

Estructura básica de un programa C++

```
#include <iostream>

int main()
{
    std::cout << "¡Hola Mundo!" << std::endl;

    return 0;
}
```

A continuación se detalla cada uno de los elementos presentes:

Línea 1

```
#include <iostream>
```

La líneas que comienzan con el signo `#` se conocen como **directivas** y son leídas e interpretadas por el **pre-procesador**. Son líneas especiales interpretadas *antes* de la compilación del programa.

Esta directiva en particular incluye una sección del código estándar de C++, conocida como *header iostream*, que permite realizar operaciones de entrada y salida, como escribir la salida `¡Hola Mundo!` a la pantalla.

Línea 3

```
int main()
```

Declaración de la función `main`. Es el punto de partida de todo programa escrito en C++.

Línea 4 y 8

```
{ }
```

Indican la apertura (`{`) y cierre (`}`) de un bloque (en este caso la función `main`). Todo lo que esté contenido entre estos dos símbolos se considera el *cuerpo* del bloque.

Línea 5

```
std::cout << "¡Hola Mundo!" << std::endl;
```

Básicamente imprime en la salida o *output* el mensaje `¡Hola Mundo!`.

Instrucción	Significado
<code>std::cout</code>	Significa s tandard c haracter o utput. En este caso se refiere a la pantalla del computador.
<code><<</code>	Operador de inserción. Indica que lo que sigue debe ser insertado en la salida.
<code>"¡Hola Mundo!"</code>	Cadena de texto que será insertada en la salida.
<code>std::endl</code>	Fin de la línea (end line). En otras palabras, inserta un salto de línea. Equivalente a escribir <code>"\n"</code> .
<code>;</code>	Se utiliza para finalizar una instrucción. Todas las instrucciones de un programa C++ deben finalizar con el signo <code>;</code> . Uno de los errores más comunes es la <i>omisión</i> de este signo.

Línea 7

```
return 0;
```

Código de retorno de la función `main`. Si no se especifica el código de retorno y el programa finaliza de forma apropiada, *implícitamente* se retornará 0, lo que significa que la ejecución terminó de forma correcta.

Otros valores (distintos a 0) pueden ser retornados por la función `main`.

Los valores retornados que se garantizan ser interpretados de la misma forma por todas las plataforma son:

Valor	Descripción
0	Ejecución exitosa
<code>EXIT_SUCCESS</code>	Ejecución exitosa (igual que 0). Este valor está definido en la directiva <code><cstdlib></code>
<code>EXIT_FAILURE</code>	Ejecución fallida. Este valor está definido en la directiva <code><cstdlib></code>

Comentarios y DocBlocks

```
// Comentario simple de una línea
```

Dos `/` indican que lo que sigue es un *comentario simple*. Se utilizan para documentar alguna funcionalidad o dejar observaciones. No influyen en el comportamiento del programa.

```
/*  
 * Comentario de  
 * múltiples líneas  
 */
```

Idem a los comentarios simples, con la diferencia de que se pueden insertar múltiples líneas dentro de los delimitadores `/* */`.

```
/**  
 * Función que suma 2 valores  
 *  
 * @autor Roberto De la Fuente <rdelafue@unap.cl>  
 * @param int a Parámetro 1  
 * @param int b Parámetro 2  
  
 * @return int  
 */  
int suma(int a, int b);
```

Los bloques de documentación se utilizan para documentar clases, funciones o estructuras. Son similares a los comentarios de múltiples líneas, con la diferencia que estos empiezan con `/**`.

Dentro del bloque se suelen utilizar metadatos especiales (comienzan con `@`), que permiten *darle forma* a la documentación de salida.

Son especialmente útiles cuando se trabaja en proyectos de gran envergadura, con múltiples desarrolladores y dentro de entornos integrados de desarrollo (IDE).