

DAM_Programación_23_24

[Taboleiro](#) / [Os meus cursos](#) / [zsDAMPR](#) / Tema 1. - Elementos básicos de programación con Java. / [PSEINT - Primeros pasos](#)

PSEINT - Primeros pasos

PSeInt_ejercicios

Conceptos vistos anteriormente

- Instrucción
- Algoritmo
- Programa - Desarrollo de un algoritmo en algún lenguaje de programación
- Aplicación

Pseudocódigo

<http://pseint.sourceforge.net/pseudocodigo.php>

Resumen pasos Algoritmo

La definición de un algoritmo debe describir tres partes: Entrada, [Proceso](#) y Salida:

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

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 cómo es representado un algoritmo. Básicamente 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 las 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 ya se ha dicho anteriormente, 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 número de alumnos, 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 .
Lógico	Se refiere a aquellos datos que pueden tomar solo dos posibles valores falso (F) o verdadero (T)	
Carácter	Asociado a aquellos datos que contienen caracteres alfanuméricos (letras, número, signos de puntuación , etc).	Nombre, DNI, telefono

Cuando se trabaja con variables un aspecto de vital importancia (además del tipo) es el **nombre** que éstas 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:

$$a = 5$$

$$b = c*d + (c-f)*m \quad z = (x+y)/(w+s)$$

$$s = (a/b)^3$$

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 éstas son las instrucciones de entrada y salida.

- **Instrucciones de entrada:**

Permite tomar uno o más datos de un medio externo (comúnmente el [teclado](#)) y asignarlos a una o más variables, su representación en pseudocódigo es:

LEER(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:

ESCRIBIR(var1,var2, ...,varN)

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	$c = 2*a*(b + c)$	$c <- 2*a*(b + c);$	La asignación en PSeint no es con igual (=) sino con flecha (<-) y al final va punto y coma (;). Pero se puede activar la opción como en clase
Entrada	LEER(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	ESCRIBIR("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 ESCRIBIR usado por conversión 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**

Una vez ejecutado, se debe abrir una ventana como la mostrada en la figura 1:

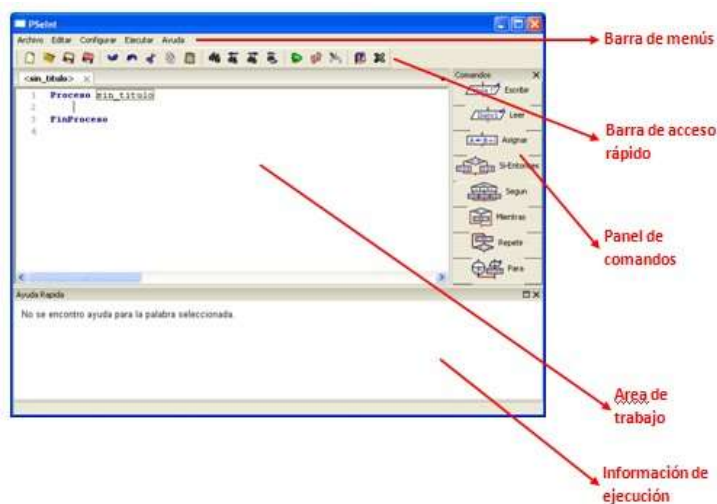


Figura 1. Ventana principal del programa PSeInt.

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:

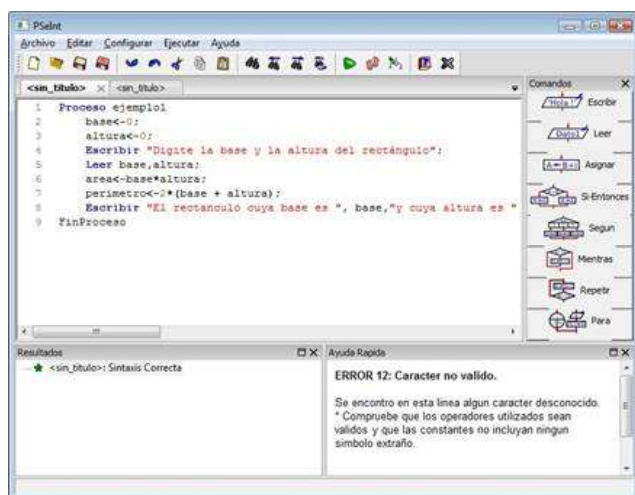
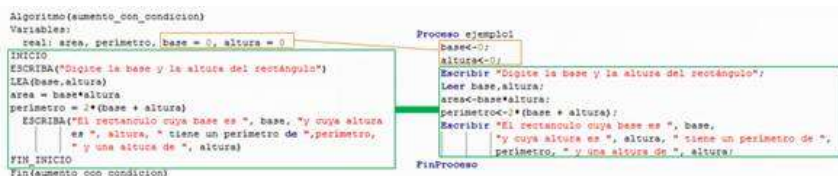


Figura 2. Adaptación del Pseudocódigo del libro al PSeInt.

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:



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 (**ESCRIBIR("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:

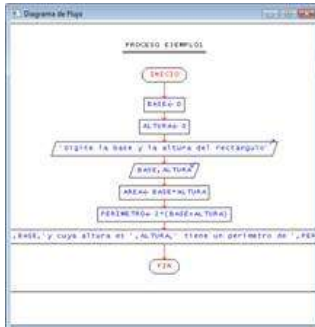


Figura 4. Diagrama de flujo del algoritmo del problema.

Guarda el diagrama de flujo anterior como una [imagen jpg](#) (puede serle útil después, por ejemplo para un [informe](#)).

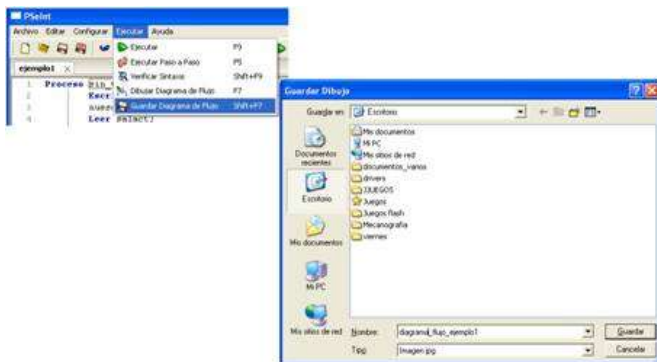


Figura 5. Guardada de la imagen.

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:

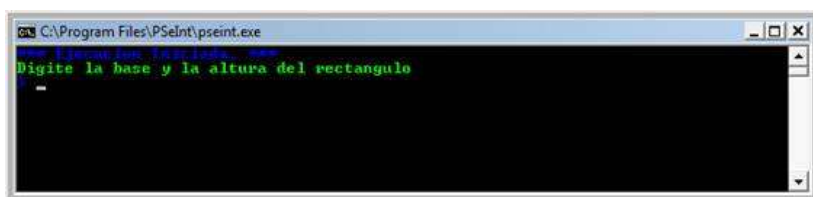


Figura 7. Ejecución del programa asociado al algoritmo (antes de ingresar el valor solicitado por teclado).

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.



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:



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.

```

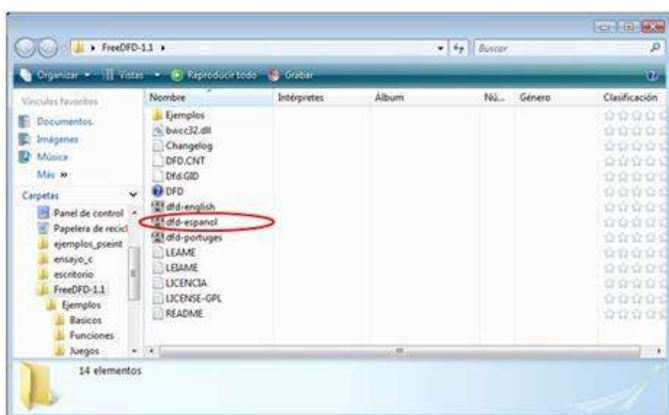
Algoritmo(nombre)
  Variables:
    alfanumerica: nom
  INICIO
    ESCRIBA("Digite el nombre")
    LEA(nom)
    ESCRIBA("Hola ", nom)
  FIN_INICIO
Fin(sumar)

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

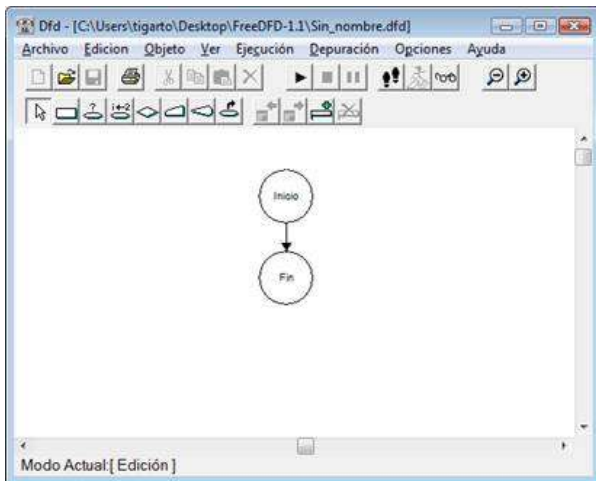
Note de la figura anterior que la codificación inicia desde el INICIO del pseudocódigo de convención. De la declaración de variables (alfanumerica: nom) no se tuvo en cuenta nada pues no hay inicialización de variable alguna.

TRABAJANDO CON FREE DFD o DÍA

Son programas que permiten realizar [diagramas de flujo](#). 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 área 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)

```

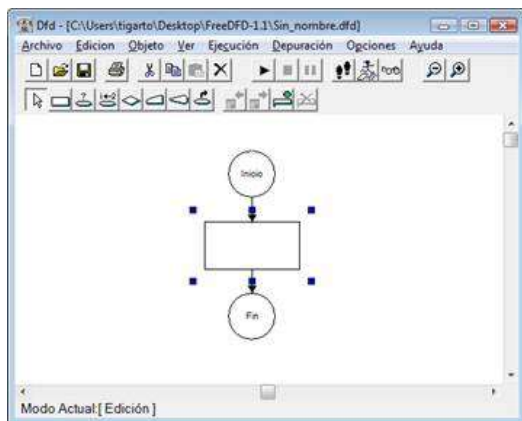
Ahora nuestro objetivo es hacer el diagrama de flujos. Al igual que en el PseInt no se declaran variables, solo se inicializan y la codificación del programa es desde la sentencia INICIO hasta la sentencia FIN_INICIO. Empecemos:

```

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:



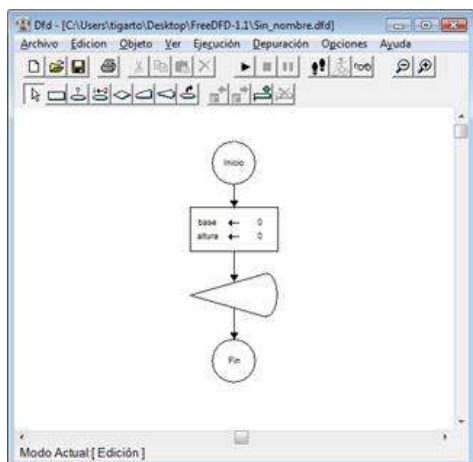
Para editar los valores del triángulo, damos doble click en el rectángulo, aparece una ventana como la mostrada a continuación, colocamos las variables y su asignación y aceptamos:

Seguimos ahora con la instrucción escriba:

```

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 rectangulo 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)
  
```

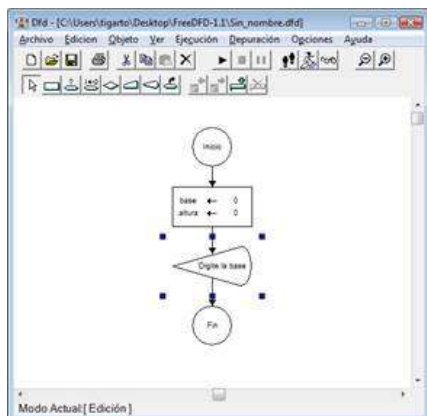
Para ello usamos el botón escriba, damos click para seleccionar el símbolo y luego en el diagrama de flujo, click para ubicarlo, queda algo como esto:



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



Al aceptar ya tenemos algo como esto:



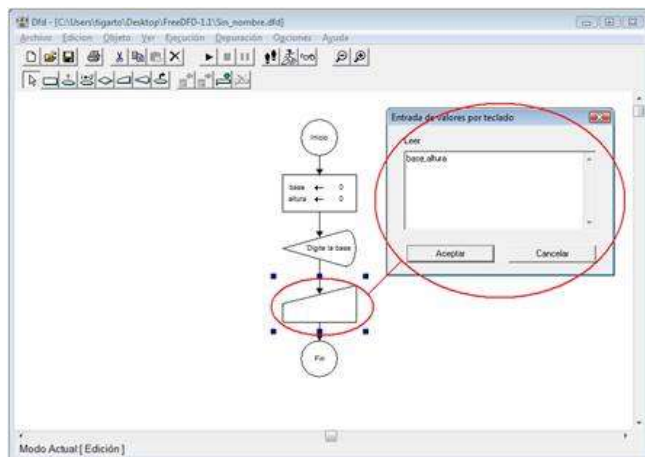
Continuando con el Pseudocódigo tenemos lo siguiente:

```

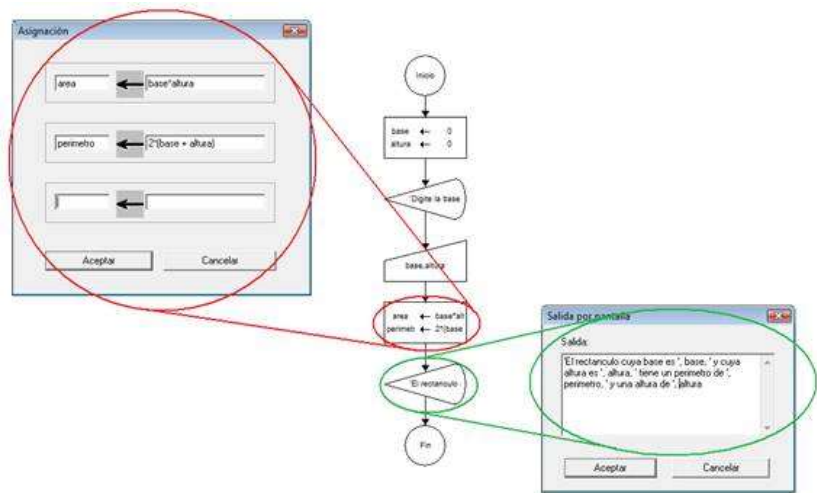
Algoritmo(ejemplo1)
  Variables:
    real: area, perimetro, base = 0, altura = 0
  INICIO
    ESCRIBA("Digite la base y la altura del rectángulo")
    LEA(base, altura)
    area = base*altura
    perimetro = 2*(base + altura)
    ESCRIBA("El rectangulo cuya base es ", base, "y cuya altura
    es ", altura, " 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.

Última modificación: Mércores, 11 de Outubro de 2023, 17:15

Manual Pseint

Ir a...

Ejercicios Pseudocódigo II ►

Vostede accedeu como Andrés Fernández Pereira (Sair)
zsDAMPR

Resumen da retención de datos
Obter a apli móbil