

Contenido

[**Objetivo.** 3](#_Toc525601804)

[Introducción 3](#_Toc525601805)

[Desarrollo 8](#_Toc525601806)

[Calculadora de 4 operaciones basicas. 8](#_Toc525601807)

[Tablas de multiplicar. 9](#_Toc525601808)

[Menú deportes 10](#_Toc525601809)

[**Conclusiones.** 11](#_Toc525601810)

[**Bibliografías** 11](#_Toc525601811)

# **Objetivo.**

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

# Introducción

En ciencias de la computación, y análisis numérico, el pseudocódigo (o falso lenguaje) es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

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.

Utiliza las convenciones estructurales de un lenguaje de programación real, pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación. Normalmente, el pseudocódigo omite detalles que no son esenciales para la comprensión humana del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas subrutinas.

El lenguaje de programación se complementa, donde sea conveniente, con descripciones detalladas en lenguaje natural, o con notación matemática compacta. Se utiliza pseudocódigo pues este es más fácil de entender para las personas que el código del lenguaje de programación convencional, ya que es una descripción eficiente y con un entorno independiente de los principios fundamentales de un algoritmo. Se utiliza comúnmente en los libros de texto y publicaciones científicas que se documentan varios algoritmos, y también en la planificación del desarrollo de programas informáticos, para esbozar la estructura del programa antes de realizar la efectiva codificación.

No existe una sintaxis estándar para el pseudocódigo, aunque los ocho IDE's que manejan pseudocódigo tengan su sintaxis propia. Aunque sea parecido, el pseudocódigo no debe confundirse con los programas esqueleto que incluyen código ficticio, que pueden ser compilados sin errores. Los diagramas de flujo y UML pueden ser considerados como una alternativa gráfica al pseudocódigo, aunque sean más amplios en papel.

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.

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 escrita

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).

Que es un pseudocódigo?

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).

El pseudocódigo tiene como características

* inicio/Fin
* Sangría
* Palabras reservadas
* No hay numeración

Tipos de datos

Entero:

Booleano:

Float:

Char:

Estructuras secuencial estructuras reservadas Leer y escribir

|  |  |
| --- | --- |
| Diagrama | Pseudocódigo |
|  | a:=b+c |
| a¡=b | a<>b |

Notación de camello (CamelCase).

CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello. El nombre CamelCase se podría traducir como Mayúsculas/Minúsculas.

promInicial

promFinal

Ejemplo:

sumaValorEntero

Estructuras de control de flujo

* 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.

INICIO

X: REAL

X:= 5.8

X:=X\*2

FIN

* 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:

SI condición ENTONCES

[Acción]

FIN SI

* La estructura condicional completa es SI-DE LO CONTRARIO:

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 seudocódigo sigue su flujo normal.

SI cond\_Booleana ENTONCES

[Acciones SI]

FIN SI

De lo contrario

[Acciones DE LO CONTRARIO]

FIN DE LO CONTRARIO

INICIO  
          a,b:ENTERO  
  
         a := 3  
          b := 5  
          SI a > b ENTONCES

             ESCRIBIR "a es mayor"  
          FIN SI

        DE LO CONTRARIO  
               ESCRIBIR "b es mayor"

        FIN DE LO CONTRARIO  
     FIN

   // >>> b es mayor

Ejercicio de menú Helados, elaborar  Pseudocódigo

Inicio

  x:Entero

  x:=1

  Escribir “Escoge un helado 1=Limón 2=fresa 3=champurrado”;

  Leer x

    seleccionar (x) en

        caso 1 ->

           Escribir “elegiste Limón”;

        caso 2 ->

           Escribir “Elegiste fresa”;

        caso 3 ->

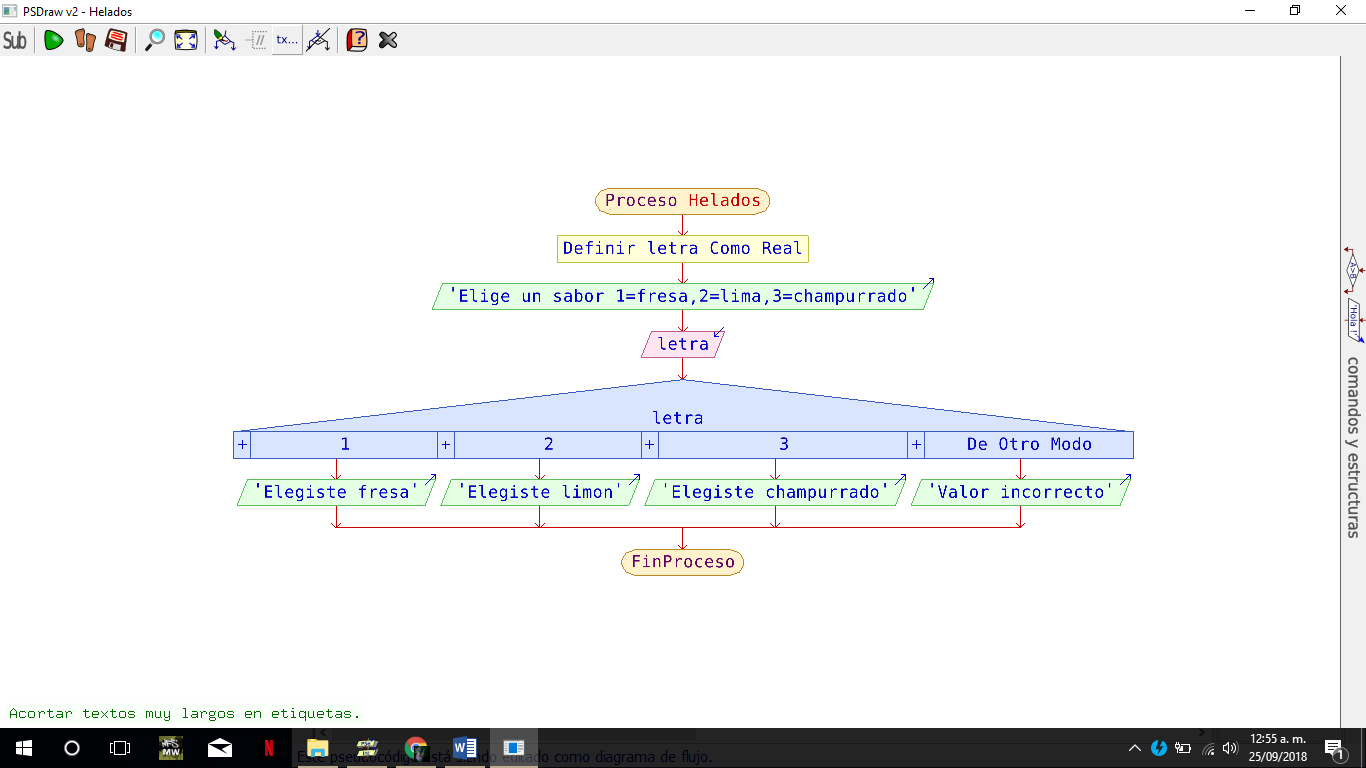
           Escribir “Elegiste champurrado”;

        caso 4 ->

           Escribir “Valor incorrecto”;

    Fin Seleccionar

Fin



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 esta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, de lo contrario rompe el ciclo y continua el flujo normal del seudocódigo.

MIENTRAS condición ENTONCES  
          [Acción]

FIN MIENTRAS

Hacer mientras: 1.-ejecuta (por lo menos una vez) y 2.- pregunta o pone condiciòn

Mientras: 1.-Pregunta/ pone condiciòn y 2.-ejecuta.

# Desarrollo

# Calculadora de 4 operaciones basicas.

Pseudocódigo

Calculadora

SubProceso multi <- multinumeros ( a,b )

multi<-a\*b

Fin SubProceso

SubProceso division <- divisionnumeros ( a,b )

division<-a/b

Fin SubProceso

SubProceso resta <- restanumeros ( a,b )

resta<-a-b

Fin SubProceso

SubProceso suma <- sumanumeros (a,b)

suma<-a+b

Fin SubProceso

Proceso calculadora\_basica\_4op

a: real

b: real

Repetir

Escribir "Digita 2 numeros para realizar 4 operaciones basicas";

Escribir "Segundo numero no puede ser menor a 0";

Leer a,b;

Hasta Que b>0;

suma<-sumanumeros(a,b)

Escribir "La suma de " a "+" b " es " sumanumeros(a,b);

resta<-restanumeros (a,b);

Escribir "La resta de " a "-" b " es " restanumeros(a,b);

multi<-multinumeros(a,b);

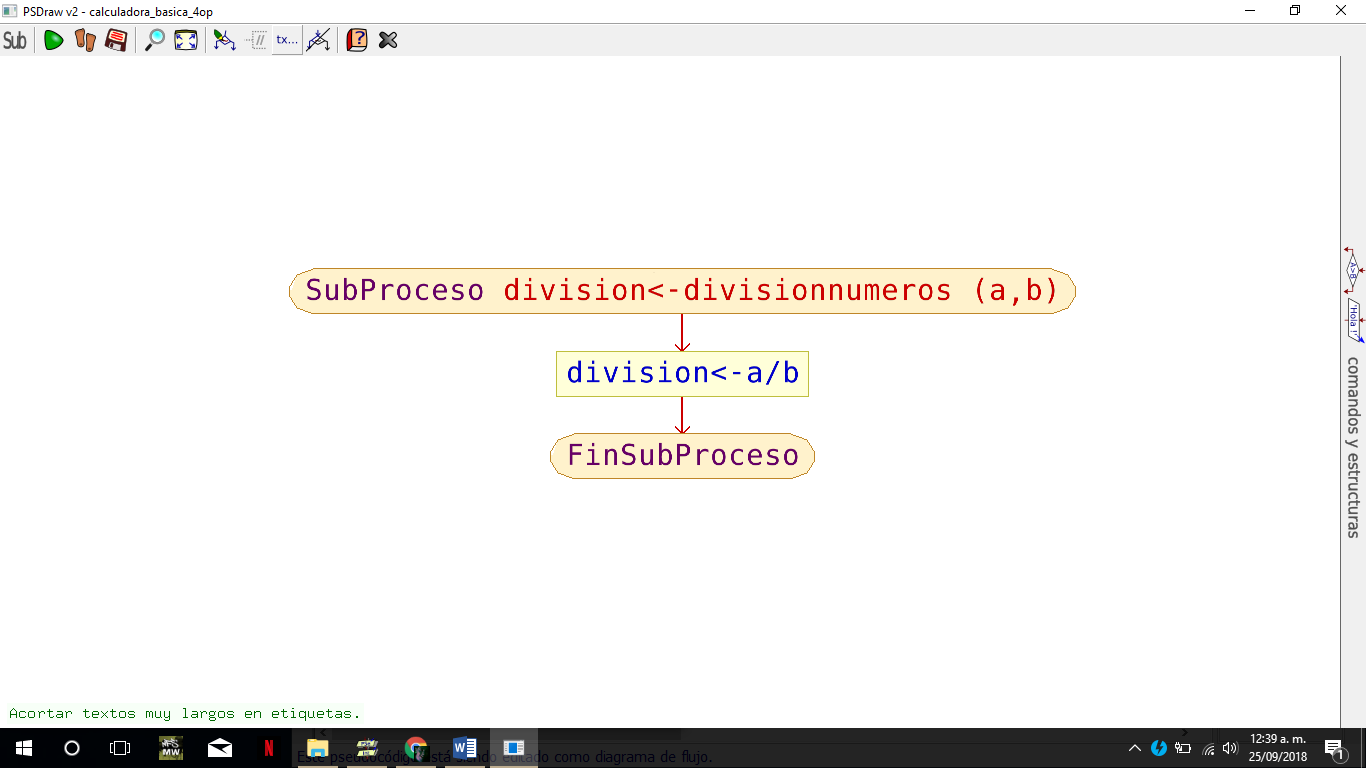
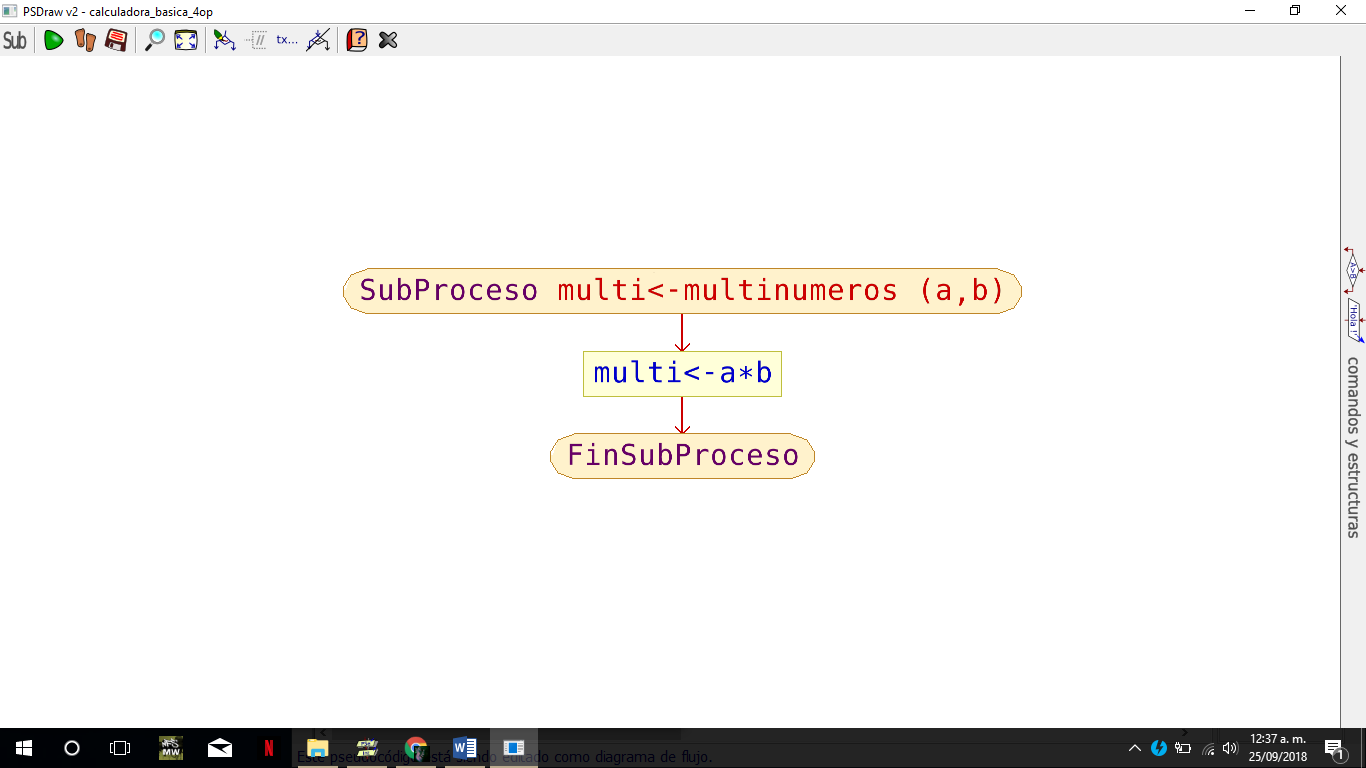
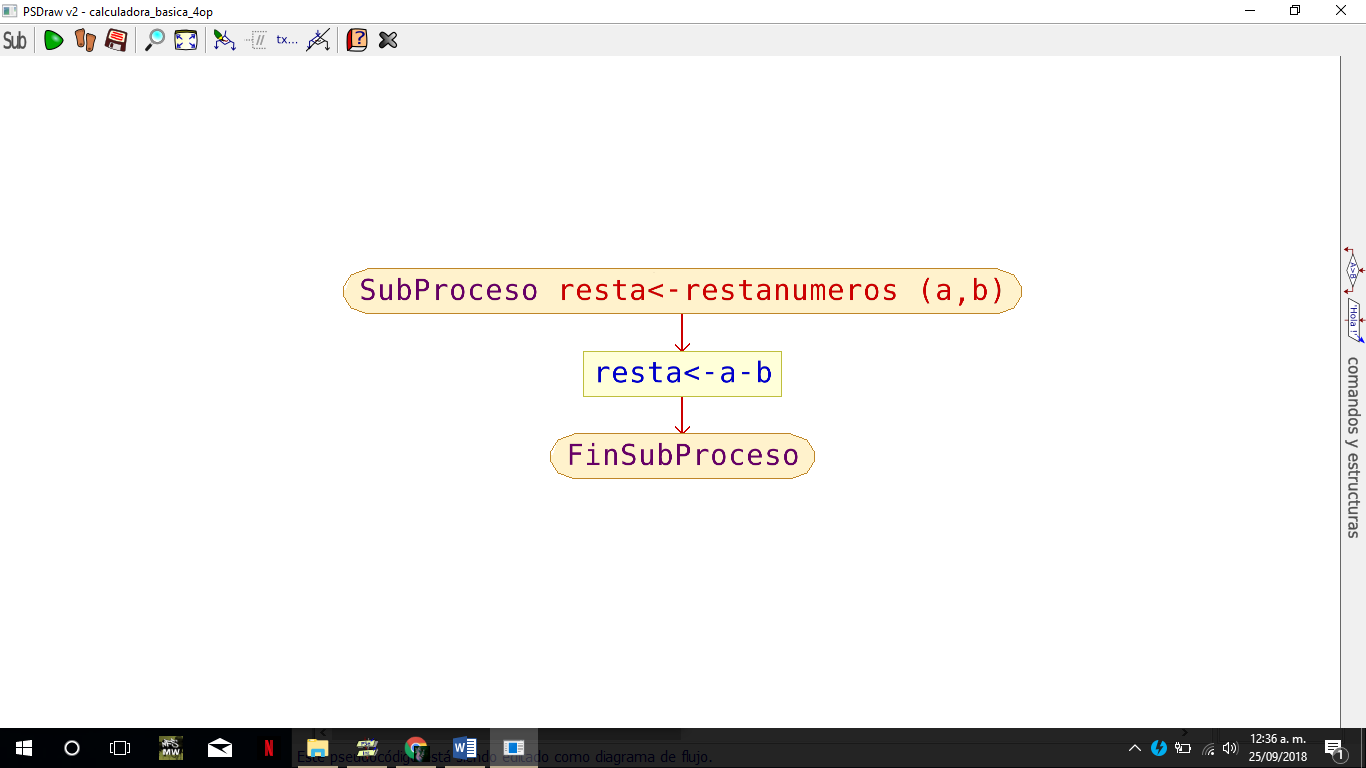
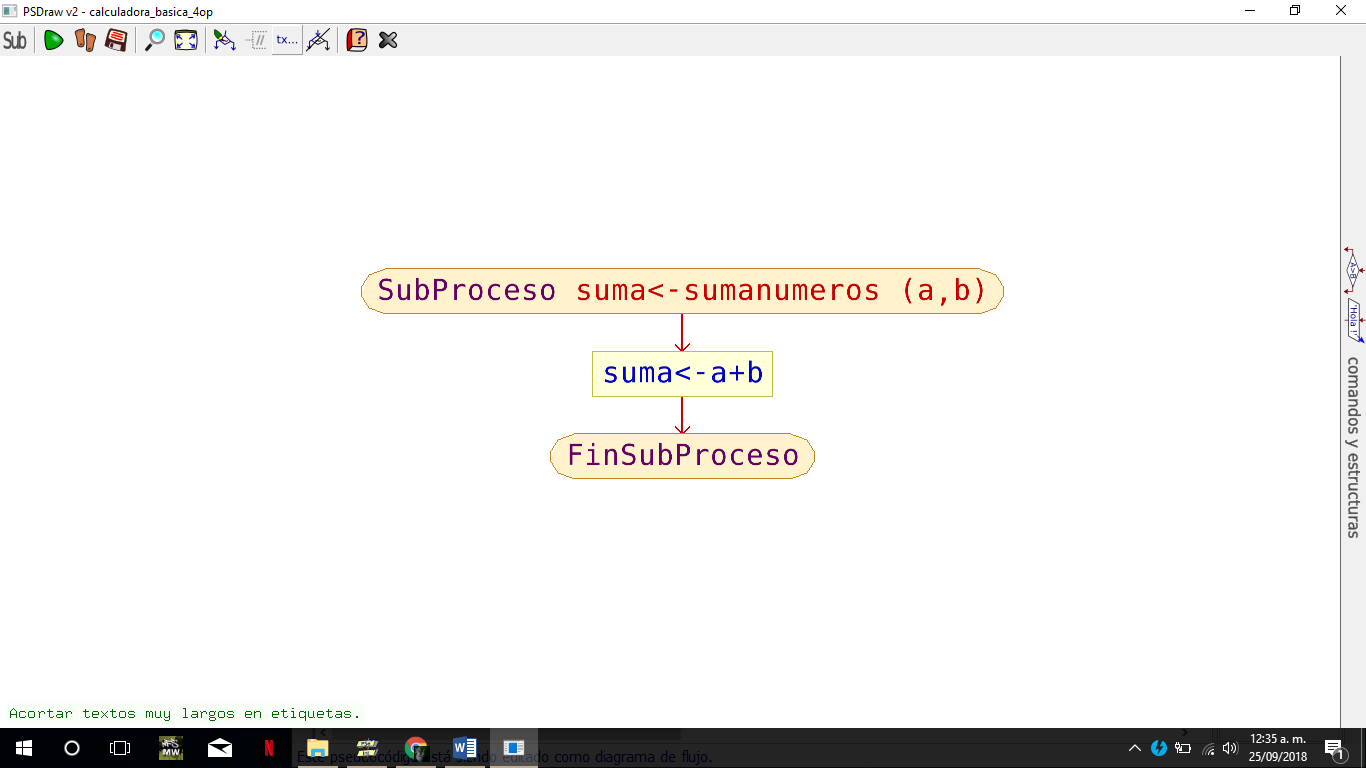
Escribir "La multiplicacion de " a "X" b " es" multinumeros(a,b)

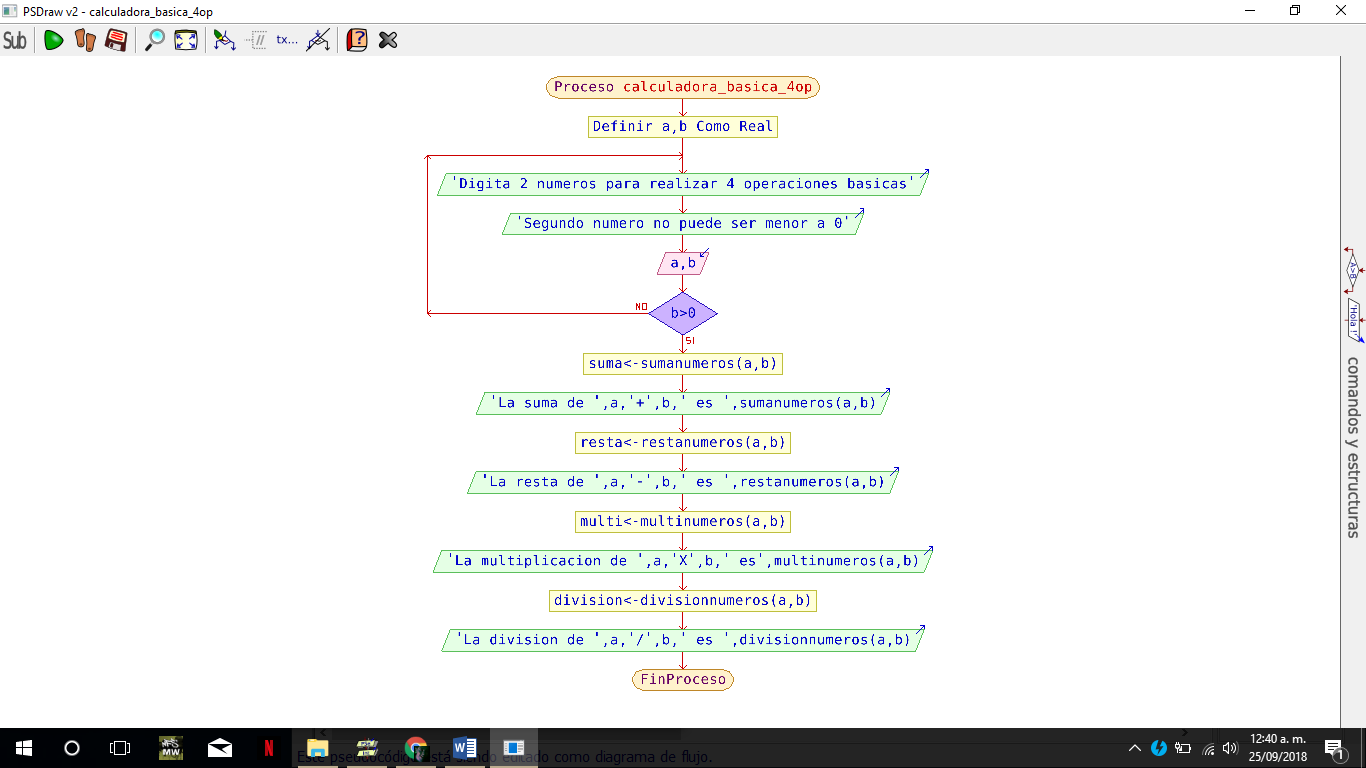
division <- divisionnumeros ( a,b )

Escribir "La division de " a "/" b " es " divisionnumeros(a,b)

FinProceso

Diagrama de flujo.





# Tablas de multiplicar.

Pseudocódigo

Proceso tablas

x:Entero

r:Entero

Escribir "Ingresa el numero a multiplicar";

Leer x;

Para cont<-1 Hasta 10 Con Paso 1 Hacer

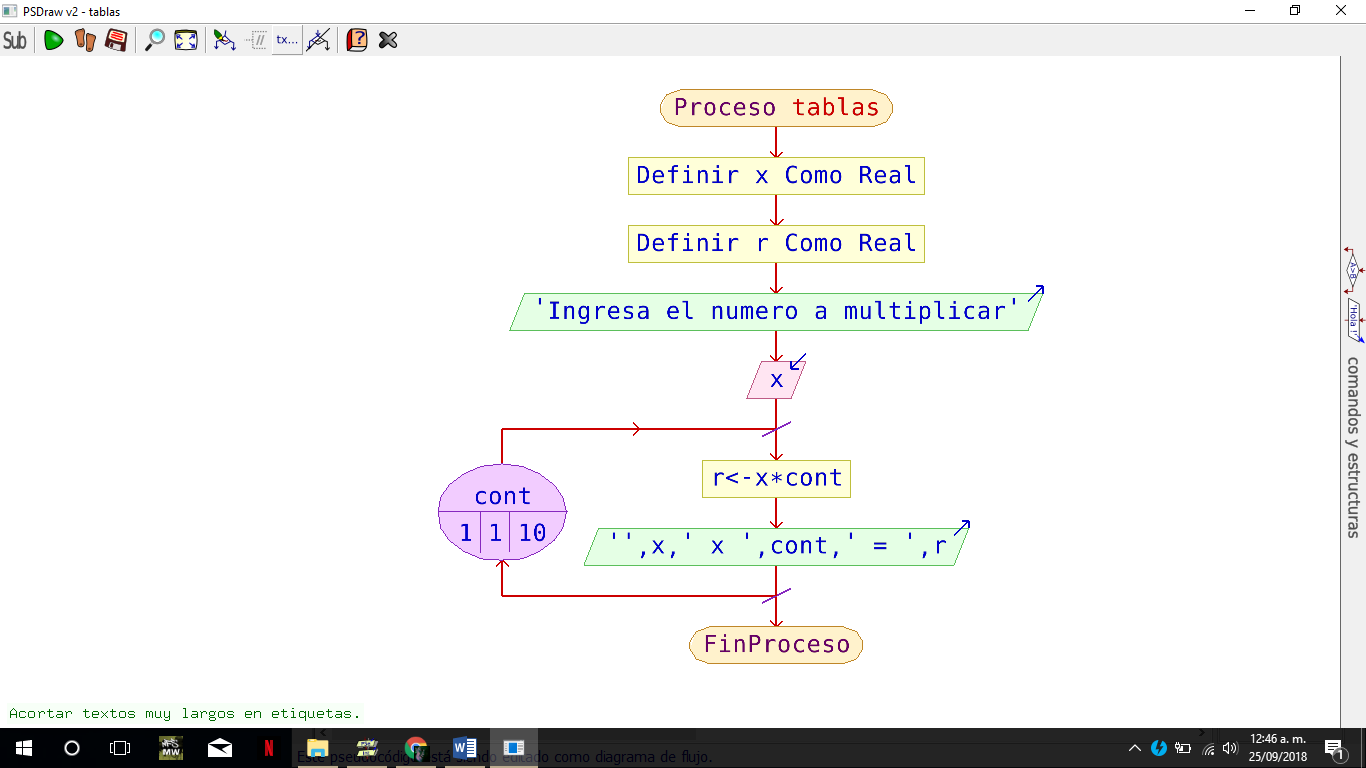
r = x\*cont;

Escribir "",x " x " ,cont, " = ",r;

FinPara

FinProceso

Diagrama de flujo



# Menú deportes

Pseudocódigo

Proceso deportes

Var: x :Entero

Escribir 'Elige un deporte: 1=futbol 2=basquetbol 3= Karate'

Leer x

Segun x Hacer

1:

Escribir “Elegiste Futbol :D”;

2:

Escribir “Elegiste Basquetbol :D”;

3:

Escribir “Elegiste Karate :D”;

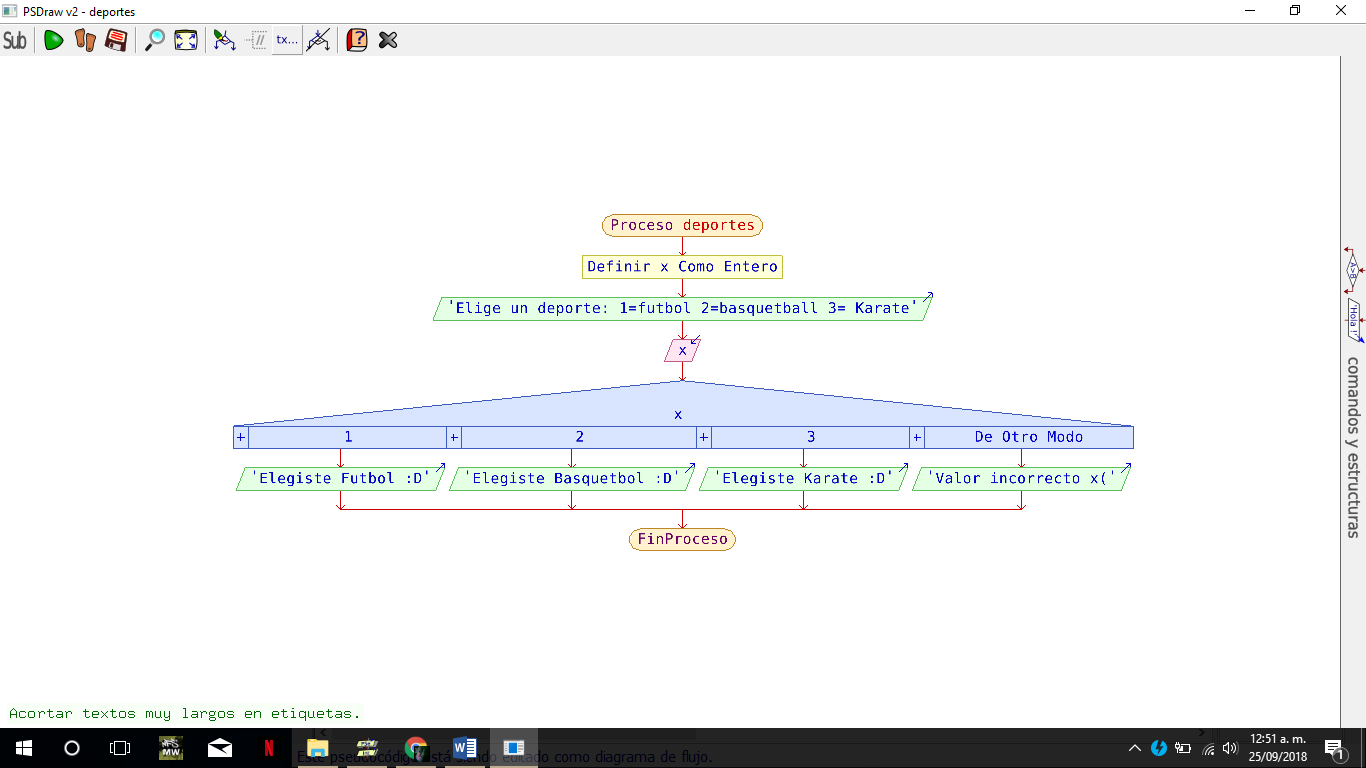
De Otro Modo:

Escribir “Valor incorrecto x(“;

FinSegun

FinProceso

Diagrama de flujo



**Conclusiones.**

Los pseudocódigos además de ser una buena herramienta, facilitan la resolución pero es un poco difícil.

**Bibliografías**

* Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.
* <http://profesores.fi-b.unam.mx/tanya/CPI/Presentaciones/Tema-V.pdf>
* http://uapas1.bunam.unam.mx/matematicas/pseudocodigo/