|  |  |  |
| --- | --- | --- |
| Imagen que contiene dibujo, taza  Descripción generada automáticamente | **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):* | Suzán Herrera Álvaro |
| *No. de lista o brigada:* | 50 |
| *Semestre:* | 1 |
| *Fecha de entrega:* | 25 octubre 2022 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

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

**Sintaxis de pseudocódigo**

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 algoritmo.

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.

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>

Los tipos de datos que se pueden utilizar son:

ENTERO -> valor entero positivo y/o negativo

REAL -> valor con punto flotante y signo

BOOLEANO -> valor de dos estados: verdadero o falso

CARACTER -> valor tipo carácter

CADENA -> cadena de caracteres

6. Operadores aritméticos: Se tiene la posibilidad de utilizar operadores aritméticos y

lógicos:

Operadores aritméticos: suma (+), resta (-), multiplicación (\*), división (/), módulo

(mod), exponenciación (^), asignación (:=).

Operadores lógicos: igualdad (=), Y-lógica o AND (&), O-lógica u OR (|), negación

o NOT (!), relaciones de orden (<, >, <=, >=) y diferente (<>).

7. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello. En la notación de camello (llamada así porque parecen las jorobas de un camello) los nombres de cada palabra empiezan con mayúscula y el resto se escribe con minúsculas. Existen dos tipos de notaciones de camello: lower camel case que en la cual la primera letra de la variable inicia con minúscula y upper camel case en la cual

todas las palabras inician con mayúscula. No se usan puntos ni guiones para separar las palabras (a excepción de las constantes que utilizan guiones bajos). Además, para saber el tipo de variable se recomienda utilizar un prefijo.

**Conceptos:**

Tipo de dato “REG” (registro) almacena una serie de preguntas que pueden recuperarse luego según el nombre del registro, se escribe como:

<nom#Reg>:<REG> + <nom.serie>:<tipo.var>

Sintaxis tipo pseudocódigo: la notación del pseudocódigo es en mayúsculas y se marca el programa por las palabras clave INICIO y FIN, estructuras:

• Sangría o tabulación  
• Estructura LEER y ESCRIBIR  
• Estructura de declaración <nom.var>:<tipo.var>  
• Nueva notación de registros previos (ciclo & condición)

**Desarrollo**

Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

**Conclusión:**

El pseudocódigo es muy importante para visualizar el proceso de una forma más técnica, a diferencia del diagrama de flujo, el pseudocódigo es más específico y requiere una estructura más puntual pues maneja niveles o tabulaciones, es más complejo que un diagrama de flujo.

El diagrama de flujo es muy visual mientras el pseudocódigo es más analítico, pero es importante aprender a usar ambos métodos.