



Programación

UDI: Introducción a la Programación



Jaime Matas Bustos
CFGS Desarrollo de Aplicaciones Multiplataforma
C.E.S. Cristo Rey



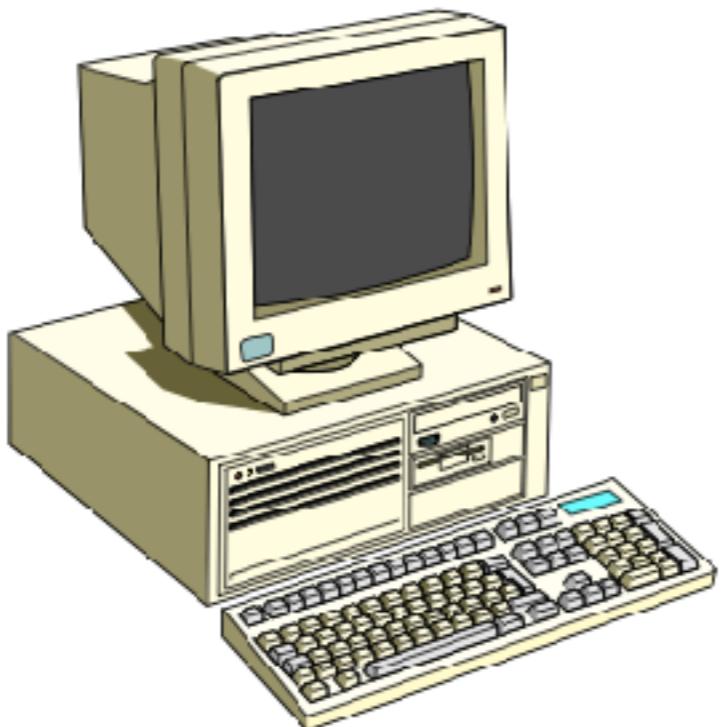


Contenido del Tema

- ① Funcionamiento de un ordenador
- ② Codificación de la información
- ③ Lenguajes de programación
 - El diseño de programas
 - El proceso de traducción
- ④ Programación en C++
 - Estructura de un programa fuente
 - Datos y expresiones
 - Operaciones



Funcionamiento de un ordenador



- Hardware
- Software
- Programa
- Lenguaje de programación
- Usuario
- Programador

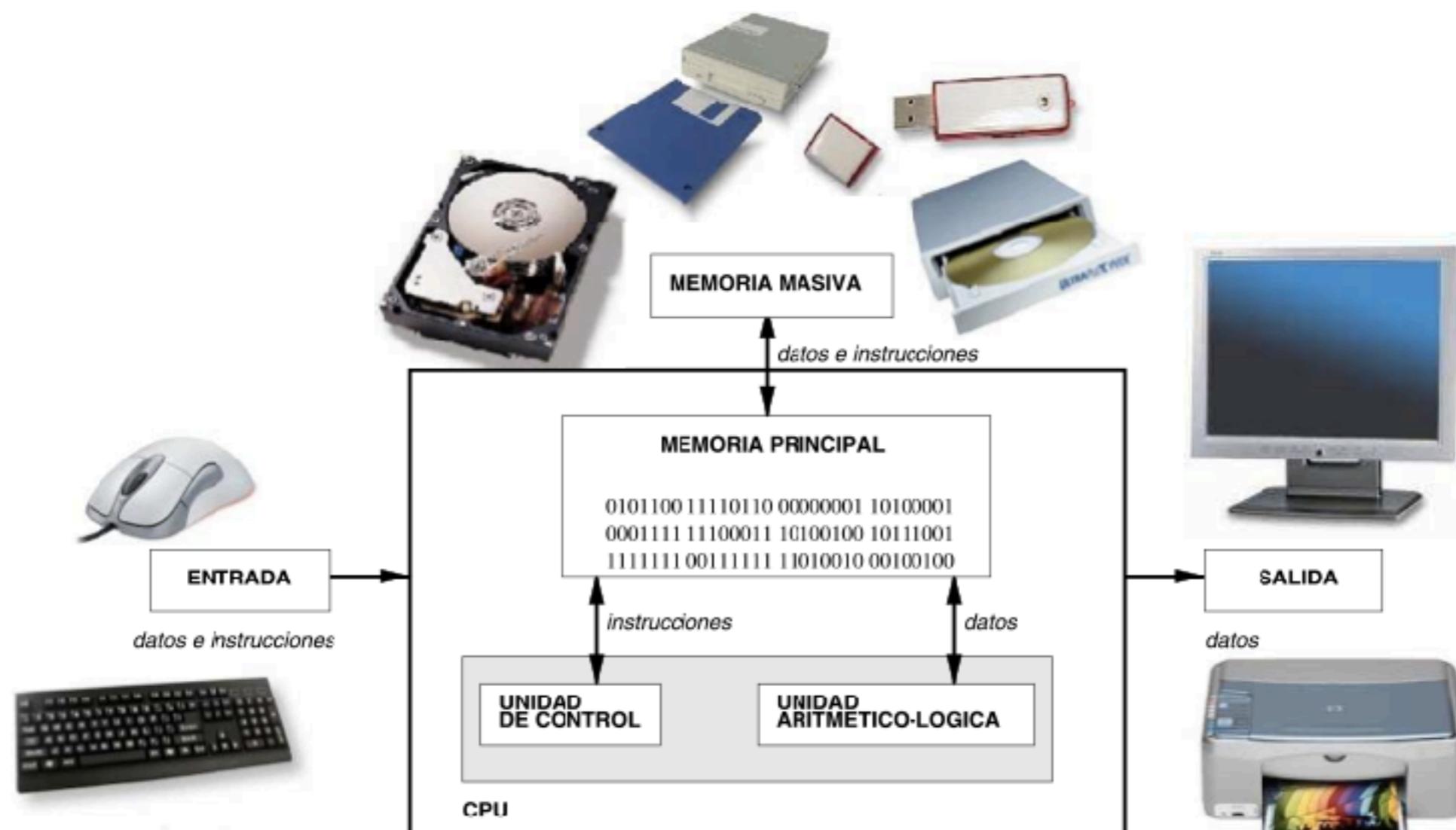
Steve Jobs, fundador de Apple Computers

Un ordenador es como una bicicleta para la mente





Esquema funcional de un ordenador





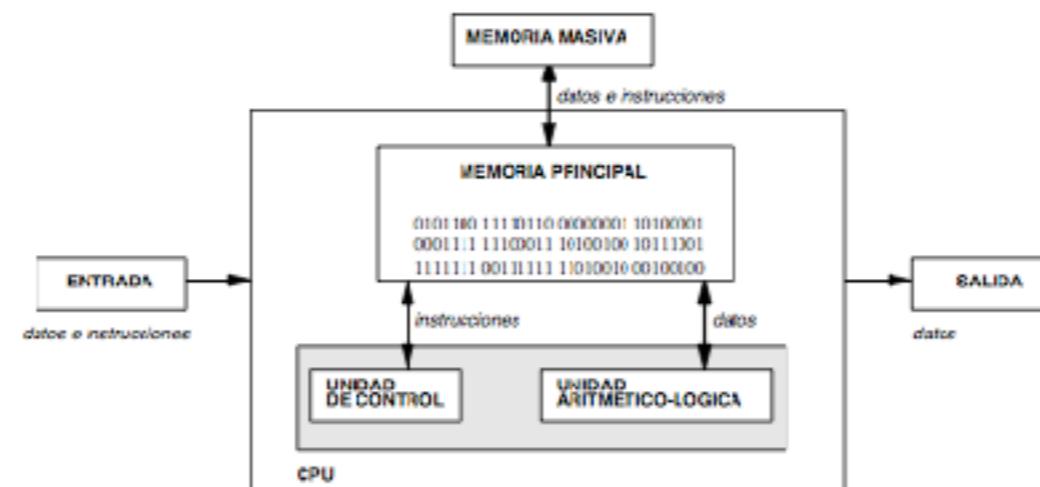
Funcionamiento de un ordenador

Otros sistemas basados en CPU (*programables*)





Codificación de la información



Datos

Juan Pérez →

1	0	1	1	..	0	1	1
---	---	---	---	----	---	---	---

75,225,813 →

1	1	0	1	..	0	0	0
---	---	---	---	----	---	---	---

0.314159×10^1 →

0	0	0	1	..	1	1	1
---	---	---	---	----	---	---	---

Instrucciones sencillas

Abrir Fichero →

0	0	0	1	..	1	1	1
---	---	---	---	----	---	---	---

Imprimir →

1	1	0	1	..	0	0	0
---	---	---	---	----	---	---	---

¡¡ Estas representaciones son incompatibles entre sí !!

"75" 75 0.75×10^2



Es necesario desarrollar programas, compuestos de instrucciones, que se ejecuten en un ordenador

Un programa simple para sumar dos números

- ① Mostrar en la pantalla un mensaje para que el usuario introduzca el primer número
- ② Leer un número y almacenarlo en memoria
- ③ Mostrar en la pantalla un mensaje para que el usuario introduzca el segundo número
- ④ Leer un número y almacenarlo en memoria
- ⑤ Calcular la suma de los dos números almacenados
- ⑥ Mostrar en la pantalla un mensaje para mostrar al usuario el resultado



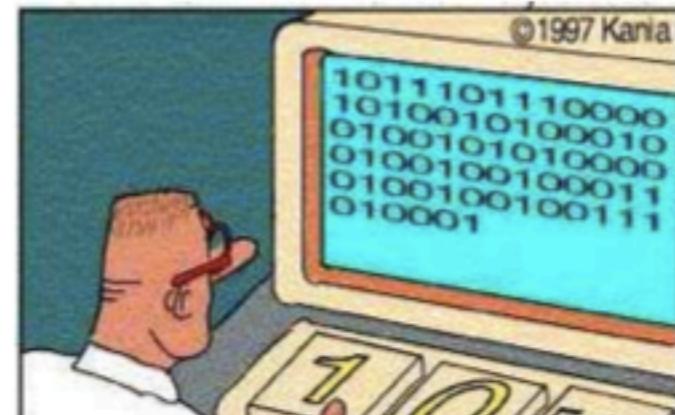
Lenguajes de programación



Pero ... ¿en qué lenguaje?

El lenguaje máquina

El conjunto de instrucciones que entiende, directamente, el ordenador.



Los lenguajes de alto nivel

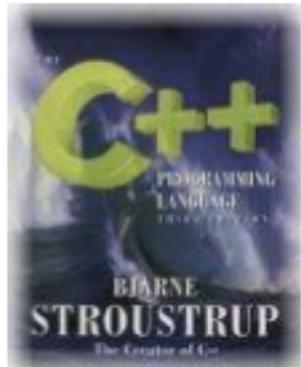


“Los lenguajes de Alto Nivel se caracterizan por el uso de macroinstrucciones, las cuales en el momento de ser ejecutadas por el computador tienen que ser traducidas a lenguaje de máquina, ya sea por medio de un compilador o intérprete”

Ada, ALGOL, Basic, C, C++, C#, Clipper, COBOL, Fortran, Java, Lexico, Logo, Object Pascal, Pascal, Perl, PHP, PL/SQL, Python, Modula-2



El lenguaje C++



- 1972 Desarrollo de programas en UNIX (proveniente de BCPL)
- 1983 Comité Técnico X3J11: ANSI C trabajos previos
- 1983 Bjarne Stroustrup: C++
- 1989 Comité de estandarización ANSI X3.159-1989 (Lenguaje de Programación C, C89)
- 1998 Comité de estandarización ANSI-ISO estandarizó C++ (ISO/IEC 14882:1998, C++98). Revisado en 2003.



El lenguaje C++: “Hola mundo” I

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     cout << "Hola mundo" << endl;
5 }
```





Lenguajes de programación

El lenguaje C++: “Hola mundo” II

Basic

```
1 10 print "Hola Mundo"
```



El lenguaje C++: “Hola mundo” III

Java

```
1 public class HolaMundo
2 {
3     public static void main(String[] args) {
4         System.out.println("¡Hola, mundo!");
5     }
6 }
7
```



El lenguaje C++: “Hola mundo” IV

PHP

```
1 <?php  
2     echo "Hola Mundo";  
3 ?>
```



Algoritmo

Un conjunto finito y ordenado de pasos o instrucciones para obtener la solución a un problema.

WIKIPEDIA
La enciclopedia libre

Muhammad ibn Musa al-Jwarizmi

(S.VIII): *Explicó que, mediante una especificación clara y concisa de cómo calcular sistemáticamente, se podrían definir algoritmos que fueran usados en dispositivos mecánicos en vez de las manos (por ejemplo, ábacos)*

Programa de ordenador

Una secuencia de sentencias que debe realizar un ordenador escritas en un lenguaje de programación determinado.



```
1 /* Fichero que calcula
2     la suma de dos números */
3 #include <iostream>
4 using namespace std;
5 int main()  {
6     int a, b, res;
7     cout << "Introduce el primer número: ";
8     cin >> a; // Este es el primer número
9     cout << "Introduce el segundo número: ";
10    cin >> b; // Y este es el segundo
11    res = a + b;
12    cout << "El resultado es: " << res << endl;
13 }
```





Programas traductores

: traducen un programa escrito en un lenguaje de alto nivel al lenguaje máquina.

- Intérpretes
- Compiladores

Programa o fichero fuente

Un fichero que contiene un programa escrito en algún lenguaje de alto nivel

Programa o fichero binario o ejecutable

Un fichero que contiene un programa escrito en lenguaje máquina

Programa fuente → Programa objeto → Programa ejecutable

Errores de compilación, de ejecución, lógicos



```
/* Breve descripción en lenguaje natural  
de lo que hace el programa */  
  
<Inclusión de archivos de cabecera>  
  
using namespace std;  
  
int main() {  
  
<Instrucciones>  
  
}
```



Dato

Un dato es una representación simbólica (numérica, alfabética, etc.), de un propiedad de una entidad que, convenientemente tratado (procesado) se puede utilizar en la realización de cálculos o toma de decisiones.

- Los lenguajes de programación disponen de distintas categorías de datos (numéricas, texto, etc.).
- Al declarar un dato, el compilador reserva una zona de memoria para trabajar con él. Ningún otro dato podrá usar dicha zona.
- Cada dato debe estar asociado a un único tipo de dato (el tipo no puede cambiarse durante la ejecución).
- El valor que se le puede asignar a un dato depende del tipo de dato con el que es declarado.
- Los datos se pueden clasificar como:
 - Variables
 - Constantes



Algunos tipos de datos básicos en C++

- **int** Datos numéricos enteros: 75
- **double** Datos numéricos reales (en coma flotante): 0.75×10^{-4}
- **char** Datos de tipo carácter: 'a'
- **bool** Datos lógicos o booleanos: true, false



Variables

Variable

Es un conjunto de celdas o posiciones de memoria que tiene asociado un nombre (**identificador**) y un contenido (**valor**) que **puede variar** durante la ejecución de un programa

Declaración de variables

Es la operación que se utiliza para crear un dato mediante la especificación del tipo del valor que almacenará y del identificador que hará referencia a ese valor

<tipo de la variable> <nombre de la variable> ;

```
1 int valor;  
2 double resultado, resultado2;  
3 char caracter;  
4 bool apto;
```



Variables

- Cuando se declara una variable, su valor es desconocido y podría ocasionar algunas interferencias en los cálculos.
- Cuando sea necesario, se puede especificar un valor inicial para una variable (**inicialización**) antes de operar con ella.

```
1 int valor=10;  
2 double resultado=0.34, resultado2=9.8;  
3 char caracter='a';  
4 bool apto=false;
```



Variables

Normas de uso de identificadores

- Deben empezar por una letra o subrayado (_)
- No pueden contener espacios en blanco ni ciertos caracteres especiales como letras acentuadas, la letra eñe, etc.
- Pueden tener una longitud máxima (por ejemplo, 31 caracteres)
- Sensibles a mayúsculas y minúsculas.
- No se podrá dar a un dato el nombre de una palabra reservada (Garrido, A., pg. 420)



Variables

Otras recomendaciones

- El identificador de un dato debe reflejar su semántica (contenido).
`11, 12, hp`
`lado1, lado2, hip o, incluso mejor, hipotenusa`
- Usaremos minúsculas, salvo la primera letra de nombres compuestos (y los identificadores de las constantes).
`precioventapublico, tasaanual`
`previoVentaPublico, precio_venta_publico`
`pvp, TasaAnual, tasa_anual`
- No se nombrarán dos datos con identificadores que difieran únicamente en la capitalización, o un sólo carácter.
`cuenta, cuentas, Cuenta`
`cuenta, colecciónCuentas, cuentaPpal`



Constante

Es un conjunto de celdas o posiciones de memoria que tiene asociado un nombre (**identificador**) y un contenido (**valor**) que **permanece fijo** durante la ejecución de un programa

Declaración de constantes

Es la operación que se utiliza para crear un dato constante mediante la especificación del tipo del valor que almacenará y del identificador que hará referencia a ese valor

```
const <tipo>    <nombre_constante> = <valor>;
```

```
1 const int NUMERO_DIAS = 5;  
2 const double PI = 3.14159;
```



```
1 /* Programa para calcular la retención a aplicar en
2 el sueldo de un empleado
3 Entradas: Salario bruto (salario_bruto) en euros
4 Salidas: La retención a aplicarle en euros,
5 calculada aplicando la fórmula:
6 Retencion = Salario Bruto * Porcentaje gravamen */
7 #include <iostream>
8 using namespace std;
9 int main() {
10     const double IRPF=0.18; // Porcentaje gravamen fiscal
11     double retencion; // Retención a aplicar, en euros
12     double salario_bruto; // Salario bruto, en euros
13     cout << "Introduzca salario bruto: ";
14     cin >> salario_bruto;
15     retencion = salario_bruto * IRPF;
16     cout << "\n Retención a aplicar: "<< retencion;
17 }
```



¿Qué ocurre si cambia el porcentaje de IRPF de un año para otro?



Literal

Es un valor concreto de un tipo de dato. Por ejemplo:

- `3` `-4300` Son literales de tipo número entero `int`.
- `-3.0` `0.999e3` Son literales de tipo número real `double`.
- `'a'` `''` Son literales de tipo carácter `char`.
- `true` `false` Son literales de tipo booleano `bool`.
- `“Luis Castillo”` `“31852257”` Son literales de tipo cadena de caracteres.



Asignación

Es un operador que permite asignar a una variable un valor concreto.

```
<variable> = <operación>;
```

```
1 valor = 3+4; // Asigna a valor el número 7  
2 resultado = 10;  
3 resultado = resultado+4; // Asigna el valor 14.0
```



Operaciones aritméticas

Son operaciones que se pueden aplicar a datos numéricos

- Operadores aritméticos binarios +, -, *, /, %
- Operador aritmético unario -

```
1 valor = 10; // 10
2 resultado = -valor; // -10.0
3 resultado = valor/4; // 2.0
4 resultado = resultado/4; // 2.5
```

- Pueden construirse agrupaciones más complejas mediante el uso de paréntesis (**expresiones**).

```
1 valor = (20+8)/4; // Asigna a valor el número 7
2 resultado = 10;
3 resultado = (3*resultado)/4; // Asigna el valor 7.5
```



Operaciones aritméticas

- Reglas de precedencia

`1 valor = 3+5*7; // ¿ 56 ó 38 ?`

Si aparecen varios operadores en la misma expresión tienen un orden de *asociatividad* o precedencia.

- 1 Los paréntesis
- 2 - (unario)
- 3 * / %
- 4 + -

En igualdad de precedencia, se evalúan de izquierda a derecha

$$\frac{a+b}{c} \quad (a+b)/c$$
$$a + \frac{b}{c} \quad a+b/c$$
$$\frac{\frac{a}{b}}{c} \quad a/b/c$$



Operaciones aritméticas

- Operaciones con operandos numéricos de distinto tipo: el tipo del resultado será el tipo del operando con un rango más amplio

```
1 int valor;
2 double resultado, resultado2;
3
4 valor = 10; // 10
5 resultado = 10; // 10.0
6 resultado2 = 1/valor; // 0
7 resultado2 = 1/resultado; // 0.1
8 resultado2 = 1.0/valor; // 0.1
```

- Asignaciones y expresiones con distinto tipo

```
1 int valor = 10;
2 double resultado = 20.5, resultado2;
3
4 resultado2 = valor; // 10.0
5 valor = resultado; // 20
```



Operaciones de entrada y salida

Permiten intercambiar datos entre el ordenador y el exterior (normalmente el usuario, pero pueden ser una impresora, un disco, etc).

Operación de entrada (**extracción**)

Permite introducir un valor desde el exterior (usuario) hacia una variable en la memoria del ordenador.

<origen_de_datos> >> <variable_destino>

El origen de datos más usual es la entrada estándar (`cin`) que suele asociarse al teclado.

```
1 int valor;  
2 char caracter;  
3  
4 cin >> valor;  
5 cin >> tecla;
```



Operaciones de entrada y salida

Operación de salida (**inserción**)

Permite enviar el valor contenido una variable en la memoria del ordenador hacia el exterior.

<destino_de_datos> << <variable>

El destino de los datos más usual es la salida estándar (cout) que suele ser la pantalla del ordenador, aunque hay otros (cerr).

```
1 int valor;  
2 char caracter;  
3  
4 cout << valor;  
5 cout << tecla;
```



Operaciones de entrada y salida

`cin, cout, cerr`

Se encuentran definidos en el fichero de cabeceras `iostream`

```
1 /* Fichero que calcula
2   la suma de dos números */
3 #include <iostream>
4 using namespace std;
5 int main() {
6     int a, b, res;
7     cout << "Introduce el primer número: ";
8     cin >> a; // Este es el primer número
9     cout << "Introduce el segundo número: ";
10    cin >> b; // Y este es el segundo
11    res = a + b;
12    cout << "El resultado es: " << res << endl;
13 }
```





```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int v;
5     double r;
6     char c;
7
8     cout << "Introduce un valor entero: ";
9     cin >> v;
10    cout << "Y ahora un valor real: ";
11    cin >> r;
12    cout << "Y finalmente un carácter: ";
13    cin >> c;
14    cout << "El primer número era: "<< v << endl;
15    cout << "El segundo número era: "<< r << endl;
16    cout << "El carácter era: "<< c << endl;
17 }
```





Operaciones de entrada y salida

Existen algunos códigos especiales para formatear la salida en la pantalla (Garrido, A., pg. 29)

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     cout << "Primera línea";
5     cout << " -> A continuación";
6     cout << "\nSiguiente línea"; //\n = salto de línea
7     cout << "\n\tPrimera tabulación"; //\t = tabulador
8     cout << "\n\t\tSegunda tabulación"; //\t = tabulador
9 }
```





Funciones de biblioteca

- Existen bibliotecas de funciones complejas a disposición de los programadores
 - Numéricas (fichero de cabeceras `cmath`)
`fabs(x)`, `sqrt(x)`, `exp(x)`, `log(x)`, `pow(x,y)`,
`tan(x)`, ...
 - De carácteres (fichero de cabeceras `cctype`)
`tolower(x)`, `toupper(x)`, ...
 - De otros tipos ...
- <http://www.icce.rug.nl/documents/cplusplus/>
- <http://c.conclase.net/librerias/index.php>



Funciones de biblioteca

```
1 /* Teorema de pitágoras: hip es la raíz
2 de la suma de los cuadrados de los catetos */
3 #include <iostream>
4 #include <cmath>
5 using namespace std;
6 int main() {
7     double cateto1, cateto2, hip;
8
9     cout << "Introduce el primer cateto: ";
10    cin >> cateto1;
11    cout << "Introduce el segundo cateto: ";
12    cin >> cateto2;
13
14    hip = sqrt(pow(cateto1, 2)+cateto2*cateto2);
15
16    cout << "La hipotenusa es: " << hip;
17 }
```





Bibliografía

[2005] Garrido,A.“Programación en C++”