**Programación**

¿Qué es programar?

Traducir una idea a un lenguaje de programación para resolver un problema

Lenguaje de alto/bajo nivel

* Alto nivel: cercano a la lengua humana
* Bajo nivel: cercano al lenguaje maquina (0 y 1)

Paradigmas de programación

Un paradigma de programación es una forma de programar

* Estructurada: programar de una forma lineal
* Orientada a objetos: programar en los recursos disponibles a tu alcance
* Orientada a aspectos
* Paralela
* Procesos e hilos
* Cíclica: programar en continuo funcionamiento

Concepto de variable

* Que varía o puede variar
* Que está sujeto a cambios frecuentes
* En programación, las variables son espacios reservados en la memoria que pueden cambiar de contenido.
  + Tipos de variables: Los diferentes lenguajes utilizan diferentes tipos de variables.
    - Tipos genéricos: Numérico, texto, lógico
    - Mas usados: int (entero), char (caracteres), float double (número real)
    - Prefijos: unsigned, short, long

Otros tipos de variables más complejas según el tipo de programación pueden ser:

* + - Vercotres (arrays)
    - Matrices
    - Objetos
  + Operaciones con variables
    - Ariméticas (+ - \* / % += -= ++ --)
    - Comparación (> >= < <= !=)
    - Lógicas o booleanas (AND, OR, NOT)
    - Entrada y salida (leer, escribir)
    - Asignación o comparación (a=1 a==1)
    - Operación con memoria del sistema (constructores)
    - Operaciones con cadenas de texto (concatenar, contar letras…)
  + Ámbito de las variables
    - Tipado fuerte o débil, dinámico o estático
    - Global
    - Local
    - Constante

Algoritmo

Conjunto de instrucciones ordenadas que conducen a la solución de un problema

Características que debe cumplir el algoritmo:

* Debe ser preciso, indicar de manera inequívoca que se debe hacer.
* Debe ser finito: número limitado de pasos
* Debe ser definido: producir los mismos resultados todo el rato (con las mismas entradas)
* Puede tener cero o más elementos de entrada
* Debe producir un resultado

Partes de un algoritmo

* Entrada de datos:
* Proceso
* Salida de resultados

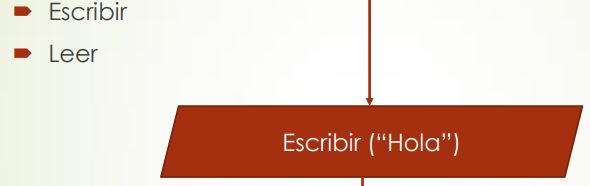
Los algoritmos se pueden representar mediante: pseudocódigo o diagramas de flujo

Traza

Indica la secuencia de acciones y el valor de las variables después de cada instrucción

Realizar un algoritmo

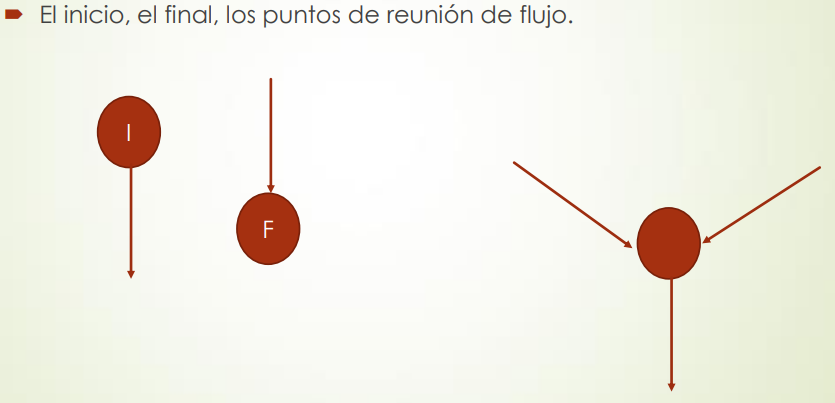
* Para realizar una operación de entrada o salida



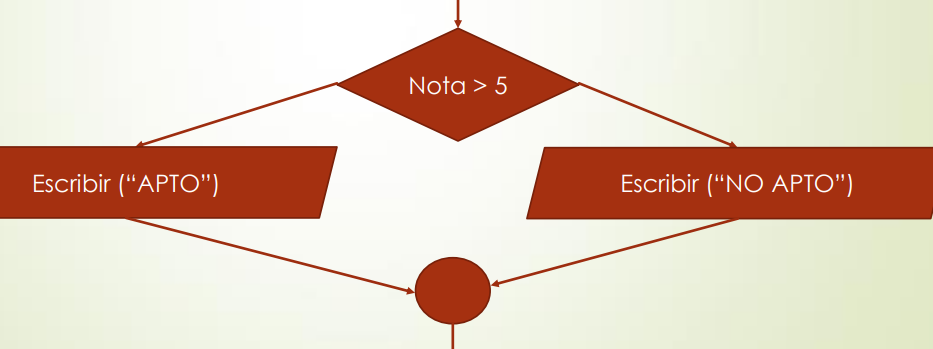
* Realizar un algoritmo (instrucción, operaciones)



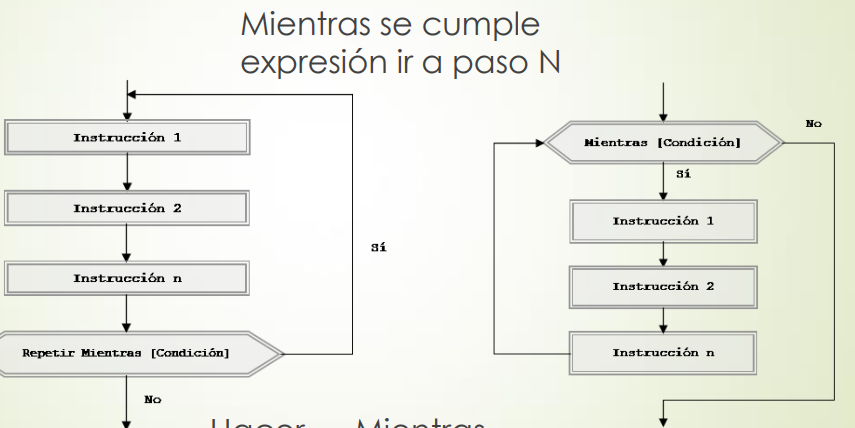
* Inicio, final o punto de reunión



* Estructura condicional



* Estructura iterativa



Del código fuente al ejecutable

Los programas se escriben en los lenguajes de programación creando así su código fuente, entonces el compilador es encargado de traducir este código para convertirlo en un fichero ejecutable.

Pueden surgir varios errores

* Errores de sintaxis: instrucciones mal escritas
* Errores léxicos: Palabras propias del lenguaje mal escritas
* Errores semánticos o lógicos: no se detectan por el compilador. El programa se ejecuta, pero no realiza su función correctamente.

Una vez compilado el código se genera un archivo objeto el cual será enlazado por un enlazador para generar el archivo ejecutable

El depurador es una herramienta que permite identificar errores en un programa. Mediante:

* Colocar puntos de parada
* Ejecutar el programa en modo depuración
* Ejecutar el programa paso a paso
* Inspeccionar variables

¿Por qué usar c?

* Es un lenguaje multiplataforma: los programas pueden ser compilados para Windows, Linus o Mac.

Partes de un fichero en C++

* Librería:
* Declaración, definición y llamada: crear una función, explicar qué va a hacer y su llamada
* Función principal: int main
* Comentarios: texto en el código que el compilador ignora