



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Saavedraa Hernández Honorato

*Asignatura:* Fundamentos de programación

*Grupo:* 01

*No de Práctica(s):* 5 Pseudocódigo

*Integrante(s):* Luis Salinas Ludwig

*Semestre:* 2018-1

*Fecha de entrega:* 18/09/2017

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo

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

## Desarrollo

Se comenzó por ver la sintaxis del pseudocódigo donde nos dice que el lenguaje del pseudocódigo tiene una gran variedad de reglas semánticas y sintácticas, las cuales son muy importantes para que nuestro programa funcione y se entienda cual será este funcionamiento. Para ello debemos tener en cuenta cual será el cuerpo de nuestro programa, así como las acciones que se deberán introducir (hay que aclarar que las palabras que pertenezcan al pseudocódigo deberán ser escritas con MAYUSCULAS) con lo que tendremos lo siguiente:

### INICIO / FIN

Aquí tendremos lo que sería nuestro programa sin estas acciones no tendríamos un programa por ello esto será lo esencial de nuestro programa. Donde INICIO es el comienzo de mi programa y FIN donde concluye este.

### ESCRIBIR “ ” / LEER

Dónde ESCRIBIR “ ” será lo que nuestro ordenador mostrara en pantalla, lo que se mostrara será el contenido existente entre las comillas y LEER será lo que se realizara y guardara de acuerdo a nuestro seguimiento en el programa (lectura de datos).

### Declaración de variables (<nombreVariable>:<tipoDeDato>)

Estas deben estar definidas por un identificador (nombre), seguido por el tipo de dato, donde los tipos de datos son distintos y pueden ser los siguientes:

ENTERO -> valor entero positivo y/o negativo

REAL -> valor con punto flotante y signo

BOOLEANO -> valor de dos estados: verdadero o falso

CARÁCTER -> valor tipo carácter

CADENA -> cadena de caracteres

Con esto se logró ver que al declarar una variable es posible declarar más de una variable y esto se debía a que se utilizan arreglos indicando la cantidad de

variables que se necesitan para declararlas, pero deberían ser de la sintaxis siguiente:

<nombreVariable>[cantidad]:<tipoDeDato>

Hay que aclarar que en la parte de cantidad siempre debe estar dentro de los corchetes ya que con ello estamos indicando la cantidad de variables.

Con los datos que tenemos hasta el momento se pueden seguir construyendo más cosas siguiendo la práctica con esto vemos que existe un dato compuesto, dónde podemos contener uno o más tipos de datos simples y diferentes, a esto se le conoce como registro o estructura donde será otra variable y la reconoceremos por REG. Su sintaxis es la siguiente:

<nombreRegistro>: REG

    <nombreVariable\_1>: <tipoDeDato>

    ...

    <nombreVariable\_N>: <tipoDeDato>

FIN REG

Para acceder a los datos del registro se hace uso del operador “ ”.

Cuando deseamos crear variables constantes lo podemos hacer con la palabra reservada CONST, la que nos indica un que nuestro identificador no cambiara su valor en todo el proceso de nuestro programa. Las constantes se deberán inicializar a momento de declararse, es decir deberán ir al comienzo y junto con nuestra declaración de variables.

Así mismo tendremos nuestros operadores los cuales serán:

Operadores aritméticos: Suma (+), Resta (-), multiplicación (\*), división real (/), división entera (div), módulo (mod), exponencial (^), asignación (:=).

Operadores lógicos: igualdad (=), y-lógica o AND (&), o-lógica u OR (|), negación o NOT (!), relaciones de orden (<, >, <=, >=) y diferente (<>).

Es importante ver como es la notación que se utiliza para nombrar variables y nombres de funciones ya que estas deben tener la notación de camello, dónde los nombres de las palabras deben de empezar con mayúscula y el resto se escribe con minúsculas. En donde la notación de camello se divide en dos la Lower camel (dónde la primera letra inicia con minúscula) y la Upper camel (en dónde las palabras inician con Mayúsculas).

Lo que continuo en la práctica fue ver las estructuras de control de flujo, que son las que nos permiten la ejecución de un conjunto de instrucciones por las que estaría constituido nuestro programa. Existen tres estructuras de control que son:

- Estructura de control secuencial  
Estas son las declaraciones que se realizan una tras de otra en el orden que se encuentran escritas. Por ejemplo; iniciamos el programa, declaramos una variable, y podemos seguir asignándole un valor a la variable mediante operaciones y posteriormente damos fin al programa.
- Estructuras de control condicionales (o selectivas)  
Estas nos permiten evaluar una expresión lógica (SI o NO), con ello dependiendo de nuestro resultado se realizara uno u otro flujo de instrucciones, dónde nuestra estructura de control de flujo más simple es la estructura condicional SI, tenemos que su sintaxis es la siguiente:

SI condición ENTONCES

[Acción]

FIN SI

Esto en nuestro programa evaluara cada acción que se le asigne y dependiendo de que si cumple la condición se ejecutaran las instrucciones del bloque SI, en caso de que no se ejecuta las instrucciones del bloque DE LO CONTRARIO, y al final de nuestro pseudocódigo sigue su flujo normal.

La estructura del flujo SI-DE LO CONTRARIO es:

SI condición ENTONCES

[Acciones]

FIN SI

DE LO CONTRARIO

[Acciones DE LO CONTRARIO]

FIN DE LO CONTRARIO

Igualmente, para la estructura condicional SELECCIONAR-CASO realiza la validación de nuestra variable y su sintaxis es la siguiente

SELECCIONAR (variable) EN

CASO valor1 -> [Acción]

CASO valor2 -> [Acción]

CASO valor3 -> [Acción]

DEFECTO -> [Acción]

FIN SELECCIONAR

- Estructuras de control repetitivas (o iterativas)  
Estas estructuras también llamadas cíclicas nos permiten ejecutar una serie de instrucciones mientras se cumpla nuestra expresión lógica donde

tendremos dos tipos de expresiones cíclicas MIENTRAS y HACER-MIENTRAS y su sintaxis es la siguientes:

- a. MIENTRAS condición ENTONCES  
    [Acción]  
    FIN MIENTRAS
- b. HACER  
    [Acción]  
    MIENTRAS condición

Por ultimo lo que se vio en la práctica fueron las funciones. Estas se dan cuando la solución de algún problema es muy compleja, con esto implica la división de un problema en sub procesos que juntos formaran la solución correcta y completa.

Las funciones están constituidas por un identificador, parámetros de entrada y un valor retorno, su sintaxis es la siguiente:

INICIO

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

        [Acciones]

    FIN FUNC

FIN

Habiendo dicho todo lo anterior, debemos saber que todas las estructuras de control de flujo deberán estar inscritas en alguna función

A lo largo de la práctica se abordó todo lo teórico y junto con ello los ejemplos que se encuentran en el manual de la práctica, pero en el transcurso de algunas clases se dieron ejemplos utilizando el pseudocódigo.

Conclusiones

El pseudocódigo es fundamental para lograr entender como es la estructura de cualquier programa y con ello saber cuál debe de ser la sintaxis correcta dependiendo del lenguaje en el que se programara en un futuro. Así mismo se conocieron ciertos comandos y cuáles son las funciones que tiene cada una de ellas, debemos comprender bien lo que es el pseudocódigo ya que desde mi punto de vista a lo largo de esta práctica es que, si no respetamos el formato, sintaxis, etc., no lograremos hacer nada ya que el ordenador realizara las acciones que le pidamos, pero estas deberán estar claras al momento de ordenárselas a nuestro ordenador