

Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Pimentel
<i>Asignatura:</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	5
<i>Integrante(s):</i>	Uno Karin Natalia
<i>No. de Equipo de cómputo empleado:</i>	37
<i>No. de Lista o Brigada:</i>	8723 #50
<i>Semestre:</i>	1
<i>Fecha de entrega:</i>	16 de septiembre del 2019
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Práctica 5:

Pseudocódigos

INTRODUCCIÓN.

El pseudocódigo es una forma de expresar los distintos pasos que va a realizar un programa, de la forma más parecida a un lenguaje de programación. Su principal función es la de representar por pasos la solución a un problema o algoritmo, de la forma más detallada posible, utilizando un lenguaje cercano al de programación. El pseudocódigo no puede ejecutarse en un ordenador ya que entonces dejaría de ser pseudocódigo, como su propio nombre indica, se trata de un código falso (pseudo = falso), es un código escrito para que lo entienda el ser humano y no la máquina.

LISTA DE PSEUDOCÓDIGOS

Instrucción	Significado
algoritmo nombre	Marca el comienzo de un algoritmo y le adjudica un nombre
Inicio	Marca el comienzo de un bloque de instrucciones
fin	Marca el final de un bloque de instrucciones
variables	
nombre_var es tipo_de_datos	Declaración de variables. Indica el identificador y el tipo de las variables que se van a usar en el algoritmo
constantes	
nombre_const = expresión	Declaración de constantes. La expresión se evalúa y su resultado se asigna a la constante. Este valor no puede modificarse a lo largo del programa.
leer (variable)	Entrada de datos. El programa lee un dato desde un dispositivo de entrada (si no se indica otra cosa, el teclado), asignando ese dato a la variable

escribir (variable)	Salida de datos. Sirve para que el programa escriba un dato en un dispositivo de salida (si no se indica otra cosa, la pantalla).
variable = expresión	Asignación. La expresión se evalúa y su resultado es asignado a la variable
si (condición) entonces inicio acciones-1 fin si_no inicio acciones-2 fin	Instrucción condicional doble. El ordenador evaluará la condición, que debe ser una expresión lógica. Si es verdadera, realiza las acciones-1, y, si es falsa, las acciones-2. Instrucción condicional simple. Es igual pero carece de la rama "si_no", de modo que, si la expresión de falsa, no se realiza ninguna acción y la ejecución continúa por la siguiente instrucción
según (expresión) hacer inicio valor1: acciones-1 valor2: acciones-2 ... valor3: acciones-N si_no: acciones- si_no fin	Instrucción condicional múltiple. Se utiliza cuando hay más de dos condiciones posibles (verdadero o falso) . Se evalúa la expresión, que suele ser de tipo entero, y se busca un valor en la lista valor1, valor2,... valorN que coincida con ella, realizándose las acciones asociadas al valor coincidente. Si ningún valor de la lista coincide con la expresión del "según", se realizan las acciones de la parte "si_no".
mientras (condición) hacer inicio acciones fin	Bucle mientras. Las acciones se repiten en tanto la condición, que debe ser una expresión lógica, sea verdadera. La condición se evalúa antes de entrar al bloque de acciones, de modo que pueden no ejecutarse ninguna vez.
repetir inicio acciones fin mientras que (condición)	Bucle repetir. Las acciones se repiten en tanto que la condición, que debe ser una expresión lógica, sea verdadera. Se parece mucho al anterior, pero la condición se evalúa al final del bucle, por lo que éste se ejecuta, como mínimo, una vez

para variable desde expr-ini hasta expr-fin hacer inicio acciones fin	Bucle para. Se evalúa la expresión expr-ini, que debe ser de tipo entero, y se asigna ese valor a la variable. Dicha variable se incrementa en una unidad en cada repetición de las acciones. Las acciones se repiten hasta que la variable alcanza el valor expr-fin.
---	--

Fuente: <http://informaticabachilleratoitea.blogspot.com/p/pseudocodigo.html>

OBJETIVO

En esta práctica, aplicando lo visto en clase se realizarán una serie de pseudocódigos, que al igual que los diagramas de flujo y algoritmos tienen el fin de solucionar un problema.

DESARROLLO/RESULTADOS

ACTIVIDAD 1. Desarrollar pseudocódigo que reciba un número obtenga su factorial.

INICIO

n>0

conta: INT

multip: INT

result: INT

result= n

conta= 1

DO:

 multi= n - conta

 result= result*multi

 conta= conta + 1

WHILE:

 multi >= 1

FIN

VERIFICAR:

- 0

n	Multi	conta	result
0	-1	1	0

NOTA. "multi" no es mayor a 1.

- 2

n	Multi	conta	result
2	1	1	2

- -4

NOTA. "n" no es positivo.

- 5

n	multi	conta	result
5	4	1	20
20	3	2	60
60	2	3	120
120	1	4	120

ACTIVIDAD 2. Desarrollar un pseudocódigo que lea 2 datos, nivel e ingreso.

INICIO

Nivel: INT POSITIVO

Ingreso: REAL

Base: REAL

Nivel= n

Ingreso= y

READ Nivel

SWITCH Nivel

Case 1:

PRINT Base (\$) 0.00

Cuota fija (\$) 0.00

Impuesto (%) 1.92

BREAK

Case 2:

PRINT Base (\$) 6,942.21

Cuota fija (\$) 133.28

Impuesto (%) 6.40

BREAK

Case 3:

PRINT Base (\$) 58,922.16

Cuota fija (\$) 3,460.00

Impuesto (%) 10.88

BREAK

Case 4:

PRINT Base (\$) 103,550.45

Cuota fija (\$) 8,315.57

Impuesto (%) 16.00

BREAK

Case 5:

PRINT Base (\$) 120,372.84

Cuota fija (\$) 11,007.14

Impuesto (%) 17.92

BREAK

Case 6:

PRINT Base (\$) 144,119.24

Cuota fija (\$) 15,262.49

Impuesto (%) 21.36

BREAK

Case 7:

PRINT Base (\$) 290,667.76

Cuota fija (\$) 46,565.26

Impuesto (%) 23.52

BREAK

Case 8:

PRINT Base (\$) 458,132.30

Cuota fija (\$) 85,952.92

Impuesto (%) 30.00

BREAK

Case 9:

PRINT Base (\$) 874,650.01

Cuota fija (\$) 210,908.23

Impuesto (%) 32.00

BREAK

Case 10:

PRINT Base (\$) 1,166,200.01

Cuota fija (\$) 304,204.21

Impuesto (%) 34.00

BREAK

Case 11:

PRINT Base (\$) 3,498,600.01

Cuota fija (\$) 1,097,220.21

Impuesto (%) 35.00

BREAK

DEFAULT "Nivel no válido"

END SWITCH

IF ingreso > base

%impuesto= $y - \text{base} * (\% \text{impuesto} / 100)$

impuesto= %impuesto + cuota fija

PRINT "Impuesto a pagar"

ELSE:

PRINT "Ingreso no válido con el nivel"

END IF

FIN

VERIFICAR:

- (1,5000) ----→ 96
- (7,8000) ----→ "Ingreso no válido"
- (12,5000000) -----→ "Nivel no válido"

CONCLUSIÓN.

Podemos concluir que al realizar un pseudocódigo se llegará a una solución del problema, siendo muy útiles para cuestiones financieras, en problemas matemáticos, entre otros. Además, sirven para que el ser humano sea capaz de comprender por medio de "comandos" similares a las de las computadoras, usando códigos fáciles de recordar; como BEGIN, END, IF, BREAK, entre otros.