

Universidad Centroamericana “José Simeón Cañas”

Fundamentos de programación Ciclo 01/2020

5 de mayo del 2020

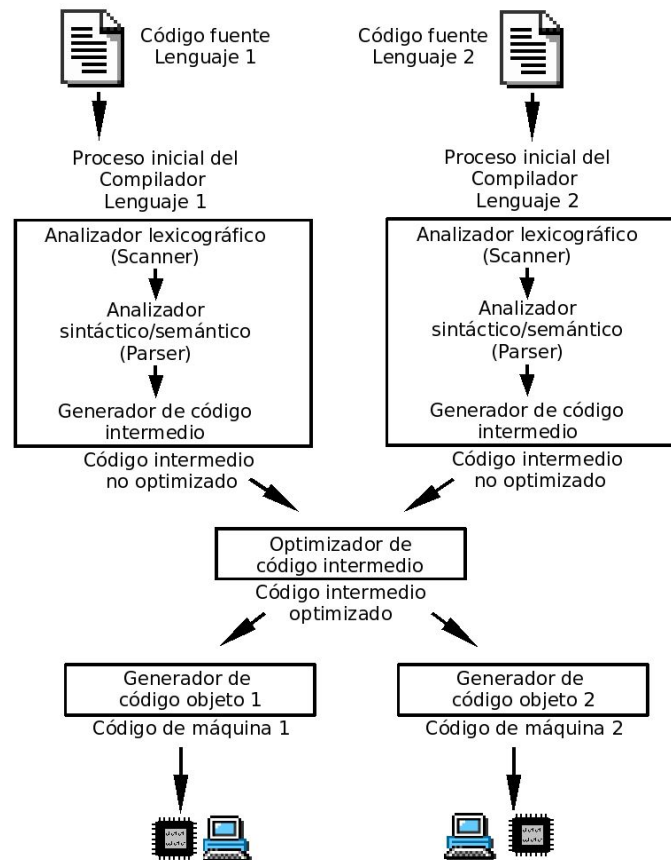
## Guia de laboratorio 3

*Programación en C++*

### Conceptos previos

- **C:** Es un lenguaje creado por Dennis Ritchie a principios de la década de 1970 y basado en el paradigma de programación estructurada. Pretende ser un lenguaje de propósito general que dé al programador control total sobre sus programas, permitiendo la programación de bajo nivel, pero a la vez proporcionando estructuras de alto nivel que hagan la programación de sistemas grandes más fácil.
- **C++:** fue creado por Bjarne Stroustrup durante la primera mitad de los años 1980 con el fin de dotar C de mecanismos que permitieran usar el paradigma de programación orientada a objetos.
- **C#:** fue creado por Anders Hejlsberg en torno al año 2000 como una versión mejorada de C++ para la programación de alto nivel. Aunque la primera versión de C# estaba fuertemente basada en C++, ya no ofrecía compatibilidad: un programa escrito en C o C++ no es válido en C#. Las versiones posteriores de C# se han ido apartando cada vez más de sus predecesores.

- **Compilador:** Un compilador es un programa informático que traduce un lenguaje de programación de alto nivel a un lenguaje de máquina.



La construcción de un compilador involucra la división del proceso en una serie de fases que variará con su complejidad. Generalmente estas fases se agrupan en dos tareas: el análisis del programa fuente y la síntesis del programa objeto.

- **Análisis:** Se trata de la comprobación del programa fuente, e incluye las fases correspondientes del análisis léxico (la descomposición del programa fuente), análisis sintáctico (agrupación de los componentes en frases gramaticales) y análisis semántico (validez semántica del programa).
- **Síntesis:** Su objetivo es la generación de la salida expresada en el lenguaje objeto.

- **Intérprete:** Un intérprete es un programa que ejecuta línea a línea las instrucciones de un programa de alto nivel. El intérprete carga el código fuente y traduce las instrucciones a un lenguaje intermedio que puede luego ser ejecutado. Los intérpretes generan un código binario que se interpreta cada vez que se ejecuta el programa *a diferencia del compilador que crea un archivo ejecutable*.

Las ventajas de utilizar un intérprete son:

- Al ejecutarse línea a línea, resulta ser más fácil de depurar y comprobar errores.
- Si el programa es muy largo, un intérprete resulta ser más rápido que un compilador porque no tiene que traducir todo el programa para ejecutarlo.

Las desventajas de utilizar un intérprete son:

- El código fuente se requiere para poder ejecutar un programa.
- Los intérpretes son más lentos que los programas compilados o ejecutables.

- **Editor de Texto:** un editor de texto es un programa informático que permite crear y modificar archivos digitales compuestos únicamente por textos sin formato, conocidos comúnmente como archivos de texto o “texto plano”. El programa lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor.
- **Sintaxis:** la sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.
- **Código Fuente:** es el archivo o archivos con las instrucciones necesarias, realizadas en un lenguaje de programación, que sirve para compilar

posteriormente un programa o programas para que puedan ser utilizados por el usuario de forma directa, tan sólo ejecutándose. Para lograr esto, los creadores de programas informáticos utilizan distintos tipos de lenguajes de programación para escribir el código fuente de un determinado software.

- **Semántica:** en informática, la semántica es considerada una aplicación de la matemática lógica. La semántica refleja el significado de programas o funciones. Los programas pueden separarse en su parte sintáctica (la estructura gramatical) y su parte semántica (el significado). Por ejemplo, las siguientes sentencias de programación utilizan diferentes sintaxis (en diferentes lenguajes), pero resultan en la misma semántica (hacen lo mismo):

- 1)  $x += y$ ; (C, Java, etc)
- 2)  $x := x + y$ ; (Pascal)
- 3)  $x = x + y$  (BASIC, Fortran)

Tipo de Dato	Tipo de Valor Almacenado	Ejemplo
int	números enteros	100
float	números con decimales	3.5
double	números con decimales de mayor tamaño	3.1415926535897932..., 2.2x10 <sup>308</sup>
char	un solo carácter	'a','1','F'
string	cadena de caracteres	"Hola Mundo"
bool (booleano)	valor binario	true, false
short	valores enteros de 16 bits	-2x10 <sup>15</sup>
long	valores enteros de 64 bits	1645x10 <sup>15</sup>

- **Ejecutar (Run):** en informática, ejecutar es la acción de iniciar la carga de un programa o de cualquier archivo ejecutable. El proceso en el cual la máquina lleva a cabo las instrucciones de un programa informático. Ejecutar un

programa implica que éste estará en estado de ejecución y, por ende, en memoria, hasta que se finalice.

- **Bug(Error):** un bug es un error o un defecto en el software que hace que un programa funcione incorrectamente. A menudo los bugs son causados por conflictos del software cuando las aplicaciones intentan funcionar. Bug es español significa insecto. Según se dice, el primer bug de ordenador fué un insecto real, descubierto en 1945 en Harvard, una polilla atrapada en la calculadora Mark II Aiken Relay hizo que la máquina entera se apagara.
- **Debug (Depurador):** es el proceso de eliminar los errores dentro de un programa informático los cuales causan problemas al momento del funcionamiento de estos. Existen programas llamado “Debuggers”, los cuales examinan el programa en busca de estos errores de forma automática.

## Ejercicios de programacion en c++

Mi primer programa en C++

```
#include <iostream>
using namespace std;

int main(){
    cout << "Hola mundo";
    return 0;
}
```

Comentarios de una línea

```
int a = 0; // Declaración de la variable a.
```

Comentarios de múltiples líneas

```
double d;
/*La declaración de la variable b se
ejecutaria, pero esto no porque es un
comentario multi-línea.*/
```

## Manejo de variables numéricas en C++

```
/*Declaración de dos variables de tipo
entero y asignandoles un valor.*/

int num1 = 6;
int num2 = 3;

/*Declaración de variables donde
almacenaremos los resultados de las
operaciones*/

int suma = num1 + num2;
int resta = num1 - num2;
int mult = num1 * num2;
int div = num1/num2;
int mod = num1%num2;
```

## Salidas y entradas

```
int num1,num2;

/*Utilizamos la palabra reservada "cin"
seguida de ">>" y luego el nombre de la
variable a ingresar , en caso de pedir
mas, ingresamos ">>" de nuevo seguido del
nombre de la siguiente variable.*/
cin >> num1 >> num2;

/*Utilizamos la palabra reservada "cout"
seguida de "<<" y luego el nombre de la
variable a mostrar.*/
int suma = num1+num2;
cout << suma;
```

## Concatenación de cadenas de caracteres

```
//Se declara la variable concatenar
string concatenar;

//Se declara la variable y se le asigna un valor
string string1 = "Hola ";

//Se declara la variable y se le asigna un valor
string string2 = "mundo";

/* Se le asigna el valor a la variable concatenar de
la concatenacion entre string1 y string 2*/
concatenar=string1+string2;
```

```
//Se declara la variable y se le asigna un valor
string string1 = "Hola ";

//Se declara la variable y se le asigna un valor
string string2 = "mundo";

/*Se manda un mensaje al usuario con la concatenacion
entre string1 y string2 */
cout << string1 << string2;
```



## Condiciones

```
#include <iostream>
using namespace std;

int main(){
    //Se declara la variable cum
    float cum;

    //Se manda al usuario un mensaje por medio de una salida
    cout << "Ingrese cum del alumno" << endl;

    //Se le pide al usuario el valor de la variable antes declarada
    cin >> cum >>;

    //Se realiza una condicion para verificar si cum es mayor o igual a 7
    if(cum>=7){
        //Se manda al usuario un mensaje
        cout<<"El estudiante pudo inscribir 21 unidades valorativas";
    }
    //Se realiza una condicion para verificar si cum es menor a 6
    else if(cum<6){
        //Se manda al usuario un mensaje
        cout<<"El estudiante pudo inscribir 10 unidades valorativas";
    }
    /*Instruccion que se ejecutara en caso que ninguna condicion
    anterior se cumpla*/
    else{
        //Se manda al usuario un mensaje
        cout<<"El estudiante pudo inscribir 16 unidades valorativas";
    }

    return 0;
}
```

## Tarea

- Realizar un programa en c++ que devuelva el promedio de 3 números enteros.
- Realizar un programa en c++ que verifique si un número es divisible entre otro.
- Realizar un programa en c++ que nos devuelva el resultado de la fórmula cuadrática.
- Realizar un programa en c++ que verifique si un número es par o impar.