

Entrada/Salida en C++

1. Salida, cout

Las operaciones printf y scanf que vimos el semestre pasado son las operaciones de salida estandar, propias de C.

En C++ se agregan operaciones más sencillas de utilizar a saber **cout** y **cin**. Estas funciones están declaradas en la biblioteca "iostream.h".

Veremos la forma en que se utilizan estas operaciones por medio de ejemplos.

Ejemplo 1.1 (Impresión) *Para imprimir el string Hola que tal y avanzar a la línea siguiente colocamos la instrucción:*

```
cout << "Hola que tal\n";
```

para avanzar a la línea siguiente podemos utilizar endl como se muestra abajo:

```
cout << "Hola que tal"<< endl;
```

que es equivalente a

```
cout << "Hola que tal";  
cout << endl;
```

Para imprimir variables simplemente las colocamos en la instrucción cout. Supongamos tenemos una variable entera i, una de carácter c y una de punto flotante f, las imprimimos como se muestra abajo:

```
cout << "Imprimo variable entera:";  
cout << i << endl;  
cout << "Imprimo variable de carácter:";  
cout << c << endl;  
cout << "Imprimo variable de punto flotante:";  
cout << f << endl;
```

Ejemplo 1.2 (Lectura) *Para leer strings en arreglos de caracteres colocamos la instrucción:*

```
cin >> a;
```

donde *a* es un arreglo de caracteres. Se coloca un `\0` en la última posición (automáticamente).

En general para leer variables simplemente las colocamos en la instrucción **cin**. Supongamos tenemos una variable entera *i*, una de carácter *c* y una de punto flotante *f*, las leemos como se muestra abajo:

```
cout << "Ingrese variable entera: ";
cin >> i;
cout << endl << "Ingrese variable de caracter: ";
cin >> c;
cout << endl << "Ingrese variable de punto flotante: ";
cin >> f;
```

Podemos leer varias variables en una sola instrucción. Supongamos tengo *i1* e *i2* de tipo entero, las puedo leer como sigue:

```
cout << "Ingrese dos valores enteros: ";
cin >> i1 >> i2;
```