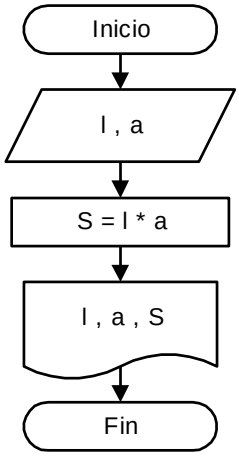
Capítulo 2
Procesos lineales

Procesos lineales

En esta primera etapa se comenzará a resolver problemas que requieren un simple proceso lineal sin bifurcaciones ni repeticiones. Estos programas son muy sencillos y su aplicación es limitada, pero constituyen el primer paso en el aprendizaje de la programación.
Tanto las bifurcaciones atadas a una condición, como las repeticiones o bucles se verán en los próximos capítulos.

Ejercicio 2.1:

Efectuar el diagrama de flujo de un programa que lea el largo l y el ancho a de un rectángulo, calcule su superficie S, e imprima el largo l, el ancho a y la superficie S obtenida.



Fórmula:
Superficie: S = l . a

Prueba de escritorio:

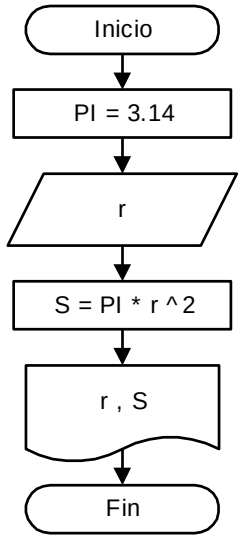
l	a	S
5,2	8,4	
		43,68

Salida por pantalla

```
 5,2     8,4     43,68
```

Ejercicio 2.2:

Efectuar el diagrama de flujo de un programa que lea el radio r de un círculo, calcule su superficie S, e imprima el radio r y la superficie S calculada.



Fórmula:
Superficie: $S = \pi . r^2$

Prueba de escritorio:

PI	r	S
3,14	10,15	
		31,871

Salida por pantalla

```
 10,15          31,871
```

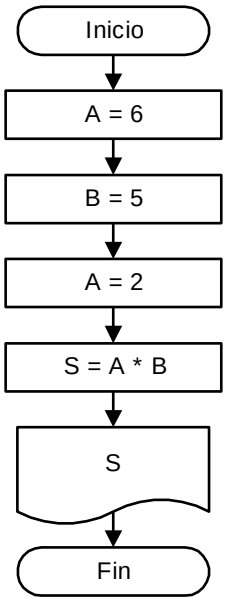
En el Ejercicio 2.2 se aprecia el uso de **una constante** PI. Una variable que mantiene su valor en un programa, y no lo cambia durante todo el proceso se conoce con el nombre de constante, y la asignación de su valor se efectúa generalmente al inicio del programa.

Prueba de escritorio y salida por pantalla

La prueba de escritorio que acompaña los dos primeros ejercicios responde a una ejecución del programa con datos hipotéticos, y en ella se analiza el valor que toman todas las variables a través del avance de cada proceso. También se agrega la salida por pantalla, que anticipa de una manera muy esquemática la información que se visualizará en el monitor.
Para distintas ejecuciones del mismo programa se pueden plantear distintos valores de datos, es decir, distintos valores del lado l y el ancho a en el primer ejercicio, o bien diferentes valores de radio r para el segundo ejercicio. **Cada ejecución tendrá su prueba de escritorio y su salida por pantalla** separada de la anterior. No se deben confundir las pruebas de escritorio de un programa lineal con un ciclo repetitivo como se verá más adelante.
La prueba de escritorio, según se ha visto hasta aquí, **es el monitoreo de todas las variables** que utiliza el programa, **y los distintos valores que van almacenando durante el proceso**. Para ello se deben anotar **todas** las variables como encabezamiento, y encolumnados hacia abajo todos los valores que almacenan dichas variables a medida que avanza el programa.
Las variables numéricas podrán almacenar valores numéricos, enteros o decimales, con puntos o con comas, y con la cantidad de decimales que les permita su definición. No pueden almacenar valores fraccionarios ni símbolos griegos, como tampoco cadenas alfanuméricas.
Las variables alfanuméricas podrán almacenar todo tipo de cadenas alfanuméricas, letras, números, símbolos, y cualquier combinación de ellos. De todas maneras, si las variables alfanuméricas almacenan números no podrán operar con ellos, es decir, no se podrán sumar o restar, etc.

Asignación de dos valores a una variable durante un programa

En los programas puede suceder que una variable reciba más de un valor durante el avance del flujo del diagrama. La variable actuará con el último valor que le haya sido asignado, ya que en cada transferencia se almacena un nuevo valor y el contenido anterior se pierde.
En el ejemplo siguiente se aprecia cómo funciona el almacenamiento de valores en variables escalares mediante un diagrama de flujo simple, con su prueba de escritorio y salida por pantalla.



Prueba de escritorio:

A	B	S
6	5	
2		
		10

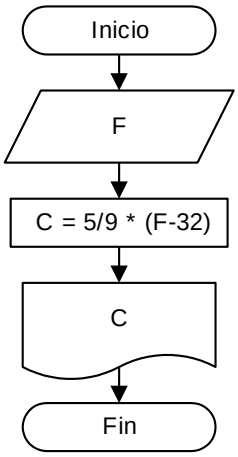
Salida por pantalla

10

El producto S = A*B ha sido efectuado con los valores 2 y 5, como resultado de la utilización de los últimos valores que tienen almacenadas las variables que participan en el producto. El valor 6 que almacenó en un principio la variable A fue desplazado por el almacenamiento de un nuevo valor igual a 2, con el cual prosigue el proceso y es el que se utilizará desde ese momento.

Ejercicio 2.3:

Efectuar el diagrama de flujo de un programa que convierta una temperatura ingresada en grados Fahrenheit a su equivalente en grados Celsius, e imprima el valor obtenido.



El procedimiento permite el ingreso de un dato y lo almacena en la variable F, luego calcula la fórmula de equivalencia y la almacena en la variable C para imprimirla posteriormente.

Fórmula:
Grados Celsius: C = 5/9 * (F-32)

Prueba de escritorio:

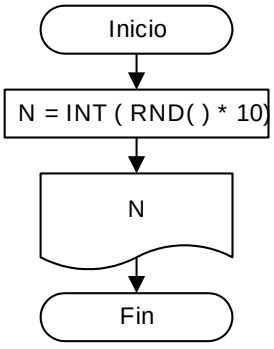
F	C
85	
	29,444

Salida por pantalla

29,444

Ejercicio 2.4:

Efectuar el diagrama de flujo de un programa que genere un número aleatorio N entre 0 y 10 en la computadora y lo imprima.



El procedimiento genera un número aleatorio en la computadora utilizando la función Rnd( ). Esta función genera valores decimales entre 0 y 1, por lo tanto se debe multiplicar el número generado por 10 y luego aplicarle la función Int ( ) que extrae la parte entera de un número con decimales, para obtener entonces un valor entero entre 0 y 10, que será almacenado en la variable N. Por último se imprime el valor de N.

Prueba de escritorio:

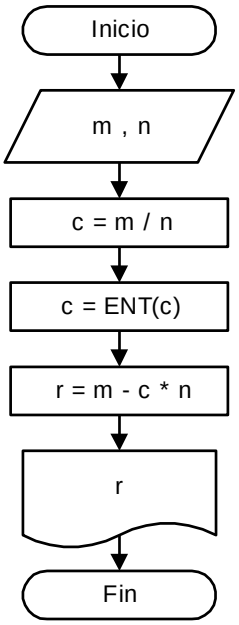
N
8

Salida por pantalla

8

Ejercicio 2.5:

Efectuar el diagrama de flujo de un programa que calcule el resto r de la división entre dos números enteros m y n, e imprima el valor del resto r obtenido.



El procedimiento consiste en calcular primero la división entre m y n, almacenándola en una variable, luego tomar la parte entera del resultado y multiplicarla nuevamente por el valor n, por último sustraer este producto del primer valor m.

Prueba de escritorio:

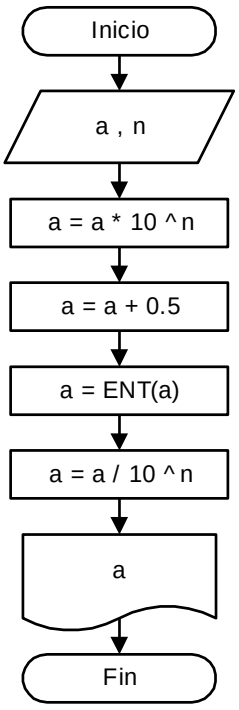
m	n	c	r
5	2		
		2,5	
		2	
			1

Salida por pantalla

```
1
```

Ejercicio 2.6:

Efectuar el diagrama de flujo de un programa que obtenga el redondeo de un número a con n decimales, e imprima el valor resultante.



El procedimiento utiliza la función ENTERO( ) como el ejercicio anterior, y además para determinar el criterio de redondeo se toma en cuenta que un número cuyo dígito decimal es 5 o superior será redondeado al valor inmediato superior.

Prueba de escritorio:

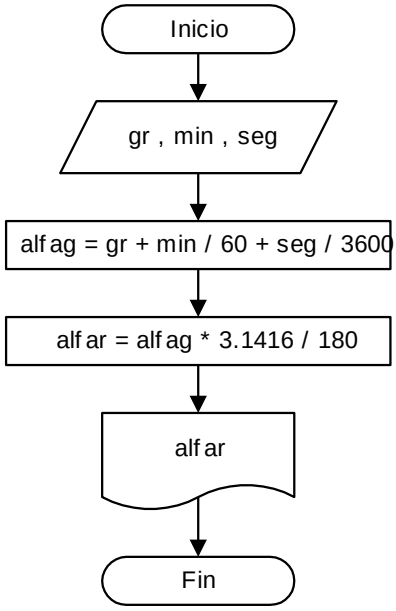
a	n
135,476	2
13547,6	
13548,1	
13548	
135,48	

Salida por pantalla

```
135,48
```

Ejercicio 2.7:

Efectuar el diagrama de flujo de un programa que calcule el valor en radianes de un ángulo, cuya medida se ha ingresado en formato sexagesimal, con grados, minutos y segundos.



El procedimiento calcula en primer lugar el valor del ángulo en formato sexagesimal incorporándole los minutos y los segundos como decimales. Luego se efectúa la conversión a radianes.

Fórmulas:

$$\alpha_{grados} = grados + \frac{\min utos}{60} + \frac{segundos}{3600}$$

$$\alpha_{radianes} = \alpha_{grados} * \frac{3.1416}{180}$$

Prueba de escritorio:

gr	min	seg	alfag	alfar .
45	27	40		
			45,461111	
				0,79345

Salida por pantalla

```
0,79345
```

Funciones del Visual Basic

El Visual Basic trae un conjunto de funciones de diferentes categorías, que resultan útiles para la solución de algunos problemas matemáticos que se ven en el curso. A continuación se verán algunas de las funciones con sus argumentos y ejemplos. El resto de las funciones se pueden consultar en el menú Ayuda del entorno de programación Visual Basic del Excel.

Función Rnd( )

Sintaxis:
Rnd[(*número*)]

Comentarios:
La función Rnd( ) devuelve un número aleatorio decimal de simple precisión.
El argumento opcional *número* es cualquier expresión numérica válida.
La función Rnd( ) devuelve un valor menor que 1 pero mayor o igual a cero. El valor del argumento *número* determina cómo la función Rnd( ) genera el número aleatorio: Para cualquier valor de semilla inicial se genera la misma secuencia de números. Esto es debido a que cada llamada sucesiva a la función Rnd( ) usará el número anterior como valor de semilla para el siguiente número de la secuencia.
Antes de llamar a la función Rnd( ), puede utilizar la instrucción **Randomize** sin argumento para inicializar el generador de números aleatorios con un valor de semilla basado en el reloj del sistema.
Para producir enteros aleatorios en un intervalo dado, use esta fórmula:

```
Int((Límite_superior - límite_inferior + 1) * Rnd( ) + límite_inferior)
```

Aquí, límite_superior es el número mayor del intervalo y límite_inferior es el número menor del intervalo.

Ejemplos:

```
A = Rnd( )                      ' Genera un valor aleatorio entre 0 y 1
B = Int((6 * Rnd) + 1)          ' Genera un valor aleatorio entre 1 y 6
N = Int(10 * Rnd)               ' Genera un valor aleatorio entre 0 y 10
MsgBox  Int((100 * Rnd) + 1)    ' Imprime un valor aleatorio entre 1 y 100
```

Instrucción Randomize

Sintaxis:
Randomize [*número*]

Comentarios:
La instrucción Randomize inicializa el generador de números aleatorios.
El argumento opcional *número* es cualquier expresión numérica válida.
La instrucción Randomize utiliza el argumento opcional *número* para inicializar el generador de números aleatorios de la función Rnd( ) y le asigna un nuevo valor de semilla. Si se omite *número*, el valor devuelto por el reloj del sistema se usa como el nuevo valor de semilla.

Si no utiliza la instrucción Randomize, la función Rnd (sin argumentos) utiliza el mismo número como valor de semilla la primera vez que se la invoca, usando después como valor de semilla el último número generado.
Nota: Para repetir secuencias de números aleatorios, llame a la función Rnd con un argumento negativo antes de utilizar la instrucción Randomize con un argumento numérico. Al utilizar la instrucción Randomize con el mismo valor de número no se repite la secuencia anterior.

Ejemplo:
En este ejemplo se utiliza la instrucción Randomize para inicializar el generador de números aleatorios. Al omitirse el argumento *número*, Randomize utiliza el valor de retorno de la función **Timer** un nuevo valor de comienzo.

```
Dim MiValor as Integer          ' Dimensiona la variable MiValor como variable entera
Randomize                       ' Inicializa el generador de números aleatorios
MiValor = Int((6 * Rnd) + 1)    ' Almacena en MiValor un número aleatorio entre 1 y 6
```

Función Int( )

Sintaxis:
Int(*número***)**

Comentarios:
La función Int( ) devuelve la parte entera de un número
El argumento *número* es cualquier expresión numérica válida, o sea, un número, una fórmula o una variable numérica decimal de precisión simple o doble.
Las funciones Int( ) y Fix( ) eliminan la fracción de un número y devuelven el valor entero resultante. La diferencia entre Int( ) y Fix( ) es que si el número es negativo, Int( ) devuelve el primer entero negativo menor o igual a número, mientras que Fix( ) devuelve el primer entero negativo mayor o igual a número. Por ejemplo, Int( ) convierte -8.4 en -9 a diferencia de Fix( ) que convierte -8.4 a -8.

Ejemplos:

```
Dim N as Integer          ' Dimensiona la variable N como entera
N = Int(99.8)             ' Asigna el valor 99 a la variable N
N = Fix(99.2)             ' Asigna el valor 99 a la variable N

N = Int(-99.8)            ' Asigna el valor -100 a la variable N
N = Fix(-99.8)            ' Asigna el valor   -99 a la variable N

MsgBox  Int(-99.2)        ' Imprime el valor -100
MsgBox  Fix(-99.2)        ' Imprime el valor   -99
```

Función Abs( )

Sintaxis:
Abs(*número***)**

Comentarios:
El valor absoluto de un número es igual a su magnitud sin el signo. La función Abs( ) devuelve un valor del mismo tipo que el que se pasó como parámetro y que especifica el valor absoluto del número. El argumento *número* puede ser cualquier expresión numérica válida, o sea, un número, una operación matemática o una variable numérica de cualquier tipo. Si el argumento *número* contiene Null, la función devolverá Null; si es una variable no inicializada, devolverá cero.

Ejemplos:

```
Dim M as Single          ' Dimensiona la variable M como decimal de simple precisión
M = Abs(50.3)            ' Almacena en M el valor 50.3
M = Abs(-50.3)           ' Almacena en M el valor 50.3

MsgBox (-24.36)          ' Imprime el valor 24.36
MsgBox (-1)              ' Imprime el valor 1
```

Operador Mod

Sintaxis:
variable = número1 **Mod** *número2*

Comentarios:
El operador Mod divide dos números y devuelve sólo el resto.

El operador de módulo o resto divide *número1* por *número2* redondeando los números decimales a enteros y devuelve sólo el resto como resultado, asignándolo a la *variable* que se encuentra a la izquierda del signo igual. En el siguiente ejemplo A recibe el valor 4.

```
A = 19 Mod 5
```

Generalmente el tipo de dato del resultado es Integer o entero. La parte fraccionaria se trunca. Sin embargo, si cualquiera de las expresiones es Null, el resultado será Null. Toda expresión Empty se considera como 0.

Ejemplos:
En estos ejemplos se utiliza el operador Mod para dividir dos números y obtener como resultado sólo el resto de la división. Si uno de los números es decimal, se redondea primero para convertirlo en un entero.

```
Dim MiResultado as Integer     ' Dimensiona la variable MiResultado como entera
MiResultado = 10 Mod 5         ' Almacena en MiResultado el valor 0
MiResultado = 10 Mod 3         ' Almacena en MiResultado el valor 1
MiResultado = 12 Mod 4.3       ' Almacena en MiResultado el valor 0
MiResultado = 12.6 Mod 5       ' Almacena en MiResultado el valor 3
```

Función Sin( )

Sintaxis:
Sin(*ángulo***)**

Comentarios:
La función Sin( ) calcula el seno de un ángulo dando como resultado un número decimal de doble precisión. El resultado entra dentro del intervalo -1 a 1.
El argumento *ángulo* es cualquier expresión numérica válida que exprese un ángulo en radianes. Para convertir grados a radianes, multiplique los grados por π/180. Para convertir radianes a grados, multiplique los radianes por 180/π.

Función Cos( )

Sintaxis:
Cos(*ángulo***)**

Comentarios:
La función Cos( ) calcula el coseno de un ángulo dando como resultado un número decimal de doble precisión. El resultado entra dentro del intervalo -1 a 1.
El argumento *ángulo* es cualquier expresión numérica válida que exprese un ángulo en radianes. Para convertir grados a radianes, multiplique los grados por π/180. Para convertir radianes a grados, multiplique los radianes por 180/π.

Función Tan( )

Sintaxis:
Tan(*ángulo***)**

Comentarios:
La función Tan( ) calcula la tangente de un ángulo dando como resultado un número decimal de doble precisión. El argumento *ángulo* es cualquier expresión numérica válida que exprese un ángulo en radianes. Para convertir grados a radianes, multiplique los grados por π/180. Para convertir radianes a grados, multiplique los radianes por 180/π.

Ejemplos:

```
Dim ang as Double       ' Dimensiona la variable ang como decimal de doble precisión
ang = 1.3               ' Almacena en la variable ang el valor del ángulo en radianes
MsgBox (Sin(ang))       ' Imprime el seno del ángulo almacenado en ang
MsgBox (Cos(ang))       ' Imprime el coseno del ángulo almacenado en ang
MsgBox (Tan(ang))       ' Imprime la tangente del ángulo almacenado en ang
```

Función Log( )

Sintaxis:
Log(*número***)**

Comentarios:
La función Log( ) devuelve el logaritmo natural de un número. El logaritmo natural es el logaritmo en base e. El valor de la constante e es 2.718282 aproximadamente. El argumento *número* es cualquier expresión numérica válida mayor que cero.
Puede calcular logaritmos en base-n para cualquier número x dividiendo el logaritmo natural de x por el logaritmo natural de n de la siguiente manera:

```
Logn(x) = Log(x) / Log(n)
```

El ejemplo siguiente ilustra el cálculo del logaritmo en base 10 de un número almacenado en x:

```
Log10 = Log(x) / Log(10#)
```

Ejemplo:

```
Dim ang as Double                    ' Dimensiona la variable ang como doble precisión
ang = 1.3                            ' Define el valor del ángulo en radianes
Shi = Log(ang + Sqr(ang * ang + 1))  ' Calcula el seno hiperbólico inverso
MsgBox (Shi)                         ' Imprime el seno hiperbólico inverso
```

Función Sqr( )

Sintaxis:
Sqr(*número***)**

Comentarios:
La función Sqr( ) calcula la raíz cuadrada de un número. El argumento *número* es cualquier expresión numérica válida mayor o igual a cero.

Ejemplos:

```
Dim n as Double      ' Dimensiona la variable n como numérica de doble precisión
n = Sqr(4)           ' Almacena un 2 en la variable n
n = Sqr(23)          ' Almacena el valor 4.79583152331272 en la variable n
n = Sqr(0)           ' Almacena el valor 0 en la variable n
n = Sqr(-4)          ' Provoca un error en tiempo de ejecución
```

Ejercicios Capítulo 2:

1. Realizar el diagrama de flujo para sumar dos números enteros e imprimir el resultado.

2. Realizar el diagrama de flujo para calcular e imprimir la superficie de un triángulo en función de la base y la altura (S=1/2 x Base x Altura).

3. Realizar el diagrama de flujo de un proceso que lea una medida en centímetros y la transforme a pulgadas e imprima el resultado. (1 pulgada = 2.54 centímetros)

4. Realizar el diagrama de flujo de un proceso que lea un ángulo en grados y los convierta a radianes e imprima el resultado. (180 grados = 3.1416 radianes)

5. Realizar el diagrama de flujo de un proceso que lea el lado de un cubo y calcule su volumen.

6. Realizar el diagrama de flujo para convertir una temperatura en grados Fahrenheit a grados Celsius. Imprimir el resultado. (C = 5/9 * (F-32))

7. Realizar el diagrama de flujo para convertir una lectura en horas, minutos y segundos a formato de hora con punto decimal. Imprimir el resultado.

8. Realizar el diagrama de flujo para convertir una lectura en horas, minutos y segundos a su valor en segundos. Imprimir el resultado.

9. Realizar el diagrama de flujo para convertir una lectura en formato de hora con punto decimal a horas, minutos y segundos. Imprimir el resultado.

10. Realizar el diagrama de flujo para convertir una lectura en segundos a su valor en horas, minutos y segundos. Imprimir el resultado.

11. Realizar el diagrama de flujo de un proceso que permita ingresar un nombre, un documento de identidad, la fecha de nacimiento y la edad de una persona, y luego imprima los datos obtenidos.

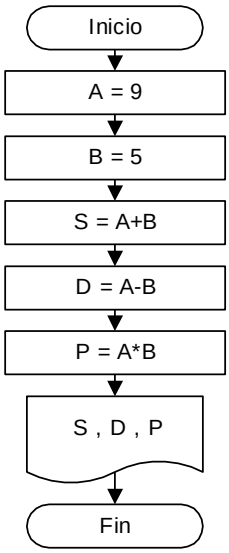
12. Realizar el diagrama de flujo de un proceso que permita calcular el precio de un artículo en el año 2020, con el ingreso del precio actual y del año actual, considerando además una tasa de interés constante anual de 22%. (La fórmula a aplicar es P = C * (1+i)^N)

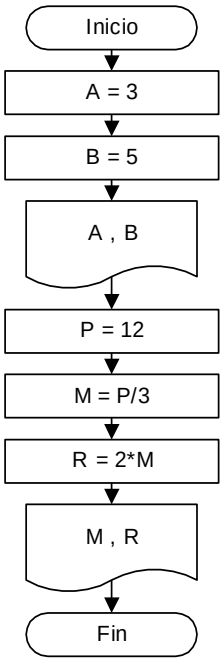
13. Realizar el diagrama de flujo que permita intercambiar los valores de dos variables A y B, ingresadas por teclado, y las imprima luego de efectuado el intercambio. Hacer la prueba de escritorio con el supuesto de A=32 y B=75.

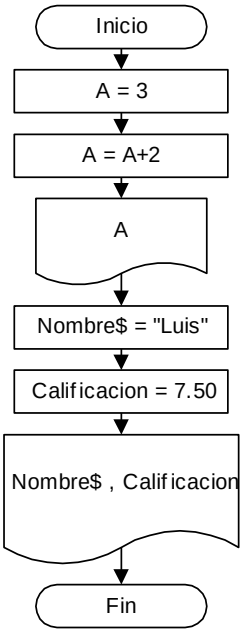
14. Realizar el diagrama de flujo de un programa que genere un número al azar entre 0 y 1.

15. Realizar el diagrama de flujo de un programa que genere un número entero al azar entre 0 y 10.

16. Realizar la prueba de escritorio y la salida por pantalla de los siguientes diagramas de flujo:







17. Realizar la prueba de escritorio y la salida por pantalla de los siguientes diagramas de flujo: