

Tema 2

Comenzando a programar en C++

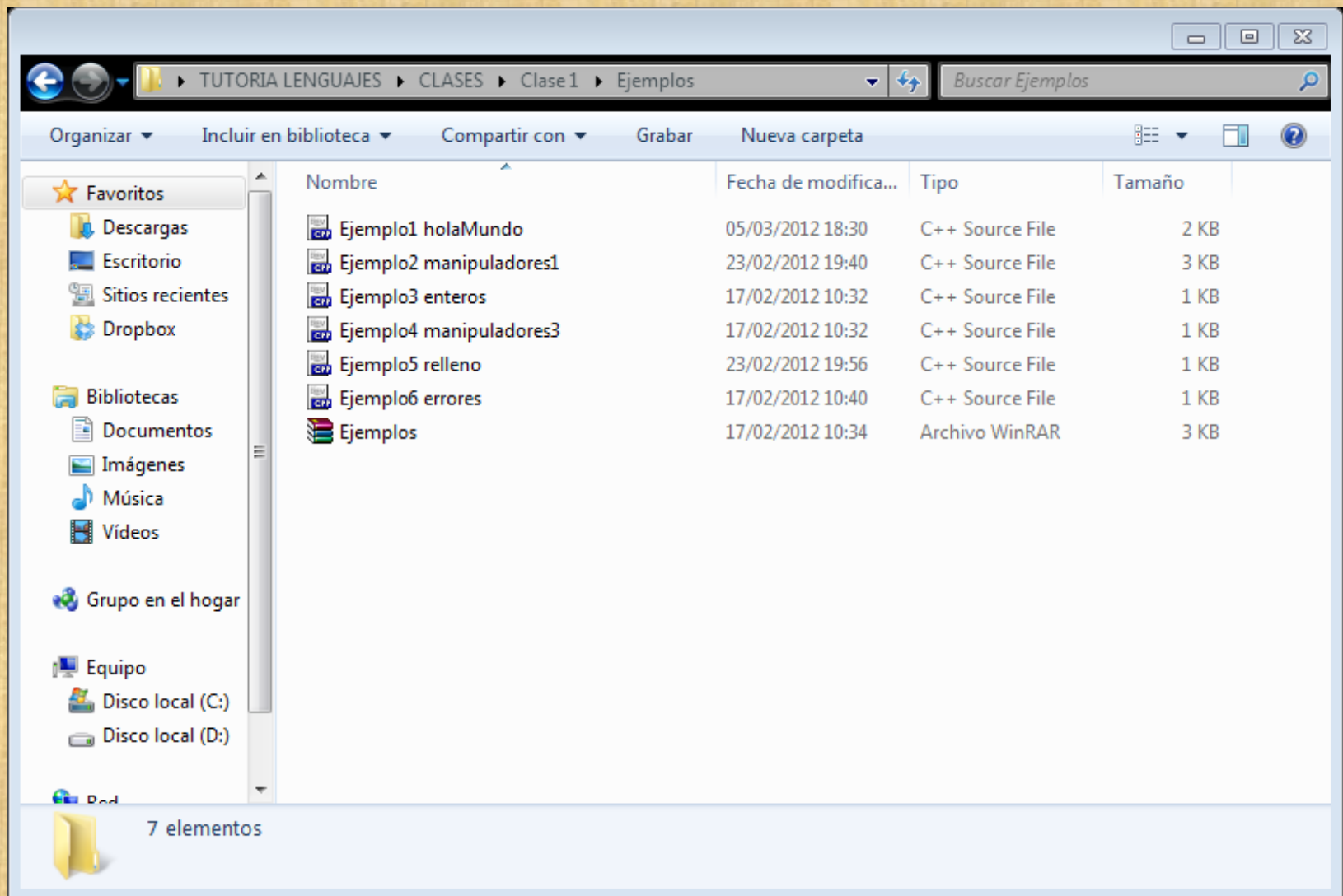
Pasos Básicos

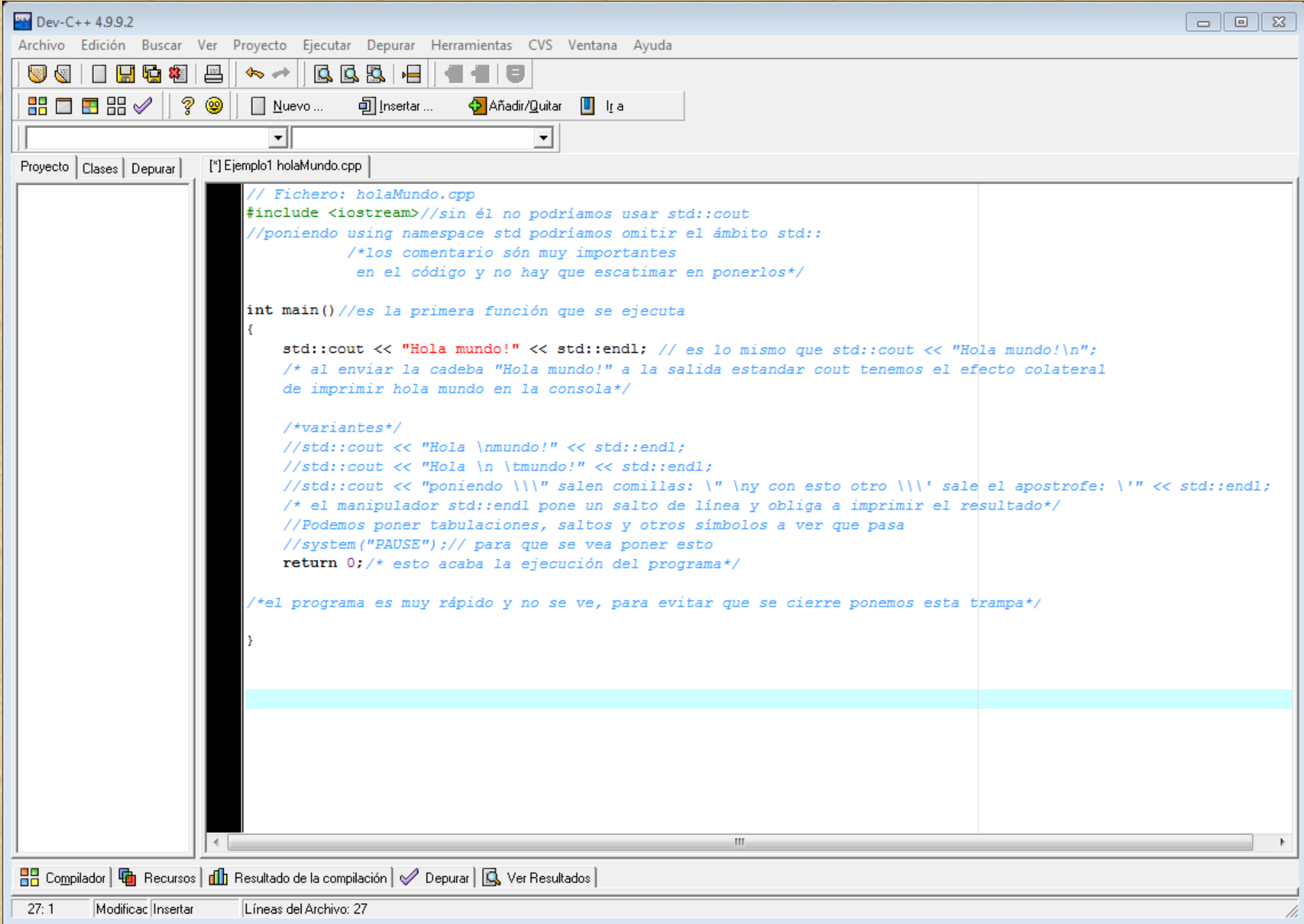
- Instalar un compilador y un entorno de desarrollo IDE
- Compilar los programas por uno mismo para ver el resultado
- Hacer cambios y experimentar para ver como funciona
- Para aprovechar los ejercicios hay que pensar antes de compilar
 - Permite comparar lo que se ha pensado y lo que el programa ha hecho
 - La depuración de los programas ayuda a aprender

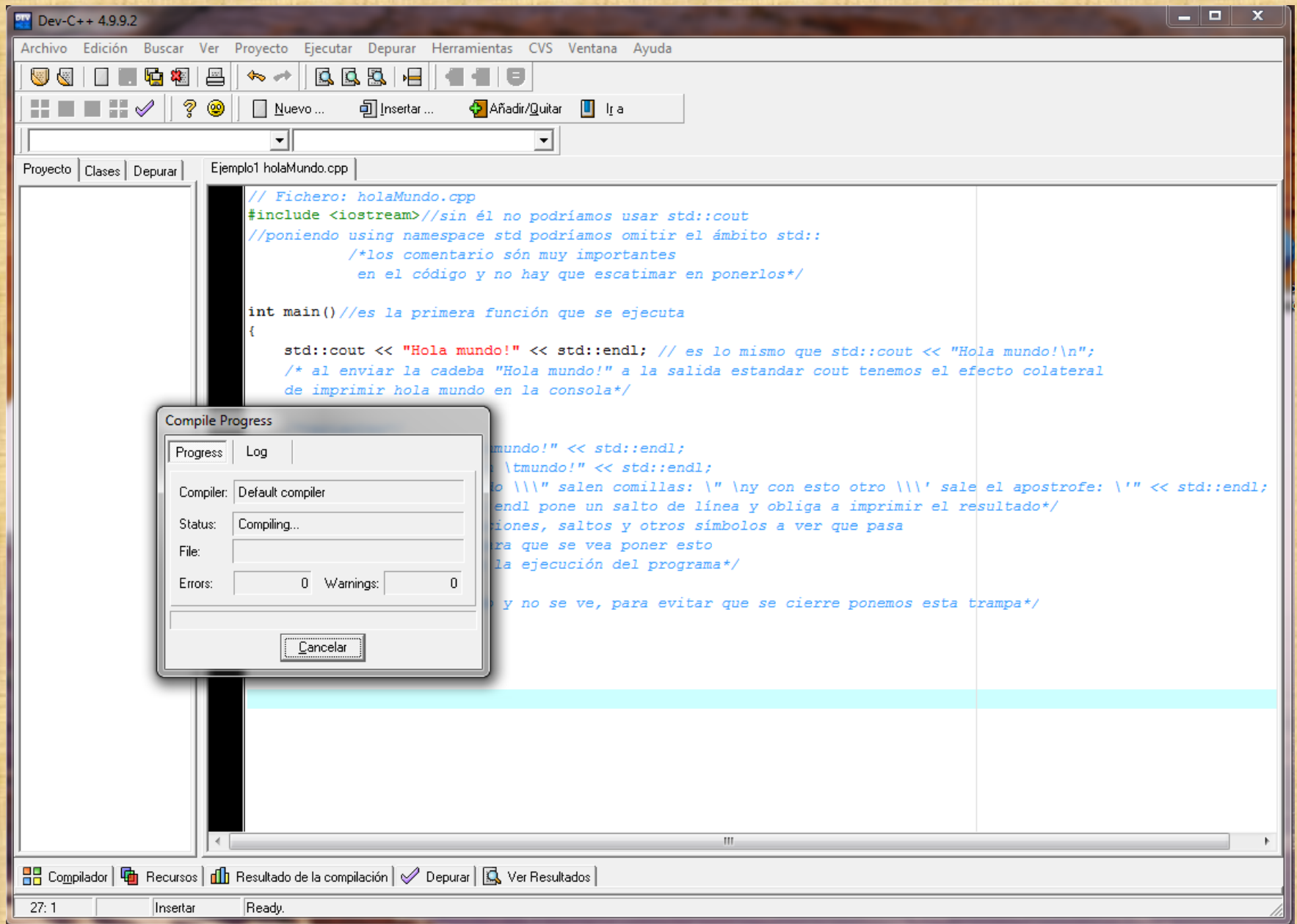
Primer programa

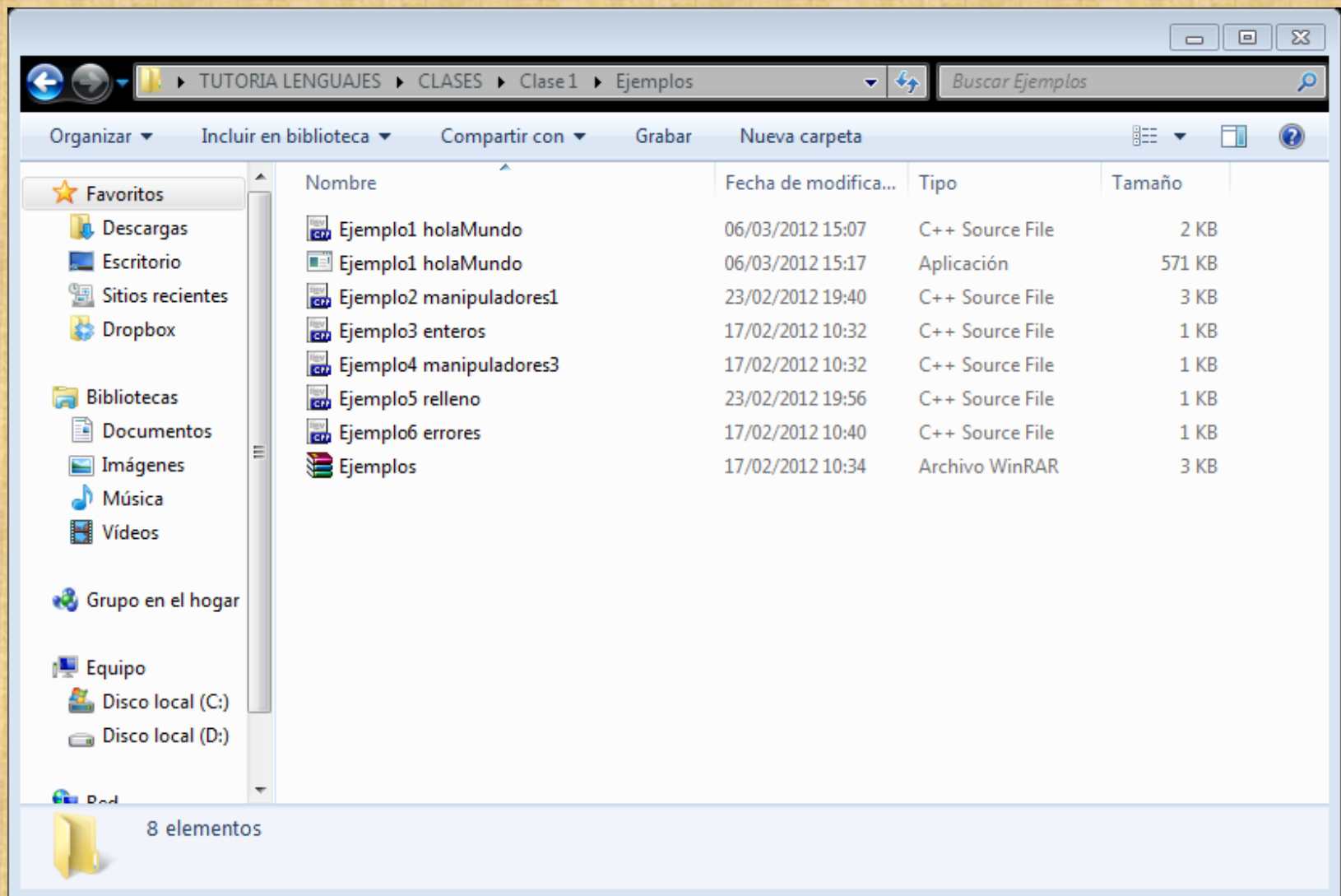
- Hola mundo
 - Como se hace un programa
 - Partes de un programa
 - Función “*main*”
 - Funciones de cabecera
 - Un *string* es un texto y se escribe entre comillas usando los modificadores siguientes
 - `\n` salto de línea
 - `\t` tabulación
 - `\”` comillas
 - `\’` apóstrofe
 - `\\` barra invertida \
- Ejemplo 1

Ejemplos

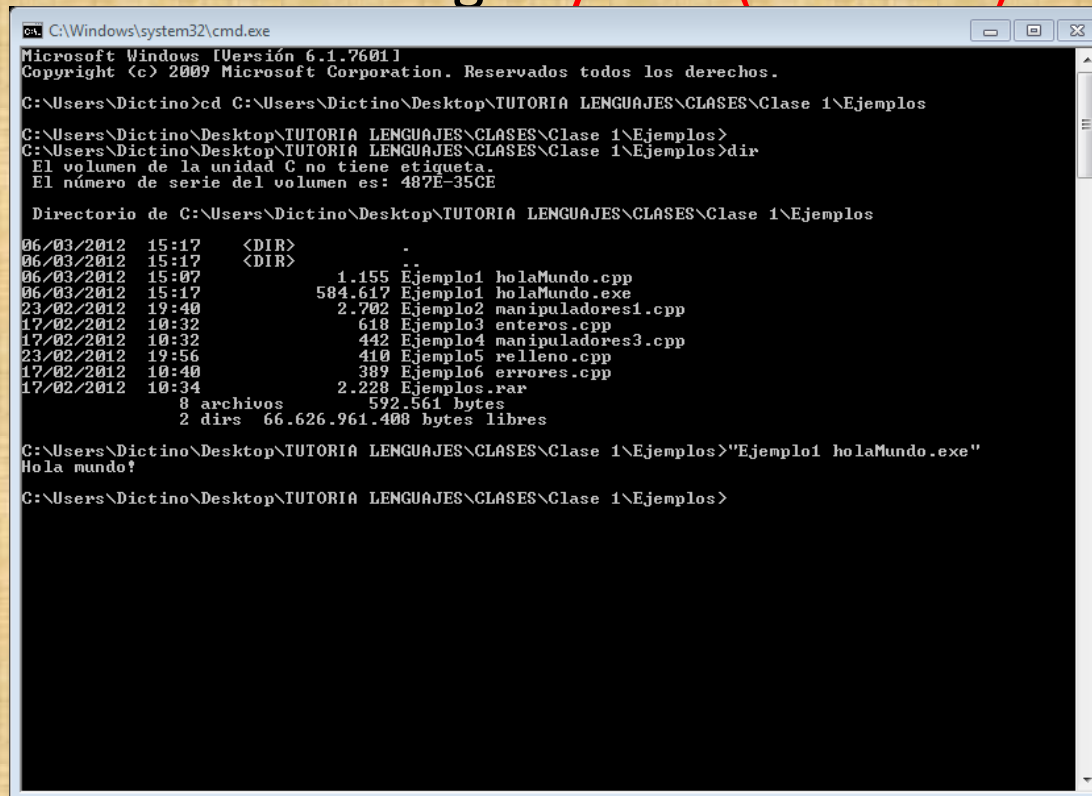








- Por defecto no se ve nada (la ventana se cierra muy deprisa) tenemos varias opciones:
 - Por línea de comandos (cmd)
 - Modificando el código `system("PAUSE")`



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

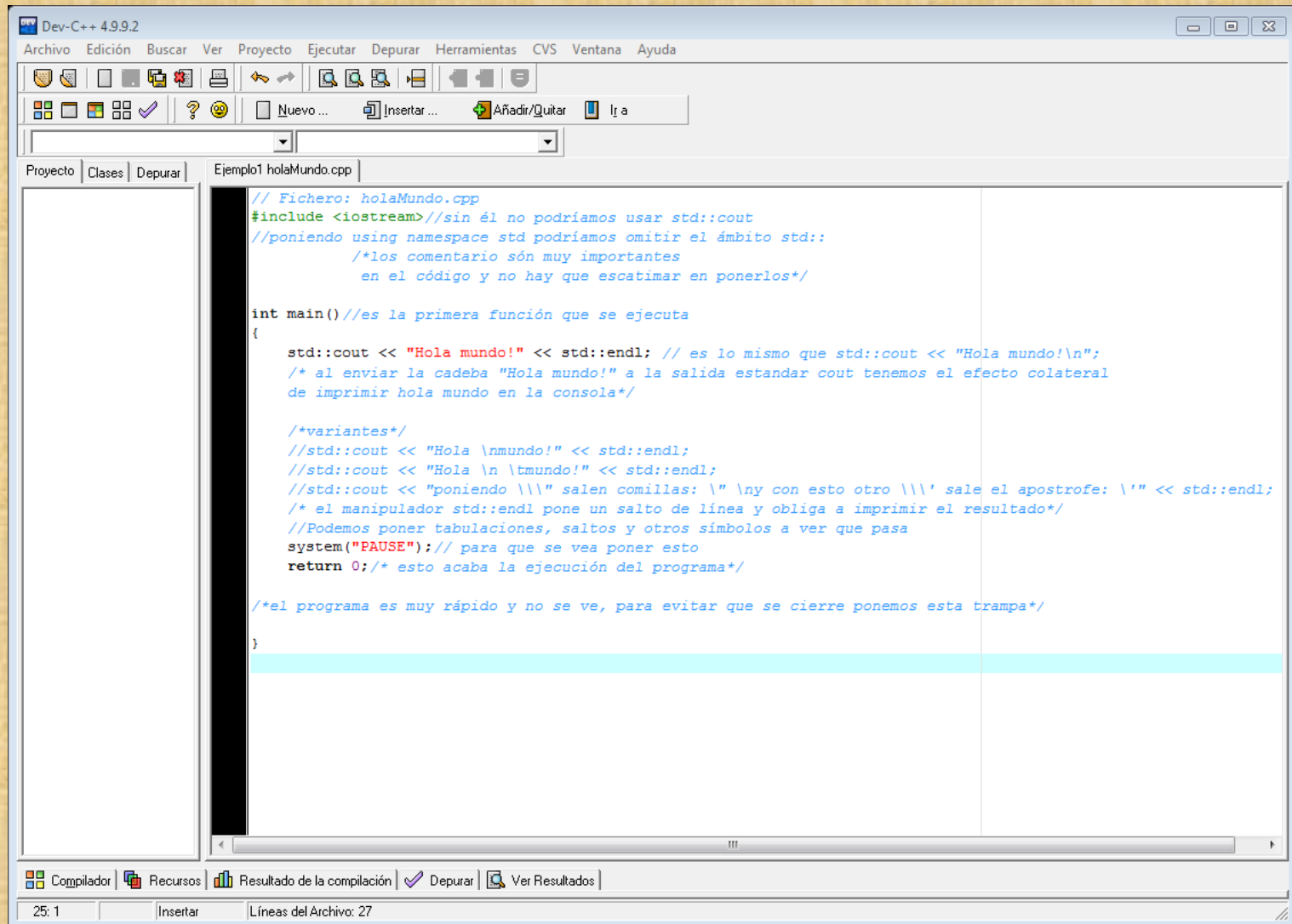
C:\Users\Dictino>cd C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos>
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 487E-35CE

Directorio de C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos
06/03/2012  15:17    <DIR>          .
06/03/2012  15:17    <DIR>          ..
06/03/2012  15:07               1.155 Ejemplo1 holaMundo.cpp
06/03/2012  15:17             584.617 Ejemplo1 holaMundo.exe
23/02/2012  19:40             2.702 Ejemplo2 manipuladores1.cpp
17/02/2012  10:32              618 Ejemplo3 enteros.cpp
17/02/2012  10:32              442 Ejemplo4 manipuladores3.cpp
23/02/2012  19:56              410 Ejemplo5 relleno.cpp
17/02/2012  10:40              389 Ejemplo6 errores.cpp
17/02/2012  10:34             2.228 Ejemplos.rar
                8 archivos             592.561 bytes
                2 dirs 66.626.961.408 bytes libres

C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos>"Ejemplo1 holaMundo.exe"
Hola mundo!

C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos>
```


Solución 2: system("PAUSE");



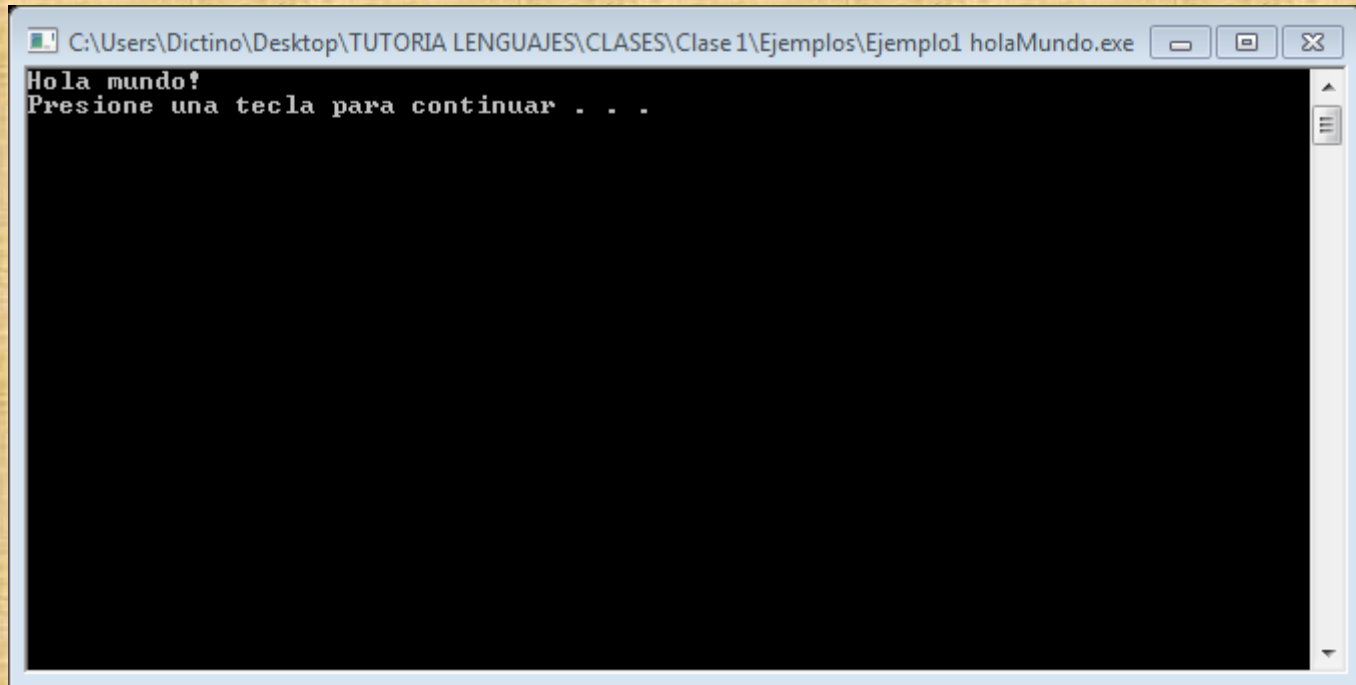
```
// Fichero: holaMundo.cpp
#include <iostream> // sin él no podríamos usar std::cout
// poniendo using namespace std podríamos omitir el ámbito std::
/* los comentarios son muy importantes
   en el código y no hay que escatimar en ponerlos */

int main() // es la primera función que se ejecuta
{
    std::cout << "Hola mundo!" << std::endl; // es lo mismo que std::cout << "Hola mundo!\n";
    /* al enviar la cadena "Hola mundo!" a la salida estándar cout tenemos el efecto colateral
       de imprimir hola mundo en la consola */

    /* variantes */
    // std::cout << "Hola \nmundo!" << std::endl;
    // std::cout << "Hola \n \tmundo!" << std::endl;
    // std::cout << "poniendo \\" salen comillas: \" \ny con esto otro \\" sale el apostrofe: \'\" << std::endl;
    /* el manipulador std::endl pone un salto de línea y obliga a imprimir el resultado */
    // Podemos poner tabulaciones, saltos y otros símbolos a ver que pasa
    system("PAUSE"); // para que se vea poner esto
    return 0; /* esto acaba la ejecución del programa */

    /* el programa es muy rápido y no se ve, para evitar que se cierre ponemos esta trampa */
}
```

Hola mundo con pausa



```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos\Ejemplo1 holaMundo.exe
Hola mundo!
Presione una tecla para continuar . . .
```

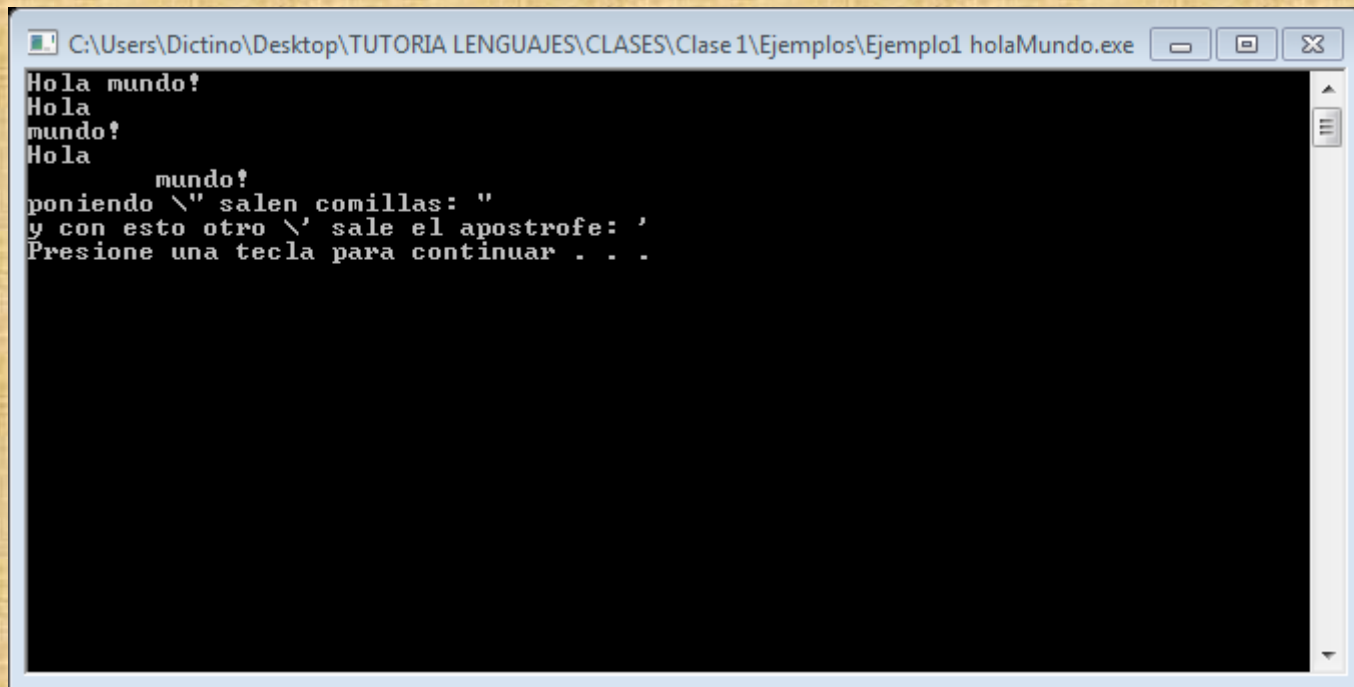
```
// Fichero: holaMundo.cpp
#include <iostream>//sin él no podríamos usar std::cout
//poniendo using namespace std podríamos omitir el ámbito std::
/*los comentario són muy importantes
en el código y no hay que escatimar en ponerlos*/

int main()//es la primera función que se ejecuta
{
    std::cout << "Hola mundo!" << std::endl; // es lo mismo que std::cout << "Hola mundo!\n";
    /* al enviar la cadeba "Hola mundo!" a la salida estandar cout tenemos el efecto colateral
de imprimir hola mundo en la consola*/

    /*variantes*/
    std::cout << "Hola \nmundo!" << std::endl;
    std::cout << "Hola \n \tmundo!" << std::endl;
    std::cout << "poniendo \\" salen comillas: \" \ny con esto otro \\' sale el apostrofe: \'\" << std::endl;
    /* el manipulador std::endl pone un salto de línea y obliga a imprimir el resultado*/
    //Podemos poner tabulaciones, saltos y otros símbolos a ver que pasa
    system("PAUSE");// para que se vea poner esto
    return 0; /* esto acaba la ejecución del programa*/

    /*el programa es muy rápido y no se ve, para evitar que se cierre ponemos esta trampa*/
}
```

Hola mundo modificado



```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos\Ejemplo1 holaMundo.exe
Hola mundo!
Hola
mundo!
Hola    mundo!
poniendo \" salen comillas: "
y con esto otro \' sale el apostrofe: '
Presione una tecla para continuar . . .
```

Salida de datos por consola

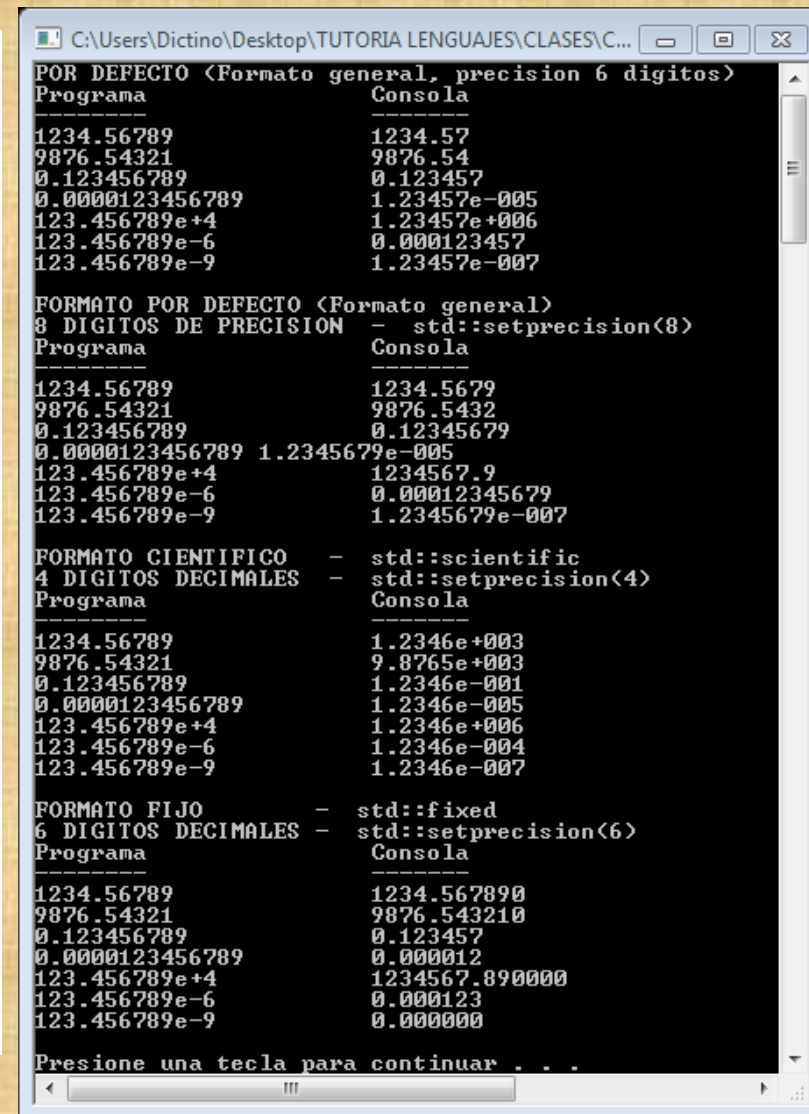
- Es la operación más importante en un programa básico
- Necesitamos
 - `#include<iostream>` (básico)
 - `#include<iomanip>` (manipulación de flujos)
- La salida es un flujo al que enviamos cosas con `<<` de vez en cuando el flujo es enviado a la consola
 - Cuando se llena el buffer
 - Cuando se pide explícitamente
- Podemos usar **manipuladores** para hacer cosas al flujo por ejemplo forzar la impresión
 - `std::endl` pone un salto de línea y fuerza la escritura
 - `std::flush` fuerza la impresión del buffer

Tipos de salida

- Punto flotante Ejemplo 2
 - Formato estándar (por defecto)
 - `std::setprecision(N)` modifica el nº de decimales N del redondeo
 - Formato científico `std::scientific`
 - Formato de punto fijo `std::fixed`

Punto flotante

```
#include <iostream>
#include <iomanip>
int main()
{ // Formato general, 6 digitos de precisión
  std::cout <<
    "POR DEFECTO (Formato general, precision 6 digitos)\n" <<
    "Programa\t\tConsola\n" <<
    "-----\t\t-----\n" <<
    "1234.56789\t\t" << 1234.56789 << "\n" <<
    "9876.54321\t\t" << 9876.54321 << "\n" <<
    "0.123456789\t\t" << 0.123456789 << "\n" <<
    "0.0000123456789\t\t" << 0.0000123456789 << "\n" <<
    "123.456789e+4\t\t" << 123.456789e+4 << "\n" <<
    "123.456789e-6\t\t" << 123.456789e-6 << "\n" <<
    "123.456789e-9\t\t" << 123.456789e-9 << "\n" <<
    std::endl;
  // Formato general, 8 digitos de precisión
  std::cout <<
    "FORMATO POR DEFECTO (Formato general)\n" <<
    "8 DIGITOS DE PRECISION - std::setprecision(8)\n" <<
    "Programa\t\tConsola\n" <<
    "-----\t\t-----\n" << std::setprecision(8) <<
    "1234.56789\t\t" << 1234.56789 << "\n" <<
    "9876.54321\t\t" << 9876.54321 << "\n" <<
    "0.123456789\t\t" << 0.123456789 << "\n" <<
    "0.0000123456789\t\t" << 0.0000123456789 << "\n" <<
    "123.456789e+4\t\t" << 123.456789e+4 << "\n" <<
    "123.456789e-6\t\t" << 123.456789e-6 << "\n" <<
    "123.456789e-9\t\t" << 123.456789e-9 << "\n" <<
    std::endl;
```



```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\C...
POR DEFECTO (Formato general, precision 6 digitos)
Programa                               Consola
-----
1234.56789                             1234.57
9876.54321                             9876.54
0.123456789                           0.123457
0.0000123456789                       1.23457e-005
123.456789e+4                         1.23457e+006
123.456789e-6                         0.000123457
123.456789e-9                         1.23457e-007

FORMATO POR DEFECTO (Formato general)
8 DIGITOS DE PRECISION - std::setprecision(8)
Programa                               Consola
-----
1234.56789                             1234.5679
9876.54321                             9876.5432
0.123456789                           0.12345679
0.0000123456789                       1.2345679e-005
123.456789e+4                         1234567.9
123.456789e-6                         0.00012345679
123.456789e-9                         1.2345679e-007

FORMATO CIENTIFICO - std::scientific
4 DIGITOS DECIMALES - std::setprecision(4)
Programa                               Consola
-----
1234.56789                             1.2346e+003
9876.54321                             9.8765e+003
0.123456789                           1.2346e-001
0.0000123456789                       1.2346e-005
123.456789e+4                         1.2346e+006
123.456789e-6                         1.2346e-004
123.456789e-9                         1.2346e-007

FORMATO FIJO - std::fixed
6 DIGITOS DECIMALES - std::setprecision(6)
Programa                               Consola
-----
1234.56789                             1234.567890
9876.54321                             9876.543210
0.123456789                           0.123457
0.0000123456789                       0.000012
123.456789e+4                         1234567.890000
123.456789e-6                         0.000123
123.456789e-9                         0.000000

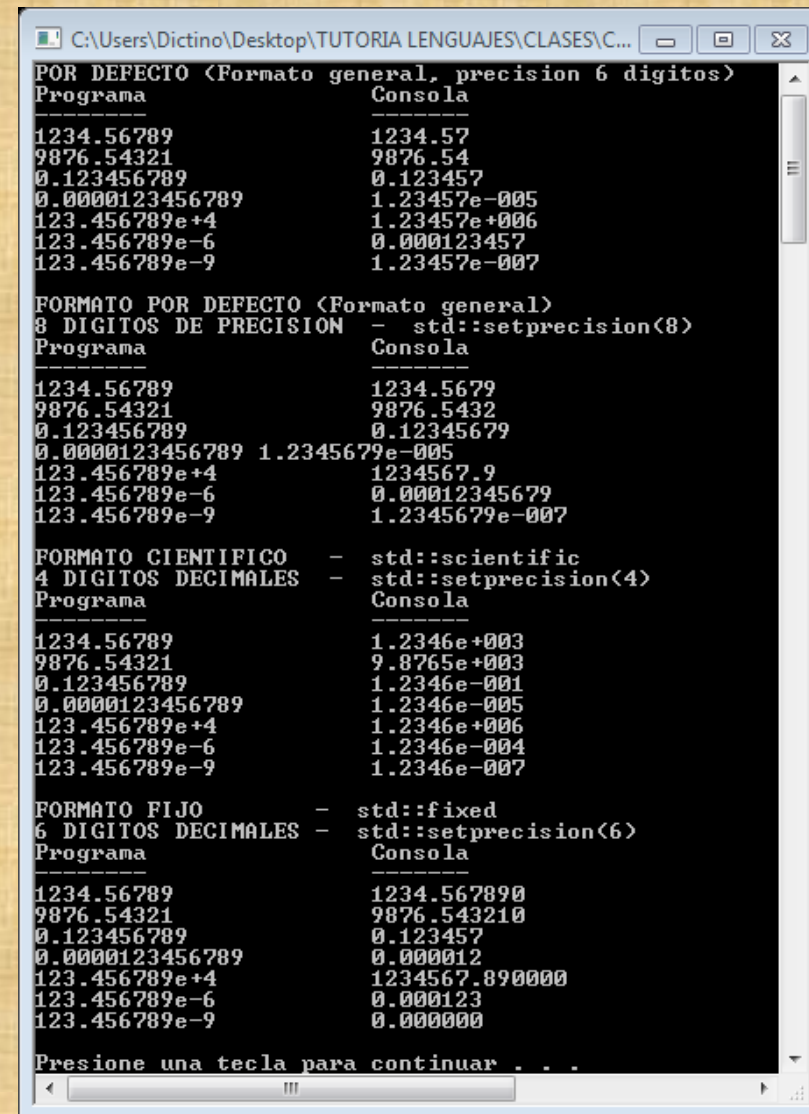
Presione una tecla para continuar . . .
```

Punto flotante



```
// Formato científico, 4 dígitos de precisión
std::cout << std::scientific << std::setprecision(4) <<
"FORMATO CIENTIFICO - std::scientific\n" <<
"4 DIGITOS DECIMALES - std::setprecision(4)\n" <<
"Programa\t\tConsola\n" <<
"-----\t\t-----\n" <<
"1234.56789\t\t" << 1234.56789 << "\n" <<
"9876.54321\t\t" << 9876.54321 << "\n" <<
"0.123456789\t\t" << 0.123456789 << "\n" <<
"0.0000123456789\t\t" << 0.0000123456789 << "\n" <<
"123.456789e+4\t\t" << 123.456789e+4 << "\n" <<
"123.456789e-6\t\t" << 123.456789e-6 << "\n" <<
"123.456789e-9\t\t" << 123.456789e-9 << "\n" <<
std::endl;

// Formato fijo, 6 dígitos de precisión
std::cout << std::fixed << std::setprecision(6) <<
"FORMATO FIJO - std::fixed\n" <<
"6 DIGITOS DECIMALES - std::setprecision(6)\n" <<
"Programa\t\tConsola\n" <<
"-----\t\t-----\n" <<
"1234.56789\t\t" << 1234.56789 << "\n" <<
"9876.54321\t\t" << 9876.54321 << "\n" <<
"0.123456789\t\t" << 0.123456789 << "\n" <<
"0.0000123456789\t\t" << 0.0000123456789 << "\n" <<
"123.456789e+4\t\t" << 123.456789e+4 << "\n" <<
"123.456789e-6\t\t" << 123.456789e-6 << "\n" <<
"123.456789e-9\t\t" << 123.456789e-9 << "\n" <<
std::endl;
system("PAUSE");
return 0;
}
```



```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\C...
POR DEFECTO (Formato general, precision 6 digitos)
Programa          Consola
1234.56789        1234.57
9876.54321        9876.54
0.123456789       0.123457
0.0000123456789   1.23457e-005
123.456789e+4      1.23457e+006
123.456789e-6      0.000123457
123.456789e-9      1.23457e-007

FORMATO POR DEFECTO (Formato general)
8 DIGITOS DE PRECISION - std::setprecision(8)
Programa          Consola
1234.56789        1234.5679
9876.54321        9876.5432
0.123456789       0.12345679
0.0000123456789   1.2345679e-005
123.456789e+4      1234567.9
123.456789e-6      0.00012345679
123.456789e-9      1.2345679e-007

FORMATO CIENTIFICO - std::scientific
4 DIGITOS DECIMALES - std::setprecision(4)
Programa          Consola
1234.56789        1.2346e+003
9876.54321        9.8765e+003
0.123456789       1.2346e-001
0.0000123456789   1.2346e-005
123.456789e+4      1.2346e+006
123.456789e-6      1.2346e-004
123.456789e-9      1.2346e-007

FORMATO FIJO - std::fixed
6 DIGITOS DECIMALES - std::setprecision(6)
Programa          Consola
1234.56789        1234.567890
9876.54321        9876.543210
0.123456789       0.123457
0.0000123456789   0.000012
123.456789e+4      1234567.890000
123.456789e-6      0.000123
123.456789e-9      0.000000

Presione una tecla para continuar . . .
```

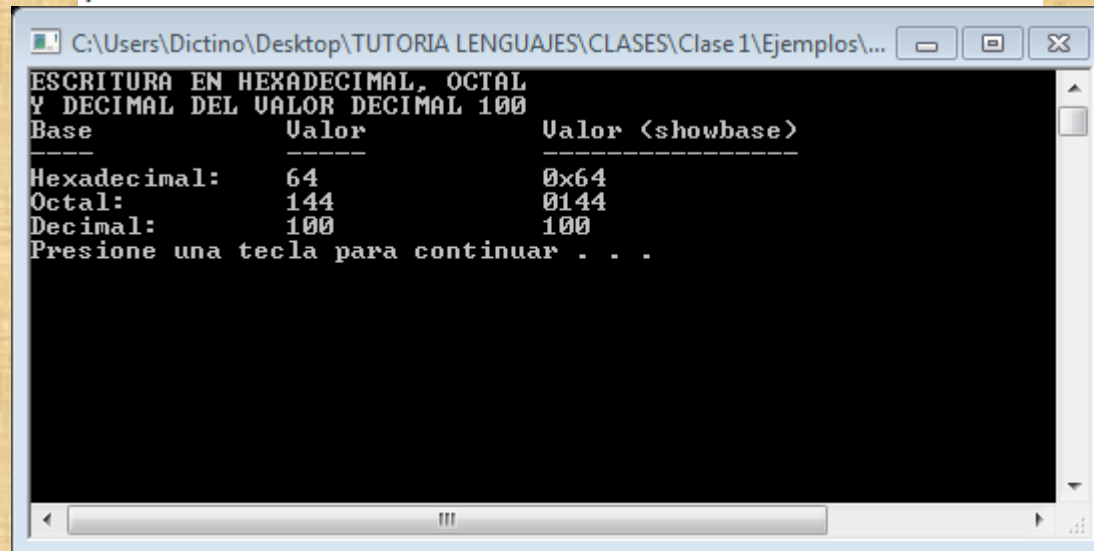
Tipos de salida

- Enteros Ejemplo 3
 - Decimal, octal y hexadecimal (`std::dec`, `std::oct`, `std::hex`)
 - Los indicadores de formato **0** y **0x** pueden mostrarse y ocultarse (`std::showbase`, `std::noshowbase`)

Enteros

```
// Fichero: manipuladores2.cpp
#include <iostream>
#include <iomanip>

int main()
{
    std::cout << "ESCRITURA EN HEXADECIMAL, OCTAL\n" <<
        "Y DECIMAL DEL VALOR DECIMAL " << 100 <<
        "\nBase\t\tValor\t\tValor (showbase)" <<
        "\n---\t\t---\t\t-----\n" <<
        "Hexadecimal:\t" << std::hex <<
        100 << "\t\t" << std::showbase << 100 << "\n" <<
        "Octal:\t\t" << std::oct << std::noshowbase <<
        100 << "\t\t" << std::showbase << 100 << "\n" <<
        "Decimal:\t" << std::dec << std::noshowbase <<
        100 << "\t\t" << std::showbase << 100 << std::endl;
    system("PAUSE");
    return 0;
}
```



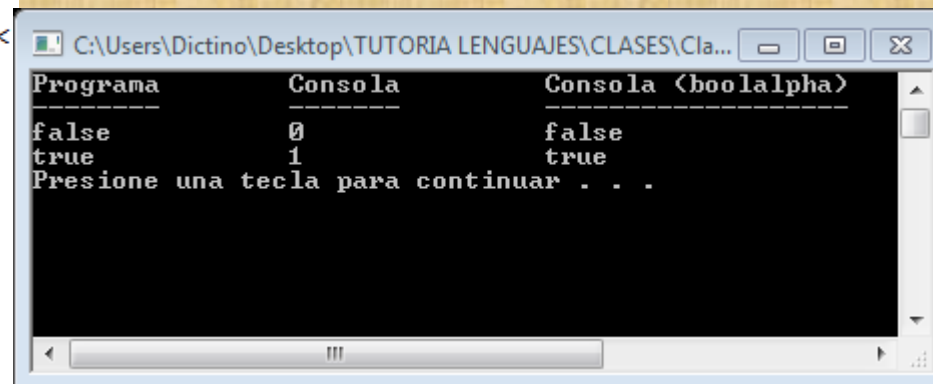
```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos\...
ESCRITURA EN HEXADECIMAL, OCTAL
Y DECIMAL DEL VALOR DECIMAL 100
Base          Valor          Valor (showbase)
---
Hexadecimal:  64          0x64
Octal:        144         0144
Decimal:      100         100
Presione una tecla para continuar . . .
```


Tipos de salida

- Booleanos Ejemplo 4
 - Por defecto se imprime 1 o 0
 - Para poner true/false se usa `std::boolalpha`
`std::noboolalpha`

```
#include <iostream>
#include <iomanip>

int main()
{
    std::cout <<
        "Programa\tConsola\t\tConsola (boolalpha)\n" <<
        "-----\t-----\t\t-----\n" <<
        "false\t\t" <<
        false << "\t\t" <<
        std::boolalpha << false << "\n" <<
        "true\t\t" <<
        std::noboolalpha << true << "\t\t" <<
        std::boolalpha << true << std::endl;
    system("PAUSE");
    return 0;
}
```



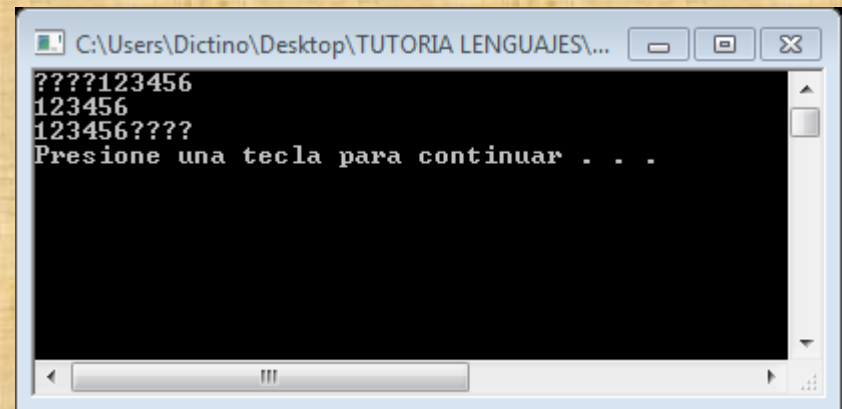
```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Cla...
Programa      Consola      Consola (boolalpha)
-----
false         0           false
true          1           true
Presione una tecla para continuar . . .
```

Relleno

- Relleno Ejemplo 5

- Aumenta el tamaño de la salida con caracteres
 - `std::setw` establece el ancho mínimo
 - `std::setfill(R)` establece el carácter (no string) de relleno R
 - `std::left` `std::right` establece por que lado rellenar
- El efecto es **local** (solo afecta a la entrada siguiente)

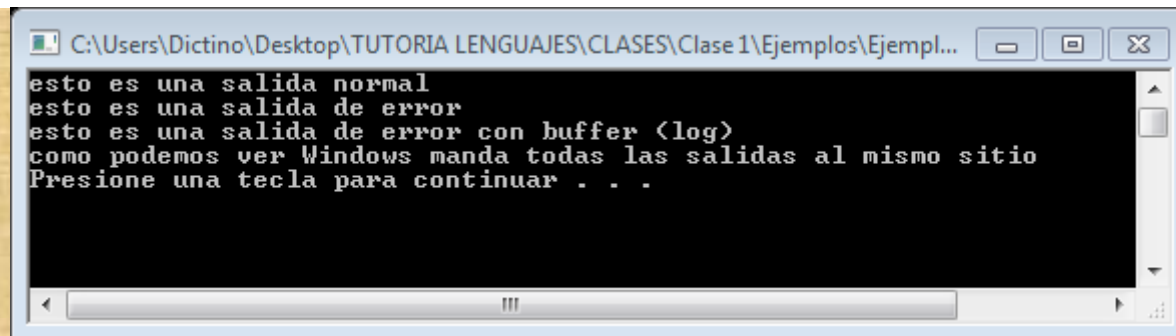
```
#include<iostream>
#include<iomanip>
int main(){
    /*esto rellena por la derecha*/
    std::cout<<std::setw(10)<<std::setfill('?')<<
    "123456" << std::endl;
    std::cout<<std::setw(5)<<std::setfill('?')<<
    "123456" << std::endl;
    /*esto rellena por la izquierda*/
    std::cout<<std::setw(10)<<std::setfill('?')<<std::left<<
    "123456" << std::endl;
    system("PAUSE");
    return 0;
}
```



Salidas especiales

- Otros flujos
 - `std::cin` entrada estándar
 - `std::cout` salida estándar
 - `std::cerr` salida de error (**inmediata**)
 - `std::clog` salida de error con buffer (actúa como `cout`)

```
#include<iostream>
#include<iomanip>
int main() {
    /*esto rellena por la derecha*/
    std::cout<<"esto es una salida normal\n";
    std::cerr<<"esto es una salida de error\n";
    std::clog<<"esto es una salida de error con buffer (log)\n";
    std::clog<<"como podemos ver Windows manda todas las salidas al mismo sitio \n";
    system("PAUSE");
    return 0;
}
```



```
C:\Users\Dictino\Desktop\TUTORIA LENGUAJES\CLASES\Clase 1\Ejemplos\Ejempl...
esto es una salida normal
esto es una salida de error
esto es una salida de error con buffer <log>
como podemos ver Windows manda todas las salidas al mismo sitio
Presione una tecla para continuar . . .
```