	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 1121

No de Práctica(s): 5

Integrante(s): Jacinto Rodríguez Moisés Rodrigo

No. de Equipo de cómputo empleado 6

Semestre: 2019-1

Fecha de entrega: 25/09/2018

Obervaciones:

CALIFICACIÓN:

Pràctica 5 Pseudocódigo

-Objetivo:

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

-Desarrollo

-¿Qué es 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).

-Tipos de datos

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

-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.

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

```
// 5 variables de tipo entero
    // 3 variables de tipo real
    // 6 variables de tipo booleano
```

-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 es la siguiente:

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 dos condiciones, el valor 0 indica falso y el valor 1 indica verdadero.

-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.

EJEMPLOS:
// variables
realAreaDelTriangulo: REAL // lower camel case
EnteroRadioCirculo: REAL // upper camel case
// funciones
calcularArea()
obtenerPerimetro()

Estructuras de control condicionales o selectivas
“Seleccionar” Condicional múltiple

-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.

*Secuencial
Son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.

EJEMPLO
INICIO
   x : REAL
   x := 5.8
   x := x * 2
FIN

*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)

*Iterativas.
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.

Función SELECCIONAR: condicional múltiple
Función seleccionar en el menú de los 3 helados

```
INICIO
  X:CARACTER
  A:= “Mamey”
  B:= “Mango”
  C:=”Chocolate”
  LEER X
  SELECCIONAR (X) EN
       CASO 1 →
            ESCRIBIR A) “Mamey”
        CASO 2 →
            ESCRIBIR B) “Mango”
        CASO 3 →
            ESCRIBIR C) “Chocolate”
       DEFECTO →
            ESCRIBIR “Opción Fallida”
  FIN SELECCIONAR
FIN
```

-Registro

```
facultad:REG
   carreraAlumno: CADENA
   númeroCuenta: ENTERO
   nombreAlumno: CADENA
FIN REG

alu:REGfacultad
aluCiencias: REGfacultad
aleIng: REGfacultad
```

TAREA.

-Suma

Análisis:

Datos de entrada: Dos números enteros

Datos de salida: La suma de esos dos números

Restricciones: Todos los números deben de ser enteros

Algoritmo:

1.- INICIO

2.- Pedir al usuario dos números enteros

3.- Si esos dos números no son enteros ir al paso 6.

4.- Leer esos dos números.

5.- R= A+B

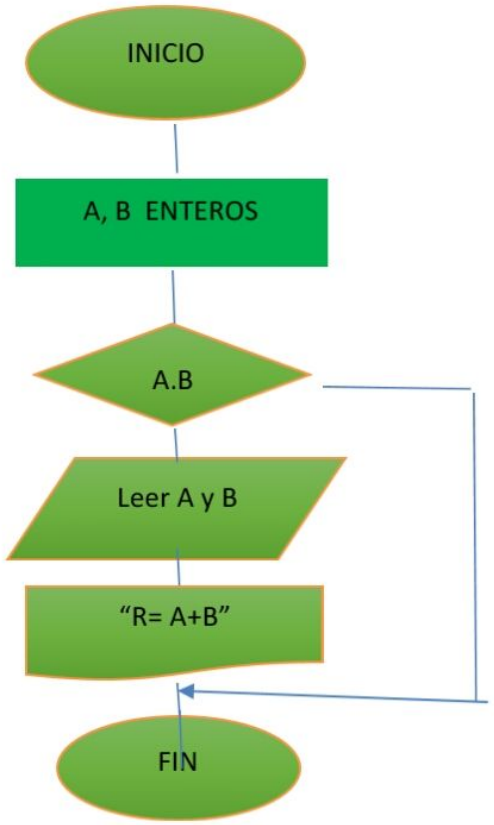
6.- FIN

Pseudocódigo

```
INICIO
    a, b: ENTERO
    SI a, b ≠ Z ENTONCES ir a FIN
    FIN SI
  R= a+b
FIN

// >>> R = a+b
```

Diagrama de flujo:



Pruebas de escritorio:

Iteración	Números	Suma	Salida
1	4 y 8	4+8	12
2	1 y 7	1+7	8
3	78 y 12	78+12	90

-Resta

Análisis:

Datos de entrada: Dos números enteros

Datos de salida: La diferencia de esos dos números enteros

Restricciones: Todos los números deben de ser enteros

Algoritmo:

1.- INICIO

2.- Pedir al usuario dos números enteros

3.- Leer esos dos números

4.- Si esos dos números no son enteros ir al paso 6

5.- R= A-B

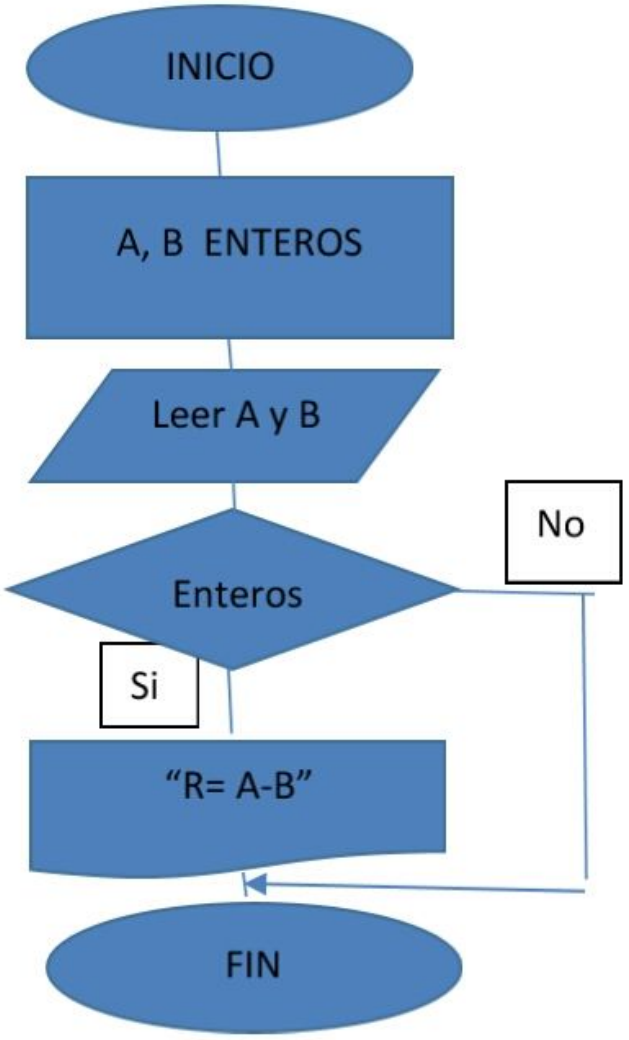
6.- FIN

Pseudocódigo:

```
INICIO
    a, b: ENTERO
    SI a, b ≠ Z ENTONCES ir a FIN
     FIN SI
   R= a-b
FIN

// >>> R = a-b
```

Diagrama de flujo:



Pruebas de escritorio

Iteración	Números	Resta	Salida
1	9 y 2	9-2	7
2	15 y 4	15-4	11
3	2 y 4	2-4	-2

-Multiplicación

Análisis:

Datos de entrada: dos números enteros

Datos de salida: el producto de esos dos números enteros

Restricciones: todos los números deben de ser enteros

Algoritmo:

1.- INICIO

2.- Pedir al usuario dos números enteros

3.- Leer esos dos números

4.- Si esos dos números no son enteros ir al paso 6

5.- Obtener el producto de esos dos números R= A*B

6.- FIN

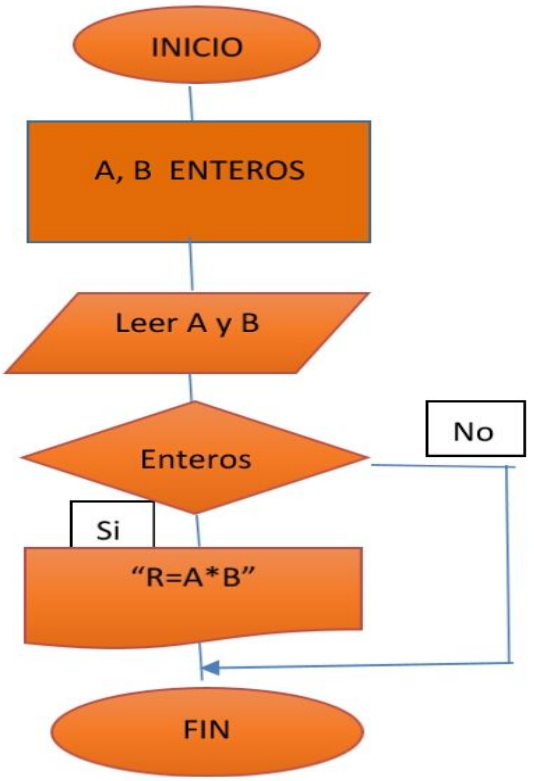
Pseudocódigo:

```
INICIO
    a, b: ENTERO
    SI a, b ≠ Z ENTONCES ir a FIN
     FIN SI
   R= a*b
FIN

// >>> R = a*b
```

Diagrama de flujo:

.



Pruebas de escritorio

Iteración	números	Multiplicación	Salida
1	5 y 4	5*4	20
2	8 y 9	8*9	72
3	3 y 11	3*11	33

-División

Análisis:

Datos de Entrada: dos números enteros

Datos de Salida: el cociente de esos dos números dados

Restricciones: todos los números deben de ser enteros

Algoritmo:

1.INICIO

2.Pedir al usuario dos números.

3.Leer los dos números dados.

4.Obtener el cociente de esos dos números.

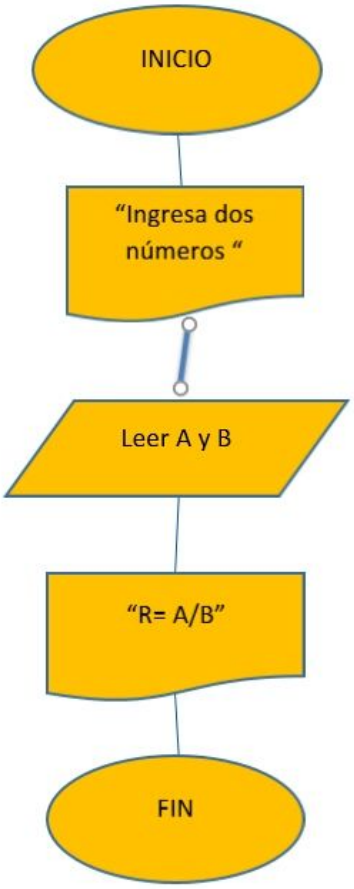
5.FIN.

Pseudocódigo:

```
INICIO
   a, b: VARIABLE
   R= a/b
FIN

// >>> R = a/b
```

Diagrama de flujo



Pruebas de escritorio:

Iteración	Números dados	Operación	Salida
1	30 y 2	30/2	15
2	1 y 2	1/2	0.5
3	25 y 4	25/4	6.25

-Función

Análisis:

Datos de entrada: Números dados y operación a realizar

Datos de salida: Resultado de la operación dada

Restricciones: todos los números deben de ser enteros

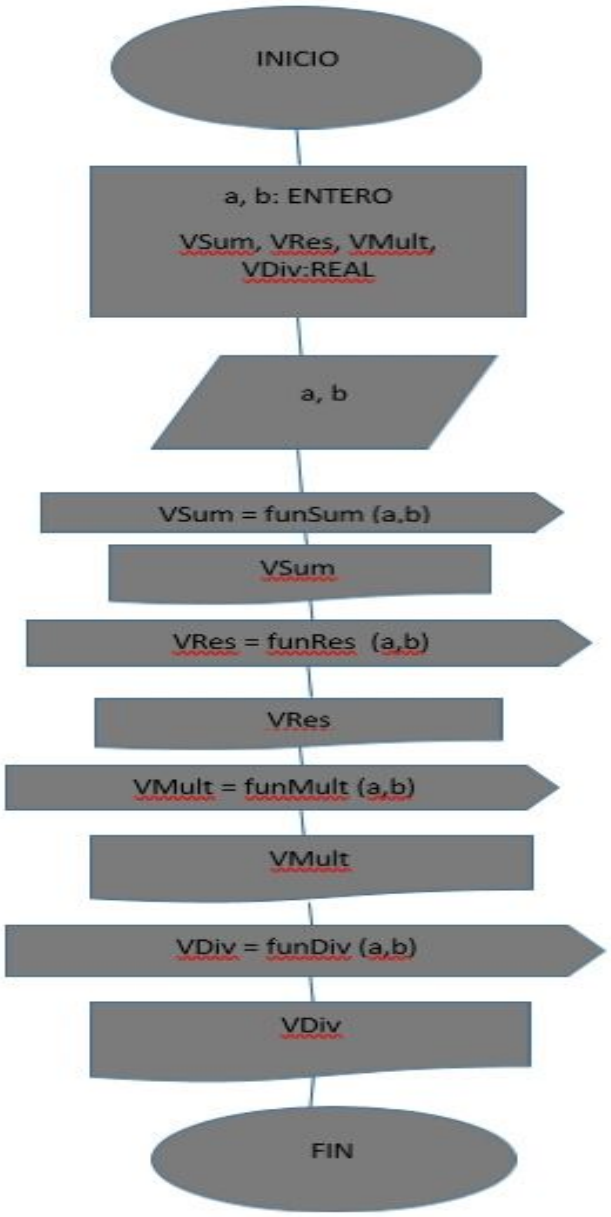
Algoritmo:

1.INICIO
2.Declarar variables
3.Leer los dos números dados
4.VSum = funSum (a, b)
5.VSum
6.VRes = funRes (a, b)
7.VRes
8.VMult = funMult (a, b)
9.VMult
10.VDiv = funDiv (a, b)
11.VDiv
12.FIN

Pseudocódigo

```
INICIO
  FUNC principal () RET:
     a, b: ENTERO
     c = sumar (a, b)
     ESCRIBIR c
     d = restar (a, b)
     ESCRIBIR d
     e = multiplicar  (a, b)
     ESCRIBIR e
     f = dividir  (a, b)
     ESCRIBIR f
  FIN FUNC
FIN
```

Diagrama de flujo



Pruebas de escritorio

Iteración	Función	Impresión	Salida
1	VSum = VSum (a,b)	VSum	R = a+b
2	VRes = VRes (a,b)	VRes	R = a-b
3	VMult = VMult (a,b)	VMult	R = a*b

-Menú de deportes

Análisis:

Datos de entrada: Selección del deporte

Datos de salida: El deporte seleccionado

Restricciones:

Algoritmo:

1.INICIO

2.Declarar variables

3.Pedir al usuario que elija un deporte de los manifestados.

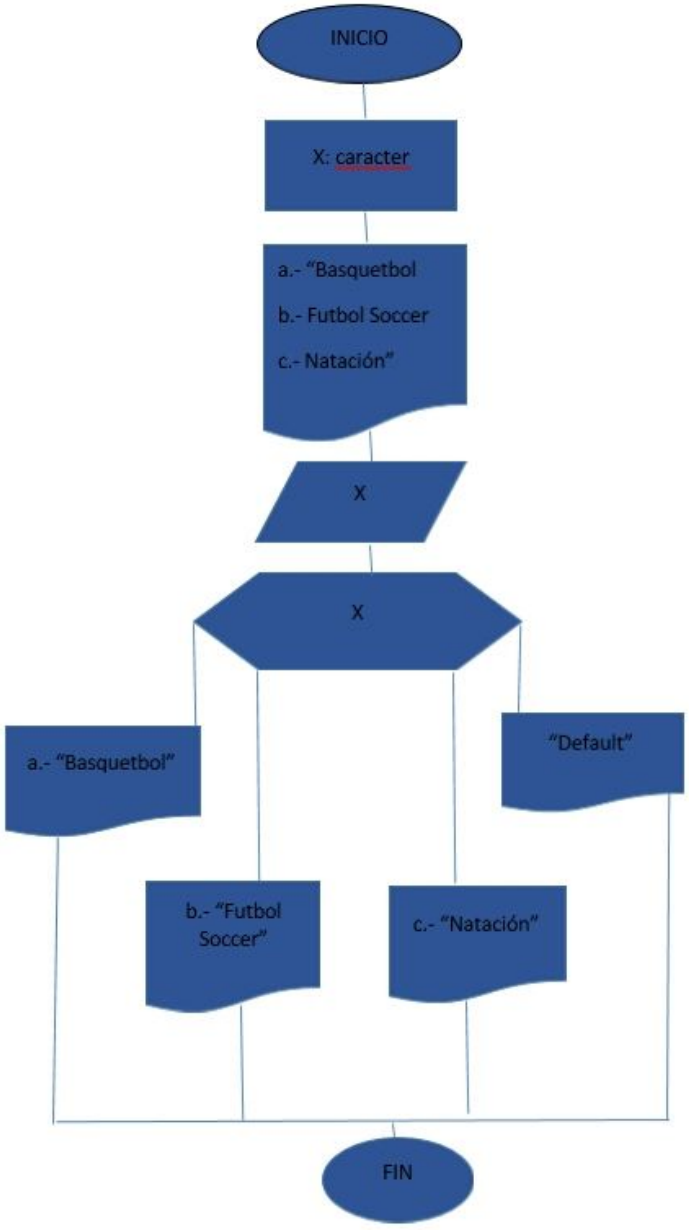
4.Manifestación del nombre del deporte

5.FIN

Pseudocódigo:

```
INICIO
   ENTERO: x
   SELECCIONAR:   (a)   EN
        CASO 1 →
            ESCRIBIR “Basquetbol”
        CASO 2 →
            ESCRIBIR “Fútbol soccer”
        CASO 3 →
            ESCRIBIR “Natación”
        DEFECTO →
            ESCRIBIR “Default”
    FIN SELECCIONAR
FIN
// >>> “Basquetbol”
```

Diagrama de flujo



-Tablas de multiplicar del 1 al 10; el usuario proporciona el valor a calcular

Análisis:

Datos de entrada: Dos números menores a 10 y mayores a 0

Datos de salida: Tablas de multiplicar del 1 al 10

Restricciones: Tablas de multiplicar únicamente del 1 al 10

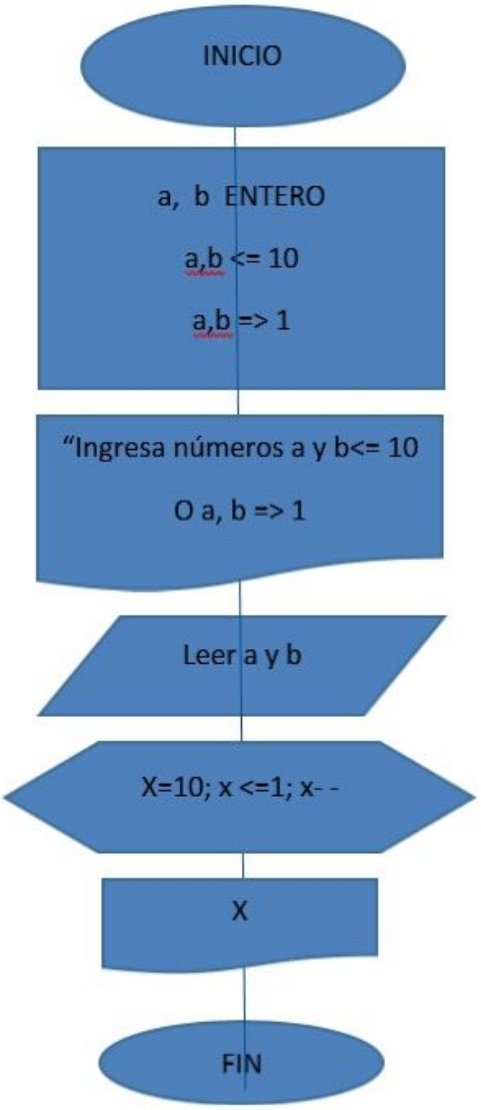
Algoritmo:

1.INICIO
2.Declarar variables
3.Pedir al usuario dos números a y b menores o iguales a 10 y mayores o iguales a 1
4.Leer a y b
5.Hacer x 10
6.Repetir con x desde 10 hasta 1
7.Escribe x
8.Hacer x x-1
9.Fin repetir
10.Fin

Pseudocódigo

```
INICIO
   ENTERO: x
       ESCRIBIR: “Ingrese dos números a = > 1 y b <= 10”
       LEER a y b
       HACER x 10
       REPETIR CON x DESDE 10 HASTA 1
       ESCRIBE x
       HACER x  x-1
    FIN REPETIR
FIN
```

Diagrama de flujo



Pruebas de escritorio.

Iteración	Números ingresados	Tabla de multiplicar	Salida
1	2 y 5	2	R=2*5
2	3 y 8	3	R=3*8
3	4 y 3	4	R=4*3