

# SETUP ENTORNO PARA DESAROLLO DE PROGRAMAS EN C++

Hugo Pelayo

Asignatura de Programación

29 de abril de 2023

C++ es un lenguaje de programación muy flexible que ofrece soporte para la programación orientada a objeto, entre otros paradigmas. Es un lenguaje compilado y como tal el código fuente debe ser convertido previamente a código máquina para ser ejecutado por nuestro CPU. A continuación, se muestran varios pasos para compilar un programa en C++ en Linux utilizando el compilador de código abierto G++.



```
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios > sudo apt install g++
[sudo] password for kate:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
g++ is already the newest version (4:11.2.0-1ubuntu1).
The following packages were automatically installed and are no longer required:
  linux-headers-5.19.0-38-generic linux-hwe-5.19.0-headers-5.19.0-38 linux-image-5.19.0-38-generic linux-modules-5.19.0-38-generic
  linux-modules-extra-5.19.0-38-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios > g++ --version
g++ (Ubuntu 12.1.0-2ubuntu1~22.04) 12.1.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios >
```

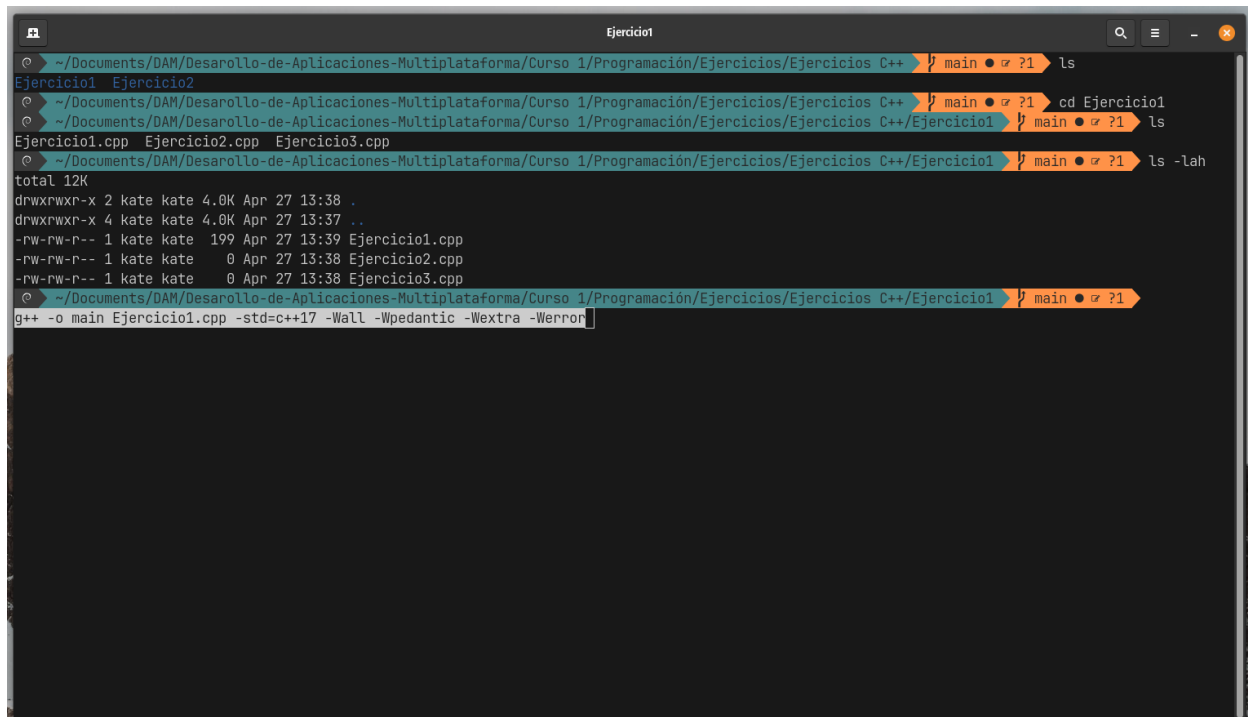
Primero instalamos el compilador en caso de que no lo tengamos en nuestro equipo. Para ello utilizamos los siguientes comandos (se asume que el lector tiene instalada una distribución de Linux basada en Ubuntu con el gestor de paquetes apt):

```
sudo apt install g++

g++ --version
```

El primer comando nos permite instalar g++, una vez instalado el segundo comando nos permitirá comprobar la versión instalada. Por lo general las distribuciones basadas en Ubuntu como Pop\_OS! tienen ya g++ instalado por defecto por lo cual el segundo comando no debería falla y nos debería salir que el comando o programa ya está instalado, pudiendo mostrar con el segundo comando la versión instalada.

Ahora ya lo tendremos instalado en nuestro sistema y podremos compilar y ejecutar nuestras aplicaciones en C++.



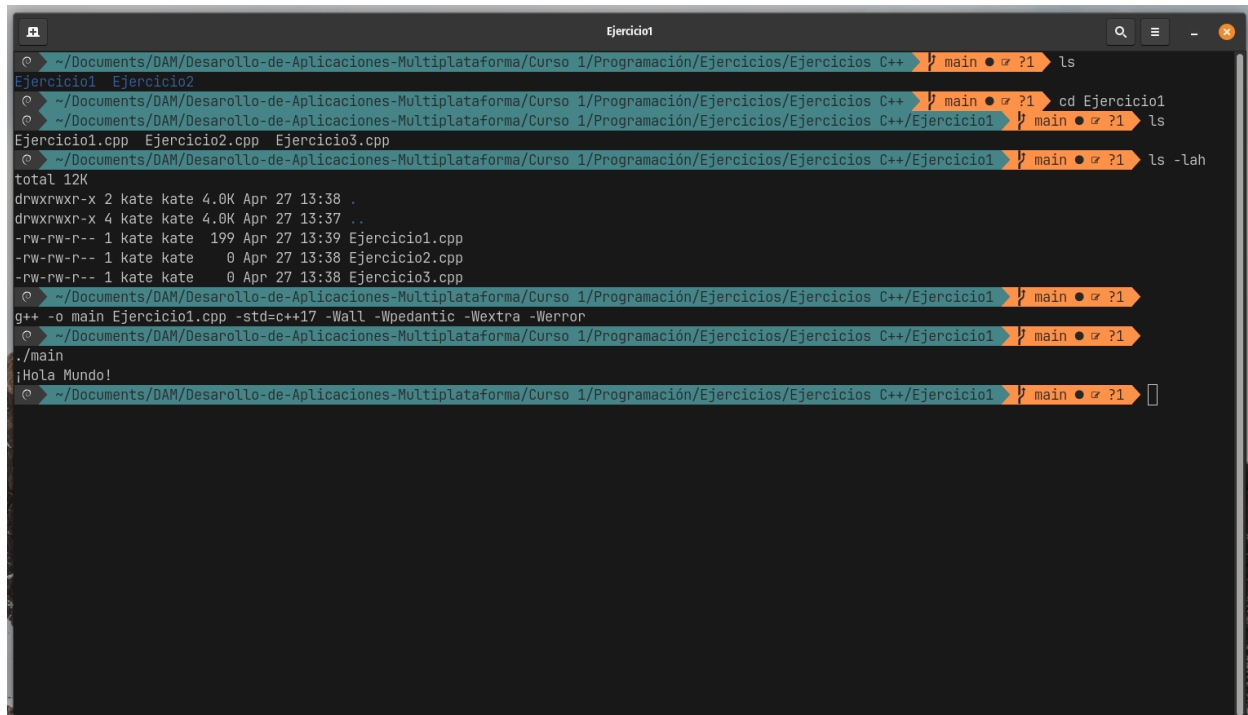
```
Ejercicio1
~/Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++ main ● ?1 ls
Ejercicio1 Ejercicio2
~/Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++ main ● ?1 cd Ejercicio1
~/Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 main ● ?1 ls
Ejercicio1.cpp Ejercicio2.cpp Ejercicio3.cpp
~/Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 main ● ?1 ls -lah
total 12K
drwxrwxr-x 2 Kate kate 4.0K Apr 27 13:38 .
drwxrwxr-x 4 Kate kate 4.0K Apr 27 13:37 ..
-rw-rw-r-- 1 Kate kate 199 Apr 27 13:39 Ejercicio1.cpp
-rw-rw-r-- 1 Kate kate 0 Apr 27 13:38 Ejercicio2.cpp
-rw-rw-r-- 1 Kate kate 0 Apr 27 13:38 Ejercicio3.cpp
~/Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 main ● ?1
g++ -o main Ejercicio1.cpp -std=c++17 -Wall -Wpedantic -Wextra -Werror
```

Nos dirigimos a la carpeta de nuestro programa. Para compilarlo nos bastaría con ejecutar el comando:

```
g++ -o main nombre_programa
```

En este comando el flag `-o` indica que queremos darle un nombre al ejecutable, si no se otorga por defecto el nombre `a.out` a nuestro ejecutable.

Ahora podemos ejecutar nuestro programa:



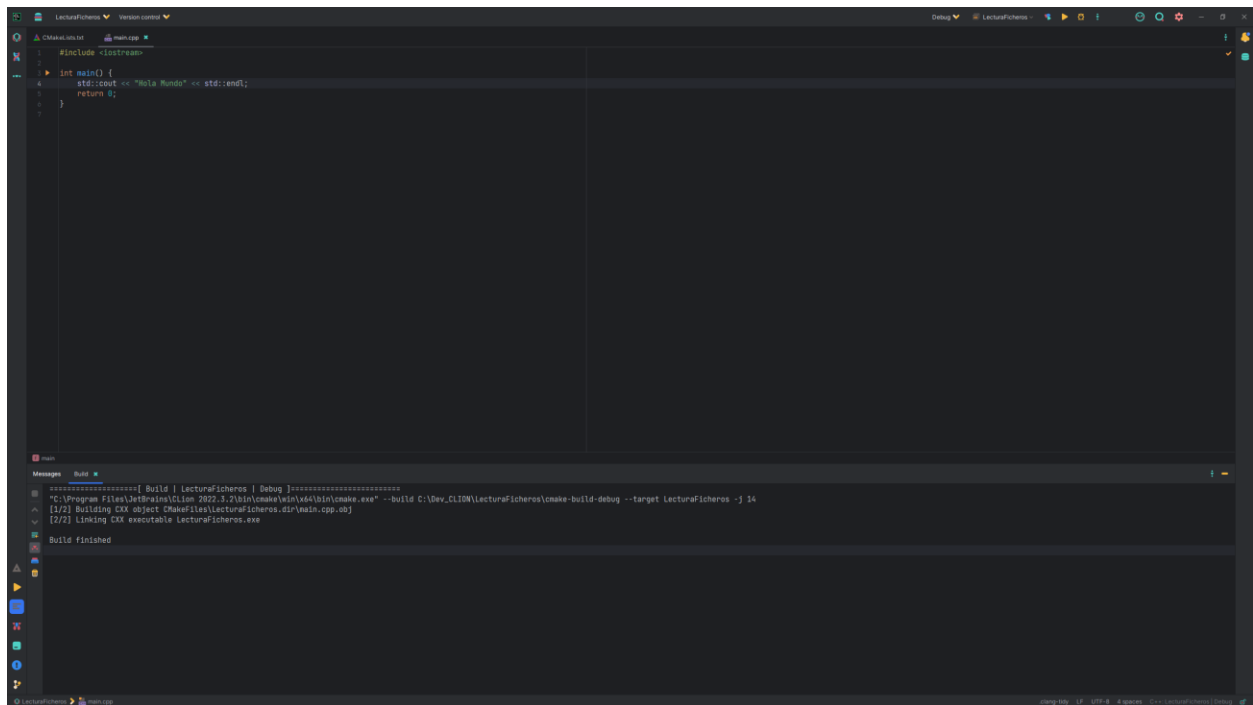
```
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++ | main ● ?1 | ls
Ejercicio1 Ejercicio2
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++ | main ● ?1 | cd Ejercicio1
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 | main ● ?1 | ls
Ejercicio1.cpp Ejercicio2.cpp Ejercicio3.cpp
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 | main ● ?1 | ls -lah
total 12K
drwxrwxr-x 2 kate kate 4.0K Apr 27 13:38 .
drwxrwxr-x 4 kate kate 4.0K Apr 27 13:37 ..
-rw-rw-r-- 1 kate kate 199 Apr 27 13:39 Ejercicio1.cpp
-rw-rw-r-- 1 kate kate 0 Apr 27 13:38 Ejercicio2.cpp
-rw-rw-r-- 1 kate kate 0 Apr 27 13:38 Ejercicio3.cpp
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 | main ● ?1 |
g++ -o main Ejercicio1.cpp -std=c++17 -Wall -Wpedantic -Wextra -Werror
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 | main ● ?1 |
./main
¡Hola Mundo!
~/.Documents/DAM/Desarrollo-de-Aplicaciones-Multiplataforma/Curso 1/Programación/Ejercicios/Ejercicios C++/Ejercicio1 | main ● ?1 |
```

Para configurar un entorno de desarrollo en Windows, se recomienda instalar el IDE Visual Studio Community. El proceso de instalación es más sencillo que configurar un entorno con VSCode. Para estos procesos [VSCode](#) y [VSCommunity](#) ofrecen tutoriales en sus respectivas páginas oficiales.

Otros métodos de compilar programas en C++, más recomendables, incluyen la compilación utilizando “build systems” que son programas que nos automatizan el proceso de compilación, destacan entre ellos:

- CMake
- Makefile (generalmente para sistemas Linux)
- Premake

En entornos de desarrollo como CLion, el build system por defecto es CMake.



Pequeño ejemplo utilizando CMake para compilar un programa en C++.

Unas de las ventajas de estos programa es que abstraen al programador del compilador por defecto de la plataforma sobre la cual se intenta compilar nuestro programa, por ejemplo, g++ es exclusivo de sistemas Linux y MSVC para Windows, a parte de otros aspectos como permitirnos indicar rutas para archivos que se tengan que incluir mediante la directiva `#include` o librerías que se tengan que enlazar a nuestro ejecutable. Todo este proceso es tedioso y largo de hacer manualmente.