



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s): 5

Integrante(s): Rivera Sosa Arlethe

No. de Equipo de cómputo empleado:

No. de Lista o Brigada: 317083033

Semestre: 2020-1

Fecha de entrega: 16/septiembre/19

Observaciones:

CALIFICACIÓN: _____

Pseudocódigo

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.

Su principal característica es la de representar un método que facilita la programación y solución del algoritmo del programa. También se caracteriza por ser una forma de representación, fácil de utilizar y de manipular, que simplifica el paso del programa, al lenguaje de programación. Otra característica que tiene el pseudocódigo es su independencia al código en el que se va a escribir el programa, proporcionando un método que facilita la posterior programación y la resolución del algoritmo del programa.

El pseudocódigo llega donde el diagrama de flujo no lo hace. La solución de un diagrama de flujo suele ser la ideal, pero no suele ser fácil de implementar al crear el programa. El pseudocódigo permite que el diseño del programa y su implementación sean muy parecidos.

El pseudocódigo, al ser independiente del lenguaje de programación, permite que su uso se pueda aplicar utilizando diferentes lenguajes y permitiendo que el programador no tenga que ser la misma persona que escribió el pseudocódigo. Una de las desventajas del uso de pseudocódigo es la falta de normas, que puede hacer que la lógica de un programa resulte complicada de ver por el programador que va a implementar este pseudocódigo. Además, en el caso de problemas muy extensos, puede llegar a ser difícil de entender.

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

Actividad1: Desarrollar pseudocódigo que reciba un número obtenga su factorial.

```

INICIO
N:INT
READ N
CONTADOR:INT
MULTIPLICADOR:INT
RESULTADO:INT
RESULTADO=N
CONTADOR=1
    MULTIPLICADOR= N-CONTADOR
    RESULTADO= RESULTADO * MULTIPLICADOR
    CONTADOR= CONTADOR + 1
DO:
IF MULTIPLICADOR > 1
    MULTIPLICADOR= N-CONTADOR
    RESULTADO= RESULTADO * MULTIPLICADOR
    CONTADOR= CONTADOR + 1
    MULTIPLICADOR > 1
ELSE MULTIPLICADOR < 0
    PRINT "ERROR"
WHILE:
    MULTIPLICADOR > 1
END IF
PRINT: RESULTADO
FIN

```

Verificar el algoritmo con los valores:

- 0
- ```

INICIO
N:INT
READ 0
CONTADOR:INT
MULTIPLICADOR:INT
RESULTADO:INT

```

```

RESULTADO=0
CONTADOR=1
 MULTIPLICADOR= 0-1
 RESULTADO= 0* (0-1)
 CONTADOR=0 + 1
DO:
IF MULTIPLICADOR > 1
MULTIPLICADOR= N-CONTADOR
RESULTADO= RESULTADO * MULTIPLICADOR
CONTADOR= CONTADOR + 1
 MULTIPLICADOR > 1
ELSE MULTIPLICADOR < 0
 PRINT "ERROR"
WHILE:
 1 > 1
END IF
PRINT: 1
FIN

```

- 2
 

```

INICIO
N:INT
READ 2
CONTADOR:INT
MULTIPLICADOR:INT
RESULTADO:INT
 RESULTADO=2
 CONTADOR=1
 MULTIPLICADOR= 2-1
 RESULTADO= 2* 1
 CONTADOR= 0+ 1
DO:
IF 1 > 1
MULTIPLICADOR= 2-1
RESULTADO= 2*1
CONTADOR= 0+ 1
 2 > 1
ELSE MULTIPLICADOR < 0
 PRINT "ERROR"
WHILE:
 MULTIPLICADOR > 1
END IF
PRINT: 2
FIN

```

- -4  
 INICIO  
 N:INT  
 READ -4  
 CONTADOR:INT  
 MULTIPLICADOR:INT  
 RESULTADO:INT  
 RESULTADO=-4  
 CONTADOR=1  
 MULTIPLICADOR= -4-1  
 RESULTADO= -4\* -5  
 CONTADOR= 0+ 1  
 DO:  
 IF MULTIPLICADOR > 1  
 MULTIPLICADOR= -5-CONTADOR  
 RESULTADO= RESULTADO \* MULTIPLICADOR  
 CONTADOR= CONTADOR + 1  
 MULTIPLICADOR > 1  
 ELSE -5< 0  
 PRINT "ERROR"  
 WHILE:  
 MULTIPLICADOR > 1  
 END IF  
 PRINT: ERROR  
 FIN

- 5  
 INICIO  
 N:INT  
 READ 5  
 CONTADOR:INT  
 MULTIPLICADOR:INT  
 RESULTADO:INT  
 RESULTADO=5  
 CONTADOR=1  
 MULTIPLICADOR= 5-1  
 RESULTADO= 5\* 4  
 CONTADOR= 1 + 1  
 DO:  
 IF 5 > 1  
 MULTIPLICADOR= 5-2  
 RESULTADO= 5\* 3  
 CONTADOR= 2+ 1  
 5> 1  
 IF 5 > 1

```

MULTIPLICADOR= 5-3
RESULTADO= 5* 2
CONTADOR= 3+ 1
 5> 1
IF 5 > 1
MULTIPLICADOR= 5-4
RESULTADO= 5* 2
CONTADOR= 4+ 1
 5> 1
IF 5 > 1
MULTIPLICADOR= 5-5
RESULTADO= 5* 4
CONTADOR= 4+ 1
 5> 1
ELSE MULTIPLICADOR < 0
 PRINT "ERROR"
WHILE:
 5 > 1
END IF
PRINT: 120
FIN

```

**Actividad 2:** Desarrollar un pseudocódigo que lea 2 datos, nivel e ingreso. El programa debe:

- Verificar que no se tiene un nivel mayor al ingreso (el ingreso debe ser mayor que la base)
- Mostrar el impuesto a pagar

INICIO

a: nivel

b: ingreso

READ a,b

IF a == nivel correcto

ELSE a != nivel correcto

PRINT "ERROR"

IF base < ingreso

Pago: ingreso – base

Impuesto: pago \* porcentaje

Final: impuesto + cuota fija

ELSE base > ingreso

PRINT "ERROR"

END IF

PRINT impuesto a pagar

FIN

Verificar el algoritmo con los pares:

- (1,5000)  
INICIO  
a: nivel  
b: ingreso  
READ a,b  
IF 1 == nivel correcto  
ELSE a != nivel correcto  
PRINT "ERROR"  
IF 0 < 5000  
Pago: 5000– 0  
Impuesto: 5000 \* 1.96%  
Final: 96 + 0  
ELSE base > ingreso  
PRINT "ERROR"  
END IF  
PRINT 96  
FIN
- (7,8000)  
INICIO  
a: nivel  
b: ingreso  
READ a,b  
IF a == nivel correcto  
ELSE a != nivel correcto  
PRINT "ERROR"  
FIN

- (12,5000000)

```
INICIO
a: nivel
b: ingreso
READ a,b
 IF 12 == nivel correcto
 ELSE a != nivel correcto
PRINT "ERROR"
 IF 3,498,600.01 < 5,000,000
Pago: 5000000 – 3,498,600.01
Impuesto: 1,501,399.99* 35%
Final: 525,489.99+ 1,097,220.21
 ELSE base > ingreso
PRINT "ERROR"
 END IF
PRINT 1,622,710.20
FIN
```

Conclusión: el pseudocódigo puede tener un lenguaje más sencillo lo que nos ayuda a entender de mejor manera un programa, al tener un lenguaje menos complejo podemos llevar acabo de manera exitosa la tarea deseada. Con más orden haciendo que nuestro pseudocódigo sea lo más parecido a nuestro programa, lo que comparando al diagrama d flujo no hace tan bien.