

Herramienta computacional



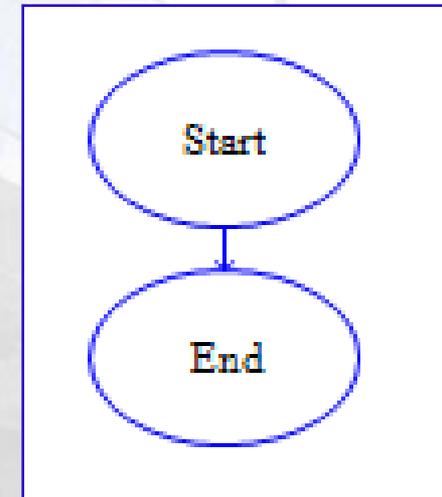
<http://raptor.martincarlisle.com/>

¿Qué es RAPTOR?

- Presenta un entorno de desarrollo de la programación visual basado en *diagramas de flujo*.
- Su lenguaje minimiza la sintaxis que se debe aprender para escribir correctamente un programa de instrucciones.
- Su objetivo es enseñar cómo diseñar y ejecutar algoritmos.

Estructura básica

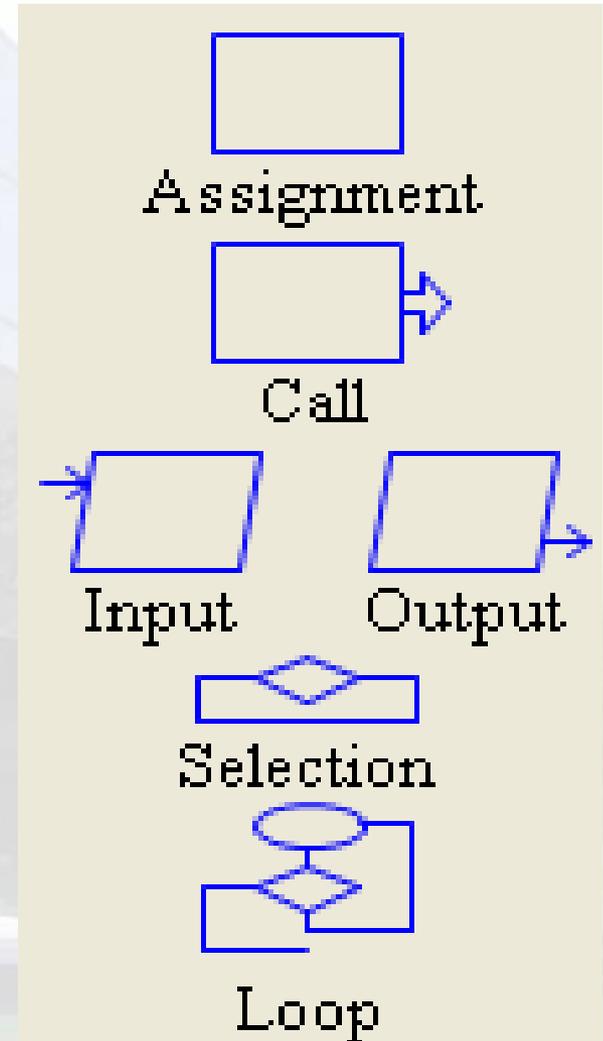
- Inicia con el símbolo **Start**
- Ejecuta el programa a través de sentencias/símbolos.
- Detiene la ejecución cuando se alcanza el símbolo **End**



Sentencias/Símbolos

Cada símbolo representa un tipo único de instrucción.

- asignación
 - llamada
 - entrada
 - salida
 - selección
 - ciclo
- Comandos básicos
- Comandos de control



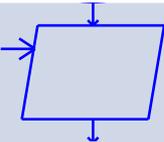
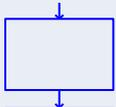
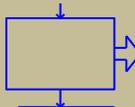
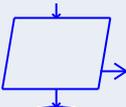
Programación estructurada

- **Secuencia**
- Selección
- Ciclo

Programa típico

- ENTRADA.- obtención de datos que se necesitan
- PROCESO.- manipular los datos para completar tareas
- SALIDA.- desplegar los valores obtenidos

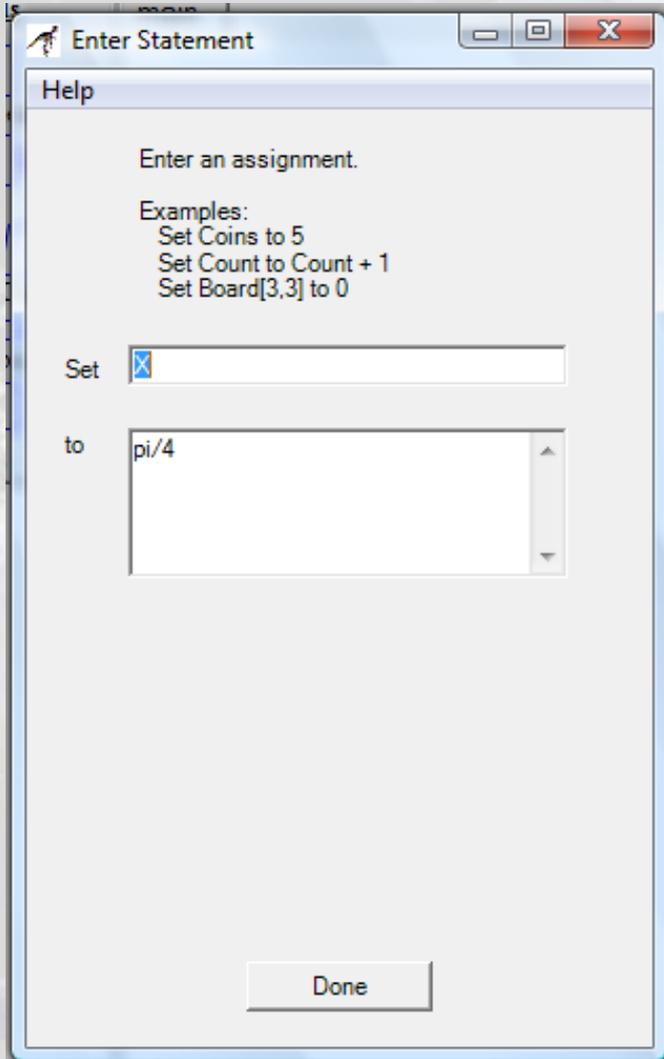
Comandos básicos

Propósito	Símbolo	Nombre	Descripción
Entrada		Sentencia de entrada	Permite que el usuario introduzca datos. Cada valor de un dato se almacena en una variable.
Proceso		Sentencia de asignación	Cambia el valor de una variable usando algún cálculo matemático.
Proceso		Llamada a procedimiento	Ejecuta un grupo de instrucciones definidas en el procedimiento nombrado. En algunos casos algunos de los argumentos de los procedimientos (es decir, variables) serán cambiadas por las instrucciones de los procedimientos.
Salida		Sentencia de salida	Muestra (o guarda en un archivo) el valor de una variable.

Variable

- Retiene el valor del dato.
- Al inicio de un programa las variables no existen.
- Se crean de manera automática cuando se usan en una sentencia.
- Su valor se puede calcular por una ecuación en una sentencia de asignación.
- Su valor se puede obtener a través de una sentencia de entrada.

Sentencia/Símbolo de Asignación



Enter Statement

Help

Enter an assignment.

Examples:
Set Coins to 5
Set Count to Count + 1
Set Board[3,3] to 0

Set

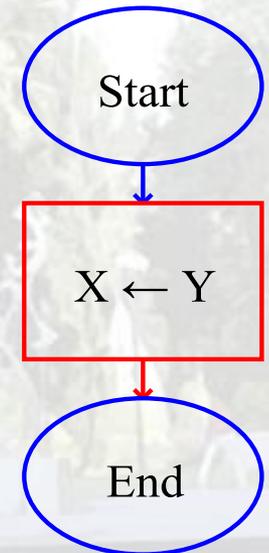
to

Done

- Se utiliza para realizar un cálculo y después almacenar los resultados en una variable.

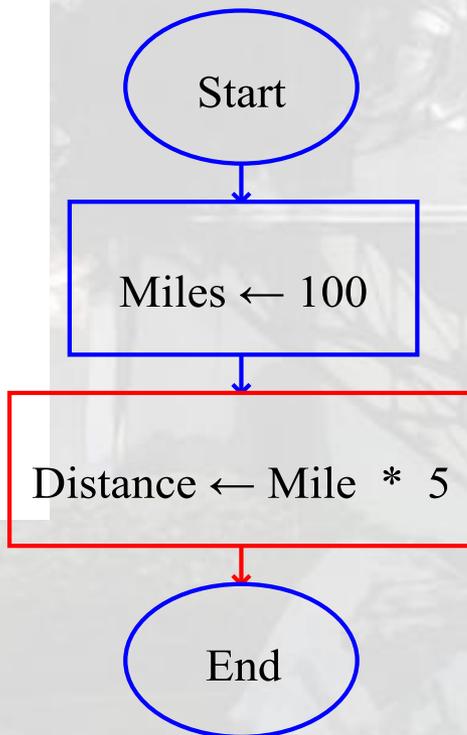
Errores comunes al usar variables

- Error 1: "Variable ____ does not have a value"
 - Hay dos razones comunes para ese error.
 - 1.- No se le dio valor a la variable



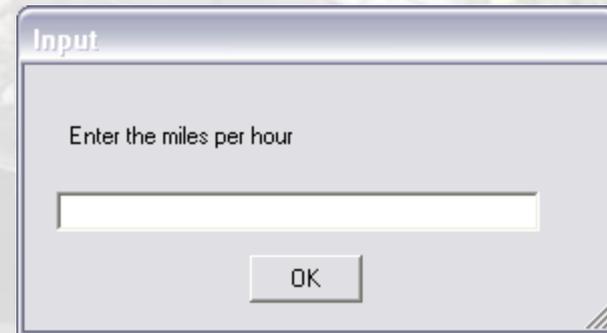
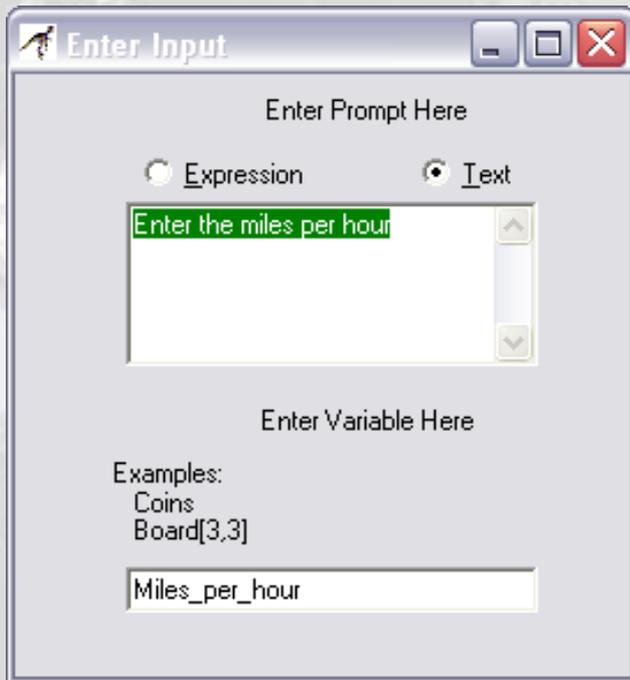
Errores comunes al usar variables

2.- El nombre de la variable fue mal escrito



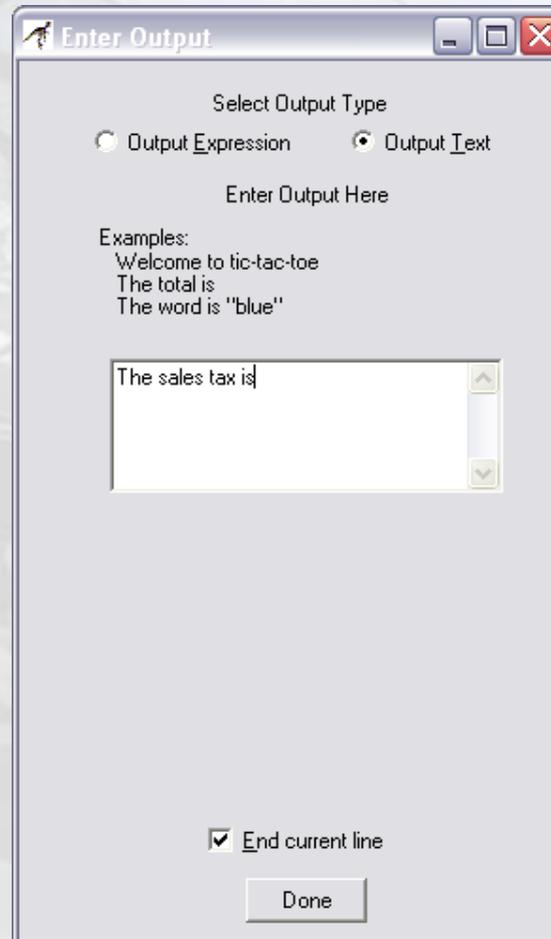
Sentencia/Símbolo de Entrada

- Permite que el usuario de un programa incorpore un valor de datos en una variable del programa durante la ejecución de programa.



Sentencia/Símbolo de Salida

- Muestra un valor en la ventana MasterConsole



Enter Output

Select Output Type

Output Expression Output Text

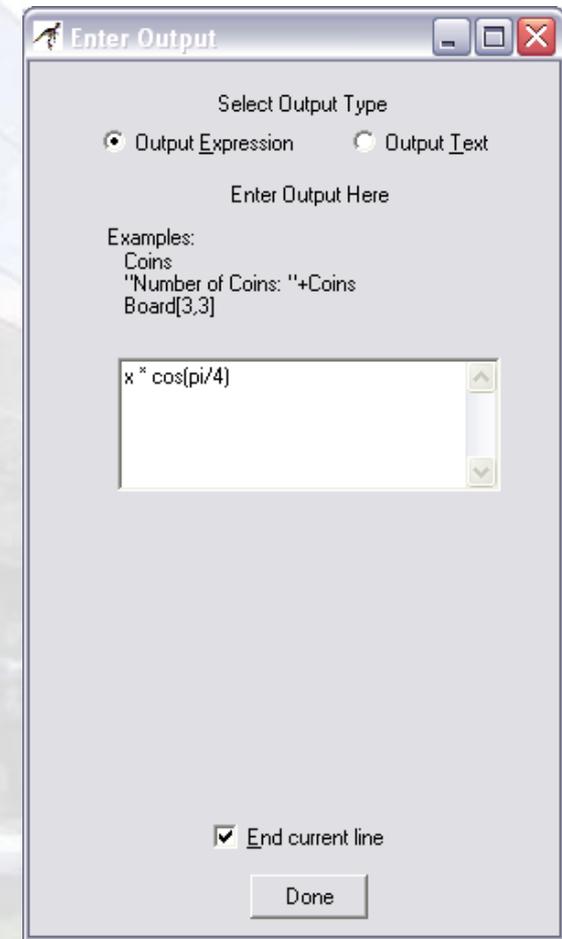
Enter Output Here

Examples:
Welcome to tic-tac-toe
The total is
The word is "blue"

The sales tax is

End current line

Done



Enter Output

Select Output Type

Output Expression Output Text

Enter Output Here

Examples:
Coins
'Number of Coins: '+Coins
Board[3,3]

$x * \cos(\pi/4)$

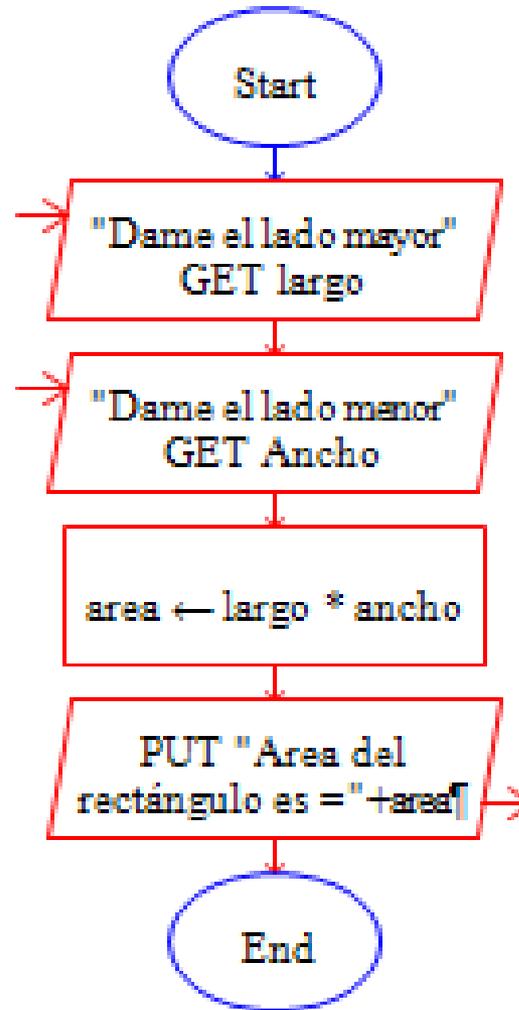
End current line

Done

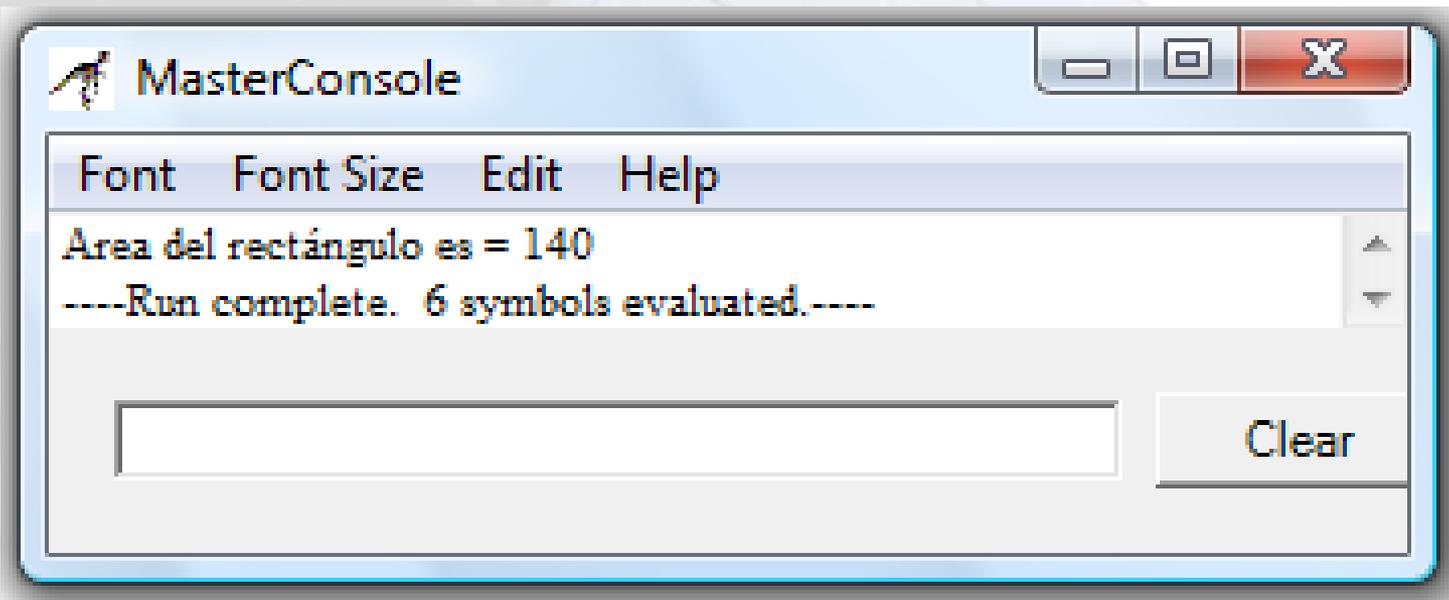
Ejercicio 1

- Diseñar un diagrama de flujo para calcular y mostrar el área de un rectángulo. El diagrama deberá de solicitar los datos necesarios y mostrar el resultado.

Diagrama de Flujo



Salida



Programación estructurada

- Secuencia
- **Selección**
- Ciclo

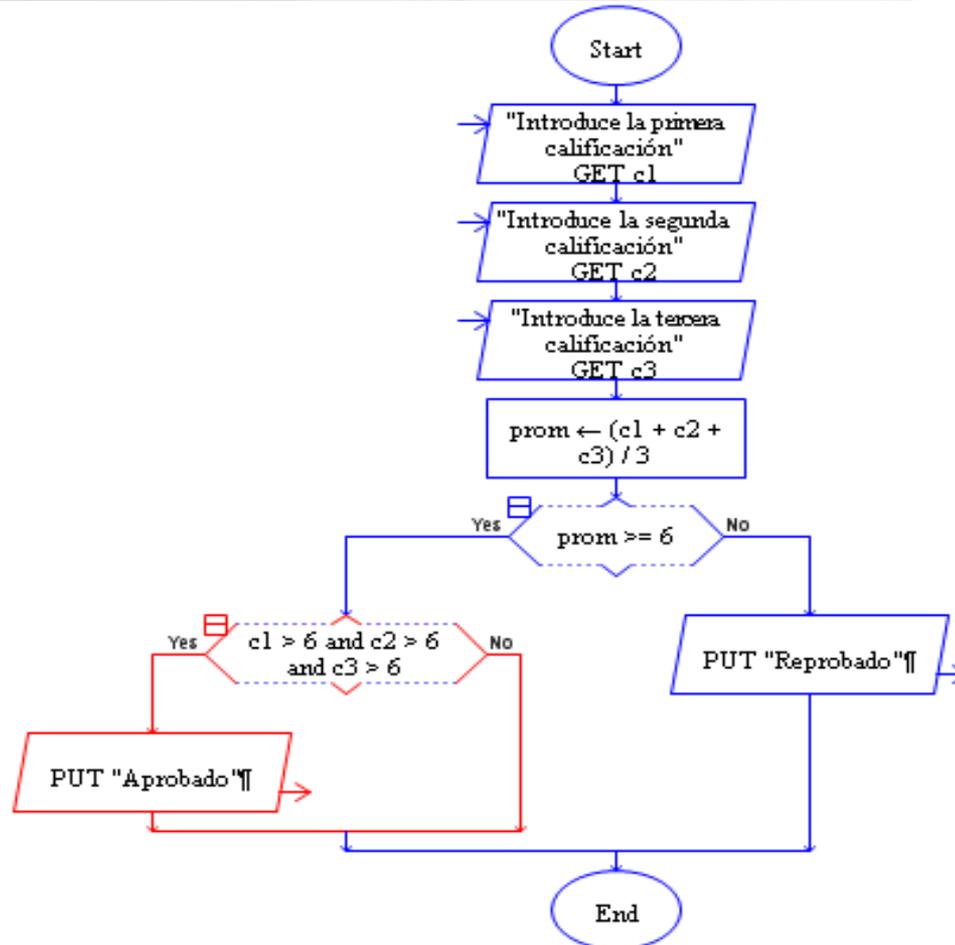
Estructura de control de selección

Le indica a la computadora qué camino seguir al tener que decidir algo, con base en si una condición es verdadera o falsa

Ejercicio 2

- Diseñar un diagrama de flujo que pida tres calificaciones, las analice y determine si el alumno reprobó o aprobó.
- El diagrama deberá de solicitar los datos necesarios y mostrar el resultado.
- Para considerar aprobado a un alumno, sus calificaciones deben cumplir las siguientes **Reglas**:
- El promedio de las 3 calificaciones debe ser Mayor o igual a 6.0.
- Ninguna de las calificaciones individuales debe ser menor que 6.

Diagrama de Flujo



Ejercicio 3

Es martes por la noche y su equipo de computación deciden quedarse a estudiar por la noche para su examen final pero el hambre los invade.

Una pizza suena bien, pero ustedes tienen poco dinero, por lo que llaman a varias pizzerías para comparar precios.

Uno llama a la pizzería VanGo, que venden una pizza redonda de 20 cm de diámetro, con dos ingredientes, por 8.99 dólares. Otro estudiante llama a un lugar llamado Venecia y descubre que tienen una pizza cuadrada, de 25 cm por lado a 11.99 dólares. ¿Cuál de las dos pizzas sería la mejor opción?

Deciden analizar un programa rápido para comparar los precios de las pizzas. Desarrollar el algoritmo y el diagrama de flujo de dicho programa.

Programación estructurada

- Secuencia
- Selección
- **Ciclo**

Estructura de control ciclos

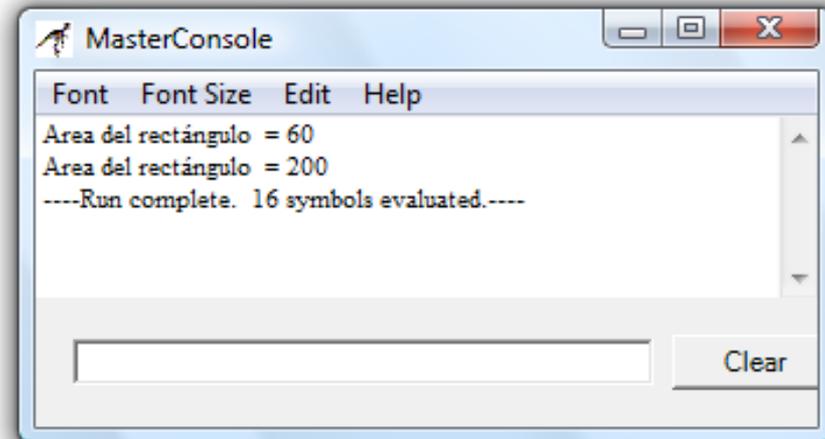
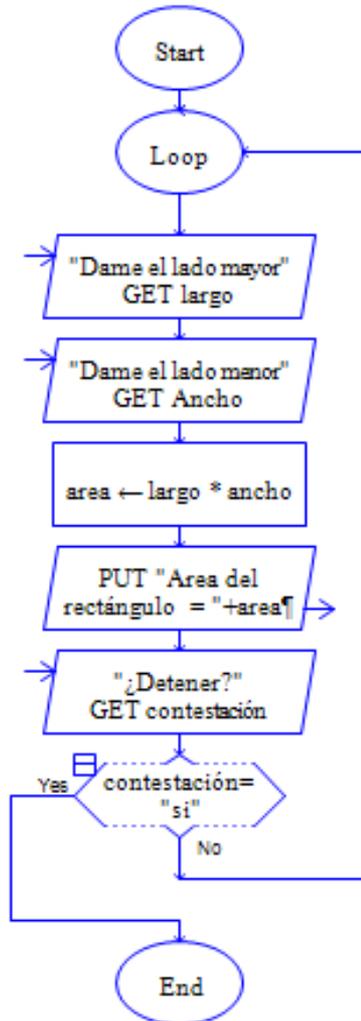
- Ciclo.- permite repetir una o más instrucciones hasta que se cumpla cierta condición.

Ejercicio 4

Diseñar un diagrama de flujo para calcular y mostrar el área de varios rectángulos que se pidan.

- El diagrama deberá de solicitar los datos necesarios y mostrar el resultado de cada rectángulo
- Se detendrá el cálculo de áreas hasta que se pida que se detenga.

Solución ejercicio 4



Ciclo con Contador

Ejecuta un bloque de instrucciones un número específico de veces.

Se debe indicar lo siguiente:

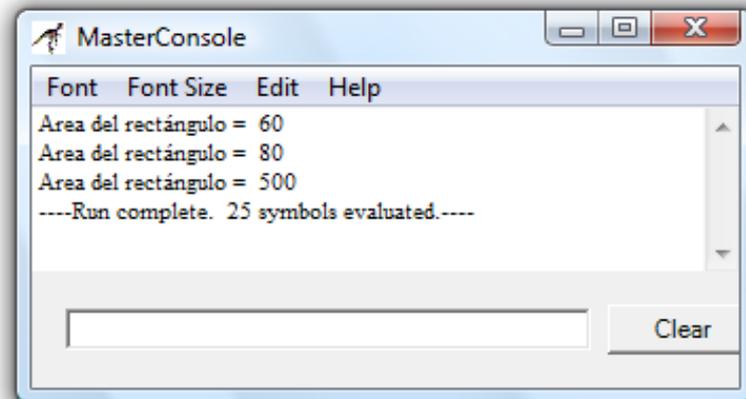
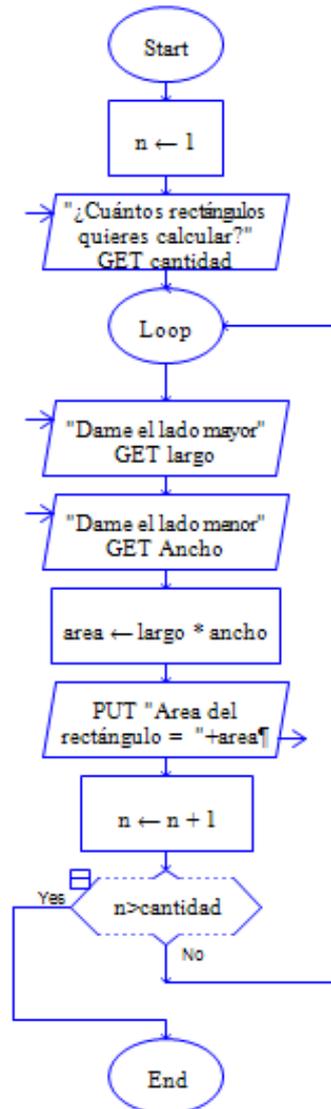
- Darle nombre al contador
- Darle un valor con el que va a iniciar el conteo.
- Ir incrementando el valor del contador
- La condición de salida es cuando se llegue al número de veces que se requiera repetir

Ejercicio 5

Diseñar un diagrama de flujo para calcular y mostrar el área de “ n ” rectángulos que se pidan.

- El diagrama deberá de solicitar los datos necesarios y mostrar el resultado de cada rectángulo
- Se detendrá el cálculo de áreas hasta que se llegue a “ n ”.

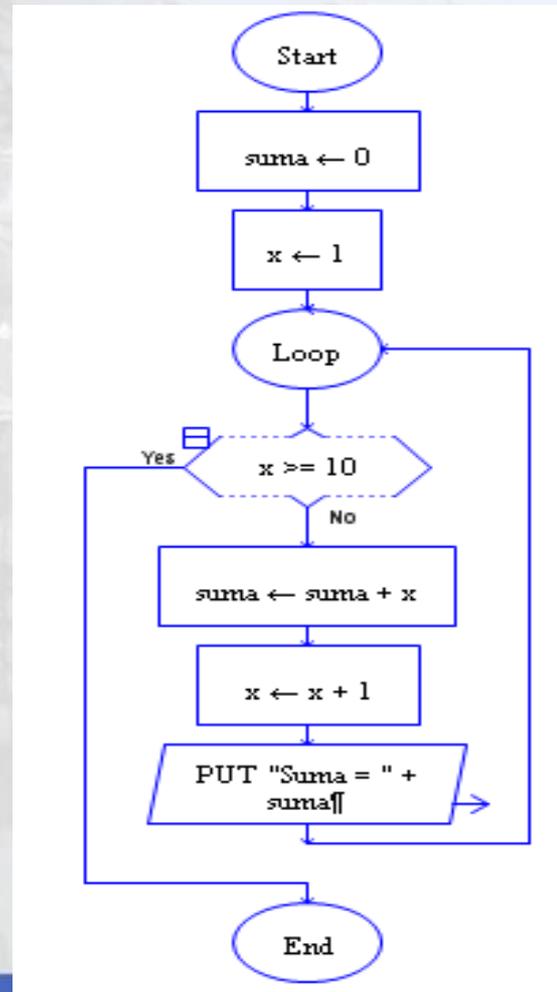
Solución ejercicio 5



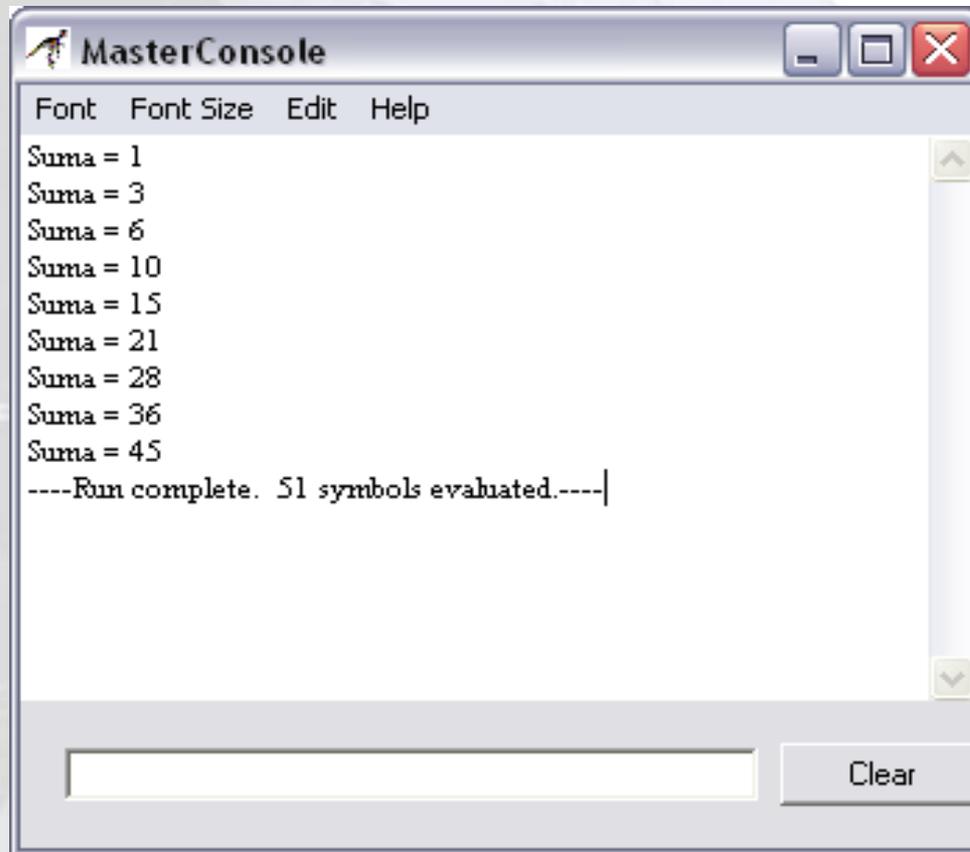
Ejercicio 6

- Diseñar un diagrama de flujo que calcule e imprima la suma de los números del 1 al 10, utilizando un ciclo automático.
El diagrama deberá de solicitar los datos necesarios y mostrar el resultado.

Diagrama de flujo ejercicio 6



Salida

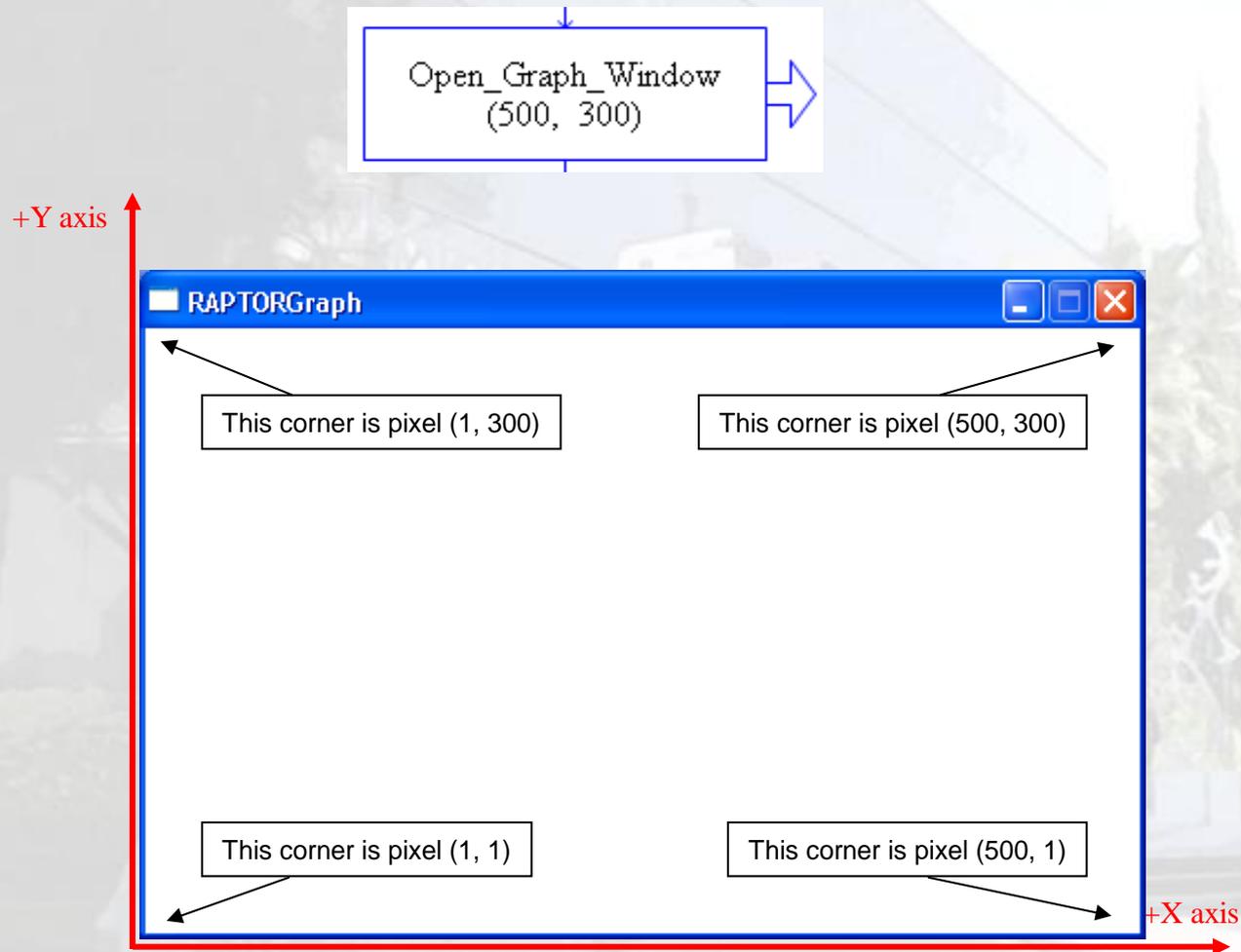


```
Font  Font Size  Edit  Help
Suma = 1
Suma = 3
Suma = 6
Suma = 10
Suma = 15
Suma = 21
Suma = 28
Suma = 36
Suma = 45
----Fun complete. 51 symbols evaluated.----|
```

Ejercicio 7

Ir dando calificaciones de alumnos hasta que se terminen las boletas que se tienen que registrar e indicar cuántos de ellos aprobaron y cuántos reprobaron (menor a 70 es reprobado, mayor o igual a 70 es aprobado)

- **Open_Graph_Window (tamañoX, tamañoY)**

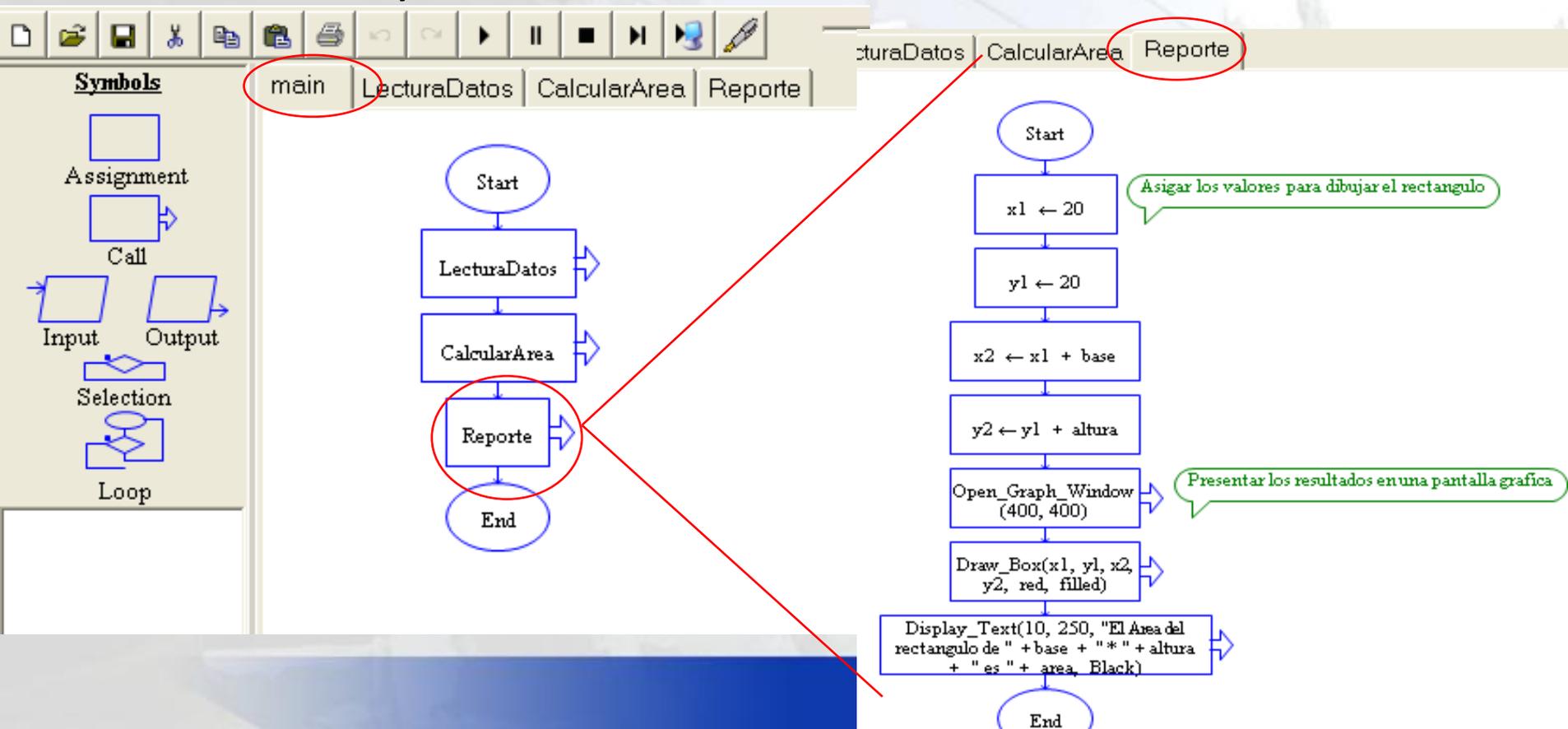


Dibujar

- **Draw_Line($X1$, $Y1$, $X2$, $Y2$, Color)**
Dibuja una recta entre $(X1, Y1)$ y $(X2, Y2)$ con el color especificado
- **Draw_Box($X1$, $Y1$, $X2$, $Y2$, Color, Filled/Unfilled)**
Dibuja un rectángulo con esquinas diagonalmente opuestas $(X1, Y1)$ y $(X2, Y2)$. *Filled* lo dibuja relleno del color
- **Draw_Circle(X , Y , Radio, Color, Filled/Unfilled)**
Dibuja un círculo dado su centro (X, Y) y su radio.
- **Draw_Ellipse($X1$, $Y1$, $X2$, $Y2$, Color, Filled/Unfilled)**
Dibuja la elipse que se inscribe en el rectángulo de esquinas diagonalmente opuestas $(X1, Y1)$ y $(X2, Y2)$.
- **Display_Text(X , Y , Texto, Color)**
Dibuja el texto de la cadena dada, con la esquina superior izquierda del primer carácter en la posición (X, Y) .

Abstracción y subprocessos

- Abstracción: Dar nombre a las cosas
 - La abstracción reduce la complejidad del problema
- Subproceso: Permite reducir la complejidad agrupando sentencias y dándoles un nombre



Ejercicio gráfico

- Diseñar un programa que solicite los lados de un rectángulo. El programa debe calcular el área del rectángulo, y luego presentar esta área y dibujar el rectángulo en una pantalla gráfica