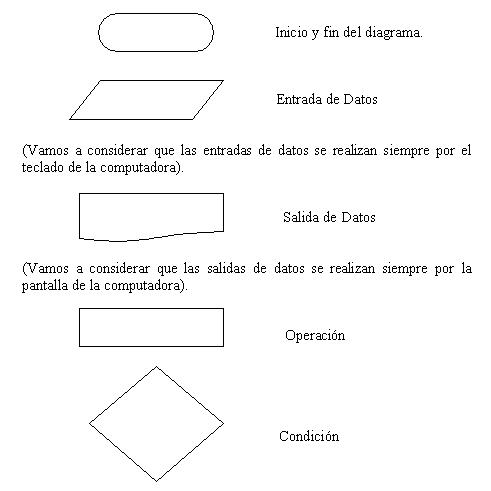
**¿Qué es un programa?**

Programa: Conjunto de instrucciones que entiende una computadora para realizar una actividad.  
Todo programa tiene un objetivo bien definido: un procesador de texto es un programa que permite cargar, modificar e imprimir textos, un programa de ajedrez permite jugar al ajedrez contra el ordenador u otro contrincante humano.  
La actividad fundamental del programador es resolver problemas empleando el ordenador como herramienta fundamental.

Para la resolución de un problema hay que plantear un algoritmo.  
Algoritmo: Son los pasos a seguir para resolver un problema.

**Diagrama de flujo**

Un diagrama de flujo es la representación gráfica de un ALGORITMO.



**Planteo de un problema utilizando diagramas de flujo.**

Para plantear un diagrama de flujo debemos tener muy en claro el problema a resolver.  
Ejemplo : Calcular el sueldo mensual de un operario conociendo la cantidad de horas trabajadas y el pago por hora.

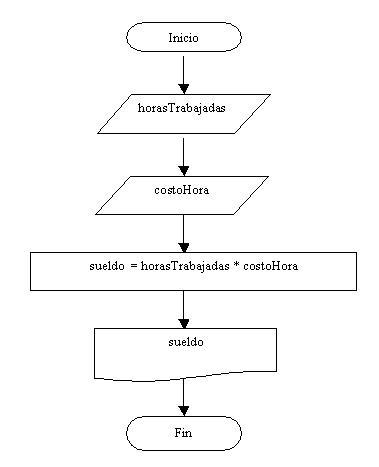
Podemos identificar:

Datos conocidos:  
Horas trabajadas en el mes.  
Pago por hora.

Proceso:  
Cálculo del sueldo multiplicando la cantidad de horas por el pago por hora.

Información resultante:  
Sueldo mensual.

Si hacemos un análisis todo problema está constituido por:  
- Datos conocidos: Datos con los que se cuenta al plantear el problema.  
- Proceso: Operaciones a realizar con los datos conocidos.  
- Información resultante: Es la información que resuelve el problema.



**Codificación del problema con el lenguaje C.**

No debemos perder de vista que el fin último es realizar un programa de computación que permita automatizar una actividad para que muchos procesos sean desarrollados por la computadora.  
El diagrama de flujo es un paso intermedio para facilitarnos la resolución del problema.  
El paso siguiente es la codificación del diagrama de flujo en un lenguaje de computación, en nuestro caso emplearemos el lenguaje C.

Lenguaje de computación: Conjunto de instrucciones que son interpretadas por una computadora para realizar operaciones, mostrar datos por pantalla, sacar listados por impresora, entrar datos por teclado, etc.

**Conceptos básicos para codificar un programa.**

Variable: Es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo.

Para el ejemplo planteado en el concepto anterior la variable horasTrabajadas almacena la cantidad de horas trabajadas por el operario. La variable valorHora almacena el precio de una hora de trabajo. La variable sueldo almacena el sueldo a abonar al operario.  
En el ejemplo tenemos tres variables.

Tipos de variable:  
Una variable puede almacenar:  
- Valores Enteros (100, 260, etc.)  
- Valores Reales (1.24, 2.90, 5.00, etc.)  
- Cadenas de caracteres ("Juan", "Compras", "Listado", etc.)

Elección del nombre de una variable:  
Debemos elegir nombres de variables representativas

Hay que tener en cuenta que el lenguaje C no se ha creado teniendo como objetivo el aprendizaje de la programación.

La codificación del diagrama de flujo la haremos dentro de la función main (la función main es la primera que se ejecuta al iniciarse un programa en el lenguaje C)

Tener en cuenta que debemos ingresar el caracter "." en lugar del caracter "," si ingresamos un costo por hora con decimales.

Conceptos que deben quedar claros:

1. Por el momento haremos todo el algoritmo dentro de la función main.
2. Si observamos el diagrama de flujos vemos que debemos definir tres variables La cantidad de horas normalmente será un valor entero (ej. 100 - 150 - 230 etc.), pero el costo de la hora es muy común que sea un valor real (ej. 5.35 - 7.50 etc.) y como el sueldo resulta de multiplicar las horas trabajadas por el costo por hora el mismo deberá ser real.

Utilizamos la palabra clave int para definir variables enteras (en C las palabras claves deben ir obligatoriamente en minúsculas, sino se produce un error sintáctico) Luego de la palabra clave debemos indicar el nombre de la variable

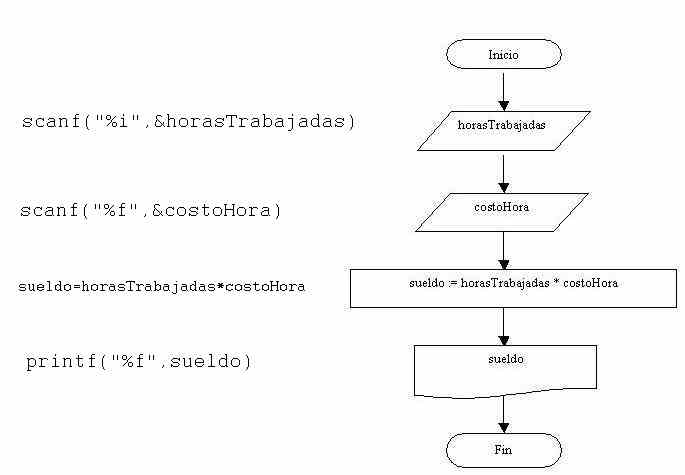
Se propone que el nombre de la variable comience con minúscula y en caso de estar constituida por dos palabras o más deben ir en mayúsculas el primer caracter (un nombre de variable no puede tener espacios en blanco, empezar con un número, ni tampoco utilizar caracteres especiales)  
Debemos buscar siempre nombres de variables que nos indiquen que almacenan

1. Para mostrar mensajes en la pantalla utilizamos la función "printf":Con esta sintaxis todo lo que se encuentra contenido entre comillas aparecerá exactamente en la pantalla.  
   Si necesitamos mostrar el contenido de una variable debemos disponer el caracter %f si mostramos una variable float o %i si mostramos una variable int
2. Para hacer la entrada de datos por teclado en C debemos utilizar la función "scanf"

La función scanf tiene un primer parámetro entre comillas dobles y dentro el caracter % y la letra i si se carga un entero y la letra f si se carga un float

El segundo parámetro es el nombre de la variable a cargar antecedida por el caracter &.

1. Las operaciones que indicamos en el diagrama de flujo mediante la figura rectángulo la codificamos tal cual
2. Por el momento veremos que siempre nuestro programa finaliza llamando a la función getch() (detiene la ejecución del programa hasta que se presiona una tecla) y la instrucción return 0 (le avisa al sistema operativo que el programa finalizó correctamente):



En el diagrama de flujo no indicamos la definición de variables

No representamos con símbolos los mensajes a mostrar previo a la carga de datos por teclado

El include permite importar librerías de funciones contenidos en otros archivos

Las líneas:

getchar();

No permite detener la ejecución del programa para ver el sueldo del operario y esperar a que se pulse una tecla.

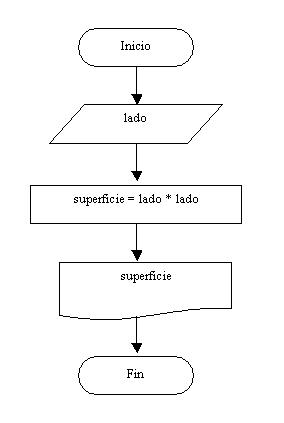
**Errores sintácticos y lógicos**

A los errores tipográficos, como por ejemplo la falta de puntos y comas, nombres de variables incorrectas, falta de paréntesis, palabras claves mal escritas, etc. los llamamos errores SINTACTICOS.  
Un programa no se puede ejecutar sin corregir absolutamente todos los errores sintácticos.

Existe otro tipo de errores llamados ERRORES LOGICOS. Este tipo de errores en programas grandes (miles de líneas) son más difíciles de localizar. Por ejemplo un programa que permite hacer la facturación pero la salida de datos por impresora es incorrecta.

**Problema:**

Hallar la superficie de un cuadrado conociendo el valor de un lado.



Hay muchos tipos de errores sintácticos que nos pueden suceder como que nos falte punto y coma al final de cada instrucción, definir variables en forma incorrecta, nombres de funciones con caracteres equivocados, etc

**Estructura de programación secuencial**

Cuando en un problema sólo participan operaciones, entradas y salidas se la denomina una estructura secuencial.  
Los problemas diagramados y codificados previamente emplean solo estructuras secuenciales.

La programación requiere una práctica ininterrumpida de diagramación y codificación de problemas.