Laboratorio de Computación
Salas A y B

Profesor:	Karina García Morales
Asignatura:	Fundamentos de Programación
Grupo:	1121
No de Práctica(s):	5
Integrante(s):	Diego Ramírez Martínez
No. de Equipo de cómputo empleado:	28
Semestre:	2019-1
Fecha de entrega:	25 de septiembre de 2018
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.

DESARROLLO

DEFINICIÓN DE PSEUCÓDIGO

La palabra pseudocódigo formada por el prefijo **“pseudo“**, significa **“falso“**, de ahí parte para que también sea conocido como **“falso lenguaje”**.

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).

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.

Este es usado previamente para usar un lenguaje de programación.

CARACTERÍSTICAS DE UN PSEUDOCÓDIGO

Se puede ejecutar en un ordenador (SLE, LPP, PilatoX, PSeInt o Seudocódigo).

Es una forma de representación sencilla de utilizar y de manipular.

Facilita el paso del programa al lenguaje de programación.

Es independiente del lenguaje de programación que se vaya a utilizar.

Es un método que facilita la programación y solución al algoritmo del programa.

SINTAXIS DE PSEUDOCÓDIGO

1) ***Alcance del programa:*** Esto quiere decir que todo pseucódigo está limitado por etiquetas INICIO Y FIN.
2) ***Sangrías:*** esto se refiere a que en un pseudocódigo debe haber alineaciones para así poder identificar con facilidad los códigos.
3) ***Palabras reservadas:*** esto se refiere a que todas las palabras usadas específicamente para un pseudocódigo serán escritas en mayúsculas.
4) ***Lectura y Escritura:*** la etiqueta **LEER** se usa para la lectura de datos y la etiqueta de **ESCRIBIR** se usa para la redacción de datos.

Por ejemplo:

```
ESCRIBIR “Ingresar nombre”
LEER nombre
```

5) ***Declaración de variables:*** cada vez que se declare una variable se debe identificar con un nombre, seguido de 2 puntos y este seguido por el tipo de dato.

Por ejemplo:

```
<nombreVariable>:<tipoDeDato>
```

Dentro de esta, existen los tipos de datos, estos son:

- ***Real:*** este tipo de dato tiene la característica de ser un valor con punto flotante y signo.
- **Entero:** este tipo dato corresponde a aquellas variables que exclusivamente pueden recibir valores positivos y/o negativos, al igual, no llevan decimales.
- ***Booleano:*** este tipo de dato se divide en 2 estados: verdadero o falso.
- ***Carácter:*** este tipo de dato se caracteriza por ser un carácter.
- ***Cadena:*** este tipo de dato se identifica, como su nombre lo dice, por ser una cadena de caracteres.

Por ejemplo:

```
contador: ENTERO
producto: REAL
continuar: BOOLEANO
```

Es posible declarar más de una variable de un mismo tipo de dato utilizando arreglos, indicando la cantidad de variables que se requieren, su sintaxis es la siguiente:

```
<nombreVariable>[cantidad]:<tipoDeDato>
```

Para una variable de 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 "**.**".

Por ejemplo:

```
nombrecompleto:REG
     nombre: CADENA
     apellidopaterno: CADENA
     apellidomaterno: CADENA
FIN REG


usuario:REG nombre    // variable llamada usuario de tipo
registro
```

```
usuario.nombre := "Diego"
usuario.apellidopaterno := “Ramírez”
usuario.apellidomaterno := "Martínez"
```

Es posible crear variables constantes con la palabra reservada **CONST**, la cual indica que un identificador no cambia su valor durante todo el pseudocódigo. Las constantes (por convención) se escriben con mayúsculas y se deben inicializar al momento de declararse.

Por ejemplo:

```
NUM_MAX := 300: REAL, CONST
```

6) ***Notación de camello:*** para poder nombrar variables y funciones se usa la notación de camello. Esta se le conoce como notación de camello, ya que la forma en la que se escribe un pseudocódigo tiene la forma de las jorobas de un camello. Y los nombres de cada palabra empiezan con mayúscula.

Existen dos tipos de notaciones de camello:

- **Lower camel case:** que en la cual la primera letra de la variable inicia con minúscula.
- **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 debe utilizar un prefijo.

Por ejemplo:

```
// variables
RealVolumenDelPentágono: REAL // lower camel case
EnteroAreaDelCuadrado: REAL // upper camel case
```

```
// funciones
calcularVolumen()
obtenerArea()
```

7) ***Operadores aritméticos y lógicos:*** 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 (<>).

TABLA DE VERDAD DE LOS OPERADORES LÓGICOS AND, OR Y NOT

A	B	A & B	A | B	!A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

***A y B son 2 condiciones, el valor 0 indica falso y el valor 1 indica verdadero**.

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:*** estas estructuras son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.

- ***Estructura de control condicional:*** este tipo de estructuras son las más simples, dentro del aspecto de la estructura condicional **SI.** Estas estructuras de control condicionales permiten evaluar una expresión lógica, esto quiere decir que se caracteriza por usa una condición que puede ser verdadera o falsa. Y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras solamente ejecutan una acción u otra.

`SI condición ENTONCES` → **Se evalúa la expresión lógica**

Si se cumple la condición

se ejecutan las instrucciones del bloque [Acción]

`[Acción]`

Si no se cumple la condición

se continúa con el flujo normal del programa

`FIN SI`

Por ejemplo:

```
INICIO
    a,b: ENTERO
    a := 3
    b := 2
    SI a > b ENTONCES
    ESCRIBIR "a es mayor"
    FIN SI
FIN
// >>> a es mayor
```

indica el resultado

- ***Estructura de control repetitivas o iterativas:*** 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.

```
MIENTRAS condición ENTONCES
    [Acción]
FIN MIENTRAS
```

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
```

EJERCICIOS DE TAREA

PSEUDOCÓDIGO DE SABORES DE NIEVE

```
INICIO
     X: Carácter
     ESCRIBIR “A) LIMÓN, B) FRESA, C) CAFÉ, D) MANGO”
     LEER X
     SELECCIONAR (X) en:
          Caso A
               ESCRIBIR “LIMÓN”
          CASO B
               ESCRIBIR “FRESA”
          CASO C
               ESCRIBIR “CAFÉ”
          CASO D
               ESCRIBIR “MANGO”
          Defecto
               “Ninguno”
     FIN SELECCIONAR
FIN
```

1.- Calculadora con las 4 operaciones básicas, para 2 variables que ingresa el usuario (+, -, *, /):

ANÁLISIS

Datos de entrada: Dos variables.

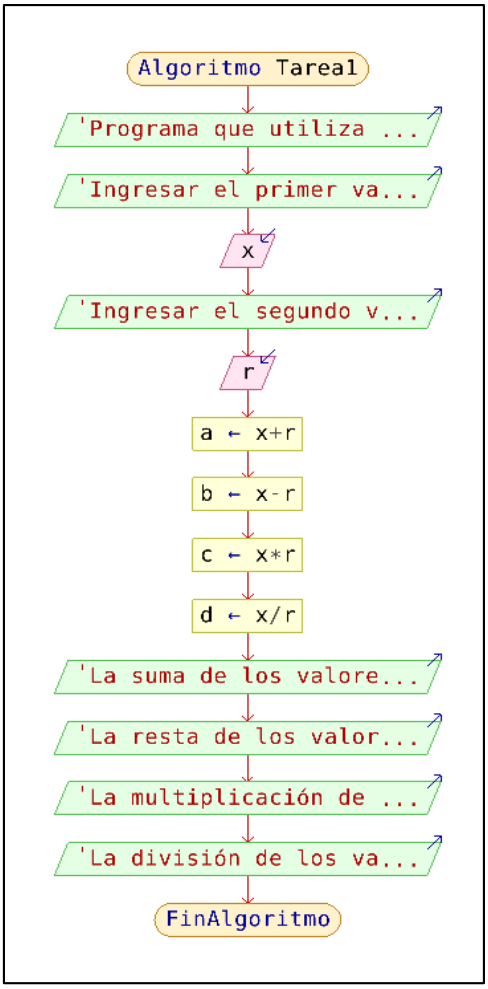
Restricciones: Para la división, el segundo número NO debe ser cero.

Datos de salida: Resultado de las operaciones aritméticas.

ALGORITMO

1) INICIO
2) A y B: ENTEROS
3) VSum, Vres, VMult, VDiv: REAL
4) Leer A y B.
5) Función VSum.
6) Imprime VSum
7) Función VRes
8) Imprime VMult
9) Función VDiv
10) Imprime VDiv
11) FIN

DIAGRAMA DE FLUJO



PSEUDOCÓDIGO

```
INICIO
    FUNC PRINCIPAL
    A,B,C: ENTERO
    C: Suma (A,B)
    C: Resta (A,B)
    C: Mult (A,B)
    C: Div (A,B)
        ESCRIBIR C
        FIN FUNC
    FIN
```

```
INICIO
    FUNC Suma
    A,B,C: ENTERO
    C: A+B
    ESCRIBIR C
    FIN FUNC Suma
FIN
```

```
INICIO
    FUNC Resta
    A,B,C: ENTERO
    C: A-B
    ESCRIBIR C
    FINFUNC Resta
FIN
```

```
    INICIO
        FUNC Mult
        A,B,C: ENTERO
        C: (A)(B)
        ESCRIBIR C
        FIN FUNC Mult
    FIN
```

```
    INICIO
        FUNC DIV
        A,B,C: ENTERO
        Si B=!0 ENTONCES
        C: A/B
        ESCRIBIR C
        FIN Si
            FIN FUNC Div
        FIN
```

2.- Menú de deportes (3 deportes) emplear condicional múltiple:

ANÁLISIS

Datos de entrada: Tres deportes a escoger.

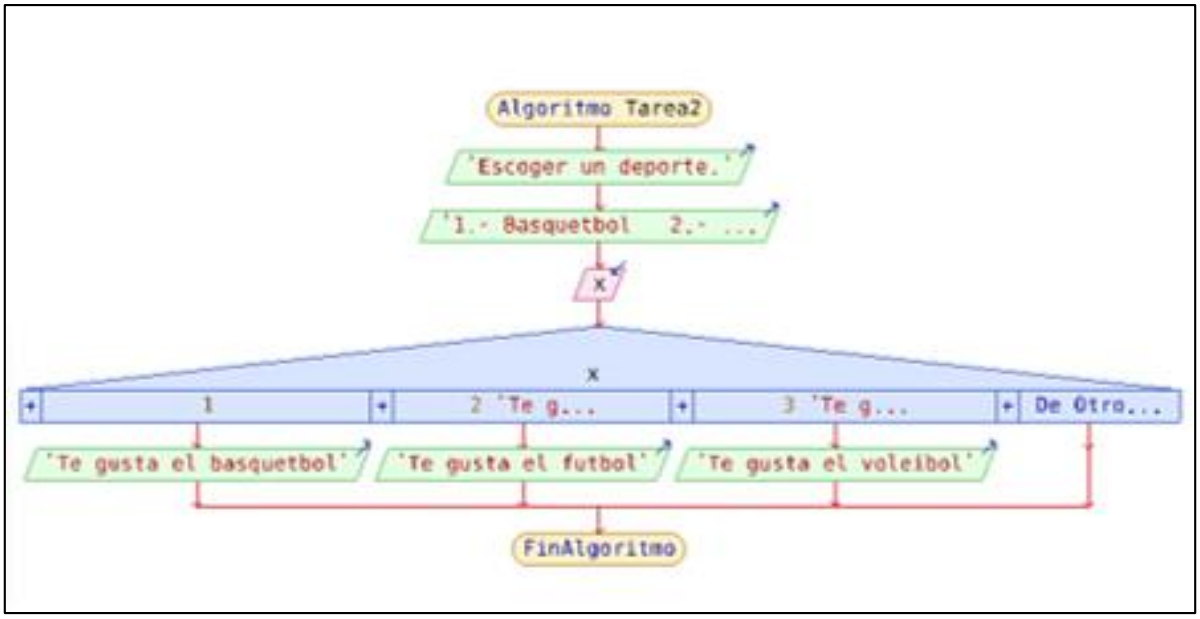
Restricciones: Solamente se escoge uno de ellos.

Datos de salida: Deporte a escoger.

ALGORITMO

1) INICIO
2) x: Carácter
3) “Seleccionar deporte favorito: A) Tenis, B) Fútbol, C) Voleibol”
4) Leer x
5) Si se escoge A), IMPRIMIR “Basquetbol” e ir a FIN
6) Si se escoge B), IMPRIMIR “Fútbol” e ir a FIN
7) Si se escoge C), IMPRIMIR “Voleibol” e ir a FIN
8) En caso contario de no escoger ninguna de las anteriores, ir a la opción defecto y e ir a FIN
9) FIN

DIAGRAMA DE FLUJO



PSEUDOCÓDIGO

```
INICIO
     X: Carácter
     ESCRIBIR “A) Tenis B) Fútbol C) Voleibol”
     Leer x
     SELECCIONAR (x) en:
     Caso A
     ESCRIBIR “Basquetbol”
     Caso B
     ESCRIBIR “Fútbol”
     Caso C
     ESCRIBIR “Voleibol”
     Defecto
     FIN SELECCIONAR
FIN
```

3.- Realizar programa que muestre la tabla de multiplicar del número que ingrese el usuario, solo debe ingresar del 1 al 10 (emplear ciclo MIENTRAS o PARA):

ANÁLISIS

Datos de entrada: Tabla que se desea calcular.

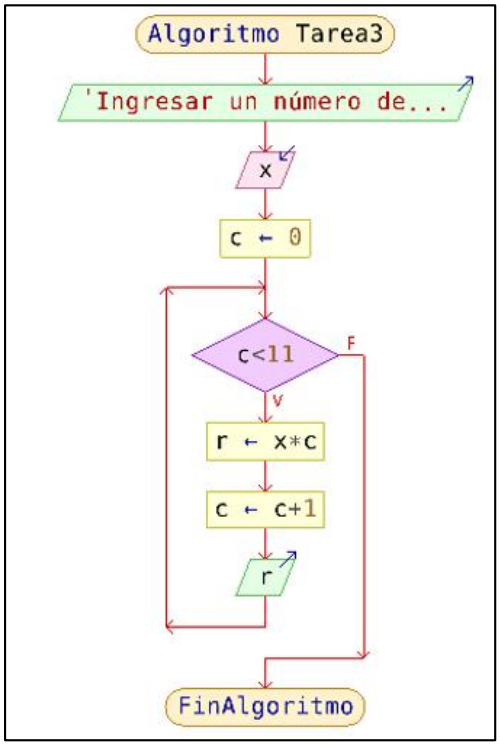
Restricciones: Hasta la tabla del 10.

Datos de salida: Valor de la tabla de multiplicar.

ALGORITMO

```
1  Algoritmo Tarea3
2      Escribir 'Ingresar un número del 1-10'
3      Leer x
4      c=0
5      Mientras c<11 Hacer
6          r<-x*c
7          c<-c+1
8          Escribir r
9      FinMientras
10 FinAlgoritmo
```

DIAGRAMA DE FLUJO



PSEUDOCÓDIGO

```
INICIO.
      R, X, M: ENTERO
      ESCRIBIR “Ingresar X”
      LEER X
      INICIO PARA
```

```
        M:1, M<=10, M++
        R: (M)(X)
        ESCRIBIR R
        FIN PARA
FIN
```

CONCLUSIONES

Gracias a que realicé esta práctica puedo reforzar mis conocimientos adquiridos durante las clases acerca del pseudocódigo, su definición, sus características y como hacer el pseudocódigo. Al igual, puedo reforzar mis conocimientos previos de los tipos de diagramas de flujo, en especial los de estructura de control.

BIBLIOGRAFÍA

http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS_FP/fp_p5.pdf