

Instalar OpenCV en Ubuntu

1. Instalar paquetes necesarios

Los paquetes que se van a necesitar son:

- GCC 4.4.x or later
- CMake 2.6 or higher
- Git
- GTK+2.x or higher, including headers (libgtk2.0-dev)
- pkg-config
- Python 2.6 or later and Numpy 1.5 or later with developer packages (python-dev, python-numpy)
- ffmpeg or libav development packages: libavcodec-dev, libavformat-dev, libswscale-dev
- [optional] libtbb2 libtbb-dev
- [optional] libdc1394 2.x
- [optional] libjpeg-dev, libpng-dev, libtiff-dev, libjasper-dev, libdc1394-22-dev

Para instalar los paquetes meter los siguientes comandos en la terminal o utilizar Synaptic

- Compilador

```
sudo apt-get install build-essential
```
- Requerido

```
sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
```
- Opcional

```
sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev
```

2. Descargar OpenCV

2.1. Descargar la última versión estable

Descargar la última versión desde [sourceforge](https://sourceforge.net/projects/opencvlibrary/) y descomprimirlo

2.2. Obtener la última versión desde el repositorio Git

Clonar la versión del repositorio mediante estos comandos

```
cd ~/<directorio de trabajo>  
  
git clone https://github.com/Itseez/opencv.git
```

3. Compilar OpenCV usando Cmake

- 1) Crear un directorio temporal, donde se va a guardar el Makefile generado, archivos de proyecto, los archivos de objetos binarios y de salida.

```
cd opencv
mkdir release
cd release
```

2) Escribir

```
sudo cmake -D CMAKE_BUILD_TYPE=RELEASE -D
CMAKE_INSTALL_PREFIX=/usr/local ..
```

3) Compilar e instalar

```
sudo make
sudo make install
```

4. Descargar los últimos paquetes necesarios

Abrir Synaptic, buscar OpenCV, marcar todos e instalarlos

5. Comprobar la instalación utilizando CMake

5.1. Crear un programa usando OpenCV

Copiar el siguiente código en un archivo llamado DisplayImage.cpp

```
#include <stdio.h>
#include <opencv2/opencv.hpp>

using namespace cv;

int main(int argc, char** argv )
{
    if ( argc != 2 )
    {
        printf("usage: DisplayImage.out <Image_Path>\n");
        return -1;
    }

    Mat image;
    image = imread( argv[1], 1 );

    if ( !image.data )
    {
        printf("No image data \n");
        return -1;
    }
    namedWindow("Display Image", WINDOW_AUTOSIZE );
    imshow("Display Image", image);

    waitKey(0);
```

```
    return 0;
}
```

5.2. Crear el archivo CMake

Crear un archivo llamado CmakeLists.txt con el siguiente texto

```
cmake_minimum_required(VERSION 2.8)
project( DisplayImage )
find_package( OpenCV REQUIRED )
add_executable( DisplayImage DisplayImage.cpp )
target_link_libraries( DisplayImage ${OpenCV_LIBS} )
```

5.3. General el ejecutable

Generar un ejecutable usando Cmake mediante los siguientes comandos

```
cd <Directorio de trabajo>
sudo cmake .
sudo make
```

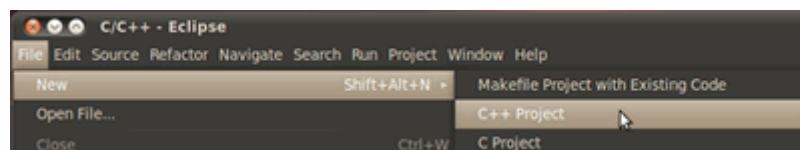
Para ejecutar el binario creado escribir

```
./DisplayImage <Imagen>.jpg
```

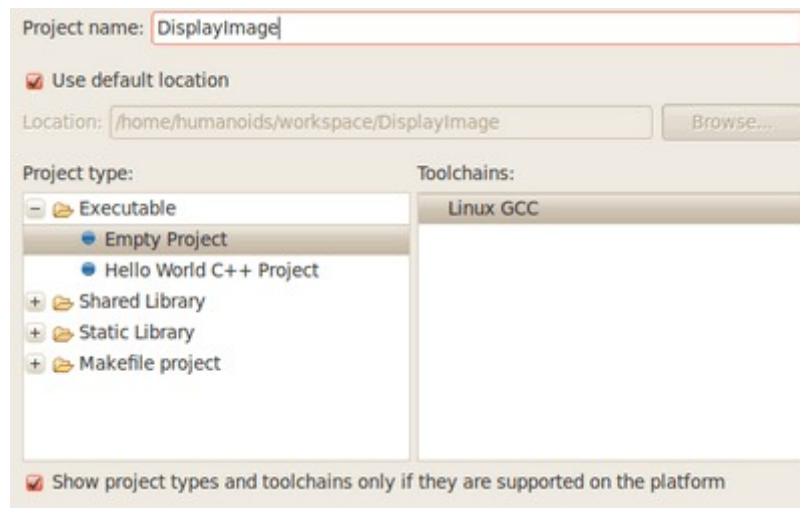
6. Comprobar la instalación utilizando Eclipse

6.1. Generar el proyecto

- 1) Abrir Eclipse
- 2) Crear un nuevo proyecto mediante **File → New → C/C++ Project**

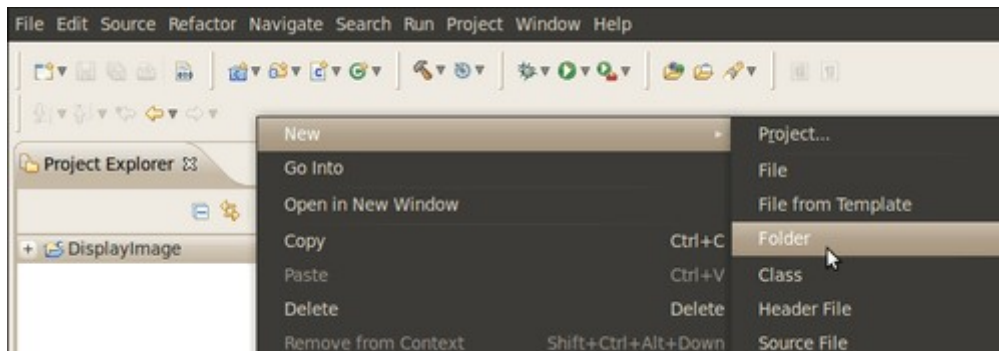


- 3) Elegir un nombre para el proyecto (por ejemplo DisplayImage), el tipo de proyecto elegirlo **Empty Project** y como compilador **Linux GCC**.



4) Dejar el resto por defecto y finalizar

5) Crear una nueva carpeta en el proyecto mediante **New → Folder** y llamar a la carpeta **src**.



6) Crear en la

carpeta un archivo fuente mediante **New → Source file** y llamarlo **DisplayImage.cpp**.



7) Copiar el siguiente código en el archivo

```

#include <cv.h>
#include <highgui.h>
#include <opencv2/objdetect/objdetect.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include "opencv2/core/core.hpp"
#include "opencv2/objdetect/objdetect.hpp"
using namespace cv;

int main( int argc, char** argv )
{
    Mat image;
    image = imread( argv[1], 1 );

```

```

if( argc != 2 || !image.data )
{
    printf( "No image data \n" );
    return -1;
}

namedWindow( "Display Image", CV_WINDOW_AUTOSIZE );
imshow( "Display Image", image );

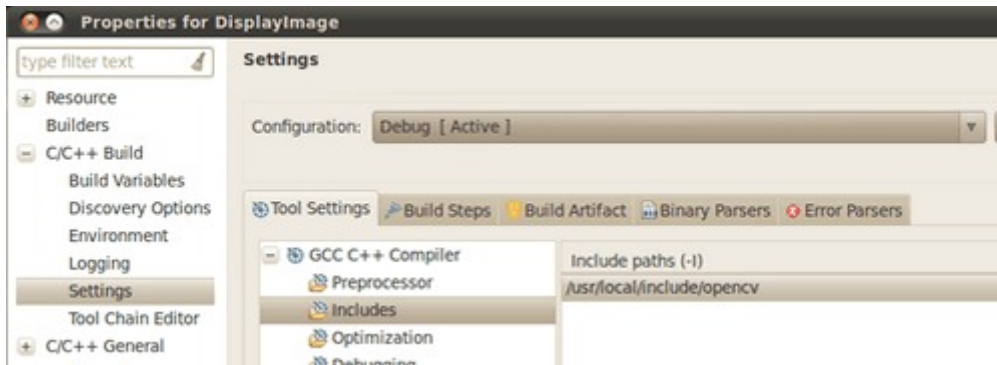
waitKey(0);

return 0;
}

```

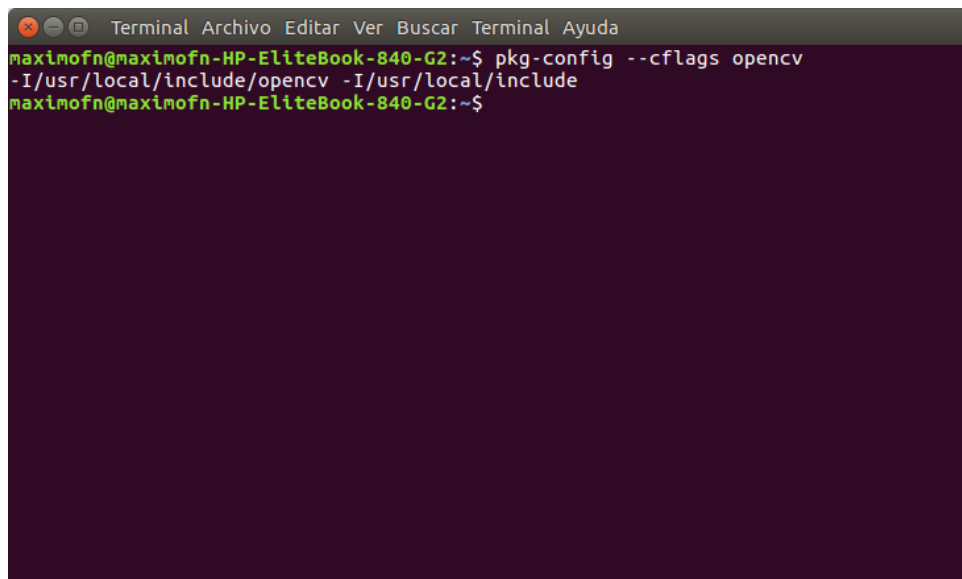
8) Configurar el proyecto

- Abrir las propiedades del proyecto mediante **Project → Properties**.
- Ir a **C/C++ Build → Settings**, ir a la pestaña **Tool Settings** y a continuación ir a **GCC C++ Compiler → Includes** y añadir la ruta dónde se ha instalado OpenCV en **Include paths (-I)**.



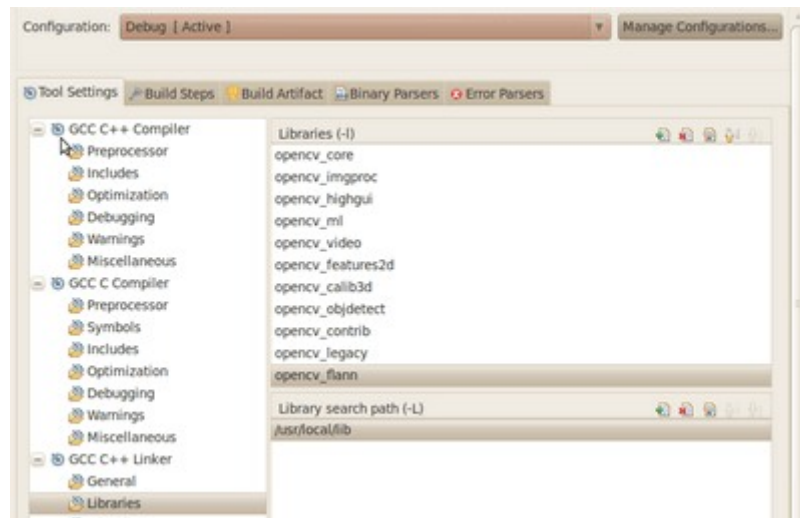
Para conocer la ruta dónde está instalado OpenCV ejecutar en una terminal

```
pkg-config --cflags opencv
```



En este caso la ruta es '/usr/local/include/opencv'

- c) A continuación **GCC C++ Linker** → **Libraries** y añadir las librerías de OpenCV en **Libraries (-l)** y la ruta de las librerías en **Library search path (-L)**.



Para saber las librerías y su ruta escribir en una terminal

```
pkg-config --libs opencv
```

```
Terminal Archivo Editar Ver Buscar Terminal Ayuda
maximofn@maximofn-HP-EliteBook-840-G2:~$ pkg-config --cflags opencv
-I/usr/local/include/opencv -I/usr/local/include
maximofn@maximofn-HP-EliteBook-840-G2:~$ pkg-config --libs opencv
-L/usr/local/lib -lopencv_shape -lopencv_stitching -lopencv_objdetect -lopencv_s
uperres -lopencv_videostab -lopencv_calib3d -lopencv_features2d -lopencv_highgui
-lopencv_videoio -lopencv_imgcodecs -lopencv_video -lopencv_photo -lopencv_ml -
lopencv_imgproc -lopencv_flann -lopencv_core
maximofn@maximofn-HP-EliteBook-840-G2:~$
```

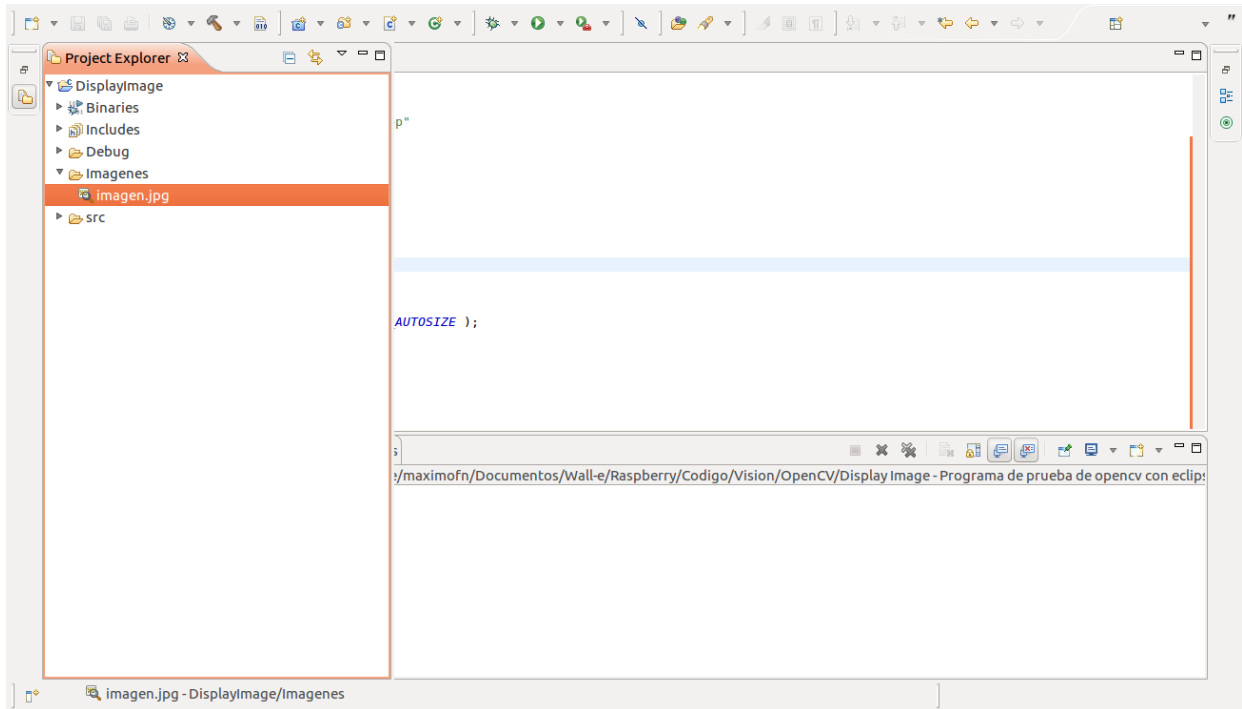
En este caso la ruta es '/usr/local/lib' y las librerías el resto

6.2. Compilar el programa

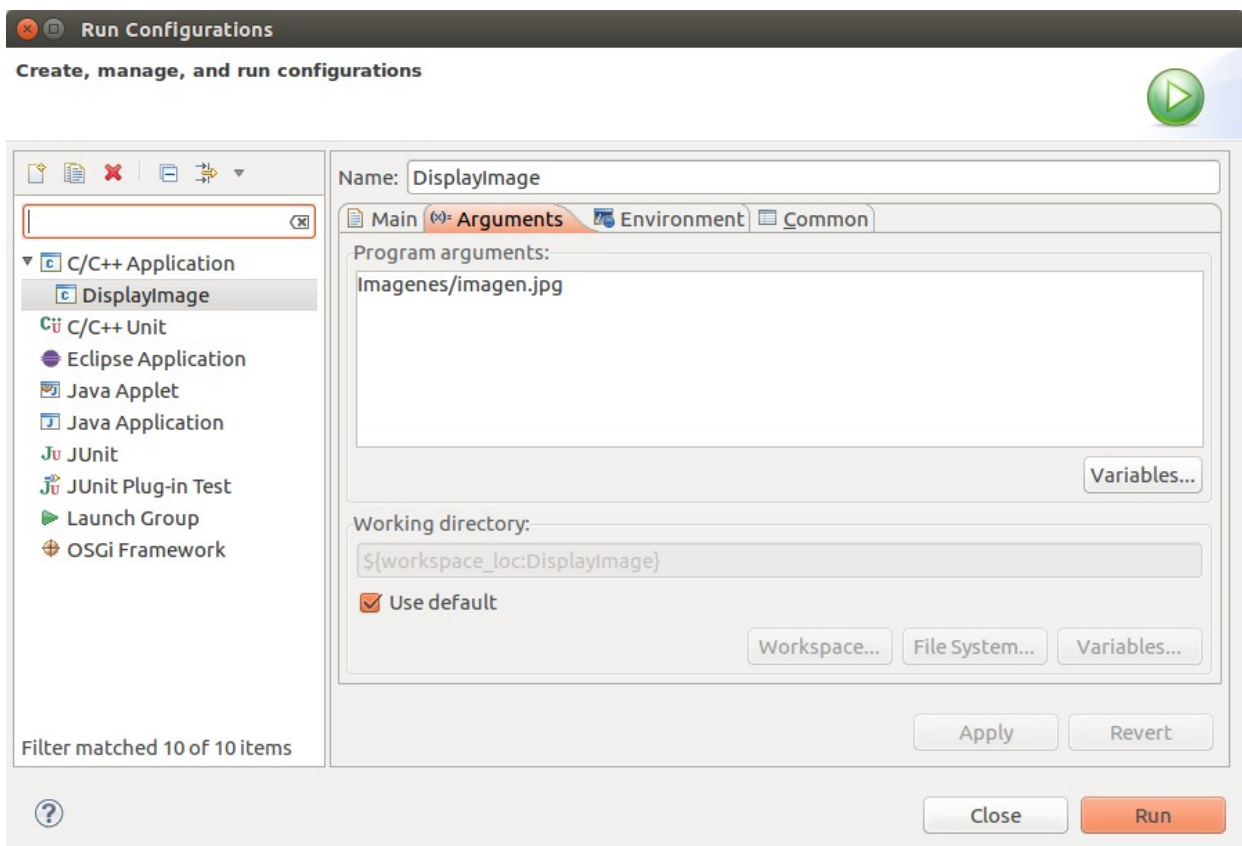
Compilar pulsando **Ctrl+B** ó **Project** → **Build all**

6.3. Ejecutar el programa

- 1) Crear una carpeta en el directorio del proyecto llamada **Imágenes**, guardar dentro una imagen, llamarla **imagen.jpg**.



- 2) Editar la configuración de ejecución mediante **Run → Run Configurations**. Ir a **C/C++ Application** → **DisplayImage**, ir a la pestaña **Arguments** y escribir la ruta relativa (al proyecto) de la imagen, es decir **Imagenes/imagen.jpg**.



- 3) Pulsar en **Run** y se ejecutará el programa