

Autoestudio 1: Programas, Compiladores y Editores

El proceso de Creación de un programa

Un algoritmo como hemos visto es una serie de instrucciones como por ejemplo:

Algoritmo para encontrar el número más grande en un conjunto

1. seleccionar el primer número del conjunto
2. marcarlo como “el número más grande”
3. comparar “el número más grande” con el siguiente número
 - a. si el siguiente es mayor marcar éste como “el número más grande”
4. Si todavía queda un número siguiente regresar a al paso 3
5. Imprimir en pantalla el número marcado como “el número más grande”

Estas series de instrucciones están en un lenguaje que los humanos podemos entender, sin embargo para que las computadoras puedan entenderlos necesitamos traducir las instrucciones al idioma de la máquina. Para esto tenemos un lenguaje intermedio, esto es un **lenguaje de programación**.

El **código fuente** del programa del algoritmo anterior se expresaría en el lenguaje de programación **C** de la siguiente forma:

```
int i, max;
int arg[] = {1,2,3,77,88,9,7,6,43,2,1,3,0,32,100,4,55,14,21,23};
max = arg[0];
for(i = 0; i < 20; i++){
    if(arg[i] > max ){
        max = arg[i];
    }
}
printf("el num mayor es %i", max);
```

Un programa se escribe en un **lenguaje de programación**, y la actividad que consiste en expresar un algoritmo en un lenguaje de programación se llama **programar**.

Cuando el programa está completo, llevamos a cabo un proceso que se llama **compilar** esto es cuando la computadora traduce automáticamente el lenguaje de programación al lenguaje en el que la computadora lo puede entender. Cuando **compilamos** un programa sin errores generamos un archivo **ejecutable**, estos ejecutables son los programas que usamos como Word (word.exe), Explorer (iexplore.exe), Fire Fox (firefox.exe), Star Craft (starcraft.exe), Angry Birds (angrybirds.exe), Itunes (itunes.exe), BitTorrent (BitTorrent.exe) etc ... Nota que todos los ejecutables tienen al final un “.exe” esto se conoce como la extensión del archivo.

Un archivo ejecutable (.exe) por dentro se ve como miles de caracteres extraños que nosotros no entendemos pero la computadora sí. Por eso nosotros trabajamos sólo hasta el nivel del lenguaje de programación y lo demás lo hace la computadora por sí misma.

Como hemos visto, para llevar a cabo un proceso en una computadora es preciso:

1. Diseñar un algoritmo que describa cómo se debe realizar el proceso.
2. Expresar el algoritmo en un lenguaje de programación.
3. Compilar el programa para ver que está bien escrito.
4. Ejecutar el programa con la computadora.

El programa

Un programa es un conjunto de instrucciones escritas en un determinado **lenguaje de programación**, cuyo objetivo es **instruir a la computadora** para que lleve a cabo una función específica.

Como en cualquier lenguaje, por ejemplo, español o inglés, un programa debe cumplir con las reglas particulares del lenguaje de programación:

- **Reglas de sintaxis:** por ejemplo, las partes de una instrucción deben seguir un orden específico.

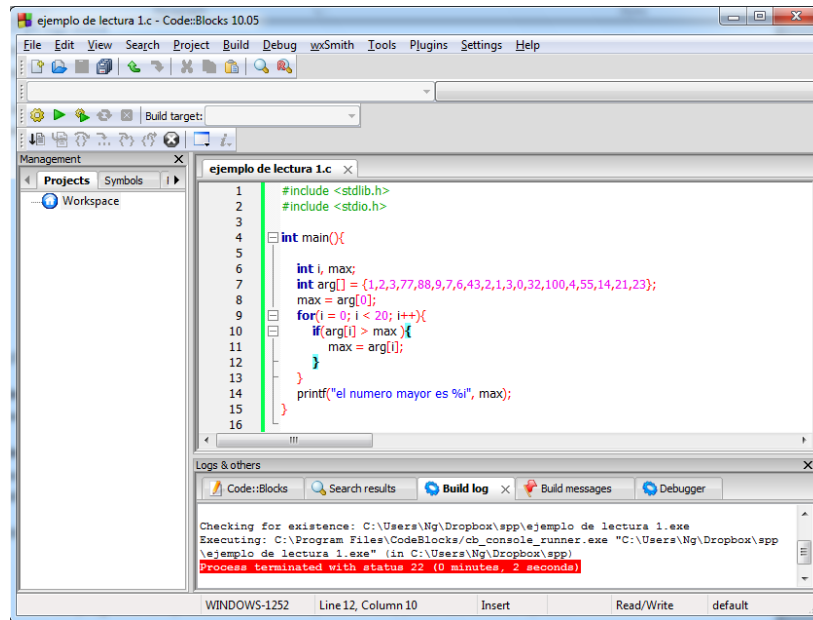
Ejemplo: quiero ir al cine. No es lo mismo que: al quiero cine ir

Ejemplo: $A + B$ No es lo mismo que: $+ A B$

- **Reglas de flujo** respecto a la secuencia de las instrucciones. Esto es en qué orden dentro del programa se llevan a cabo las instrucciones.

¿Qué se requiere para escribir un programa?

Un editor. Es una herramienta de software que facilita la escritura de programas. Es un software parecido a un “procesador de palabras” (notepad). En nuestro caso usaremos **codeblocks**. Y se ve así:



Un compilador. Es una herramienta que traduce un código fuente al lenguaje máquina para posteriormente enlazarlo con las librerías de C, y obtener así un archivo ejecutable. En nuestro caso utilizaremos el compilador **gcc** que viene dentro de [MinGW](#).

Tipos de Lenguajes

Hay muchos lenguajes de programación, cada uno está diseñado para programar en ambientes diferentes, por ejemplo, lenguajes de alto nivel más conocidos son **C** y **Java** estos serían el equivalente a lenguajes genéricos, pero también hay lenguajes enfocados a matemáticas como **MATLAB**, **R** y **MATHEMATICA**, lenguajes de programación lógica **Scheme**, **Prolog**, lenguajes script para desarrollos en internet **python**, **php**, **javascript**, lenguajes para el manejo de gráficas como **kuda**, lenguajes para el desarrollo de plataformas móviles **objectiveC** o **andriod**, etc... Estos son sólo algunos ejemplos, pero todos estos lenguajes se pueden clasificar en dos grupos: Interpretados y Compilados.

- Un **programa interpretado** se traduce y ejecuta instrucción por instrucción
- Un **programa compilado** se traduce completamente y posteriormente se ejecuta

Los interpretados no generan un ejecutable, los compilados sí. Los interpretados son utilizados normalmente en lenguajes de tipo script y su uso predomina en la Internet. Los compilados predominan en las aplicaciones de escritorio.

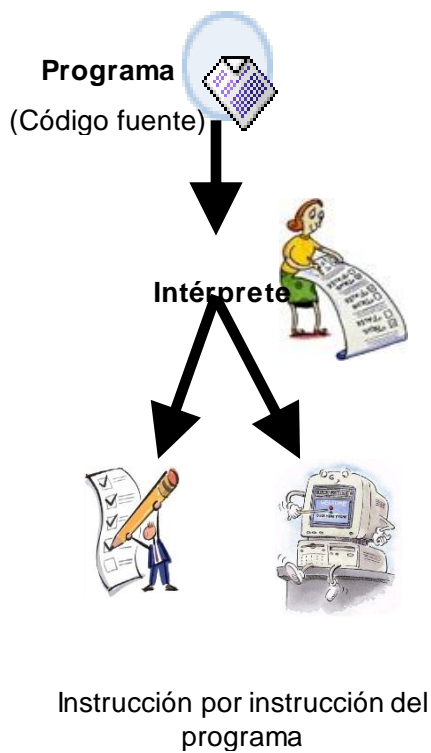
programa interpretado:	programa compilado:
Traduce a lenguaje máquina instrucción por instrucción y las va ejecutando (mientras traduce ejecuta)	Traduce a lenguaje máquina todo el programa, pero no lo ejecuta
No genera un archivo ejecutable (*.exe)	Genera un archivo ejecutable (*.exe)

Si lo quieres volver a ejecutar necesitas que el intérprete lo vuelva a traducir y ejecutar instrucción por instrucción

Si lo quieres ejecutar basta invocar al archivo ejecutable

En la siguiente página se presenta el diagrama de proceso de ejecución para programas interpretados y para programas compilados. Nosotros crearemos **algoritmos** que **programaremos** en **C** que es un **lenguaje de programación compilado** usando el editor **CodeBlocks** y el compilador **gcc**.

Programa Interpretado



Programa compilado

