

Capítulo 1

Introducción

Pasos a seguir para resolver un problema

1- Definición del problema

Consiste en identificar el problema y enunciarlo de una manera comprensible. Se puede partir de las preguntas: ¿Cuál es el problema? ¿En qué consiste el problema? ¿Cuál es el objetivo al cual quiero llegar?

2- Análisis del problema

Consiste en identificar con claridad los datos con los que se cuenta, las incógnitas que se deben calcular, seleccionar el método de trabajo, utilizar las distintas herramientas de las que se disponen, como ser diagramas de flujo o algoritmos, y verificar su comportamiento con una prueba de escritorio.

3- Programación

Consiste en llevar el diagrama de flujo o el algoritmo a un lenguaje de programación mediante la codificación. Para ello se deberán tener en cuenta los tipos de datos que se utilizarán, los periféricos de entrada y salida y la sintaxis de los comandos del lenguaje seleccionado. Una vez codificado el proceso se lo llevará a una computadora y se lo verifica, haciendo las correcciones que sean necesarias.

4- Resolución del problema

La resolución se hará mediante el uso de la computadora, ejecutando el proceso programado, con los datos reales del problema. Como resultado obtendremos la información buscada.

5- Documentación

Consiste en guardar adecuadamente el diagrama de flujo, la codificación y algunos ejemplos de ejecución, agregando todas las notas y aclaraciones que pudieran resultar útiles para un posterior análisis o modificación del proceso desarrollado.

Diagramas de flujo

Un diagrama de flujo es **la representación gráfica de los pasos lógicos a seguir** para resolver un problema, que puede ser de procesamiento de datos, científico, etc.

El diagrama de flujo se compone de gráficos estandarizados, que tienen su significado y alcance, unidos por líneas de flujo, que marcan el recorrido del proceso.

Variables

Las variables son espacios de memoria con un nombre de identificación, que almacenan valores que pueden ir cambiando durante la ejecución del programa. Las variables pueden ser variables simples escalares o variables de tipo matricial.

Las variables simples, también conocidas como variables escalares, pueden almacenar un solo valor a la vez, y el almacenamiento de un nuevo valor desplaza al anterior provocando su pérdida. Este es el tipo de variables con las que se comienza a trabajar en las primeras unidades.

Las variables matriciales subindicadas o arrays se utilizarán en las unidades finales, ya que son variables con una estructura interna que permite el almacenamiento de más de un valor, y su manejo es más complejo.

Las variables pueden ser **numéricas o alfanuméricas**, es decir, algunas almacenan **valores numéricos enteros o decimales**, y otras almacenan **cadena alfanuméricas** formadas por letras, números y otros caracteres del código ASCII encerrados entre comillas. Las variables numéricas, a su vez, pueden tener diferente precisión, así se tienen las variables numéricas enteras, las variables numéricas para valores decimales con simple precisión y doble precisión, variables numéricas para fechas, etc. No pueden almacenar números fraccionarios.

En la primera etapa de la materia se utilizarán variables alfanuméricas y variables numéricas sin precisar exactamente su tipo, pero con el comienzo de las unidades de codificación se especificarán en forma exacta los tipos de variables que requiere cada programa.

Ejemplos:

n = 5

c = 1

Nombre\$ = "José Luis"

Direccion\$ = "Av. Alberdi 201"

Indice = 10

j = - 420.12

det = 32.02562546

opcion\$ = "sí"

R\$ = "NO"

DNI\$ = "14.628.224"

promedio = acumulador / n

superficie = base * altura

En los ejemplos anteriores se muestran nombres válidos de variables y la asignación de valores que pueden ser almacenados en ellas, de acuerdo al tipo de variable en cada caso. Las variables alfanuméricas llevan el símbolo \$ al final de su nombre, para distinguirlas de las variables numéricas, según las normas de nombres válidos para variables en el lenguaje Visual Basic.

Las reglas para la denominación de las variables se explican a continuación en este mismo capítulo. El estudio de todos los tipos de variables, la precisión de cada una y el espacio de memoria que utilizan se verá en capítulos posteriores, ya que tiene estrecha vinculación con el lenguaje de programación seleccionado para codificar los diagramas de flujo.

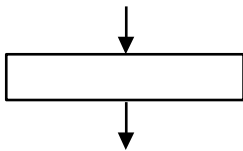
Reglas de asignación de nombres de variables en Visual Basic

Para dar nombre a constantes y variables en un módulo de Visual Basic han de seguirse las siguientes reglas:

- El primer caracter debe ser una letra.
- En el nombre no se pueden utilizar espacios, puntos (.), ni el símbolo @.
- No se puede utilizar una letra griega, por ejemplo: α , β , γ , λ , ξ , π , como nombre de variable. Tampoco se pueden utilizar letras griegas en operaciones matemáticas.
- El nombre no puede tener más de 255 caracteres de longitud.
- Como regla general, no se deben usar nombres iguales a los de los procedimientos Sub, Function, instrucciones y métodos de Visual Basic, es decir todo tipo de palabras reservadas por el lenguaje Visual Basic.
- Los nombres no se pueden repetir dentro del mismo nivel de alcance. Por ejemplo, no se pueden declarar dos variables con el nombre edad dentro del mismo procedimiento. Sin embargo, se puede declarar una variable privada edad y una variable de nivel de procedimiento llamada edad dentro del mismo módulo.
- Visual Basic no diferencia entre mayúsculas y minúsculas, pero respeta la forma en que se escriben las instrucciones de declaración de nombres. En otras palabras, para Visual Basic la variable Ancho, la variable ANCHO o la variable ancho hacen referencia a la misma variable.

Gráficos del diagrama de flujo. Significado y ejemplos

Proceso

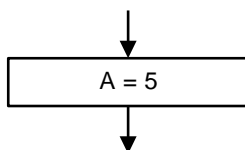


Representa un punto en el problema en que se resuelve una operación, se asigna un valor a una variable, ya sea en forma directa o a través de una operación matemática o lógica.

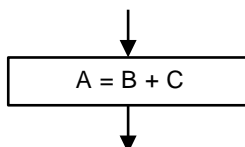
En el interior del gráfico debe aparecer una variable a la izquierda y un signo igual que significa transferencia o asignación, además de un valor, una operación o una función en el lado derecho, que será resuelta en primer lugar y luego su resultado será asignado a la variable de la izquierda para su almacenamiento.

El gráfico de proceso tiene una entrada y una salida de flujo.

Ejemplos:



La variable A recibe un valor igual a 5

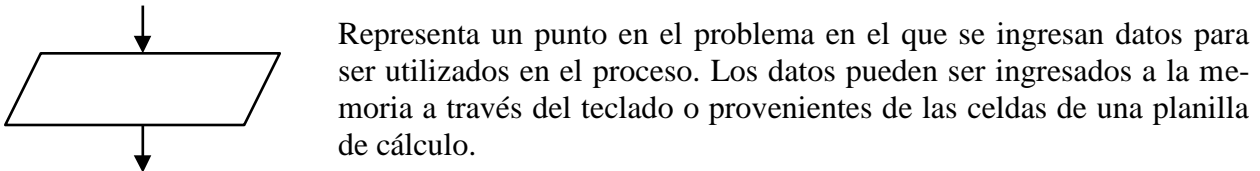


La variable A recibe el resultado de la suma de los valores almacenados en las variables B y C



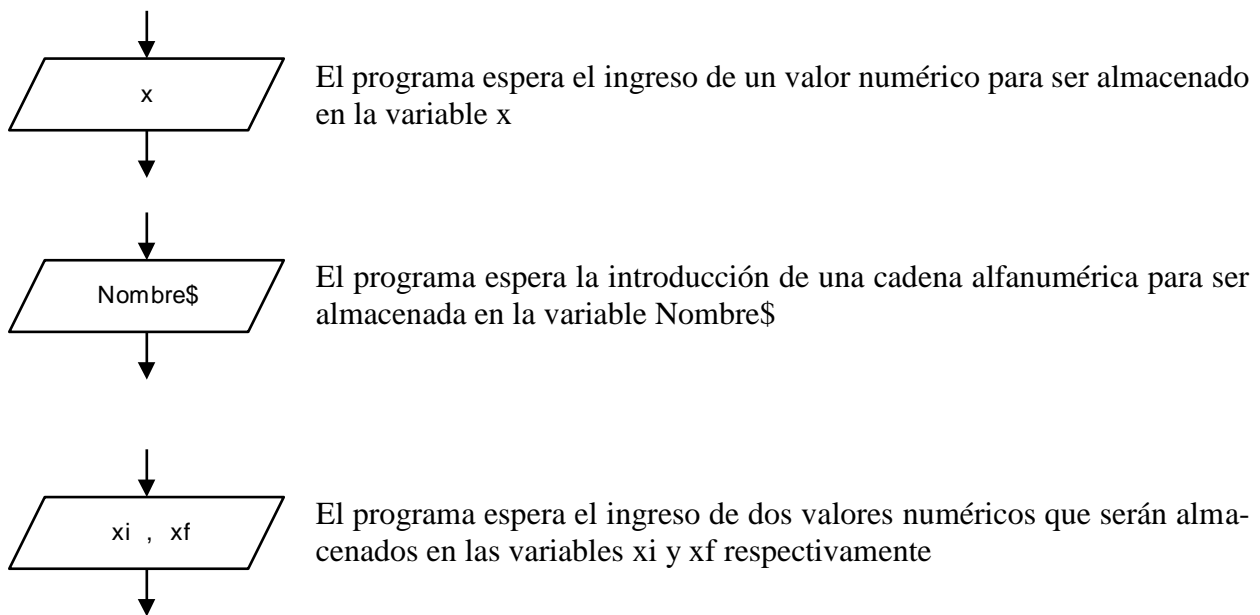
Como se ve en los ejemplos, las variables almacenan distintos tipos de valores, como por ejemplo números enteros, números decimales, palabras o cadenas de caracteres, etc. Para cada tipo de dato almacenado existe un tipo de variable adecuado como se explicó anteriormente.

Entrada de datos

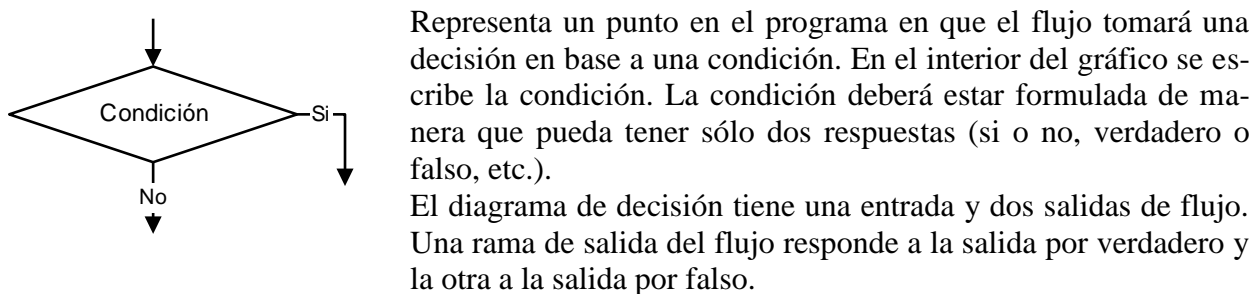


En este gráfico se debe colocar la variable que recibe los datos para almacenarlos. En el mismo gráfico se pueden colocar varias variables separadas por comas. El diagrama de entrada de datos tiene una entrada y una salida de flujo.

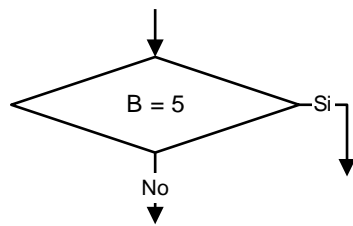
Ejemplos:



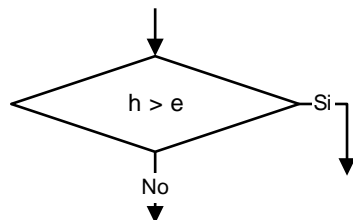
Decisión



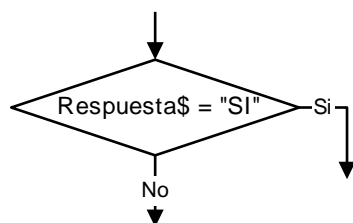
Ejemplos:



El programa verifica si la variable B contiene un valor igual a 5 y selecciona la rama de salida correspondiente.



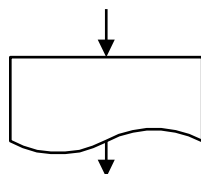
El programa realiza la comparación de los valores almacenados en las variables h y e, y luego selecciona la rama de salida correspondiente.



El programa compara la cadena alfanumérica almacenada en la variable Respuesta\$ con la cadena alfanumérica "SI" y toma la rama de salida correspondiente.

En el diagrama de decisión el signo igual (=) no significa transferencia, como en un proceso, sino igualdad, y se lo utiliza como condición para comparar si la variable de la izquierda es igual a la variable o al valor de la derecha.

Salida

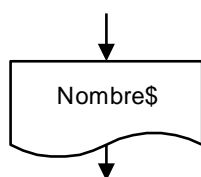


Representa un punto del problema en el que se visualiza una determinada información contenida en memoria. La información puede ser mostrada en pantalla, en una impresora, o ser escrita en las celdas de una hoja de cálculo.

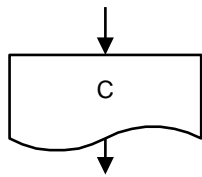
En el gráfico pueden colocarse variables de diferentes tipos, operaciones matemáticas, valores numéricos y cadenas alfanuméricas. En cada caso la salida será diferente. Más adelante se verá en detalle cómo visualizar los valores contenidos en una variable de manera básica y también cómo generar mensajes de salida combinando palabras y valores numéricos, facilitando la interpretación para el usuario que utiliza el programa.

El gráfico de salida tiene una entrada y una salida de flujo.

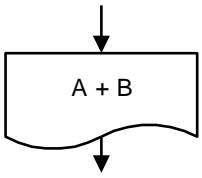
Ejemplos:



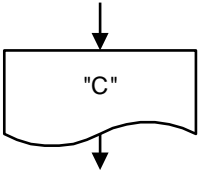
El programa imprime el contenido almacenado en la variable Nombre\$



El programa imprime el contenido almacenado en la variable C

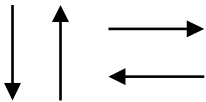


El programa imprime el resultado de la operación A+B



El programa imprime la letra "C"

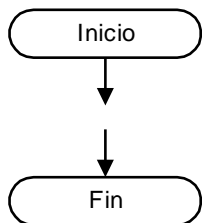
Líneas de flujo



Las líneas de flujo marcan el recorrido en que será procesado el diagrama de flujo. El sentido normal es de arriba hacia abajo y de izquierda a derecha. Este sentido no necesita flechas.

Cuando se desea recorrer de abajo hacia arriba o de derecha a izquierda es necesario el uso de las flechas.

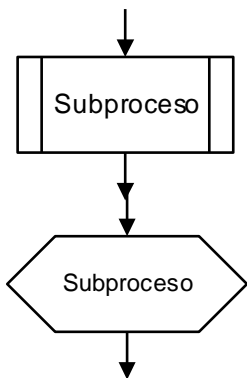
Inicio y fin



Estos gráficos marcan el inicio y el fin del programa, por ello tienen una sola línea de flujo, que sale o que llega al gráfico, según corresponda.

Es aconsejable que los programas tengan un solo gráfico de fin, como se verá más adelante.

Subproceso

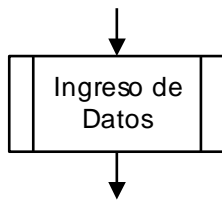


Representa un punto en el programa en el que se ejecutará un conjunto de instrucciones repetitivas, que han sido desarrolladas y programadas por separado. Esto se utiliza por una cuestión de orden y de claridad en el flujo principal del proceso, dejando por separado la diagramación que se repite varias veces en el transcurso del programa y sólo se la invoca desde el flujo principal del programa.

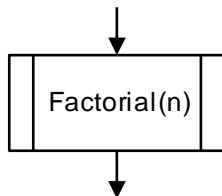
Otro uso muy frecuente consiste en la división de los programas extensos en módulos, que se van programando por separado y se los invoca desde el flujo principal del programa. Esto se conoce como programación modulada.

En Visual Basic se pueden utilizar dos tipos de subprocesos, los Procedimientos Sub() y los Procedimientos Function(). El estudio de estos procedimientos así como de los argumentos que admiten se verá en detalle más adelante.

Ejemplos:

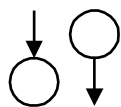


Este subproceso se encarga de ingresar los datos necesarios para la ejecución del programa principal. Ha sido resuelto y programado por separado, y en el programa principal sólo se lo invoca por su nombre.

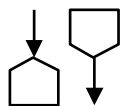


Este subproceso calcula el factorial de un número n que se suministra cuando se hace su llamado. Los cálculos para obtener el factorial han sido desarrollados y programados por separado, con la ventaja de que este subproceso podrá ser invocado varias veces dentro del mismo programa suministrando en cada caso un número diferente para obtener su factorial.

Conectores



Conectores dentro de una misma página



Conectores para cambio de página

Se utilizan para interrumpir el flujo del proceso y retomarlo en otro lado. En su interior se colocan números o letras iguales, donde se corta y donde se retoma el flujo. Su objetivo es darle continuidad a las líneas de flujo cuando se cambia de hoja, o bien mejorar la claridad de un diagrama de flujo en la misma hoja evitando el entrecruzamiento de líneas.

Operadores Aritméticos

Las operaciones matemáticas se transcriben en Visual Basic respetando la sintaxis exigida por el lenguaje. A continuación se presenta una lista de las operaciones más utilizadas y el operador correspondiente.

Suma	+
Resta	-
Producto	*
División	/
Exponenciación	^

Las operaciones se resuelven según la jerarquía de los operadores matemáticos, es decir, los operadores de menor jerarquía se resuelven primero, como la exponenciación, el producto y la división, y luego los de mayor jerarquía, como la suma y la resta. A igualdad de jerarquía la expresión se resuelve de izquierda a derecha.

Para cambiar el orden de resolución de expresiones matemáticas se deben utilizar paréntesis, empleando todos los que sean necesarios, uno a continuación de otro, ya que **no están permiti-**

dos los corchetes ni las llaves como en la notación matemática. A continuación se pueden ver algunos ejemplos de operaciones y de uso de varios paréntesis.

Ejemplos:

$$a = 22 * z ^ 3$$

$$y = - 4 / (2 * x)$$

$$z = 2 * x + 3 * y$$

$$x1 = (-b + (b^2 - 4 * a * c)) / (2 * a)$$

$$f = n * (n-1) * (n-2)$$

$$\text{delta} = (y2 - y1) / (x2 - x1)$$

$$\text{distancia} = ((x2-x1)^2 + (y2-y1)^2) ^ (1/2)$$

Ejercicios Capítulo 1:

1. Determinar qué nombres de las siguientes variables son válidos en Visual Basic:

- | | |
|-------------------------|-----------------|
| 1) TotalVentasAnuales | 2) %Intereses |
| 3) NumeroDecimal | 4) π |
| 5) Tasa IVA | 6) 14Valores |
| 7) Nombre\$ | 8) Error ξ |
| 9) Fecha\$ | 10) λ |
| 11) Superficie | 12) alfa |
| 13) Dia de Nacimiento\$ | 14) radio |
| 15) ϵ | 16) Direccion\$ |

2. Determinar qué valores se pueden asignar a una variable numérica en Visual Basic:

- | | |
|-----------|-------------|
| 1) 14.5 | 2) 225\$ |
| 3) 47.540 | 4) 14225 kg |
| 5) 12 cm | 6) 15,24 |

3. Determinar qué procesos de asignación contienen errores y en qué consiste el error cada caso:

- | | |
|--------------------------------|------------------------------|
| 1) A = 1.000 | 2) Q = P/R |
| 3) Promedio = Acumulador / n | 4) lado * ancho = Superficie |
| 5) Direccion = "Las Heras 740" | 6) Nombre\$ = "José Gómez" |

4. Determinar qué valores se asignarán a las variables en las siguientes expresiones, si trabajamos con los siguientes valores: A = 8; B = 4 y C = 3:

- | | |
|------------------|-----------------------|
| 1) M = A + B + C | 2) N = C / 3 |
| 3) B = B + 1 | 4) Superficie = B * C |
| 5) C = A - 1 | 6) C = C + 2 |

5. Escribir las siguientes expresiones algebraicas en forma interpretable por el Visual Basic.

$$r = \frac{x^2 + y^2}{z^2}$$

$$r = \sqrt{b^2 - 4ac}$$

$$r = \frac{4x + 2y}{3z}$$

$$r = \frac{a + b}{c - d}$$

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$s = \frac{4}{3}\pi r^3$$

$$r = ax^2 - 5x + 10$$

$$r = \frac{x + y}{z} - \frac{3x}{4} + 4y$$

$$r = \frac{ab}{cd}$$

$$r = \frac{y_2 - y_1}{x_2 - x_1}$$

$$d = 2\pi r$$

$$s = 4\pi r^2$$

6. Escribir las siguientes expresiones Visual Basic como expresiones algebraicas.

a) $y = 3*x^4 - 4*x^3 + x^2 - 10$

b) $y = B^2 - 4*A*C$

c) $y = (A^2 + B^2)^2$

d) $y = (A^2 + B^2)^{1/3}$

e) $n = (x+y) / (z+5)$

7. Evaluar las siguientes expresiones trabajando con los siguientes valores: A=4; B=5 y C=1:

a) $Z = 3*A - B^2 / 4*C$

b) $Z = (A+B) / 3^2$

c) $Z = (((B+C) / 2*A + 10) * 3 * B) - 6$

8. Evaluar las siguientes expresiones para valores de x=2; y=3 y z=2:

a) $d = x^y^z$

b) $d = x^{(y^z)}$

c) $d = x-y/z$

d) $d = (x-y)/z$

e) $d = 4*x+y+z$

f) $d = 4*(x+y)+z$

g) $d = 4*(x+y+z)$