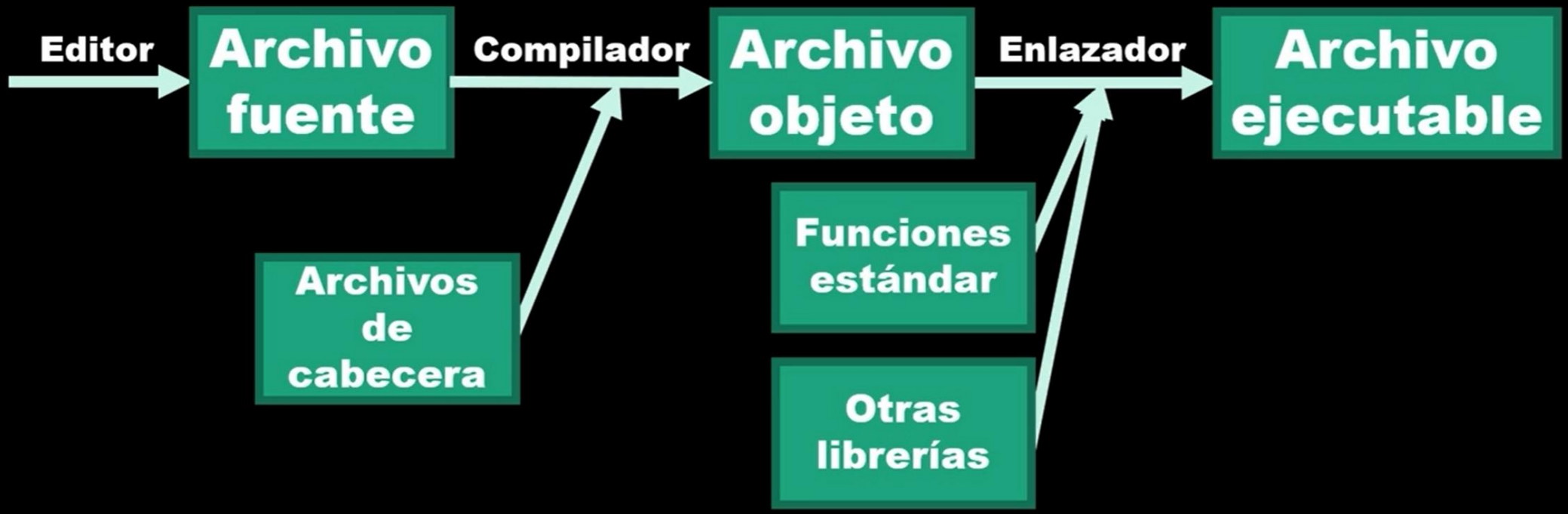
C++

Introducción

Características

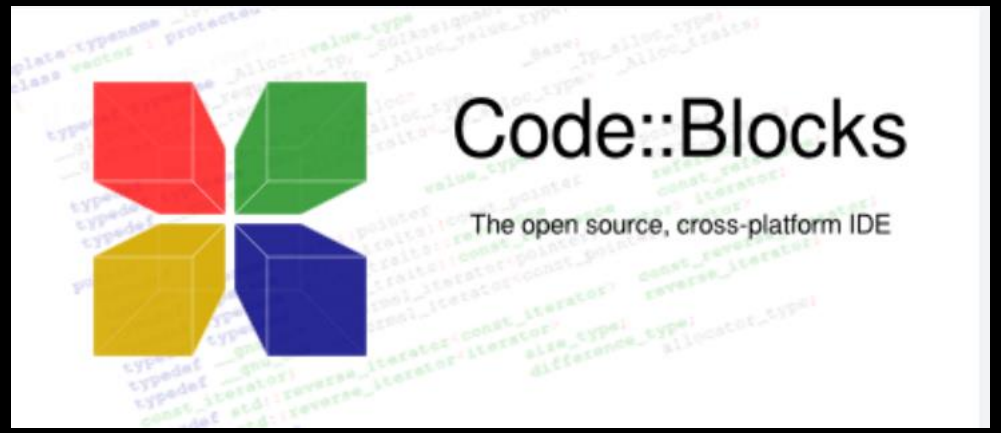
- Creado en 1980 como una extensión de C
- Es de alto rendimiento
- Actualizado y extendido
- Multiplataforma
- Bastante extendido
- Es orientado a objetos
- Aplicaciones y usos:
  - Bases de datos
  - Navegadores web
  - Sistemas operativos
  - Compiladores
  - Videojuegos

Desarrollo de un programa en C++



Entorno de desarrollo integrado

- **Objetivo**: Simplificar el desarrollo y evitarnos procesos repetitivos como compilar y enlazar.
- Provee un entorno para desarrollar y ejecutar un programa de manera más amigable.
- Utilizaremos Code::Blocks.



Instalar Code::Blocks

- https://www.codeblocks.org/downloads/
- Pulsamos en *Download the binary release*
- Elegimos una versión *mingw-setup* porque ya viene con el compilador y nos evita hacer instalaciones adicionales.
- Abrimos el ejecutable y mediante el asistente dejamos las opciones por defecto.
- Finalmente instalamos.
- Al abrirlo aceptamos las opciones para aceptar el compilador.

Variables

- Una variable es una ubicación en memoria donde guardamos información.
- Esa ubicación está asociada a un nombre único.
- El valor de la variable puede ir cambiando a lo largo del programa.
- Para declarar una variable, debemos indicar el tipo de variable, su nombre y asignarle un valor.
- Una variable debe declararse antes de usarse.
- Reglas en los nombres:
  - Puede conformarse de letras, números y _
  - El primer carácter debe ser una letra o _
  - No se pueden usar palabras reservadas. Por ejemplo, no podemos llamar a una variable *namespace* ni *int.*
  - C++ es *case sensitive.* No es lo mismo llamar a una variable *numero* o *Numero.*

Tipos de datos

- Textos
  - **char**: Son para almacenar un único carácter.
  - **string**: Son para almacenar una cadena de caracteres. Son más utilizados
- Números
  - Enteros
    - **short**: Almacena 2 bytes.
    - **long**: Almacena 4 bytes.
    - **Int**: Se adapta al que vamos utilizando
  - Decimales
    - **float**: Almacena 4 bytes. Reserva 6 dígitos para la parte decimal.
    - **double**: Almacena 8 bytes. Reserva 15 dígitos para la parte decimal.
    - **long double**: Almacena 10 bytes. Reserva 19 dígitos para la parte decimal.
  - Booleanos
    - **bool**: Almacena solo dos posibles valores: *true* o *false*.

Tabla de tipos de datos

Nombre	Descripción	Tamaño*	Rango de valores*
char	Carácter o entero pequeño	1byte	con signo: -128 to 127 sin signo: 0 a 255
short int (short)	Entero corto	2bytes	con signo: -32768 a 32767 sin signo: 0 a 65535
int	Entero	4bytes	con signo: -2147483648 a 2147483647 sin signo: 0 a 4294967295
long int (long)	Entero largo	8bytes	con signo: -2147483648 a 2147483647 sin signo: 0 a 4294967295
bool	Valor booleano. Puede tomar dos valores: verdadero o falso	1byte	true o false
float	Número de punto flotante	4bytes	3.4e +/- 38 (7 digitos)
double	De punto flotante de doble precisión	8bytes	1.7e +/- 308 (15 digitos)
long double	Long de punto flotante de doble precisión	8bytes	1.7e +/- 308 (15 digitos)

Constantes

- Las constantes son variables a las que no podemos modificar su valor inicial.
- Mediante la palabra reservada *const,* podemos declarar una constante.
- Por convención las constantes se declaran con su nombre en mayúsculas.

Funciones

- Una función es un bloque de código que se va a ejecutar cuando se invoque.
- Para invocar una función, simplemente hay que llamarla por su nombre.
- Una función *void* ejecuta algo y no retorna nada.
- Una función que tiene un tipo está obligada a retornar un valor de ese tipo.
- Una función a menudo necesitará valores de entrada para poder desempeñar su funcionalidad. Esos valores serán los parámetros.

Operadores aritméticos

Nombre del operador	Sintaxis
Suma	`a + b`
Suma y asignación	`a += b`
Resta	`a - b`
Resta y asignación	`a -= b`
Multiplicación	`a * b`
Multiplicación y asignación	`a *= b`
División	`a / b`
División y asignación	`a /= b`
Módulo	`a % b`
Módulo y asignación	`a %= b`
Más unario (promoción entera)	`+a`
Menos unario (opuesto)	`-a`
Incremento prefijo	`++a`
Incremento postfijo	`a++`
Decremento prefijo	`--a`
Decremento postfijo	`a--`

Operadores relacionales y lógicos

Nombre del operador	Sintaxis
Menor que	`a < b`
Mayor que	`a > b`
Menor o igual que	`a <= b`
Mayor que o igual que	`a >= b`
Igual que	`a == b`
Diferente que / No igual que	`a != b`

Nombre del operador	Sintaxis
Negación lógica (NOT)	`!a`
Y lógico (AND)	`a && b`
O lógico (OR)	`a || b`