**Tipos de dato en C++**

[C++](http://ehack.info/glosario/return-c/) admite 7 tipos fundamentales de tipos de dato: *char, int, bool, float, double, void y enum*, algunos además definen subtipos que modifican el rango del tipo de dato, de esta forma podemos definir una serie de “familias” en los tipos de dato.

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Familia** | **Tipo Base** | **Subtipos** | **Rango\* (Compliladores bajo i386)** | **Comentario** |
| Enteros | char | char, wchar\_t | char – 1 byte  wchar\_t – | char – Almacena caracteres ASCII, es equivalente a un entero  wchar\_t – Almacena caracteres Unicode |
| Booleanos | bool | bool | 0-1 | Representa valores que pueden ser true o false. |
| Enteros | int | short, int, long, long long | short – 2 byte | Los objetos del tipo short se pueden declarar como signed short o unsigned short.Signed short es un sinónimo de short int. |
| int – 4 byte | Los objetos del tipo int se pueden declarar como signed int o unsigned int.Signed int es un sinónimo de int. |
| long – 8 byte | Los objetos del tipo long se pueden declarar como signed long o unsigned long.Signed long es un sinónimo de long. |
| long long – 16 byte | Los objetos del tipo long long se pueden declarar como signed long long o unsigned long long.Signed long long es un sinónimo de long long. |
| De punto flotante | double | float, double, long double | float – 4 byte | El tipo float es el tipo de punto flotante más pequeño. |
| double – 8 byte | El tipo double es un tipo flotante superior o igual al tipo float, pero inferior o igual al tamaño del tipo long double. Opción predeterminada para los valores de punto flotante. |
| long double – 16 byte | El tipo long double es un tipo de punto flotante que es superior o igual al tipo double. |
|  | void |  | Null | El tipo void describe un conjunto de valores vacío. No se puede especificar ninguna variable de tipo void; se utiliza principalmente para declarar funciones que no devuelven ningún valor o para declarar punteros genéricos a datos sin tipo o con un tipo arbitrario. |
|  | enum |  | varía | Una enumeración es un tipo definido por el usuario que consta de un conjunto de constantes enteras con nombre conocidas como enumeradores. |

**Modificadores de tipos de dato:  short, long, signed y unsigned**

Como puedes apreciar los tipos base (concretamente los int y double) pueden modificar su rango con el uso de los modificadores, en este caso:

### **short y long**

* short in: Entero corto
* long int: Entero Largo
* long double: Largo doble (punto flotante)

De forma predeterminada los tipos enteros y char son con signo, tal como se definen en álgebra, de este modo está implícito en su definición el modificador:

### **signed** (con signo)

* signed char es equivalente a char
* signed int es equivalente a int
* signed long es equivalente a long
* signed long long es equivalente a long long

Podemos ampliar el “tamaño hacia el lado positivo” de una variable entera utilizando el modificador unsigned

### **unsigned** (sin signo)

* unsigned short
* unsigned int
* unsigned long
* unsigned long long

En un sistema con arquitectura i386, los rangos definidos para una variable entera pueden ser:

int a;

a, puede tomar valores entre –2,147,483,648 a 2,147,483,647

unsigned int a;

a, puede tomar valores entre 0 a 4,294,967,295

Como puedes apreciar los tipos base (concretamente los int y double) pueden modificar su rango con el uso de los modificadores, en este caso:

### **short y long**

* short in: Entero corto
* long int: Entero Largo
* long double: Largo doble (punto flotante)

De forma predeterminada los tipos enteros y char son con signo, tal como se definen en álgebra, de este modo está implícito en su definición el modificador:

### **signed** (con signo)

* signed char es equivalente a char
* signed int es equivalente a int
* signed long es equivalente a long
* signed long long es equivalente a long long

Podemos ampliar el “tamaño hacia el lado positivo” de una variable entera utilizando el modificador unsigned

### **unsigned** (sin signo)

* unsigned short
* unsigned int
* unsigned long
* unsigned long long

En un sistema con arquitectura i386, los rangos definidos para una variable entera pueden ser:

int a;

a, puede tomar valores entre –2,147,483,648 a 2,147,483,647

unsigned int a;

a, puede tomar valores entre 0 a 4,294,967,295

### Valores de los Tipos de dato en C++

Con esta información en mente podemos presentar los valores numéricos de los tipos de dato fundamentales en [C++](http://ehack.info/glosario/return-c/)

| **Nombre de tipo** | **Bytes** | **Otros nombres** | **Intervalo de valores** |
| --- | --- | --- | --- |
| int | 4 | signed | De –2,147,483,648 a 2,147,483,647 |
| unsigned int | 4 | unsigned | De 0 a 4,294,967,295 |
| bool | 1 | ninguna | false o true |
| char | 1 | ninguna | De –128 a 127 de manera predeterminada |
| signed char | 1 | ninguna | De –128 a 127 |
| unsigned char | 1 | ninguna | De 0 a 255 |
| short | 2 | short int, signed short int | De –32,768 a 32,767 |
| unsigned short | 2 | unsigned short int | De 0 a 65,535 |
| long | 4 | long int, signed long int | De –2,147,483,648 a 2,147,483,647 |
| unsigned long | 4 | unsigned long int | De 0 a 4,294,967,295 |
| long long | 8 | ninguno | De –9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 |
| unsigned long long | 8 | ninguno | De 0 a 18,446,744,073,709,551,615 |
| enum | varía | ninguna | Vea más adelante en este artículo |
| float | 4 | ninguna | 3.4E +/- 38 (7 dígitos) |
| double | 8 | ninguna | 1.7E +/- 308 (15 dígitos) |
| long double | igual que double | ninguna | Igual que double |
| wchar\_t | 2 | ninguna | De 0 a 65,535 |
| void |  | ninguna |  |
| enum | Variable | ninguna | Variable |

### **Prioridad y asociatividad de los operadores de C++**

| **Descripción del operador** | **Operador** |
| --- | --- |
| Group 1 precedence, no associativity |  |
| [Resolución de ámbito](https://msdn.microsoft.com/es-es/library/b451xz31.aspx) | :: |
| Group 2 precedence, left to right associativity |  |
| [Selección de miembro (objeto o puntero)](https://msdn.microsoft.com/es-es/library/b930c881.aspx) | . or –> |
| [Subíndice de matriz](https://msdn.microsoft.com/es-es/library/th52k9c8.aspx) | [ ] |
| [Llamada a función](https://msdn.microsoft.com/es-es/library/kdte99aa.aspx) | ( ) |
| [Incremento de postfijo](https://msdn.microsoft.com/es-es/library/e1e3921c.aspx) | ++ |
| [Decremento de postfijo](https://msdn.microsoft.com/es-es/library/e1e3921c.aspx) | –– |
| [Nombre de tipo](https://msdn.microsoft.com/es-es/library/fyf39xec.aspx) | typeid( ) |
| [Conversión de tipos constante](https://msdn.microsoft.com/es-es/library/bz6at95h.aspx) | const\_cast |
| [Conversión de tipos dinámica](https://msdn.microsoft.com/es-es/library/cby9kycs.aspx) | dynamic\_cast |
| [Conversión de tipos reinterpretada](https://msdn.microsoft.com/es-es/library/e0w9f63b.aspx) | reinterpret\_cast |
| [Conversión de tipos estática](https://msdn.microsoft.com/es-es/library/c36yw7x9.aspx) | static\_cast |
| Group 3 precedence, right to left associativity |  |
| [Tamaño de objeto o tipo](https://msdn.microsoft.com/es-es/library/4s7x1k91.aspx) | sizeof |
| [Incremento de prefijo](https://msdn.microsoft.com/es-es/library/dy3d35h8.aspx) | ++ |
| [Decremento de prefijo](https://msdn.microsoft.com/es-es/library/dy3d35h8.aspx) | –– |
| [Complemento a uno](https://msdn.microsoft.com/es-es/library/dxt4z71k.aspx) | ~ |
| [NOT lógico](https://msdn.microsoft.com/es-es/library/1k6w8551.aspx) | ! |
| [Negación unaria](https://msdn.microsoft.com/es-es/library/wxxx8d2t.aspx) | - |
| [Unario más](https://msdn.microsoft.com/es-es/library/ewkkxkwb.aspx) | + |
| [Dirección de](https://msdn.microsoft.com/es-es/library/w7049scy.aspx) | & |
| [Direccionamiento indirecto](https://msdn.microsoft.com/es-es/library/fw63e3c3.aspx) | \* |
| [Crear objeto](https://msdn.microsoft.com/es-es/library/kewsb8ba.aspx) | new |
| [Destruir objeto](https://msdn.microsoft.com/es-es/library/h6227113.aspx) | delete |
| [Conversión de tipos explícita](https://msdn.microsoft.com/es-es/library/ts48df3y.aspx) | Cast: () |
| Group 4 precedence, left to right associativity |  |
| [Puntero a miembro (objetos o punteros)](https://msdn.microsoft.com/es-es/library/k8336763.aspx) | .\* or –>\* |
| Group 5 precedence, left to right associativity |  |
| [Multiplicación](https://msdn.microsoft.com/es-es/library/ty2ax9z9.aspx) | \* |
| [División](https://msdn.microsoft.com/es-es/library/ty2ax9z9.aspx) | / |
| [Módulo](https://msdn.microsoft.com/es-es/library/ty2ax9z9.aspx) | % |
| Group 6 precedence, left to right associativity |  |
| [Adición](https://msdn.microsoft.com/es-es/library/ktz6ya2s.aspx) | + |
| [Resta](https://msdn.microsoft.com/es-es/library/ktz6ya2s.aspx) | – |
| Group 7 precedence, left to right associativity |  |
| [Desplazamiento a la izquierda](https://msdn.microsoft.com/es-es/library/336xbhcz.aspx) | << |
| [Desplazamiento a la derecha](https://msdn.microsoft.com/es-es/library/336xbhcz.aspx) | >> |
| Group 8 precedence, left to right associativity |  |
| [Menor que](https://msdn.microsoft.com/es-es/library/yf3abyat.aspx) | < |
| [Mayor que](https://msdn.microsoft.com/es-es/library/yf3abyat.aspx) | > |
| [Menor o igual que](https://msdn.microsoft.com/es-es/library/yf3abyat.aspx) | <= |
| [Mayor o igual que](https://msdn.microsoft.com/es-es/library/yf3abyat.aspx) | >= |
| Group 9 precedence, left to right associativity |  |
| [Igualdad](https://msdn.microsoft.com/es-es/library/c35t2ffz.aspx) | == |
| [Desigualdad](https://msdn.microsoft.com/es-es/library/c35t2ffz.aspx) | != |
| Group 10 precedence left to right associativity |  |
| [AND bit a bit](https://msdn.microsoft.com/es-es/library/z0zec0b2.aspx) | & |
| Group 11 precedence, left to right associativity |  |
| [OR exclusivo bit a bit](https://msdn.microsoft.com/es-es/library/3akey979.aspx) | ^ |
| Group 12 precedence, left to right associativity |  |
| [OR inclusivo bit a bit](https://msdn.microsoft.com/es-es/library/edc0fscw.aspx) | &#124; |
| Group 13 precedence, left to right associativity |  |
| [AND lógico](https://msdn.microsoft.com/es-es/library/c6s3h5a7.aspx) | && |
| Group 14 precedence, left to right associativity |  |
| [OR lógico](https://msdn.microsoft.com/es-es/library/f355wky8.aspx) | &#124;&#124; |
| Group 15 precedence, right to left associativity |  |
| [Condicional](https://msdn.microsoft.com/es-es/library/e4213hs1.aspx) | ? : |
| Group 16 precedence, right to left associativity |  |
| [Asignación](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | = |
| [Asignación y multiplicación](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | \*= |
| [Asignación y división](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | /= |
| [Asignación y módulo](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | %= |
| [Asignación y suma](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | += |
| [Asignación y resta](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | –= |
| [Asignación y desplazamiento a la izquierda](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | <<= |
| [Asignación y desplazamiento a la derecha](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | >>= |
| [Asignación AND bit a bit](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | &= |
| [Asignación OR inclusivo bit a bit](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | &#124;= |
| [Asignación OR exclusivo bit a bit](https://msdn.microsoft.com/es-es/library/5bk6ya5x.aspx) | ^= |
| Group 17 precedence, right to left associativity |  |
| [Expresión Throw](https://msdn.microsoft.com/es-es/library/6dekhbbc.aspx) | throw |
| Group 18 precedence, left to right associativity |  |
| [Coma](https://msdn.microsoft.com/es-es/library/zs06xbxh.aspx) | , |

## Bibliografia

## <http://ehack.info/tipo-datos-basicos-c/>

## <https://msdn.microsoft.com/es-es/library/126fe14k.aspx>