBIR "Ingresar la altura del polígono" LEER altura**

5. Declaración de variables: la declaración de variables la definen un identificador (nombre), seguido de dos puntos, seguido del tipo de dato, es decir:

**<nombreVariable>:<tipoDeDato>**

**Funciones**

Cuando la solución de un problema es muy compleja se suele ocupar el diseño descendente (divide y vencerás). Este diseño implica la división de un problema en varios subprocesos más sencillos que juntos forman la solución completa. A estos subprocesos se les llaman métodos o funciones. Una función está constituida por un identificador de función (nombre), de cero a n parámetros de entrada y un valor de retorno:

INICIO

FUNC identificador (var:TipoDato,..., var:TipoDato) RET: TipoDato

[Acciones]

FIN FUNC

FIN

El identificador es el nombre con el que llama a la función. Las funciones pueden o no recibir algún(os) parámetro(s) (tipo(s) de dato(s)) como entrada; si la función recibe alguno se debe incluir entre los paréntesis. Todas las funciones pueden regresar un valor al final de su ejecución (el resultado). Todas las estructuras de control de flujo (secuencial, condicional y repetitivas o iterativas) deben ir dentro de alguna función.

**Ejemplos**

1. **Área del círculo**

INICIO

rad, area: REAL

ESCRIBIR “Ingrese el radio del circulo”

LEER rad

area: (3.1416)\*(rad\*rad)

ESCRIBIR “El area del circulo es: “ area

FIN

// >>> El area del circulo es **area**Fin

1. **Edad**

INICIO

edad: ENTERO

ESCRIBIR “Ingrese su edad”

LEER edad

SI edad >= 18

ESCRIBIR “Eres mayor de edad”

FIN SI

FIN

**edad >= 18** // >>> Eres mayor de edad

**edad < 18** //>>>

1. **Valores de X**

INICIO

x,y : REAL

ESCRIBIR “Ingrese el valor de x ”

LEER x

SI x < 2

y:= (2\*x\*x)+(x)+(8)

FIN SI

DE LO CONTRARIO

y:= (x\*x)+(3\*x)-(2)

FIN DE LO CONTRARIO

ESCRIBIR “El valor de y es: “ y

FIN

// >>> El valor de y es: **y**

1. **Menú de altas bajas y cambios**

INICIO

sel : ENTERO

ESCRIBIR “1) Altas 2) Bajas 3)Cambios”

LEER sel

SELECCIONAR (sel) EN

CASO 1 ->

ESCRIBIR “ Estas en Altas”

CASO 2 ->

ESCRIBIR “Estas en Bajas”

CASO 3->

ESCRIBIR “Estas en Cambios”

DEFECTO ->

ESCRIBIR “Opción inexistente”

FIN DE SELECCIONAR

FIN

1. **Mostrar los números del 1 al 100**

INICIO

ini : ENTERO

ini := 1

HACER

ESCRIBIR ini” “

ini := ini +1

MIENTRAS ini <= 10

FIN

// >>> 1 2 3 4 5 6 7 8 9 10

**Conclusión**

En conclusión, un pseudocódigo es una herramienta que nos permite describir textual mente como se va a comportar un programa, utilizando una mezcla de lenguaje de programación con palabras del lenguaje común. Como dije anterior mente con el uso de pseudocódigos podemos explicar el trabajo de una aplicación a desarrollar. Para un programador es indispensable el uso de pseudocódigos puesto que, al diseñar una aplicación se necesita tener bien en claro las funciones, variables y constantes que va a usar la misma para así proceder con la digitación y compilación de esta. Actualmente existen una serie de sistemas de interpretación de pseudocódigos en la red, aplicaciones que son diseñadas por motivos educacionales para las diferentes plataformas ya sea Linux Mac o Windows. PSeInt es un intérprete de pseudocódigos que ahora usan muchas universidades acá en Venezuela. Después de haber analizado todos los conceptos y definiciones acerca de los pseudocódigos, en conclusión podríamos decir que es muy importante para nosotros estudiantes universitarios aprender muchísimo más afondo el uso de los pseudocódigos, aprender a emplearlos y practicar con aplicaciones intérpretes, puesto que la algorítmica y la programación son materias de análisis y lógica programática, es decir que tenemos que esforzarnos en la práctica y en el análisis de los mismos para poder avanzar mucho más en nuestra carrera.