Documento creado por EMILIO RIVAS

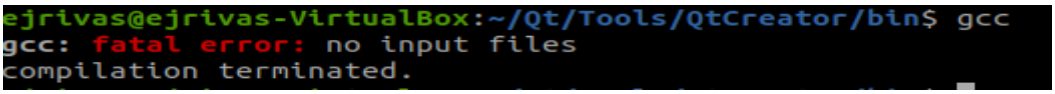
1. Problemas de configuración en Linux Virtualización VirtualBox Ubuntu 20.04.1 LTS:

1. Requisitos previos a la instalación:

Antes de instalar comprobamos que tenemos los compiladores, el make y el debugger instalado, además de comprobar las librerías de GL y los includes.

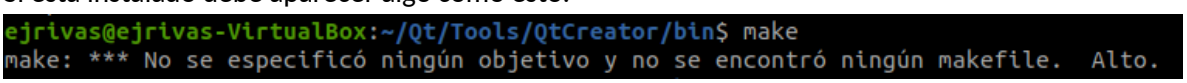
1. Instalar los compiladores de C++ si no se tienen, es decir si se hace gcc o g++ en un terminal y no encuentra ese comando
   Para instalar:
   sudo apt install g++

   Si está instalado debe aparecer algo como esto:



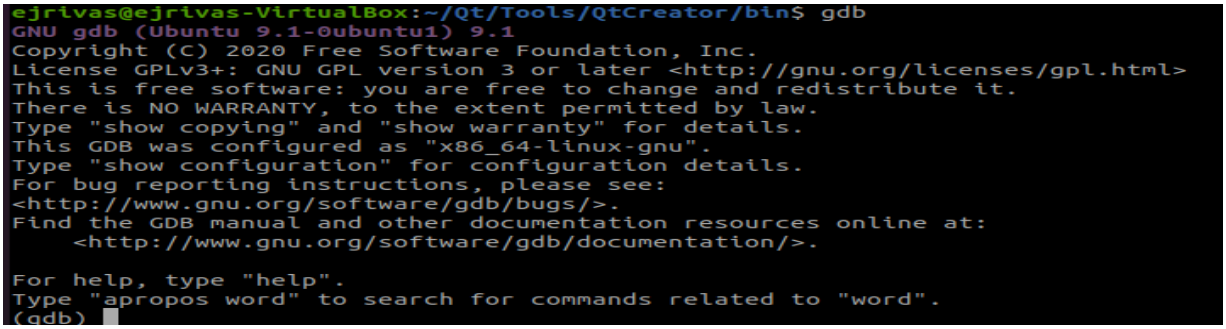
2. Instalar make si no se encuentra instalado, si no se encuentra el comando make
   Para instalar:
   sudo apt-get install make

   Si está instalado debe aparecer algo como esto:

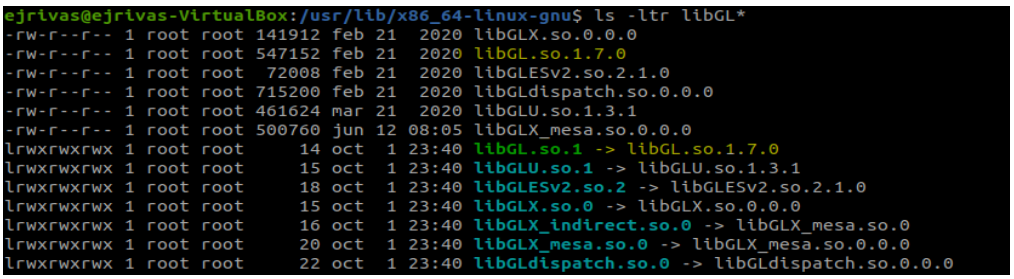


3. Instalar el debugger si no se tiene, es decir, si no se obtiene nada al ejecutar gdb

Para instalar:
sudo apt-get install gdb
Si está instalado debe aparecer algo como esto:



4. Comprobar que tenemos la 'libGL.so' exactamente con este nombre y si no la tenemos hay que crear un enlace simbólico.
   Comprobamos : en /usr/lib/x86_64-linux-gnu/
   ls -ltr libGL*

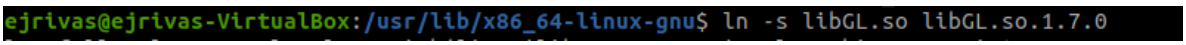


Como podemos ver en la pantalla de antes no aparece la librería con el nombre exacto “libGL.so” , sino que aparecen librerías con libGL.so .XXXX pero con números de versiones tras el ”.so”.

Para que nos aparezca la librería 'libGL.so' nos creamos un enlace simbólico a la versión de la librería física que tenemos en mi caso es la libGL.so.1.7.0, en amarillo en la foto anterior, pero otros podréis tener una versión diferente, haría igual habría que cambiar solo el nombre.

sudo ln –s libGL.so libGL.so.XXXX (donde XXXX es el valor de tu versión)

en mi caso sería: (usa “sudo” delante de “ln” para hacerlo como root porque sino no te deja),



Y comprobamos de nuevo:

```
ejrivas@ejrivas-VirtualBox:/usr/lib/x86_64-linux-gnu$ ls -ltr libGL*
-rw-r--r-- 1 root root 141912 feb 21  2020 libGLX.so.0.0.0
-rw-r--r-- 1 root root 547152 feb 21  2020 libGL.so.1.7.0
-rw-r--r-- 1 root root  72008 feb 21  2020 libGLESv2.so.2.1.0
-rw-r--r-- 1 root root 715200 feb 21  2020 libGLdispatch.so.0.0.0
-rw-r--r-- 1 root root 461624 mar 21  2020 libGLU.so.1.3.1
-rw-r--r-- 1 root root 500760 jun 12 08:05 libGLX_mesa.so.0.0.0
lrwxrwxrwx 1 root root     14 oct  1 23:40 libGL.so.1 -> libGL.so.1.7.0
lrwxrwxrwx 1 root root     15 oct  1 23:40 libGLU.so.1 -> libGLU.so.1.3.1
lrwxrwxrwx 1 root root     18 oct  1 23:40 libGLESv2.so.2 -> libGLESv2.so.2.1.0
lrwxrwxrwx 1 root root     15 oct  1 23:40 libGLX.so.0 -> libGLX.so.0.0.0
lrwxrwxrwx 1 root root     16 oct  1 23:40 libGLX_indirect.so.0 -> libGLX_mesa.so.0
lrwxrwxrwx 1 root root     20 oct  1 23:40 libGLX_mesa.so.0 -> libGLX_mesa.so.0.0.0
lrwxrwxrwx 1 root root     22 oct  1 23:40 libGLdispatch.so.0 -> libGLdispatch.so.0.0.0
lrwxrwxrwx 1 root root     14 oct  2 19:41 libGL.so -> libGL.so.1.7.0
```

Ahora si vemos la librería con el nombre correcto 'libGL.so' que es un enlace simbólico a la versión de la librería de openGL que tengo en mi sistema, en mi caso como he comentado la libGL.so.1.7.0

5. Copiar en el directorio de Include los includes proporcionados por el profesor en el caso de no tenerlos.
   Comprobamos en /usr/include
   Si no tenemos el directorio GL como aparece abajo, copiamos el directorio entero que tenemos adjunto en la practica 1

```
ejrivas@ejrivas-VirtualBox:/usr/include$ ls -d *
aio.h          c++         envz.h      fnmatch.h  gnumake.h
aliases.h      complex.h   err.h       fstab.h    gnu-versio
alloca.h       cpio.h      errno.h     fts.h      grp.h
argp.h         crypt.h     error.h     ftw.h      gshadow.h
argz.h         ctype.h     execinfo.h  gcalc-2    iconv.h
ar.h           dirent.h    fcntl.h     gci-2      ifaddrs.h
arpa           dlfcn.h     features.h  gconv.h    inttypes.h
asm-generic    drm         fenv.h      getopt.h   iproute2
assert.h       elf.h       finclude    GL         langinfo.h
byteswap.h     endian.h    fmtmsg.h    glob.h     lastlog.h
ejrivas@ejrivas-VirtualBox:/usr/include$
```

   Además tenemos que darle permisos dentro del directorio GL
   chmod 777 *

```
ejrivas@ejrivas-VirtualBox:/usr/include$ ls -ltr GL
total 2900
-rwxrwxrwx 1 root root  11246 jul 16  2012 glxtokens.h
-rwxrwxrwx 1 root root  80979 jul 16  2012 glxproto.h
-rwxrwxrwx 1 root root   2086 jul 16  2012 glxmd.h
-rwxrwxrwx 1 root root   4706 jul 16  2012 glxint.h
-rwxrwxrwx 1 root root 656589 ago 24  2012 glext.h.old
-rwxrwxrwx 1 root root   4468 jul  5  2013 wmesa.h
-rwxrwxrwx 1 root root  41377 jul  5  2013 wglext.h
-rwxrwxrwx 1 root root  51274 jul  5  2013 vms_x_fix.h
-rwxrwxrwx 1 root root   8497 jul  5  2013 osmesa.h
-rwxrwxrwx 1 root root   3463 jul  5  2013 glx_mangle.h
-rwxrwxrwx 1 root root  17170 jul  5  2013 glx.h
-rwxrwxrwx 1 root root  43887 jul  5  2013 glxext.h
-rwxrwxrwx 1 root root   3315 jul  5  2013 glu_mangle.h
-rwxrwxrwx 1 root root  17255 jul  5  2013 glu.h
-rwxrwxrwx 1 root root 128943 jul  5  2013 gl_mangle.h
-rwxrwxrwx 1 root root  84468 jul  5  2013 gl.h
-rwxrwxrwx 1 root root 656589 jul  5  2013 glext.h
-rwxrwxrwx 1 root root    639 feb 27  2014 glut.h
-rwxrwxrwx 1 root root  26523 feb 27  2014 freeglut_std.h
-rwxrwxrwx 1 root root    681 feb 27  2014 freeglut.h
-rwxrwxrwx 1 root root   9007 feb 27  2014 freeglut_ext.h
-rwxrwxrwx 1 root root  62206 feb 27  2014 wglew.h
-rwxrwxrwx 1 root root  68423 feb 27  2014 glxew.h
-rwxrwxrwx 1 root root 925388 feb 27  2014 glew.h
drwxrwx--- 2 root root   4096 oct  1 01:31 internal
```

2. Ahora instalamos QT siguiendo los pasos que comenta el profesor en la guía de instalación en Unix,

Pero si al pulsar sobre el icono de QT creator, no hace nada…

es decir se hace doble clic y doble clic y parece que no se ejecuta nada, tenemos que ver si hay algún problema con la carga de librerías dinámicas.

Hay que abrir un terminal y posicionarnos donde está el binario

```
ejrivas@ejrivas-VirtualBox:~/Qt/Tools/QtCreator/bin$ pwd
/home/ejrivas/Qt/Tools/QtCreator/bin
```

Para ver exactamente la ejecución y que nos dé más detalle del posible problema de la carga de librerías, lanzamos a mano la ejecución, vemos que obtenemos un Core, por eso no se está ejecutando y si pulsamos el icono no nos hace nada:

```
ejrivas@ejrivas-VirtualBox:~/Qt/Tools/QtCreator/bin$ ./qtcreator
```

qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.

This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb.

Aborted (core dumped)

Para obtener más información lo que hacemos es exportar en ese terminal que QT se arranque en modo debug

Ejecutamos la sentencia

export QT_DEBUG_PLUGINS=1

y volvemos a ejecutar el binario a mano en vez de con el icono.

Ahora nos da más detalles :

Got keys from plugin meta data ("xcb")

QFactoryLoader::QFactoryLoader() checking directory path "/home/ejrivas/Qt/Tools/QtCreator/bin/platforms" ...

Cannot load library /home/ejrivas /Qt/Tools/QtCreator/lib/Qt/plugins/platforms/libqxcb.so: (libxcb-xinerama.so.0: cannot open shared object file: No such file or directory)

QLibraryPrivate::loadPlugin failed on "/home/ejrivas/Qt/Tools/QtCreator/lib/Qt/plugins/platforms/libqxcb.so" : "Cannot load library /home/ejrivas /Qt/Tools/QtCreator/lib/Qt/plugins/platforms/libqxcb.so: (libxcb-xinerama.so.0: cannot open shared object file: No such file or directory)"

qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.

This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.

SOLUCIÓN: Buscando por internet en foros he visto una solución instalar la librería que no se está encontrando, con este comando:

sudo apt-get install libxcb-xinerama0

Ahora si hacemos un ldd sobre el binario vemos que no tenemos ningún error en la carga de librerías, no tengo pantallazo pero antes de hacer el **sudo apt-get install libxcb-xinerama0, en el link en amarillo me aparecía un file not found.**

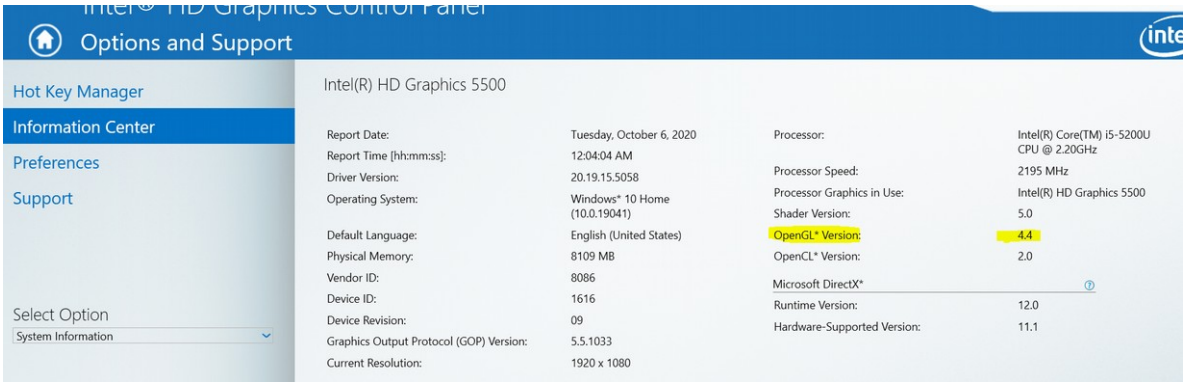
Y ya si lo lanzamos a mano o con el icono se cargara el programa correctamente, al menos a mi así fue como lo solucioné.

```
ejrivas@ejrivas-VirtualBox:~/Qt/Tools/QtCreator/bin$ ldd qtcreator
        linux-vdso.so.1 (0x00007ffd55def000)
        libExtensionSystem.so.4 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/qtcreator/libExtensionSystem.so.4 (0x00007fd1d07b6000)
        libAggregation.so.4 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/qtcreator/libAggregation.so.4 (0x00007fd1d05b0000)
        libUtils.so.4 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/qtcreator/libUtils.so.4 (0x00007fd1d010c000)
        libQt5Widgets.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libQt5Widgets.so.5 (0x00007fd1cf8aa000)
        libQt5Gui.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libQt5Gui.so.5 (0x00007fd1cef79000)
        libQt5Concurrent.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libQt5Concurrent.so.5 (0x00007fd1ced70000)
        libQt5Network.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libQt5Network.so.5 (0x00007fd1ce9c3000)
        libQt5Core.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libQt5Core.so.5 (0x00007fd1ce1cd000)
        libGL.so.1 => /lib/x86_64-linux-gnu/libGL.so.1 (0x00007fd1ce133000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fd1ce110000)
        libstdc++.so.6 => /lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007fd1cdf2f000)
        libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fd1cddde000)
        libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007fd1cddc3000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fd1cdbd1000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fd1cdbcb000)
        libQt5Qml.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/qtcreator/../Qt/lib/libQt5Qml.so.5 (0x00007fd1cd54e000)
        libQt5Xml.so.5 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/qtcreator/../Qt/lib/libQt5Xml.so.5 (0x00007fd1cd30e000)
        libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1 (0x00007fd1cd2f2000)
        libgssapi_krb5.so.2 => /lib/x86_64-linux-gnu/libgssapi_krb5.so.2 (0x00007fd1cd2a5000)
        libicui18n.so.56 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libicui18n.so.56 (0x00007fd1cce0c000)
        libicuuc.so.56 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libicuuc.so.56 (0x00007fd1cca54000)
        libicudata.so.56 => /home/ejrivas/Qt/Tools/QtCreator/bin/./../lib/Qt/lib/libicudata.so.56 (0x00007fd1cb06f000)
        libgthread-2.0.so.0 => /lib/x86_64-linux-gnu/libgthread-2.0.so.0 (0x00007fd1cb06a000)
        libglib-2.0.so.0 => /lib/x86_64-linux-gnu/libglib-2.0.so.0 (0x00007fd1caf41000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fd1d09fd000)
        libGLdispatch.so.0 => /lib/x86_64-linux-gnu/libGLdispatch.so.0 (0x00007fd1cae89000)
        libGLX.so.0 => /lib/x86_64-linux-gnu/libGLX.so.0 (0x00007fd1cae55000)
        librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007fd1cae48000)
        libkrb5.so.3 => /lib/x86_64-linux-gnu/libkrb5.so.3 (0x00007fd1cad6b000)
        libk5crypto.so.3 => /lib/x86_64-linux-gnu/libk5crypto.so.3 (0x00007fd1cad3a000)
        libcom_err.so.2 => /lib/x86_64-linux-gnu/libcom_err.so.2 (0x00007fd1cad33000)
        libkrb5support.so.0 => /lib/x86_64-linux-gnu/libkrb5support.so.0 (0x00007fd1cad24000)
        libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007fd1cacaf000)
        libX11.so.6 => /lib/x86_64-linux-gnu/libX11.so.6 (0x00007fd1cab72000)
        libkeyutils.so.1 => /lib/x86_64-linux-gnu/libkeyutils.so.1 (0x00007fd1cab6b000)
        libresolv.so.2 => /lib/x86_64-linux-gnu/libresolv.so.2 (0x00007fd1cab4f000)
        libxcb.so.1 => /lib/x86_64-linux-gnu/libxcb.so.1 (0x00007fd1cab25000)
        libXau.so.6 => /lib/x86_64-linux-gnu/libXau.so.6 (0x00007fd1cab1d000)
        libXdmcp.so.6 => /lib/x86_64-linux-gnu/libXdmcp.so.6 (0x00007fd1cab15000)
        libbsd.so.0 => /lib/x86_64-linux-gnu/libbsd.so.0 (0x00007fd1caafb000)
ejrivas@ejrivas-VirtualBox:~/Qt/Tools/QtCreator/bin$
```

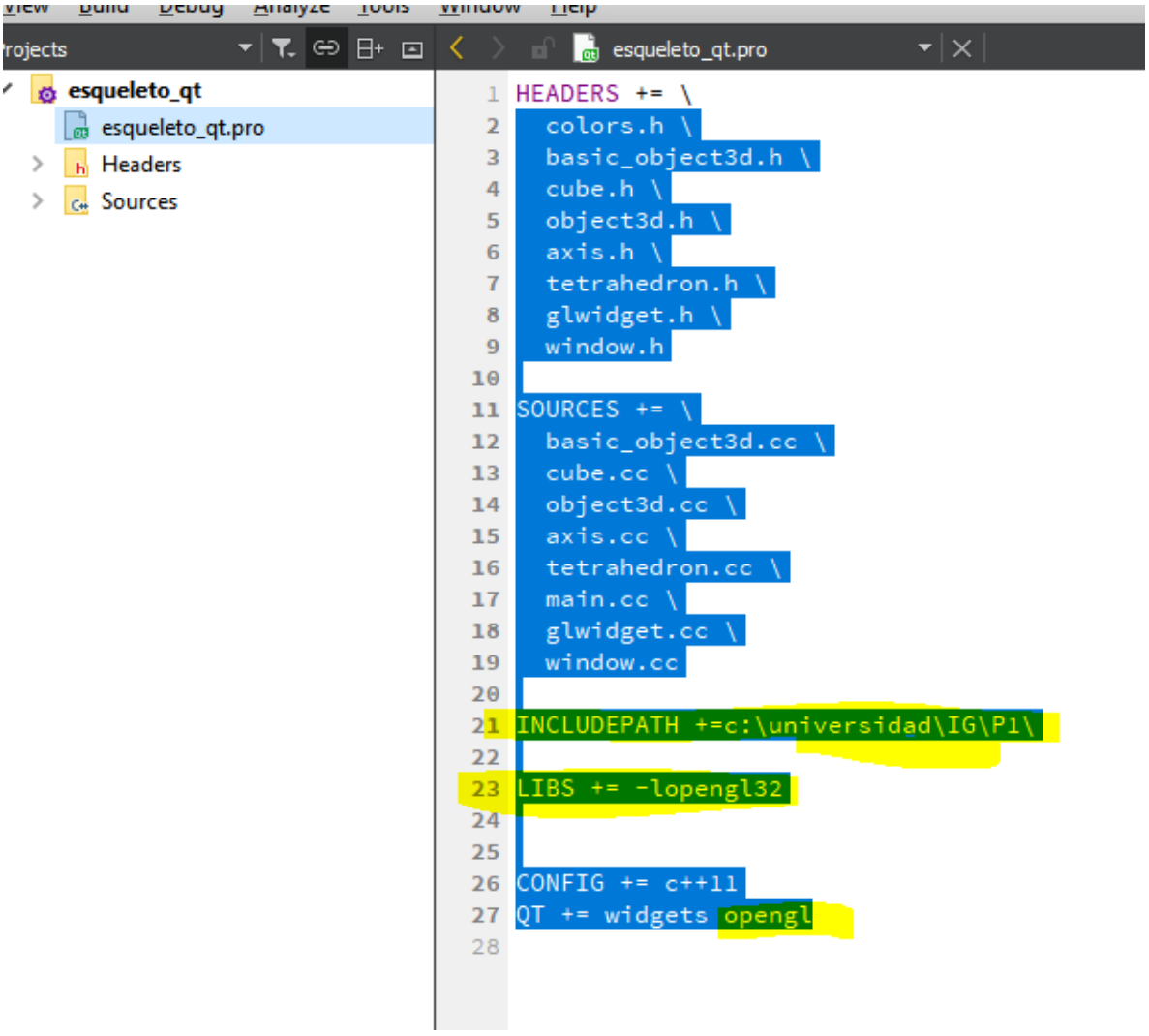
2. Problemas de configuración en windows 10

1. Si no tenemos compilador instalado lo recompensable es descargarse y instalarse visual studio y en el proceso de instalación seleccionare que instale todo lo relacionado con c++,para que genere un entorno de compilación en nuestro Windows.

2. Comprobar que tenemos open GL instalado viendo la información de la tarjeta grafica. CONTROL + ALT +F12



3. Instalarse el QT según la guía del profesor.
4. Dejar en un directorio específico el directorio GL que se proporciona en la P1 con los includes de open GL.
5. Modificar el esqueleto_qt.pro proporcionado por el profesor para que detecte dónde se encuentra la librería de QT y los includes, ya que es diferente que en Linux.
   A mí me funcionó con estas modificaciones. Pongo en amarillo lo que sería diferente.
   En mi caso el directorio con los includes lo tengo en c:\universidad\IG\P1\GL\
   De ahí que INCLUDEPATH +=c:\universidad\IG\P1\



Documento creado por MIGUEL FERNÁNDEZ

Sistema Operativo: MacOSX

-No se puede instalar QT a partir de la versión 5 si no está el sistema operativo actualizado como mínimo a macOS Mojave 10.14.6.

-Hay que actualizar Xcode a la versión más reciente del SO utilizado, ya que es el compilador nativo de mac y contiene las herramientas de compilación.

-Comprobar en la terminal de mac que con la actualización de Xcode hemos actualizado e instalado bien:

cmake --version
clang++ --version

***Si necesitamos cmake —> https://cmake.org/download/**

-Descargar la librería GLFW:
http://www.glfw.org/download.html

-Compilarla e instalarla en la terminal:

cd glfw-....
cmake -DBUILD_SHARED_LIBS=ON .
make
sudo make install

-Instalar solamente la herramienta de QT desktop 5.12.9 clang 64 bit, para evitar problemas.

-En el archivo "esqueleto_qt.pro", dos opciones:

1. Comentar el LIBS:

**LIBS += **
#-L/usr/X11R6/lib64 -lGL

y entonces copiar la carpeta GL de prado dentro del proyecto.

2. Buscar la librería en /usr/X11R6/.

***aconsejable #DEFINE para opciones de sistemas operativos.**

-Errores de compilación al enlazar .o —> click derecho sobre el proyecto y pulsar clean (es como un make clean para borrar los objetos que había creados), y después volver a compilar.