

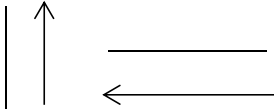
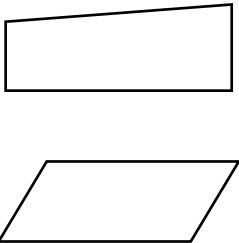
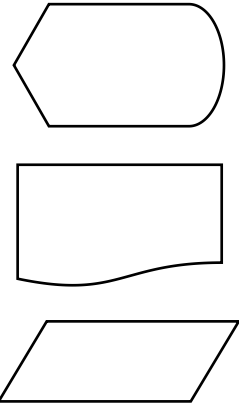


Elementos de diagrama de flujo

Ahora bien, todo módulo de programa, antes de ser escrito en un lenguaje de programación de computadoras, debe ser pensado, analizado y diseñado. Existen ciertas herramientas gráficas que nos ayudan a construir su diseño, lo que nos ayuda al análisis y validación de la propuesta de solución del problema que queremos resolver.

Estos elementos gráficos y su significado, se muestran en la tabla que se presenta a continuación:

Elemento	Descripción
	Inicio y fin de algoritmo / programa
	Proceso
	Flujo de los datos
	Entrada de datos Desde teclado Símbolo genérico para entrada/salida
	Salida de datos Monitor Impresora Símbolo genérico para entrada/salida

Conectores	
○	Dentro de la misma página
◩	Fuera de página

En general, un programa de computadora lo hemos representado de la siguiente manera:

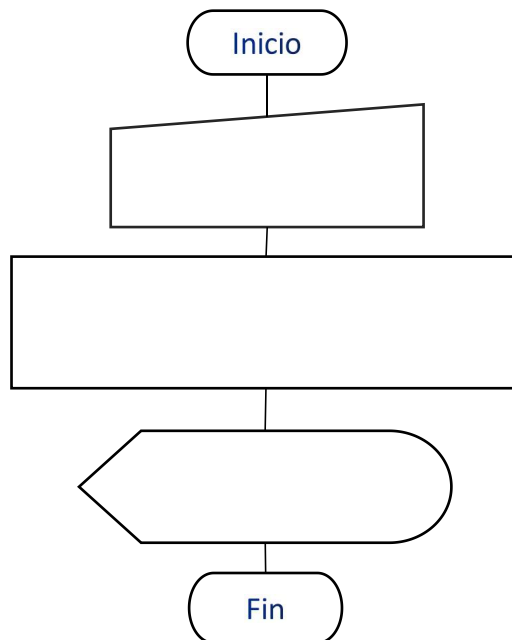


Pero para su diseño detallado utilizamos los elementos de diagrama de flujo arriba presentados, considerando los siguientes convencionalismos:

- Arriba del diagrama se coloca un nombre que sea representativo de la tarea que realiza. Este es el nombre del algoritmo.
- El inicio y el fin del proceso se representan por dos **elipses**: una al inicio con la palabra **Inicio** en su interior y otra al final con la palabra **Fin** en su interior.
- Por convención, los diagramas de flujo crecen verticalmente, de arriba hacia abajo y horizontalmente de izquierda a derecha.
- Para representar la secuencia del flujo y procesamiento de los datos dentro de un módulo de programa, los diferentes elementos presentados se conectan por medio de las **líneas de flujo de datos**, desde el inicio hasta el final.
- Entre dos elementos contiguos del diagrama, si el que sigue está por debajo del anterior, no es necesario colocar una **punta de flecha** en el extremo de la línea de flujo. Pero si el que le sigue está colocado por encima, la **punta de flecha** sí es conveniente colocarla.
- Entre dos elementos contiguos del diagrama, si el que sigue está a la derecha del anterior, no es necesario colocar una **punta de flecha** en el extremo de la línea de flujo. Pero si el que le sigue está colocado a la izquierda, la **punta de flecha** sí es conveniente colocarla.
- En el interior de elementos, como los de entrada, salida y proceso se colocan las correspondientes variables y expresiones involucradas del problema. Dentro de los símbolos de entrada se listarán las **variables que capturan los datos de entrada**, dentro de los símbolos de salida se colocarán las **variables que se utilizan en las instrucciones de despliegue** y dentro de los símbolos de proceso se colocarán las **expresiones de asignación** utilizadas.

Por ejemplo, los programas que hasta ahora hemos escrito, cuyo propósito ha sido demostrar el uso de los operadores fundamentales, han sido relativamente sencillos y todos tienen en común que se esbozan gráficamente utilizando el siguiente esquema de diagrama de flujo:

Nombre del algoritmo/módulo



Lo que diferencia a uno de otro son los detalles que se escriben dentro de cada elemento del diagrama.

Por ejemplo, los diagramas de flujo de dos programas que se solicitan en una hoja de Laboratorio son los siguientes:

Elabore un programa que reciba el radio de un círculo y calcule su perímetro y área.

```
#include "iostream"
#include "cmath"

using namespace std;

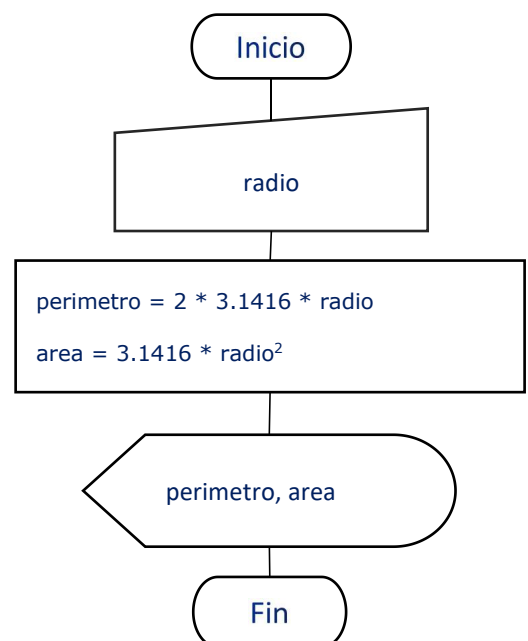
int main(void)
{
    float radio, perimetro, area;

    cout << "CÁLCULO DEL PERÍMETRO Y ÁREA DE UN CÍRCULO" << endl << endl;

    cout << "Digite el radio de la circunferencia: ";
    cin >> radio;

    perimetro = 2 * 3.1416 * radio;
    area = 3.1416 * pow(radio, 2);

    cout << "El perímetro es: " << perimetro << endl;
    cout << "El área es: " << area << endl;
}
```



Elabore un programa que reciba los datos sobre un producto comprado: nombre, precio y la cantidad de unidades compradas. Luego calcule y muestre el total a pagar.

```
#include "iostream"
#include "cmath"
#include "string"

using namespace std;

int main(void)
{
    string nombreProducto;
    float precio, totalPagar;
    int cantidad;

    cout << "TOTAL A PAGRA POR COMPRA DE UN PRODUCTO" << endl << endl;

    cout << "Nombre del producto: ";
    cin >> nombreProducto;
    cout << "Precio: ";
    cin >> precio;
    cout << "Cantidad: ";
    cin >> cantidad;

    totalPagar = cantidad * precio;

    cout << "Por la compra de " << cantidad << endl;
    cout << "unidades de " << nombreProducto << endl;
    cout << "debe pagar: $" << totalPagar << endl;
}
```

