

FUNDAMENTOS DE PROGRAMACIÓN

Inicio Clase 01

Profesor: Carlos Díaz

Clase 01: Introducción a la programación

- Lenguaje máquina
- Lenguajes ensambladores
- Lenguajes de bajo y alto nivel
- Lenguaje interpretado y compilado
- Creación de un programa C++ ejecutable
- Lenguajes orientados a procedimientos y a objetos
- Desarrollo del C++

Lenguaje máquina

- Es el único lenguaje que en realidad entiende la computadora.
- Consisten en una secuencia de instrucciones compuestas por números binarios. Ejemplo:

11000000 0000000000001 0000000000010

11110000 0000000000010 0000000000011

operación dirección1 dirección2

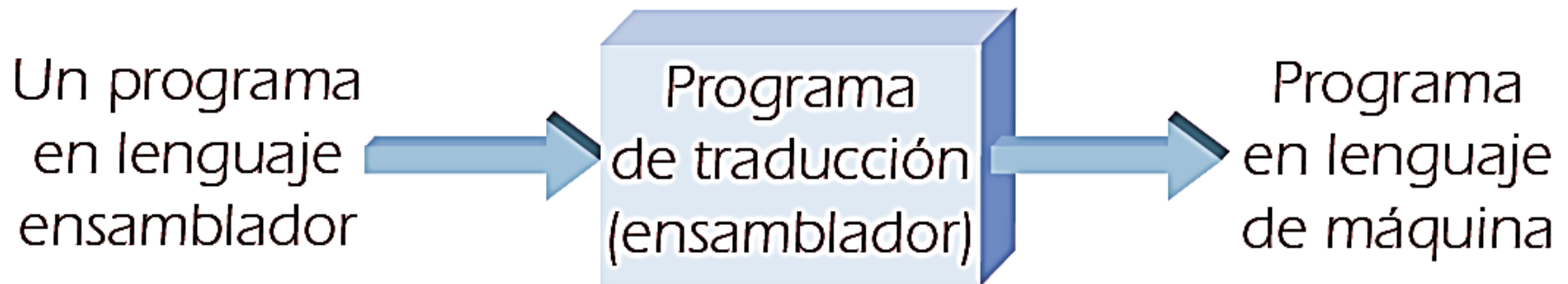
- El conjunto de bits izquierdo de la instrucción indica la operación a realizar (sumar, multiplicar, etc.), mientras que los derechos especifican las direcciones de la memoria donde están los datos.

Lenguajes ensambladores

- Programar en lenguaje máquina es muy lento y tedioso.
- Uno de los primeros avances en la programación fue la sustitución de los números binarios por palabras y números hexadecimales.

Ejemplo:

ADD	1	2
MUL	2	1
operación	dirección1	dirección2



Lenguajes de bajo y alto nivel

- Los lenguajes de máquina y ensambladores se denomina **lenguajes de bajo nivel** porque utilizan instrucciones que se vinculan directamente con un tipo de computadora.
- Esto puede ocasionar problemas porque el programa en bajo nivel solo funcionará en ese tipo de computadora y no en otra.
- En contraste los **lenguajes de alto nivel** usan instrucciones que se parecen a los lenguajes escritos, como el inglés, y pueden ejecutarse en una variedad de computadoras: Visual Basic, C, C++, Java, etc.

Ejemplo:

resultado = (primero + segundo) * tercero;

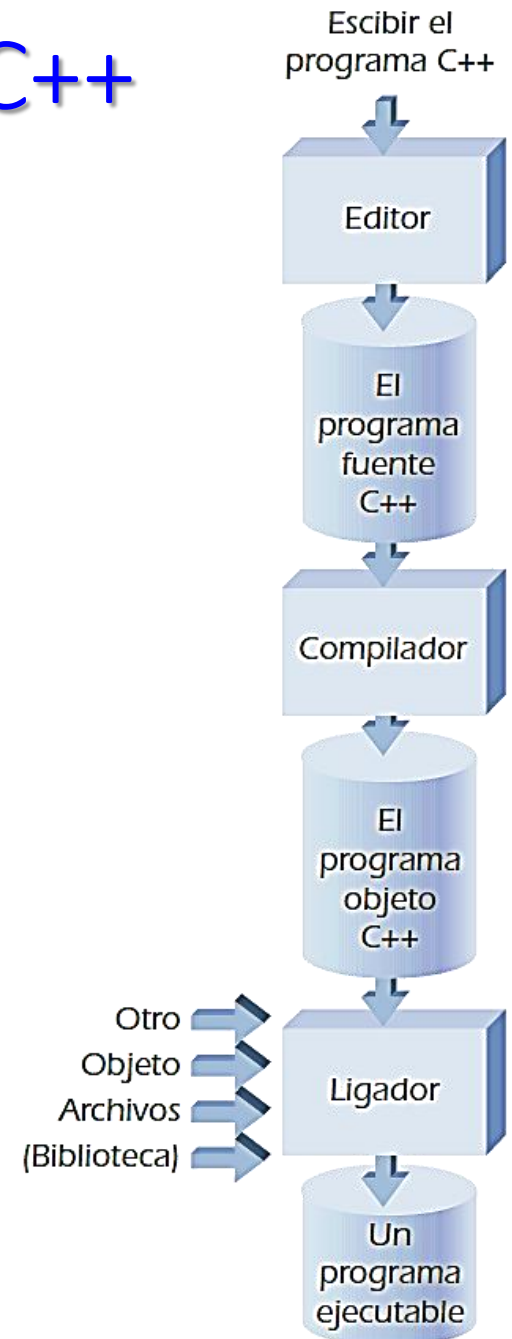
- Los programas escritos en un lenguaje de alto o bajo nivel se conocen como **programa fuente** o **código fuente**.

Lenguajes interpretado y compilado

- Una vez escrito un programa en lenguaje de alto nivel también debe traducirse al lenguaje máquina. Esto puede lograrse de dos formas.
- 1. Cuando cada instrucción del código fuente es traducida de manera individual y ejecutada inmediatamente después de la traducción. Este tipo de lenguaje se llama **lenguaje interpretado**.
- 2. Cuando todas las instrucciones del código fuente son traducidas a una unidad completa antes que cada instrucción sea ejecutada. Este tipo de lenguaje se llama **lenguaje compilado**.
- Pueden existir versiones compiladas como interpretadas de un mismo lenguaje.
- C++ es predominantemente un lenguaje compilado.

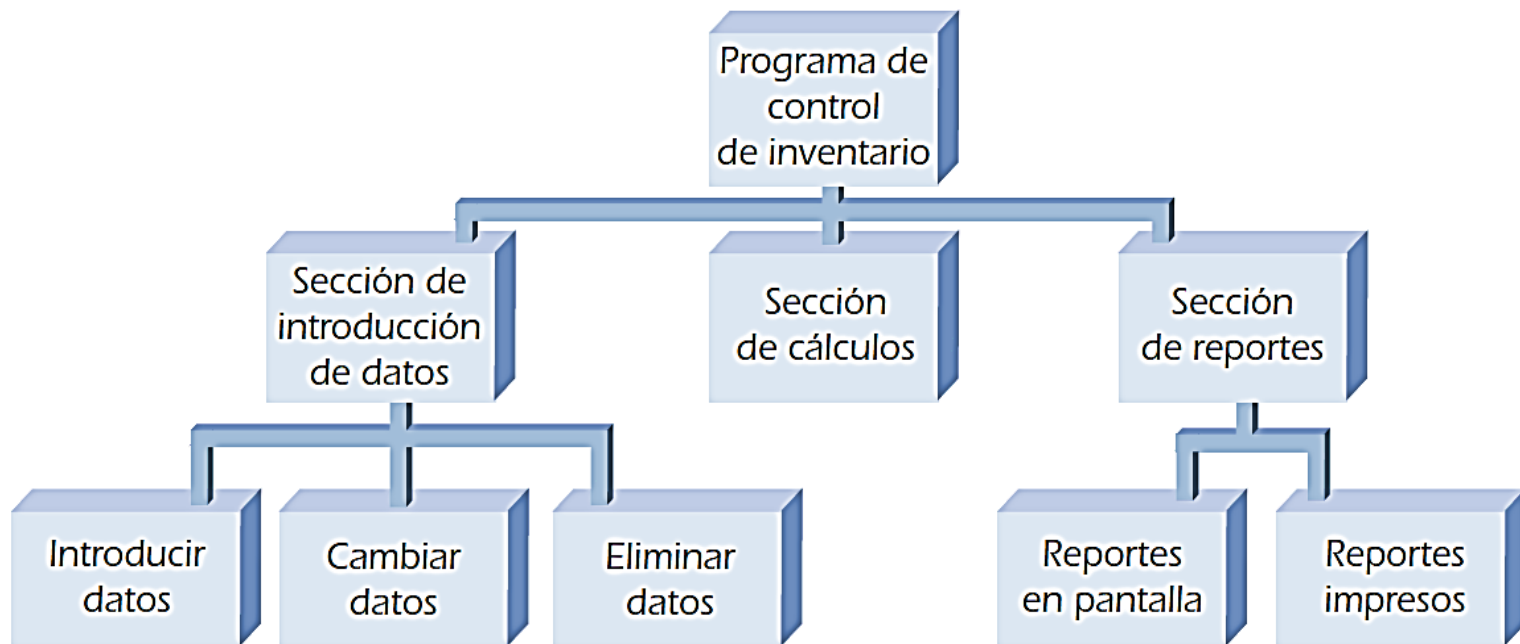
Creación de un programa C++ ejecutable

- El programa fuente se introduce utilizando un editor de texto.
- La traducción del programa en lenguaje máquina comienza por el compilador.
- La salida del compilador se llama **programa objeto**, el cual es una versión en lenguaje máquina del código fuente.
- En casi todos los casos, su código fuente usará código preprogramado existente en C++ o código que usted ya haya escrito con anterioridad.
- Esta es la tarea del **ligador** que combina este código existente con su programa objeto.
- El resultado de este ligamiento es un programa en lenguaje máquina completo. Llamado **programa ejecutable**.



Lenguajes orientados a procedimientos y a objetos

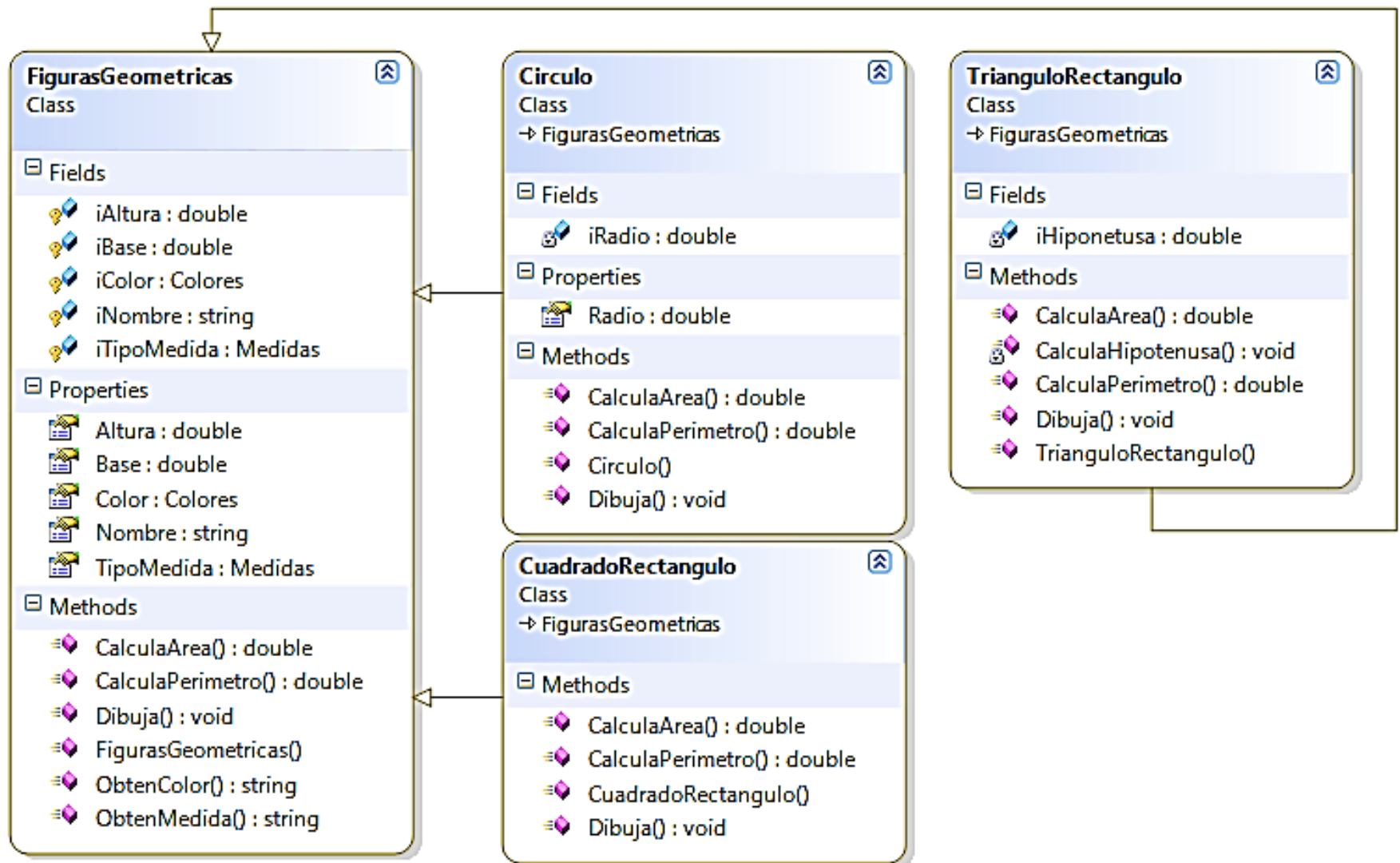
- Además de clasificar los lenguajes como de alto y bajo nivel, también se clasifican por su orientación a procedimientos u objetos.
- En un **lenguaje orientado a procedimientos** las instrucciones se usan para crear unidades independientes, conocidas como **procedimientos**. El propósito de un procedimiento es aceptar datos como entrada y transformarlos de alguna manera para producir un resultado específico como una salida.



Lenguajes orientados a procedimientos y a objetos

- Hasta la década de los años 90 la mayor parte de los lenguajes de programación de alto nivel eran orientados a procedimientos.
- Actualmente predominan los lenguajes orientados a objetos.
- En un **lenguaje orientado a objetos**, el programa debe definir primero los objetos que manipulará, incluyendo una descripción de sus características.
- C++ se considera un lenguaje orientado a objetos. Sin embargo contiene características que se encuentran en lenguajes orientados a procedimientos.

Ejemplo de objetos



El desarrollo del C++

- **FORTAN**: El primer lenguaje de procedimientos, su nombre deriva de **FORmula TRANslation**, fue desarrollado por IBM en 1957 y usado para aplicaciones científicas y de ingeniería, fue popular hasta los principios de la década de los 70.
- **ALGOL**: Fue desarrollado en 1958, su nombre proviene de **ALGOrithmic Language**, un lenguaje de procedimientos muy popular en las universidades en los años 60 pero sin éxito comercial.
- **COBOL**: Apareció en 1958, también un lenguaje de procedimientos y permaneció vigente hasta la década de los 80, su nombre deriva de **COmmon Business-Oriented Language** y fue desarrollado para aplicaciones de negocios.

El desarrollo del C++

- **BASIC**: Su nombre deriva de **Beginners All-purpose Symbolic Instruction Code**, originalmente un lenguaje de procedimientos, fue desarrollado en 1964 y pretendía ser un lenguaje introductorio para estudiantes universitarios, no imponía un enfoque estructurado y esa era su principal desventaja. Aún es popular en su versión orientada a objetos Visual Basic. NET.
- **Pascal**: Presentado en 1970, se le puso ese nombre en honor al **matemático Blaise Pascal**, su objetivo era crear un lenguaje de procedimientos estructurado que facilitará el aprendizaje en los estudiantes remediando la falta de diseño estructurado del BASIC.

El desarrollo del C++

- **C:** El lenguaje Pascal estaba tan fuertemente estructurado que a veces impedía programar muchos proyectos del mundo real. En 1972 se crea el C, un lenguaje de procedimientos estructurado, que no solo se escribía como un lenguaje de alto nivel, sino que también podía acceder a característica de nivel de máquina de la computadora. Se convirtió en el lenguaje dominante de la década de los 80.
- **C++:** Apareció en 1983, con el objetivo de extender el exitoso lenguaje de programación C. Por tanto C++ conserva las capacidades estructuradas y de procedimientos proporcionadas por C y le agrega mecanismos para manipular objetos para convertirse en un verdadero lenguaje orientado a objetos. C++ se puede usarse desde programas interactivos simples, hasta programas de ingeniería y científicos sofisticados y complejos.

FUNDAMENTOS DE PROGRAMACIÓN

Fin Clase 01

Profesor: Carlos Díaz