

TUTORIAL SENCILLO MANEJO DE PSEINT Y FREE DFD

Resumen

El siguiente tutorial muestra de manera sencilla como manejar el PSEINT, para ello se toma un ejemplo resuelto del libro del libro de ude@, se modifiica un poco el pseudocódigo del libro y se muestra como llevarlo en PSEINT.

Repaso teórico resumido

Cuando nos enfrentamos a un problema en la vida cotidiana, su resolución requiere que sigamos una serie de pasos; por ejemplo, un problema típico puede ser **terminarle a la novia**, para tal fin, se requieren realizar una serie de pasos los cuales se muestran a continuación:

- **Paso 1:** Llamar a la novia y decirle que es solicitada para hablarle de algo importante.
- **Paso 2:** Ponerle una hora de encuentro, 6:00 PM por ejemplo, eso para que por lo menos llegue a las 7:00 PM.
- **Paso 3:** Si no ha llegado a las 7:00 tomar cerveza, pero si llega poner cara seria.
...
- **Paso N:** ...

Folclóricamente hablando, el conjunto ordenado de pasos (anteriormente descritos) seguidos con el fin de resolver un problema o lograr un objetivo es conocido como algoritmo.

Mas formalmente, un algoritmo es un conjunto de instrucciones que especifica la secuencia de operaciones a realizar, en orden, para resolver un problema específico; en otras palabras, un algoritmo **es una fórmula para la resolución de un problema**.

La definición de un algoritmo debe describir tres partes: Entrada, Proceso y Salida, asi:

- **Entrada:** Información dada al algoritmo, o conjunto de instrucciones que generen los valores con que ha de trabajar.
- **Proceso:** Cálculos necesarios para que a partir de un dato de entrada se llegue a los resultados.
- **Salida:** Resultados finales o transformación que ha sufrido la información de entrada a través del proceso.



Por ejemplo, un ejemplo típico es una receta de cocina (Por ejemplo la receta para hacer una bandeja paisa), tomando la receta como un algoritmo (pues describe una secuencia de pasos), las partes asociadas a esta son:

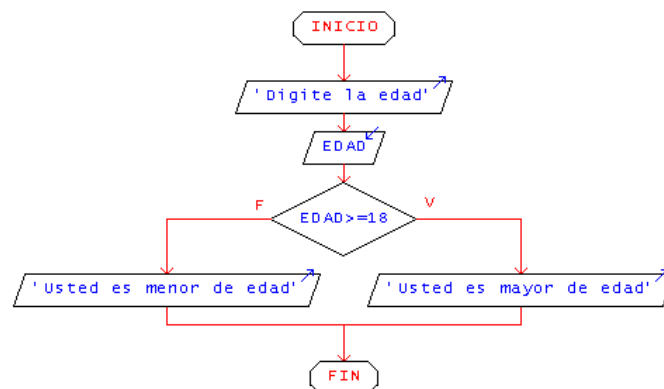
- **Entrada:** Ingredientes y utensilios empleados.
- **Proceso:** Elaboración de la receta en la cocina.
- **Salida:** Terminación del plato.

Cuando se formula un algoritmo el objetivo es ejecutar este en un computador, sin embargo, para que este entienda los pasos para llevar a cabo nuestro algoritmo debemos indicárselo siguiendo un conjunto de instrucciones y reglas que este entienda, y estas instrucciones son abstraídas en lo que conocemos como **lenguaje de programación**.

Un algoritmo codificado siguiendo un lenguaje de programación es conocido como **programa**. Antes de aprender un lenguaje de programación es necesario aprender la metodología de programación, es decir la estrategia necesaria para resolver problemas mediante programas.

Como punto de partida se aborda la manera como es representado un algoritmo. Basicamente analizamos dos formas, la representación usando **pseudocódigo** y la representación usando **diagramas de flujo**.

Un **diagrama de flujo** es un diagrama que utiliza símbolos (cajas) estándar y que tiene los pasos del algoritmo escritos en esas cajas unidas por flechas, denominadas líneas de flujo, que indican la secuencia que debe ejecutar el algoritmo



Por otro lado, el **pseudocódigo** es un lenguaje de especificación (descripción) de algoritmos. El uso de tal lenguaje hace el paso de codificación final (traducción al lenguaje de programación) relativamente fácil, por lo que este es considerado un primer borrador de la solución del programa.

```
Proceso pseudocodigo
  Escribir "Digite la edad";
  Leer edad;
  Si edad >= 18 Entonces
  ..... Escribir "Usted es mayor de edad";
  Sino
  ..... Escribir "Usted es menor de edad";
  FinSi
FinProceso
```

Pseudocódigo en pocas palabras

Como habíamos dicho antes, el pseudocódigo es un lenguaje de descripción de algoritmos por lo que un primer paso consiste en familiarizarnos con este lenguaje. Como punto de partida tenemos que tener en cuenta lo siguiente:

- **Anatomía de un algoritmo:** Un algoritmo es finito por lo que tiene un principio y un fin. La siguiente plantilla muestra la forma básica de un algoritmo:

```
Algoritmo(Nombre_algoritmo)
  Declaracion_de_variables
  INICIO
    INSTRUCCIÓN_1
    INSTRUCCIÓN_2
    INSTRUCCIÓN_3
    .
    .
    INSTRUCCIÓN_N
  FIN_INICIO
Fin(Nombre_algoritmo)
```

- **Las variables:** Cuando nos referimos a variables nos referimos a lugares de memoria en los cuales se almacena algún tipo de información, por ejemplo el numero de gallinas, la altura, la edad, el nombre y el peso. Existen diferentes tipos de datos tal y como se muestra en la siguiente tabla:

Tipo de dato	Descripción	Ejemplo
entero	Tipo de dato asociado a cantidades enteras. No poseen parte decimal. Ejemplo: 5, 6, -15, 199,...	Numero de vacas, edad.
real	Tipo de dato asociado a cantidades con parte decimal. Por ejemplo: 0.06, -3.4, 2.16, 1000.345,...	Estatura, peso, volumen.
lógicos	Se refiere a aquellos datos que pueden tomar solo dos posibles valores falso (F) o verdadero (T)	
alfanuméricos	Asociado a aquellos datos que contienen caracteres alfanuméricos (letras, número, signos de puntuación, etc).	Nombre, cedula, telefono

Cuando se trabaja con variables un aspecto de vital importancia (además del tipo) es el nombre que estas van a tener, se recomiendan nombres relacionados con la información que van a manejar, por ejemplo, si se va a manejar un **salario**, un nombre apropiado para una variable puede ser **sal**.

- **Instrucción de asignación:** Escribe sobre una variable el valor de una expresión. Así:

variable = expresión

Donde, una expresión es una combinación de valores, variables y operadores, los siguientes son algunos ejemplos de expresiones:

$a = 5$
 $b = c*d + (c-f)*m$
 $z = (x+y)/(w+s)$
 $s = (a/b)^3$

Existen diferentes tipos de operadores. La siguiente tabla muestra los operadores aritméticos.

Operador	Significado
^	Potenciación
+	Suma
-	Resta
*	Multiplicación
/	División

- **Instrucciones de entrada y salida:** Para que un programa pueda interactuar con el usuario deben haber un conjunto de instrucciones que permitan especificar tal interacción, y estas son las instrucciones de entrada y salida.

- ✓ **Instrucciones de entrada:** Permite tomar uno o mas datos de un medio externo (comúnmente el teclado) y asignarlos a una o mas variables, su representación en pseudocódigo es:

LEA(var1, var2, ..., varN)

- ✓ **Instrucciones de salida:** Permite mostrar de variables y constante en un medio externo (comúnmente la pantalla). En pseudocódigo la instrucción asociada a la salida tiene la siguiente forma:

ESCRIBA(var1,var2, ..., varN)

Ejemplo 1:

Codifique un algoritmo que solicite el nombre y devuelva como salida el mensaje: **Hola nombre_ingresado**. Por ejemplo, si el usuario digita ramón, el mensaje desplegado será: **Hola ramón**.

Solución:

La codificación en Pseudocódigo del algoritmo se muestra a continuación:

```
Algoritmo (nombre)
  Variables:
    alfanumerica: nom
  INICIO
    ESCRIBA("Digite el nombre")
    LEA (nom)
    ESCRIBA("Hola ", nom)
  FIN_INICIO
Fin (sumar)
```

Ejemplo 2:

Realice un algoritmo que solicite dos números entero, realice su suma y muestre el resultado.

Solución: A continuación se muestra la codificación del problema en Pseudocódigo:

```

Algoritmo (suma)
  Variables:
    entero: a, b, c
  INICIO
    ESCRIBA("Digite el primer numero (entero)")
    LEA(a)
    ESCRIBA("Digite el segundo numero (entero)")
    LEA(b)
    c = a + b
    ESCRIBA("La suma es: ",c)
  FIN_INICIO
Fin (sumar)

```

Nota: Cuando se codifica en Pseudocódigo, la declaración de variables no es necesaria (es mas la gran mayoría de libros nunca declaran variables), sin embargo en los ejemplos anteriormente mostrados se realizo pues esta si es necesaria cuando se trabaja en un lenguaje de programación.

Diagramas de flujo en pocas palabras

El diagrama de flujo es la representación grafica del algoritmo, de modo que lo único que es necesario es conocer la equivalencia de este con el pseudocódigo. La siguiente tabla resume esto:

Instrucción	Representación en Pseudocódigo	Representación en Diagrama de flujo
Asignación	$a = b + c$	
Entrada	LEA(nom, tel)	
Salida	ESCRIBA("Hola",nom)	

Con base en lo anterior es fácil representar los dos anteriores Pseudocódigos, es casi como reemplazar en una formula. Vale notar que en el diagrama de flujos no existe la declaración de variables:

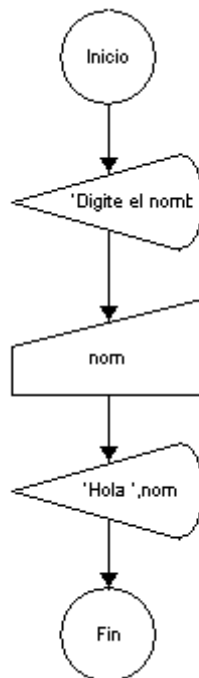
Ejemplo 3:

Realizar el diagrama de flujo del algoritmo codificado en el ejemplo 1, a continuación se muestra el Pseudocódigo asociado a este:

```
Algoritmo (nombre)
  Variables:
    alfanumerica: nom
  INICIO
    ESCRIBA("Digite el nombre")
    LEA (nom)
    ESCRIBA("Hola ", nom)
  FIN_INICIO
Fin (sumar)
```

Solución:

Como el diagrama de flujo no posee declaración e variables, el diagrama de flujo equivalente se toma de las instrucciones que se encuentran entre las sentencias INICIO y FIN_INICIO.



Ejemplo 4:

Realizar el diagrama de flujo del algoritmo codificado en el ejemplo 2, a continuación se muestra el Pseudocódigo asociado a este:

```
Algoritmo (suma)
  Variables:
    entero: a, b, c
  INICIO
    ESCRIBA("Digite el primer numero (entero)")
    LEA (a)
    ESCRIBA("Digite el segundo numero (entero)")
    LEA (b)
    c = a + b
    ESCRIBA("La suma es: ", c)
  FIN_INICIO
Fin (sumar)
```

Solución: El diagrama de flujo codifica las instrucciones que se encuentran entre **INICIO** y **FIN_INICIO**, así:



Enunciado del problema

Ejemplo resuelto

Realizar un algoritmo que calcule el perímetro y el arrea de un rectángulo dadas la base y la altura del mismo.

Solución en Pseudocodigo del libro

A continuación se detalla la solución en Pseudocodigo tomada del libro, esta se modifico un poco respecto a la original.

Algoritmo (ejemplo1)

Variables:

real: area, perímetro, base = 0, altura = 0

INICIO

ESCRIBA("Digite la base y la altura del rectángulo")

LEA(base,altura)

```

area = base*altura
perímetro = 2*(base + altura)
ESCRIBA("El rectanculo cuya base es ", base, "y cuya altura
        es ", altura, " tiene un perímetro de ",perímetro,
        " y una altura de ", altura)
FIN_INICIO
Fin(aumento_con_condicion)

```

Codificación en PSEINT

Ya se tiene el pseudocódigo del programa codificado en nuestro pseudocódigo por convención sin embargo la idea es probarlo para ver cómo funciona. Existe una herramienta llamada PSeint el cual es un pseudointerprete, es decir entiende Pseudocódigo, sin embargo en el momento de codificar el anterior programa en PSeInt debemos tener en cuenta que el pseudocódigo manejado en le PSeInt es un poco diferente, la siguiente tabla muestra esto en detalle:

Instrucción	Pseudocódigo propio	Pseudocódigo PSeint	Observaciones
Asignación	$c = 2 * a * (b + c)$	$c \leftarrow 2 * a * (b + c);$	La asignación en PSeInt no es con igual (=) sino con flecha (<-) y al final va punto y coma (;).
Entrada	LEA(a,b,c)	Leer a,b,c;	La instrucción de entrada en PSeInt se llama Leer no hace uso de paréntesis y termina con signo de punto y coma (;).
Salida	ESCRIBA("Hola ",nombre)	Escribir "Hola",nombre;	La instrucción de entrada en PSeInt se llama Escribir no hace uso de paréntesis y termina con signo de punto y coma (;). Al igual que en el ESCRIBA usado por convesion la parte del mensaje que no cambia (que es constante) va entre comillas (""), y la parte variable va sin comillas ("").

Ejecución del PSeInt

Abra el programa **PSEINT (Inicio > todos los Programas > PSeInt)**. Una vez ejecutado, se debe abrir una ventana como la mostrada en la figura 1:

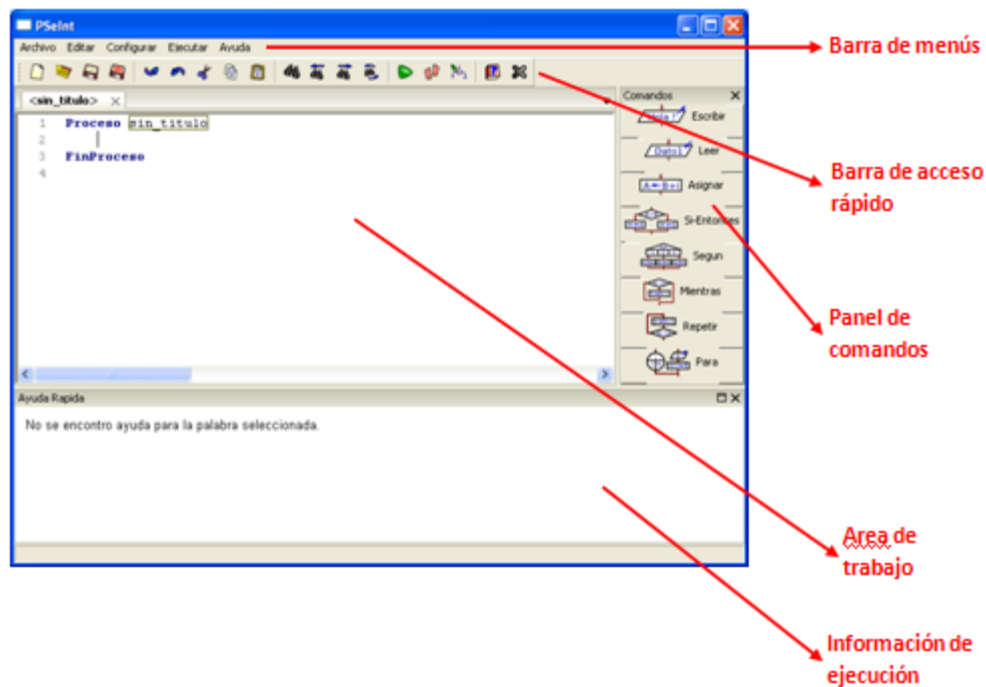


Figura 1. Ventana principal del programa PSeInt.

Codificación del algoritmo

Explore la herramienta dando click en los diferentes botones del panel de comando, observe el efecto en el área de trabajo, una vez que se haya familiarizado un poco con la herramienta intente adaptar el código mostrado en el pseudocódigo del problema anterior en el PseInt:

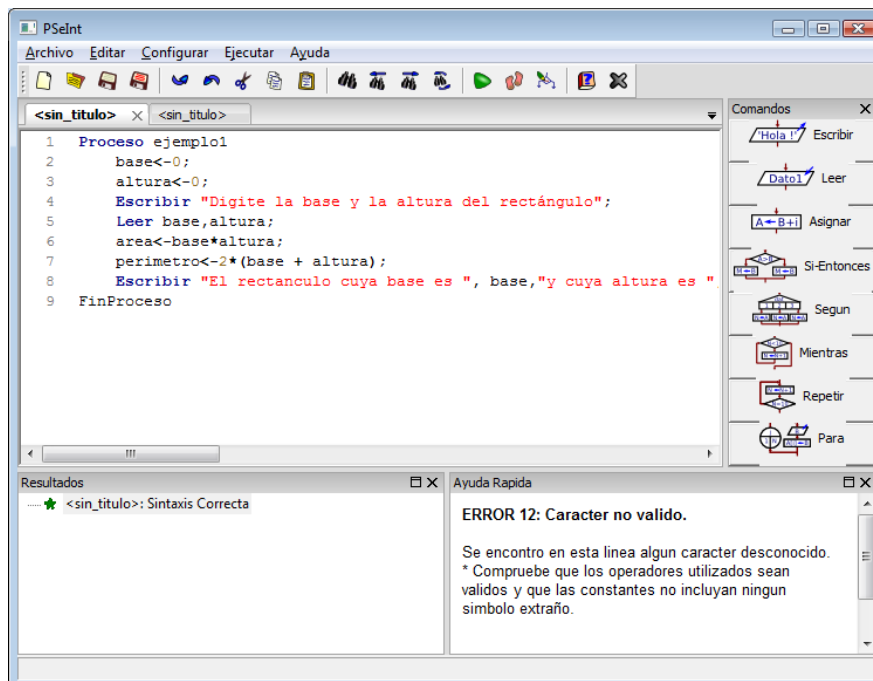


Figura 2. Adaptación del Pseudocódigo del libro al PSeInt.

Una vez codificado el pseudocódigo (ayudado de los botones del panel de comandos) en el área de trabajo guarde el archivo como **ejemplo1** En una ruta conocida.

Nota sobre la codificación en PSeInt

La siguiente figura muestra una comparación entre el Pseudocódigo convención y el Pseudocódigo del Pseint:

```
Algoritmo(aumento_con_condicion)
Variables:
  real: area, perimetro, base = 0, altura = 0
INICIO
  ESCRIBA("Digite la base y la altura del rectángulo")
  LEA(base,altura)
  area = base*altura
  perimetro = 2*(base + altura)
  ESCRIBA("El rectanculo cuya base es ", base, "y cuya altura
  es ", altura, " tiene un perimetro de ",perimetro,
  " y una altura de ", altura)
FIN_INICIO
Fin(aumento_con_condicion)
```

```
Proceso ejemplo1
base<-0;
altura<-0;
Escribir "Digite la base y la altura del rectángulo";
Leer base,altura;
area<-base*altura;
perimetro<-2*(base + altura);
Escribir "El rectanculo cuya base es ", base,
"y cuya altura es ", altura, " tiene un perimetro de ",
perimetro, " y una altura de ", altura;
FinProceso
```

Note lo siguiente:

- En el Pseint no hay declaración de variables (es decir enunciar el tipo y la lista de variables de dicho tipo, para el ejemplo en el pseudocódigo convención la declaración es: **real: area, perimetro, area = 0, altura = 0**) por lo tanto la codificación del algoritmo en PSeint empieza desde la parte de INICIO del algoritmo, sin embargo cuando una variable es inicializada esto si deba ser tenido en cuenta en el Pseint antes de iniciar la codificación del programa desde el INICIO. Como se puede notar en el pseudocódigo convención las variables base y altura están inicializadas ambas en cero ($base = 0$, $altura = 0$), por ello antes de empezar la primera instrucción después del INICIO (**ESCRIBA("Digite la base y la altura del rectángulo")**) es necesario codificar en PSeint dicha inicialización por ello las líneas **base<-0;** y **altura <-0;** antes del Escribir.
- El Pseudocódigo que se codifica es propiamente el que se encuentra entre las sentencias **INICIO** y **FIN_INICIO** (Ver parte resaltada en verde), salvo cuando se tiene que tener en cuenta la nota anterior.

Una vez realizado lo anterior obtenga el diagrama de flujo asociado al pseudocódigo para ello presione el dibujar diagrama de flujo.



Figura 3. Botón para obtener el diagrama de flujo.

Si lo anterior está bien, se generará un diagrama como el mostrado en la siguiente figura:

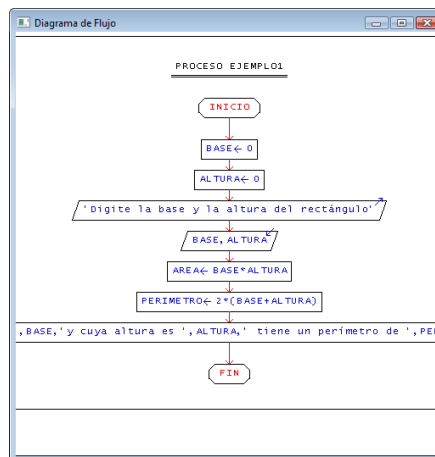


Figura 4. Diagrama de flujo del algoritmo del problema.

Guarde el diagrama de flujo anterior como una imagen *jpg* (puede serle útil después, por ejemplo para un informe).

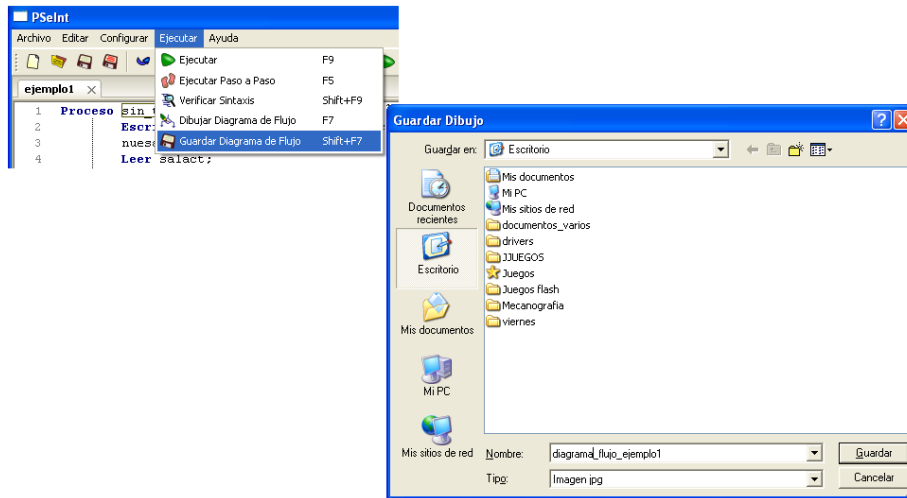


Figura 5. Guardada de la imagen.

Ejecución del algoritmo

Una vez guardado el programa anterior, proceda a realizar la prueba del algoritmo presionando el botón ejecutar.

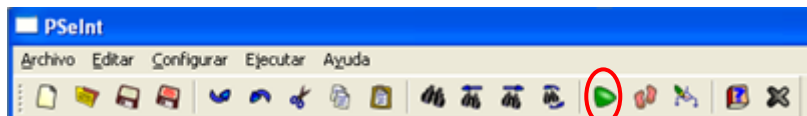


Figura 6. Botón de ejecución del algoritmo.

Deberá aparecer una ventana como la siguiente asociada al programa:

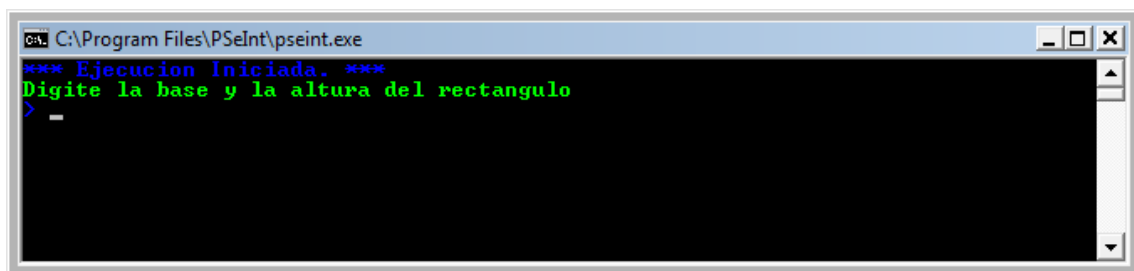
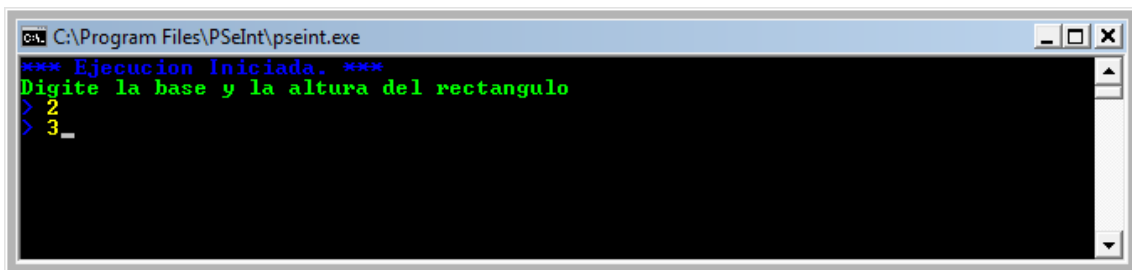


Figura 7. Ejecución del programa asociado al algoritmo (antes de ingresar el valor solicitado por teclado).

Lo anterior se debe a la instrucción **Escribir "Digite la base y la altura del rectángulo"**;

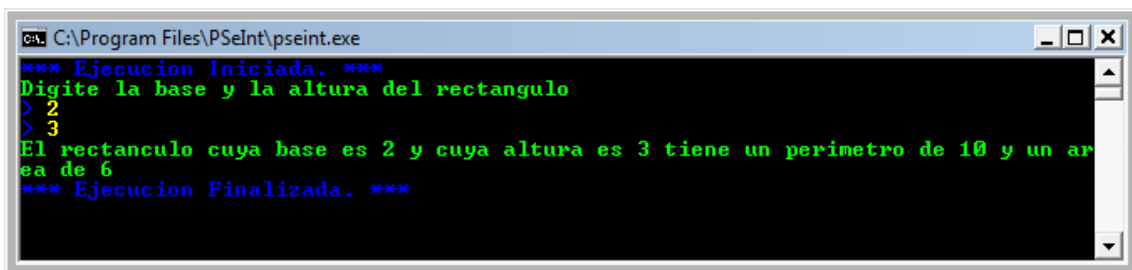
Si lo nota el cursor se queda titilando esperando a que sean introducidos los valores para la altura y la base, esto debido a la instrucción **Leer base, altura;**

Introduzca el valor de 2 como valor para la base y 3 como valor para la altura.



```
C:\Program Files\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite la base y la altura del rectangulo
> 2
> 3
_
```

Note que cada vez que introduce un valor por teclado debe presionar enter. Una vez que presione el enter después de digitar el segundo valor aparece algo como lo siguiente:



```
C:\Program Files\PSeInt\pseint.exe
*** Ejecucion Iniciada. ***
Digite la base y la altura del rectangulo
> 2
> 3
El rectangulo cuya base es 2 y cuya altura es 3 tiene un perimetro de 10 y un area de 6
*** Ejecucion Finalizada. ***
```

Figura 8. Ejecución del programa asociado al algoritmo (después de ingresar el valor solicitado por teclado).

Después de que aparece la ventana anterior si damos enter esta se cierra. Intente nuevamente ejecutar el algoritmo pero esta vez de 6 como valor para la base y 7 como valor para la altura.

Ejercicio de refuerzo

Con el fin de obtener un poco de familiaridad con el Pseint, se muestra a continuación el pseudocódigo del ejemplo 1 y su codificación en PSeInt. Codifique dicha codificación en el PSeInt, genere el diagrama de flujos y ejecute el programa.

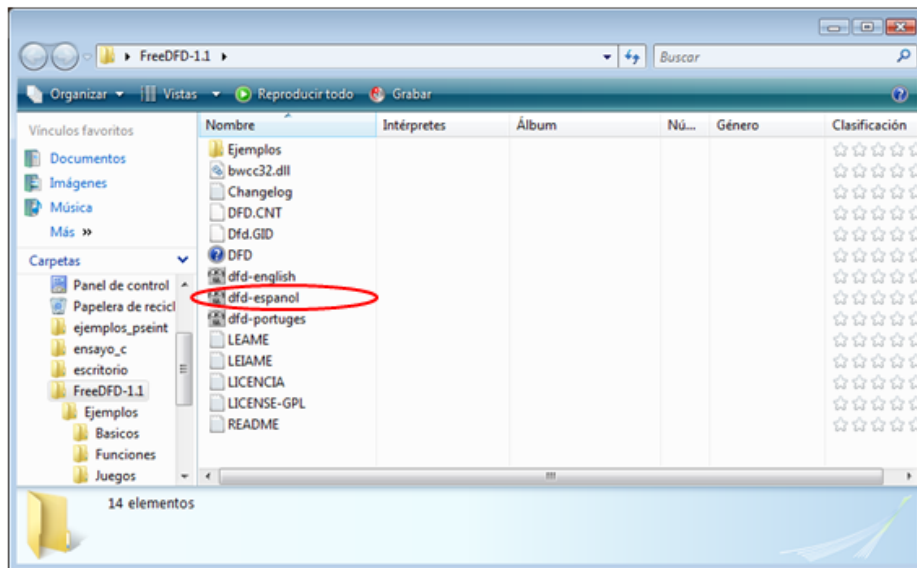
```
Algoritmo(nombre)
  Variables:
  .....
  alfanumerica: nom
  INICIO
  .....
  ESCRIBA("Digite el nombre")
  LEA(nom)
  .....
  ESCRIBA("Hola ",nom)
  FIN_INICIO
Fin(sumar)

Proceso nombre
  Escribir "Digite el nombre";
  Leer nom;
  Escribir "Hola",nom;
FinProceso
```

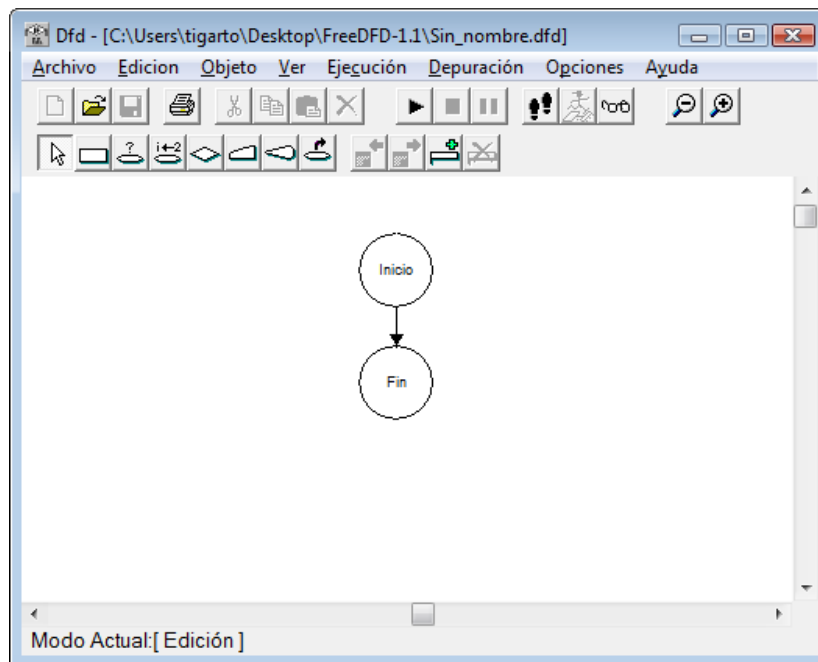
Note de la figura anterior que la codificación inicia desde el INICIO del pseudocódigo de convención. De la declaración de variables (alfanumérica: nom) no se tuvo en cuenta nada pues no hay inicialización de variable alguna.

Trabajando con el Free DFD

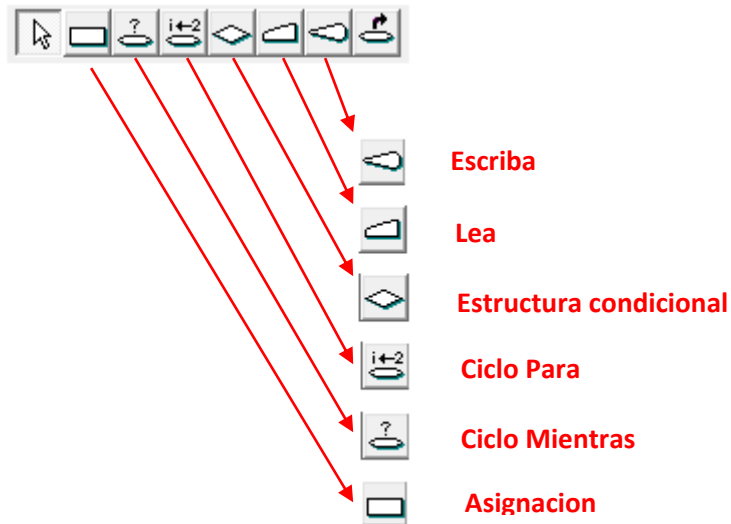
Free DFD es un programa que permite realizar diagramas de flujo, asumiendo, a diferencia del PSeInt este no cuenta con un instalador, ya viene como archivo ejecutable. Asumiendo que este se descargó en el escritorio el primer paso consiste en meternos a la carpeta de este e invocar el DFD:



Una vez que este es invocado, aparece una ventana como la siguiente:



Para ingresar los símbolos del diagrama de flujo se usan principalmente los botones mostrados a continuación:



Por el momento solo nos interesa lo básico, así que para los problemas que hemos trabajado solo vamos a hacer uso de los botones; Escriba, asignación y Lea. Retomando el Pseudocódigo del problema del area y del perímetro del rectángulo tenemos:

Algoritmo (ejemplo1)

Variables:

real: area, perímetro, base = 0, altura = 0

INICIO

ESCRIBA("Digite la base y la altura del rectángulo")

LEA(base, altura)

area = base*altura

perímetro = 2*(base + altura)

ESCRIBA("El rectanculo cuya base es ", base, "y cuya altura es ", altura, " tiene un perímetro de ", perímetro, " y una altura de ", altura)

FIN_INICIO

Fin(aumento_con_condicion)

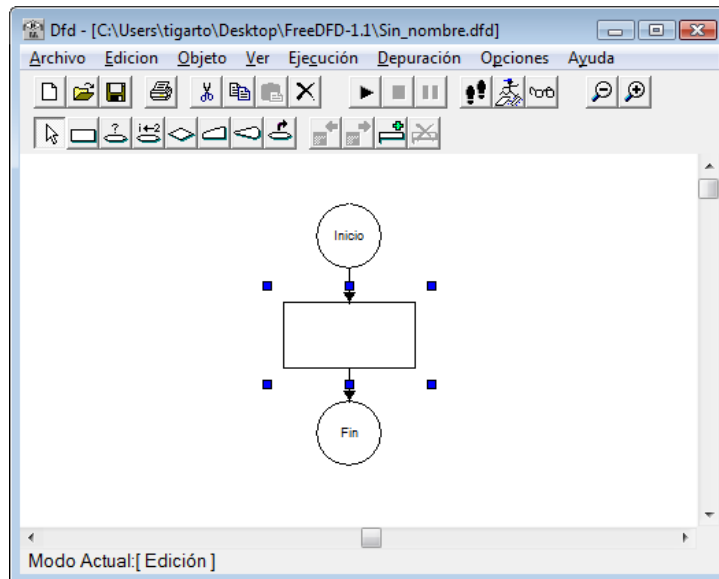
Ahora nuestro objetivo es hacer el diagrama de flujos. Al igual que en el Pselnt no se declaran variables, solo se inicializan y la codificación del programa es desde la sentencia INICIO hasta la sentencia FIN_INICIO. Empecemos:

```

Algoritmo (ejemplo1)
Variables:
  real: area, perímetro, base = 0, altura = 0
INICIO
  ESCRIBA("Digite la base y la altura del rectángulo")
  LEA(base, altura)
  area = base*altura
  perímetro = 2*(base + altura)
  ESCRIBA("El rectanculo cuya base es ", base, "y cuya altura
    es ", altura, " tiene un perímetro de ", perímetro,
    " y una altura de ", altura)
FIN_INICIO
Fin(aumento_con_condicion)

```

Vamos a agregar la parte resaltada, para ello damos click en el botón de asignación, y ubicamos el símbolo en el diagrama de flujo, quedando algo como la siguiente figura:



Para editar los valores del triangulo, damos doble click en el rectángulo, aparece una ventana como la mostrada a continuación, colocamos las variables y su asignación y aceptamos:

The screenshot shows a dialog box titled 'Asignación'. It contains three rows of input fields. The first row has 'base' in a text box on the left and '0' in a text box on the right, with a left-pointing arrow between them. The second row has 'altura' in a text box on the left and '0' in a text box on the right, also with a left-pointing arrow. The third row has two empty text boxes with a left-pointing arrow between them. At the bottom of the dialog are two buttons: 'Aceptar' and 'Cancelar'.

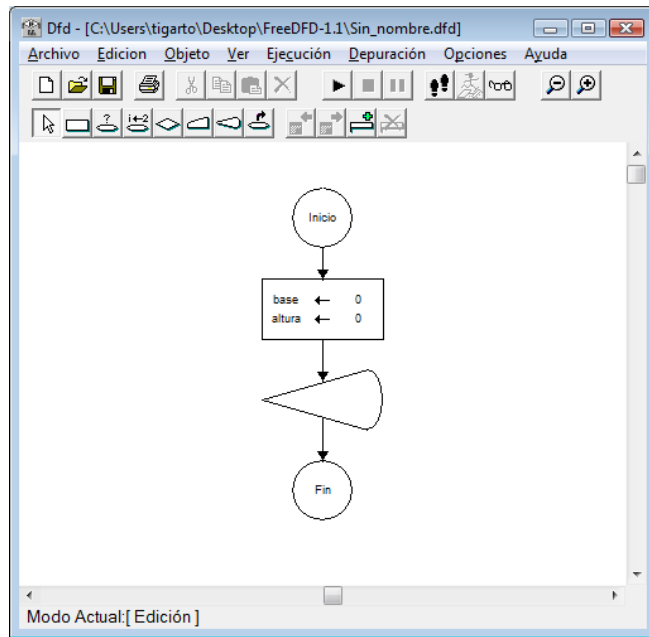
Seguimos ahora con la instrucción escribe:

```

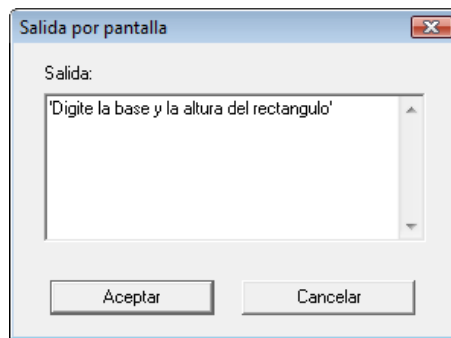
Algoritmo (ejemplo1)
  Variables:
    real: area, perimetro, base = 0, altura = 0
  INICIO
    ESCRIBA("Digite la base y la altura del rectángulo")
    LEA(base, altura)
    area = base*altura
    perimetro = 2*(base + altura)
    ESCRIBA("El rectanculo cuya base es ", base, "y cuya altura
    es ", altura, " tiene un perimetro de ", perimetro,
    " y una altura de ", altura)
  FIN_INICIO
Fin(aumento_con_condicion)

```

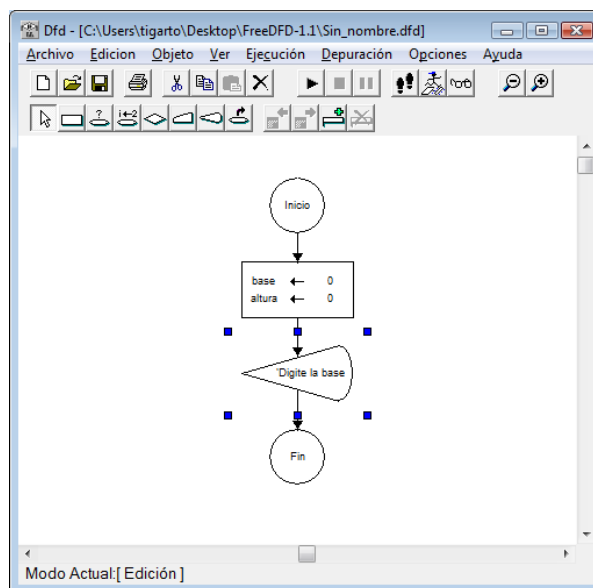
Para ello usamos el botón escribe, damos click para seleccionar el símbolo y luego en el diagrama de flujo, click para ubicarlo, queda algo como esto:



Luego damos en el símbolo de escriba doble click para editarlo, colocamos lo que deseamos que se vea en pantalla **entre comillas sencillas ('')**. Así:



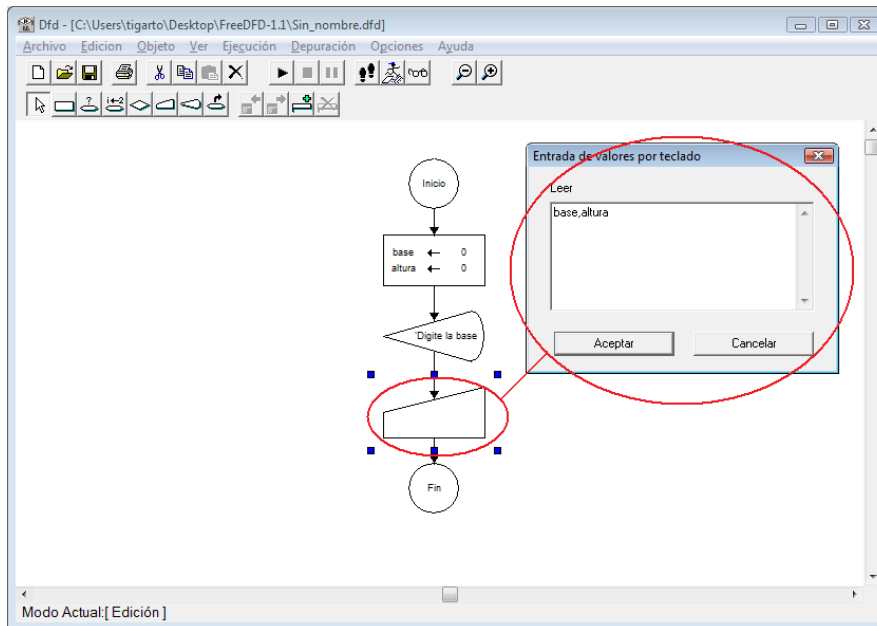
Al aceptar ya tenemos algo como esto:



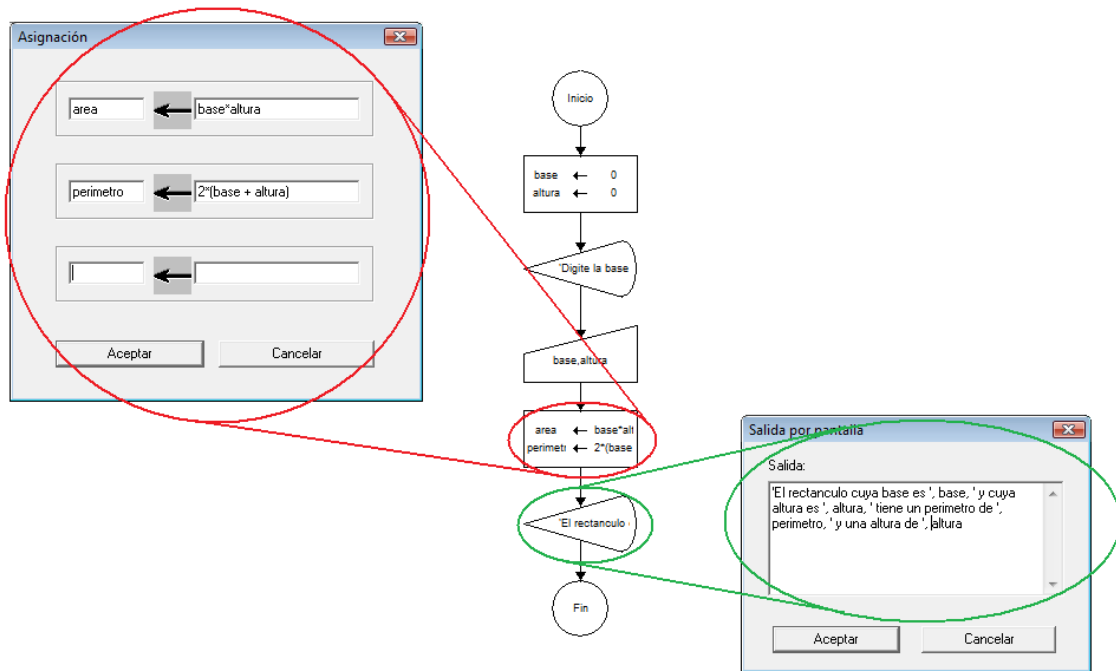
Continuando con el Pseudocódigo tenemos lo siguiente:

```
Algoritmo (ejemplo1)
  Variables:
    real: area, perimetro, base = 0, altura = 0
  INICIO
    ESCRIBA("Digite la base y la altura del rectángulo")
    LEA(base, altura)
    area = base*altura
    perimetro = 2*(base + altura)
    ESCRIBA("El rectanculo cuya base es ", base, "y cuya altura
      es ", altura, " tiene un perimetro de ", perimetro,
      " y una altura de ", altura)
  FIN_INICIO
Fin(aumento_con_condicion)
```

Para ello invocamos el botón lea, y procedemos a editar su contenido:



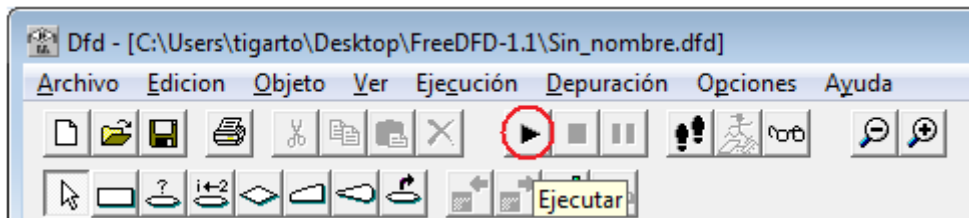
Se procede de igual manera hasta completar el diagrama tal y como se muestra a continuación:



Una vez editado el diagrama de flujo guárdelo.

Ejecutando el algoritmo

Una vez realizado el diagrama de flujo se presiona el botón ejecutar:



Ejecute el programa ingresando 2 y 3 como valore para la base y la altura respectivamente.

Referencias

- www.google.com
- Aguilar, Luis. Fundamentos de programación. Algoritmos y Estructura de Datos. Mc Graw Hill.
- <http://pseint.sourceforge.net/manual.pdf>
- <http://pseint.sourceforge.net/reference.pdf>
- <http://pseint.sourceforge.net/pseudocodigo.php>
- <http://wiki.freaks-unidos.net/freedfd/index>