|  |
| --- |
| Laboratorio de Computación  Salas A y B |
|  |

|  |  |
| --- | --- |
| Profesor: | García Morales Karina |
| Asignatura: | Fundamentos de Programación |
| Grupo: | 1121 |
| No de Práctica(s): | 05 |
| Integrante(s): | Valle Olivas Guillermo |
|  |  |
|  |  |
|  |  |
| No. de Equipo de cómputo empleado: |  |
| Semestre: | 2019-1 |
| Fecha de entrega: |  |
| Observaciones: |  |
|  |  |

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

**Pseudocódigo**

**Objetivo:**

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

**Actividades:**

x 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.

x 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.

**¿Qué es un pseudocódigo?**

La palabra pseudocódigo formada por el prefijo “[pseudo](http://dle.rae.es/?id=XkBx392)“, que según el diccionario de la RAE (Real Academia Española) significa “falso“, de ahí parte para que también sea conocido como “falso lenguaje”.

Dentro de la programación se basa en un lenguaje de programación real, donde se permite expresar las instrucciones en un lenguaje común (ejemplo: español, inglés u otro idioma) para facilitar la escritura y lectura, debido a que está en lenguaje común si requiere una modificación es fácil hacerla. La escritura sólo puede ser comprendida por el ser humano y no por la máquina, por ende, tampoco puede ejecutarla. La finalidad de un pseudocódigo es representar la solución a un algoritmo (problema) de la forma más detallada posible, utilizando acciones sucesivas. Según Analía Lanzillotta: “Considerado como un lenguaje falso el pseudocódigo, que es un lenguaje intermedio entre nuestro lenguaje y el de programación, debido a que quien lo utiliza se guía por una serie de normas, pero sin llegar a usar una estructura tan rígida como la del lenguaje de programación”.[*Liga*](http://informaticabachilleratoitea.blogspot.mx/p/pseudocodigo.html)

En otras palabras, el [pseudocódigo](http://definicion.de/pseudocodigo/) es un lenguaje simplificado entre el programador y la máquina, hecho por el programador en su propio idioma, para describir un algoritmo y  poder comprender mejor la estructura de dicho programa, donde el lenguaje simplificado no puede ser compilado, ejecutado ni corrido por la máquina. Siendo una herramienta que se encuentra previa al lenguaje formal de programación

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).

.

**Características del pseudocódigo**

Las características de este pseudolenguaje fueron propuestas en 2001 por el responsable de la asignatura Fundamentos de Programación de la carrera de Ingeniería Informática de la FICH-UNL. Las premisas son: 

 Sintaxis sencilla

 Manejo de las estructuras básicas de control

 Solo 3 tipos de datos básicos: numérico, caracter /cadenas de caracteres y lógico (verdadero-falso).

 Estructuras de datos: arreglos

**Tipos de datos válidos en un pseudocódigo**

ENTERO valor entero positivo y/o negativo

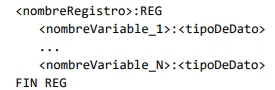
REAL valor con punto flotante y signo

BOOLEANO valor de dos estados: verdadero o falso

CARACTER valor tipo carácter

CADENA cadena de caracteres

Existe un tipo de dato compuesto, es decir, que puede contener uno o más tipos de datos simples diferentes. Este tipo de dato se conoce como registro o estructura y su sintaxis es la siguiente:



Para crear una variable tipo registro se debe indicar el nombre del registro y el nombre de la variable. Para acceder a los datos del registro se hace uso del operador ".".

**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 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.

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

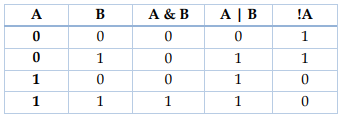


6. Operadores aritméticos: Se tiene la posibilidad de utilizar operadores aritméticos y lógicos:

Operadores aritméticos: suma (+), resta (-), multiplicación (\*), división real (/), división entera (div), módulo (mod), exponenciación (^), asignación (:=).

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

La tabla de verdad de los operadores lógicos AND, OR y NOT se describe a continuación:



NOTA: A y B son dos condiciones, el valor 0 indica falso y el valor 1 indica verdadero.

7. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

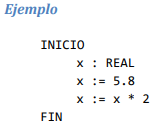
En la notación de camello (llamada así porque parecen las jorobas de un camello) los nombres de cada palabra empiezan con mayúscula y el resto se escribe con minúsculas. Existen dos tipos de notaciones de camello: lower camel case que en la cual la primera letra de la variable inicia con minúscula y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

**Estructuras de control de flujo**

Las estructuras de control de flujo permiten la ejecución condicional y la repetición de un conjunto de instrucciones. Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

**Estructura de control secuencial**

Las estructuras de control secuenciales son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.



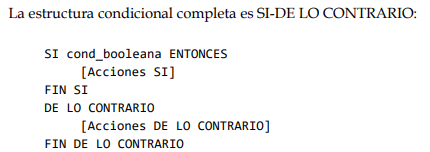
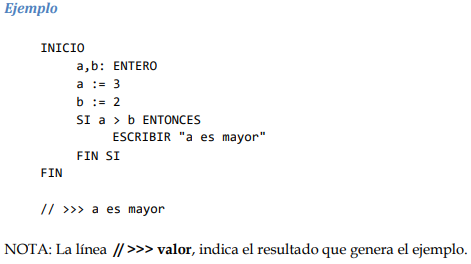
**Estructuras de control condicionales (o selectivas)**

Las estructuras de control condicionales permiten evaluar una expresión lógica (condición que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes (o se ejecuta una acción o se ejecuta la otra)

La estructura de control de flujo más simple es la estructura condicional SI, su sintaxis es la siguiente:

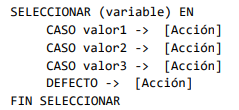


Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se ejecutan las instrucciones del bloque [Acción]. Si no se cumple la condición, se continúa con el flujo normal del programa.



Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se ejecutan las instrucciones del bloque SI [Acciones SI]. Si no se cumple la condición se ejecutan las instrucciones del bloque DE LO CONTRARIO [Acciones DE LO CONTRARIO]. Al final el pseudocódigo sigue su flujo normal.

La estructura condicional SELECCIONAR-CASO valida el valor de la variable que está entre paréntesis y comprueba si es igual al valor que está definido en cada caso. Si la variable no tiene el valor de ningún caso se va a la instrucción por defecto (DEFECTO).



**Estructuras de control iterativas o repetitivas**

Las estructuras de control de flujo iterativas o repetitivas (también llamadas cíclicas) permiten ejecutar una serie de instrucciones mientras se cumpla la expresión lógica. Existen dos tipos de expresiones cíclicas MIENTRAS y HACER- MIENTRAS.

La estructura MIENTRAS (WHILE en inglés) primero valida la condición y si ésta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, de lo contrario rompe el ciclo y continúa el flujo normal del pseudocódigo.



El final de la estructura lo determina la etiqueta FIN MIENTRAS.

La estructura HACER-MIENTRAS primero ejecuta las instrucciones descritas en la estructura y al final valida la expresión lógica.

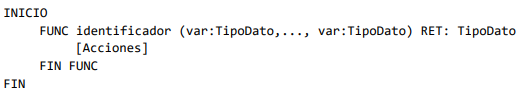


Si la condición se cumple vuelve a ejecutar las instrucciones de la estructura, de lo contrario rompe el ciclo y sigue el flujo del pseudocódigo. Esta estructura asegura que, por lo menos, se ejecuta una vez el bloque de la estructura, ya que primero ejecuta y después pregunta por la condición.

**Funciones**

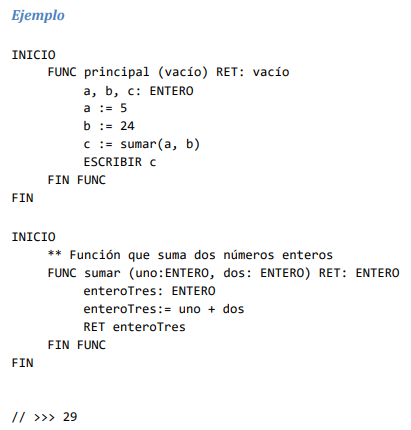
Cuando la solución de un problema es muy compleja se suele ocupar el diseño descendente (divide y vencerás). Este diseño implica la división de un problema en varios subprocesos más sencillos que juntos forman la solución completa. A estos subprocesos se les llaman métodos o funciones.

Una función está constituida por un identificador de función (nombre), de cero a n parámetros de entrada y un valor de retorno:



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.



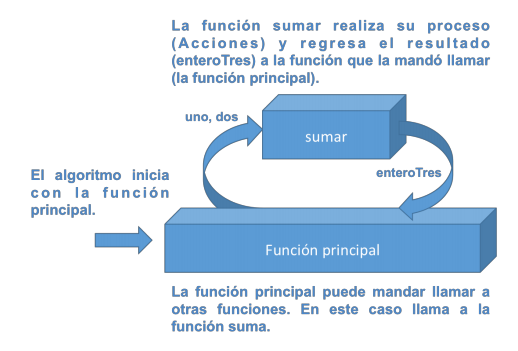
NOTA: Los dos asteriscos (\*\*) dentro de un pseudocódigo se utilizan para hacer un comentario y, por tanto, lo que esté en la misma línea y después de los \*\* no es parte del algoritmo y no se toma en cuenta. Es una buena práctica realizar comentarios sobre una función o sobre un bloque del algoritmo para guiar sobre el funcionamiento del mismo.

Descripción

La primera función que se ejecuta es 'principal', ahí se crean las variables (uno y dos) y, posteriormente, se manda llamar a la función 'sumar'. La función 'sumar' recibe como parámetros dos valores enteros y devuelve como resultado un valor de tipo entero, que es la suma de los valores que se enviaron como parámetro.

Para la función 'principal' los pasos que realiza la función 'sumar' son transparentes, es decir, solo manda a llamar a la función y espera el parámetro de retorno.

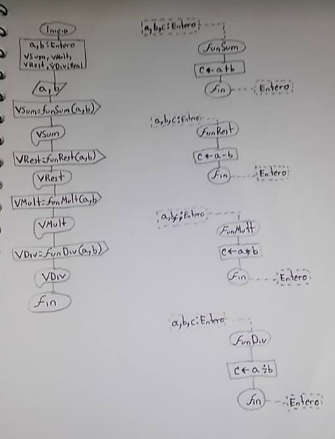
La siguiente figura permite analizar el pseudocódigo a través del tiempo. El algoritmo inicia con la función principal, dentro de esta función se hace una llamada a una función externa (sumar). Sumar realiza su proceso (ejecuta su algoritmo) y devuelve un valor a la función principal, la cual sigue su flujo hasta que su estructura secuencial (las instrucciones del pseudocódigo) llega a su fin.



**Ejercicios de Tarea:**

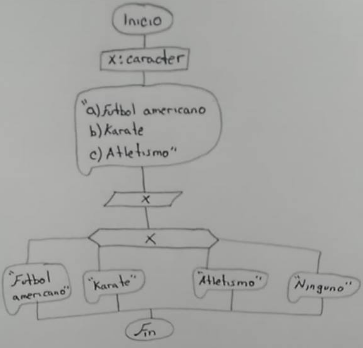
Realizar el pseudocódigo de los ejercicios de tarea de la práctica 4.

1.- Calculadora para dos variables (+,-,\*,/)



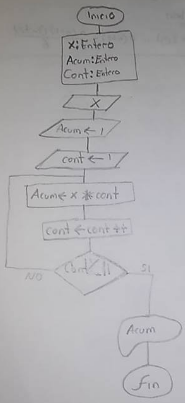
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INICIO  FUNCprincipal  a,b,c: Entero  c: Suma (a,b)  c: Resta (a,b)  c: Mult (a, b)  c: Div (a,b)  ESCRIBIR c  FIN FuncPrincipal  FIN | INICIO  FUNCsuma  a,b,c: Entero  c:=a+b  ESCRIBIR c  FIN FUNC  FIN | INICIO  FUNCResta  a,b,c: Entero  c:=a-b  ESCRIBIR c  FIN FUNC  FIN | INICIO  FUNCMult  a,b,c: Entero  c:=a\*b  ESCRIBIR c  FIN FUNC  FIN | INICIO  FUNCDiv  a,b,c: Entero  c:=a/b  ESCRIBIR c  FIN FUNC  FIN |

2.-Menú de deportes (3 deportes)



|  |
| --- |
| INICIO  x:Carácter  ESCRIBIR ( “ a) Futbol Americano b) Karate c) Atletismo” )  LEER: x  SELECCIONAR (x)  CASO a  ESCRIBIR “Futbol Americano  CASO b  ESCRIBIR “Karate”  CASO c  ESCRIBIR “Atletismo”  DEFECTO  ESCRIBIR “Ninguno”  FIN SELECCIONAR  FIN |

3.- Tablas de multiplicar del 1 al 10, el usuario proporciona el valor a calcular (Para o Mientras)



|  |
| --- |
| INICIO  x: Entero  Acum: Entero  Cont: Entero  LEER x  Cont:=1  Acum:=1  INICIOMientras  Acum:=x\*Cont  Cont:=Cont++  Cont<11  ESCRIBIR Acum  FINMientras  FIN |

**Conclusiones:**

Con esta práctica me fue posible practicar el declarar variables y tipos de datos en un pseudocódigo, siento que eso es lo que más me falla al momento de hacer el pseudocódigo y prefiero practicarlo desde ahora que arrastrar este tipo de errores al momento de hacer el código.

**Fuentes**

[**http://michelletorres.mx/que-es-el-pseudocodigo/**](http://michelletorres.mx/que-es-el-pseudocodigo/)

[**http://pseint.sourceforge.net/pseudocodigo.php**](http://pseint.sourceforge.net/pseudocodigo.php)

Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.

Metodología de la programación a través de pseudocódigo. Miguel Ángel Rodríguez Almeida, primera edición, McGraw Hill