**Instalacion de Kinect**

**1. Abrir una terminal y ejecutar los siguientes comandos**

Ejecutar el siguiente comando para abir el administrador

sudo su

contraseña 1234admin

sudo apt-get update

sudo apt-get upgrade

**2. Instalar las dependencias necesarias**

sudo apt-get install git-core cmake freeglut3-dev pkg-config build-essential libxmu-dev libxi-dev libusb-1.0-0-dev

**3. Clonamos la librería del repositorio**

git clone git://github.com/OpenKinect/libfreenect.git

**4. Instalar el libfreenect**

cd libfreenect

mkdir build

cd build

cmake -L ..

make

sudo make install

sudo ldconfig /usr/local/lib64/

**5. Para usar Kinect como usuario no root haga lo siguiente**

sudo adduser $USER video

sudo adduser $USER plugdev

**6) También crea un archivo con reglas para el administrador de dispositivos Linux – nano es editor de texto que utiliza ubuntu**

sudo nano /etc/udev/rules.d/51-kinect.rules

**A continuación, copie y pegue lo siguiente y guárdelo**

# ATTR{product}=="Xbox NUI Motor"

SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02b0", MODE="0666"

# ATTR{product}=="Xbox NUI Audio"

SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ad", MODE="0666"

# ATTR{product}=="Xbox NUI Camera"

SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ae", MODE="0666"

# ATTR{product}=="Xbox NUI Motor"

SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02c2", MODE="0666"

# ATTR{product}=="Xbox NUI Motor"

SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02be", MODE="0666"

# ATTR{product}=="Xbox NUI Motor"

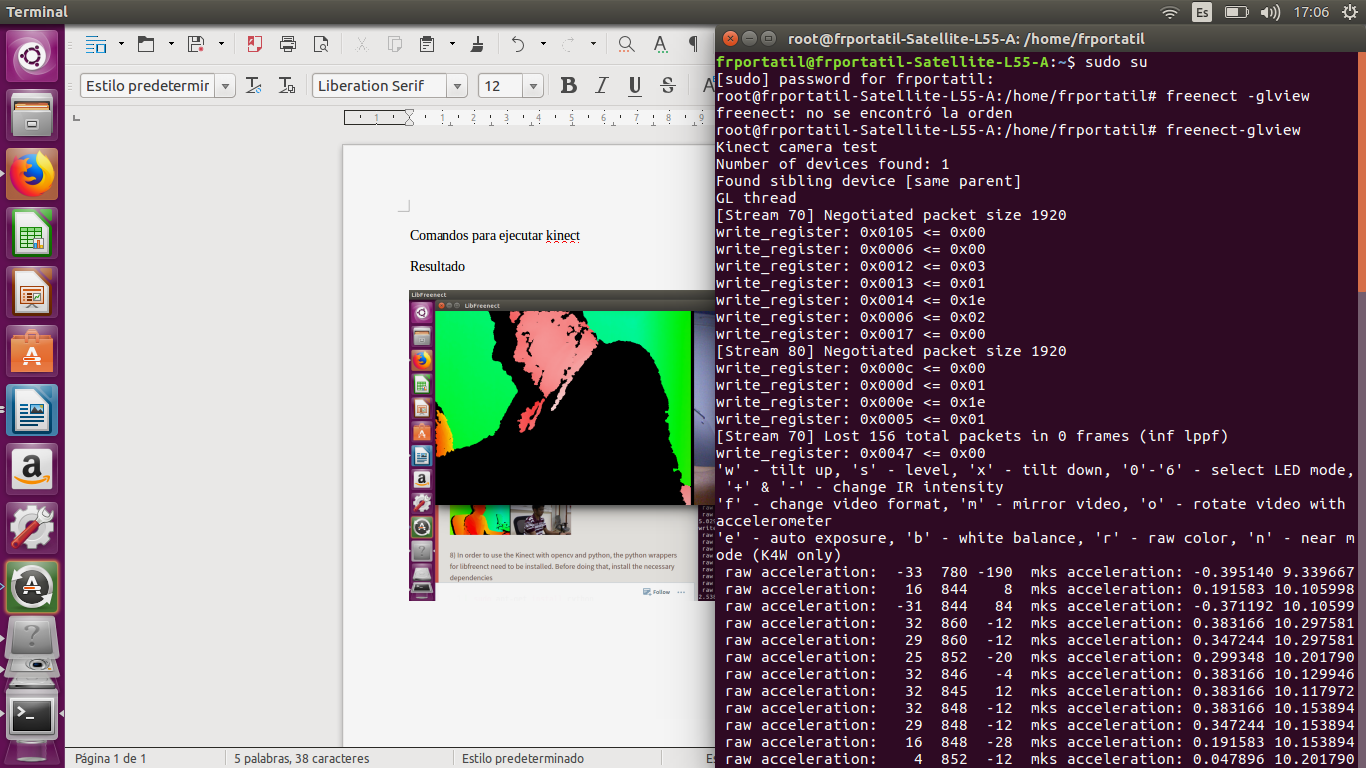
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02bf", MODE="0666"

**7) Desconéctese y vuelva a iniciarse. Ejecute el siguiente comando en un terminal para probar si libfreenect está instalado correctamente.**

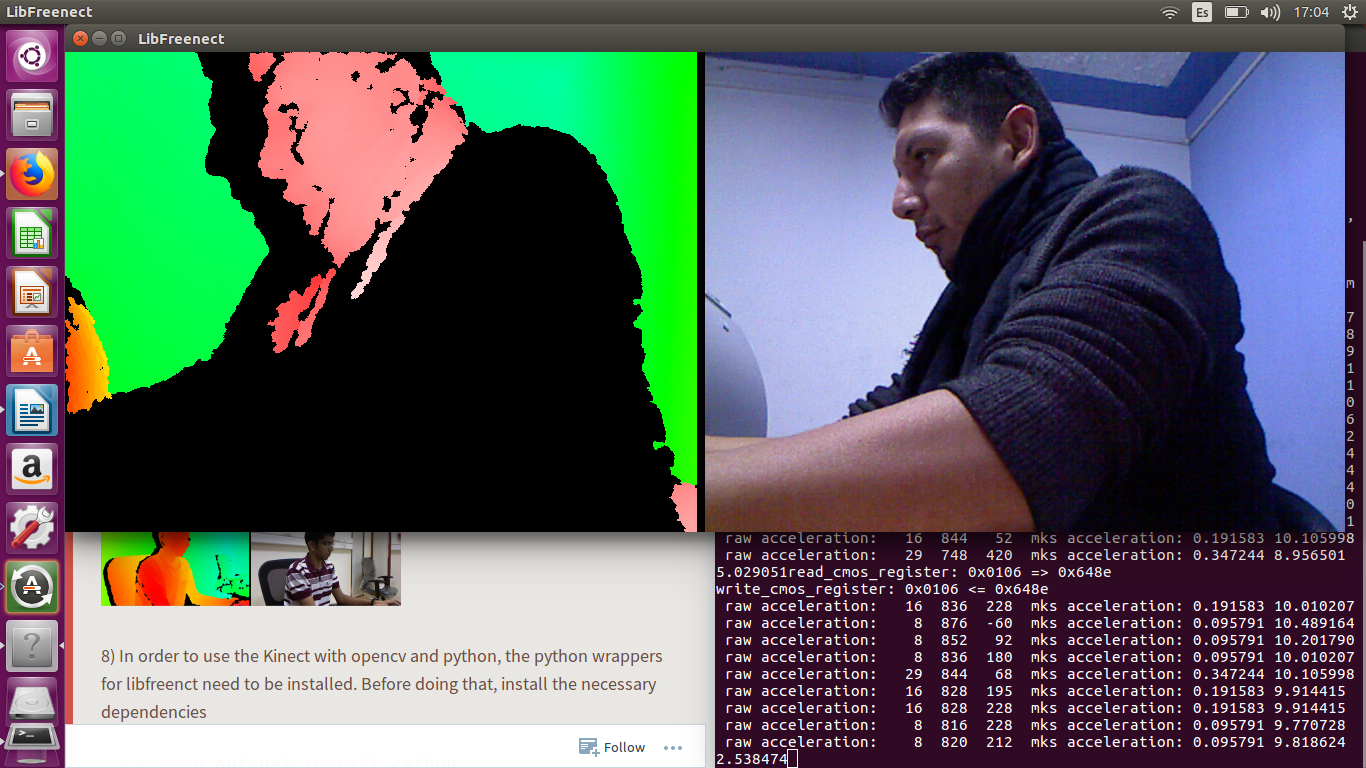
**Comando para ejecutar kinect**

Esto debería hacer que aparezca una ventana que muestra la profundidad y las imágenes RGB. Al presionar 'w' en el teclado, el kinect se inclina hacia arriba y al presionar 'x' el kinect se inclina hacia abajo. Hay varias otras opciones de control que se enumeran en la terminal cuando se ejecuta "freenect-glview"

freenect-glview



debe abir una ventana con la imagen.

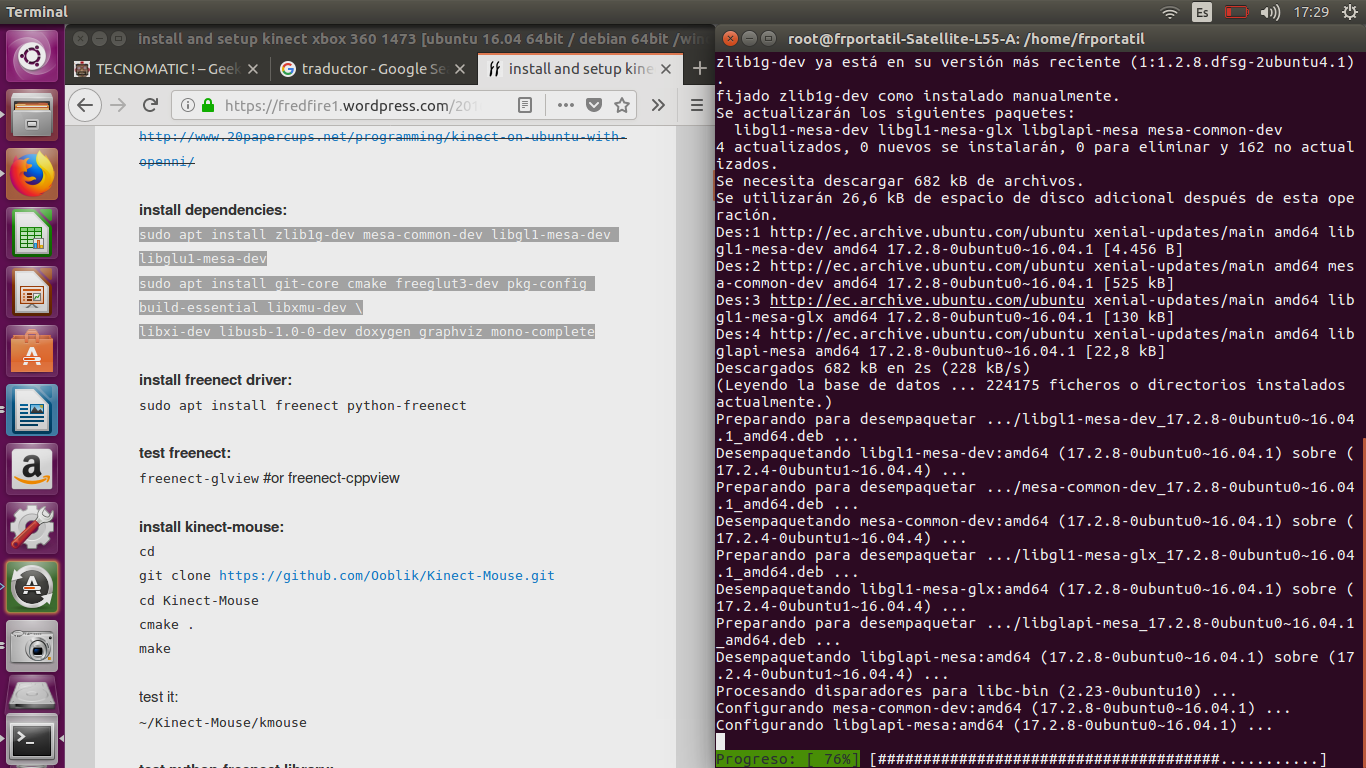


Referencia:

<https://naman5.wordpress.com/>

**Instalacion de las dependencias**

sudo apt install zlib1g-dev mesa-common-dev libgl1-mesa-dev libglu1-mesa-dev  
sudo apt install git-core cmake freeglut3-dev pkg-config build-essential libxmu-dev \  
libxi-dev libusb-1.0-0-dev doxygen graphviz mono-complete



**Instalar freenect driver:**

sudo apt install freenect python-freenect

Probar el freenect escribimos el siguiente comando

freenect-glview o freenect-cppview

**escribir el comando para descargar opencv**

sudo apt install python-opencv

**Aparece la pregunta s/n y escribimos s**

**para Probar opencv sobre python**

**wget : descarga dedee linea de comandos**

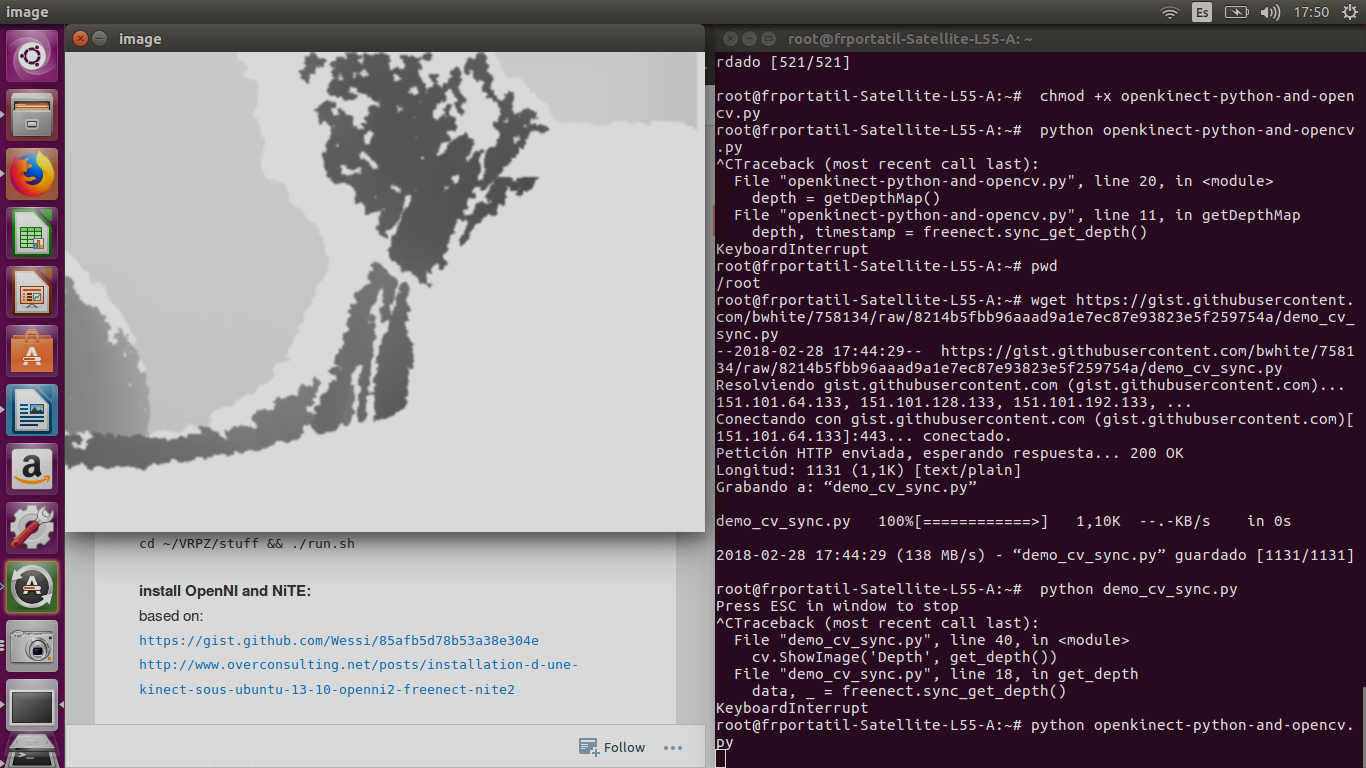
wget <https://gist.githubusercontent.com/anonymous/126c50cf9233ade3c79b35e4e22f7629/raw/62bea505b6d9d24079903bf210c873f7d84e4cd6/openkinect-python-and-opencv.py>

**chmod: asigna permisos de descarga**

chmod +x openkinect-python-and-opencv.py

**ejecuta la aplicación que funciona con opencv**

python openkinect-python-and-opencv.py



**Instalacion de pygame**

sudo apt install python-pygame

**aparece un mensaje de s/n escribimos s**

**Instalamos xlib**

sudo apt-get install python-xlib

abrimos en el navegador de internet el codigo para el reconocimiento del mause con la mano.

<http://code.activestate.com/recipes/578104-openkinect-mouse-control-using-python/>

**Abrimos una nueva terminal en ingresamos a cada una de las carpetas con cd libfreenect, cd wrappers, cd python**

**crear el archivo**

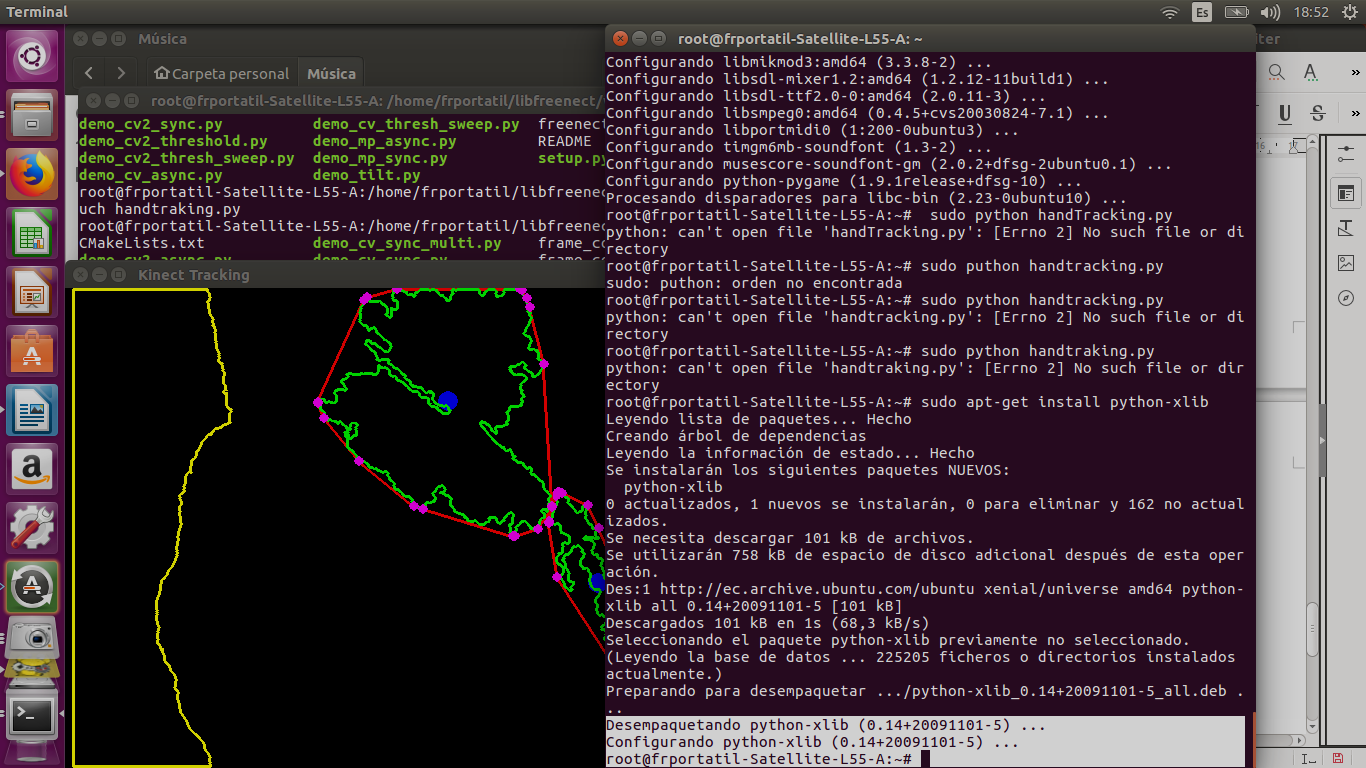
libfreenect/wrappers/python/ touch handtracking.py

**editamos,copiamos el codigo del sitio y pegamos, salimos con ctrl** + x

libfreenect/wrappers/python/ nano handtracking.py

**ejecutamos con el siguiente comando**

sudo python handtracking.py



**8. Para utilizar Kinect con opencv y python, es necesario instalar los wrappers de Python para libfreenct. Antes de hacerlo, instale las dependencias necesarias**

sudo apt-get install cython

sudo apt-get install python-dev

sudo apt-get install python-numpy

**aparece un mensaje s/n escribir s**

**9. Ir al directorio ……./libfreenect/wrappers/python and run the following command**

**escribir lo siguiente**

cd ..

cd home

cd frportatil este es el nombre del usuario

cd libfreenect/wrappers/python

sudo python setup.py install

**10. crear un nuevo archivo**

**crear el archivo**

libfreenect/wrappers/python/ touch kinect\_test.py

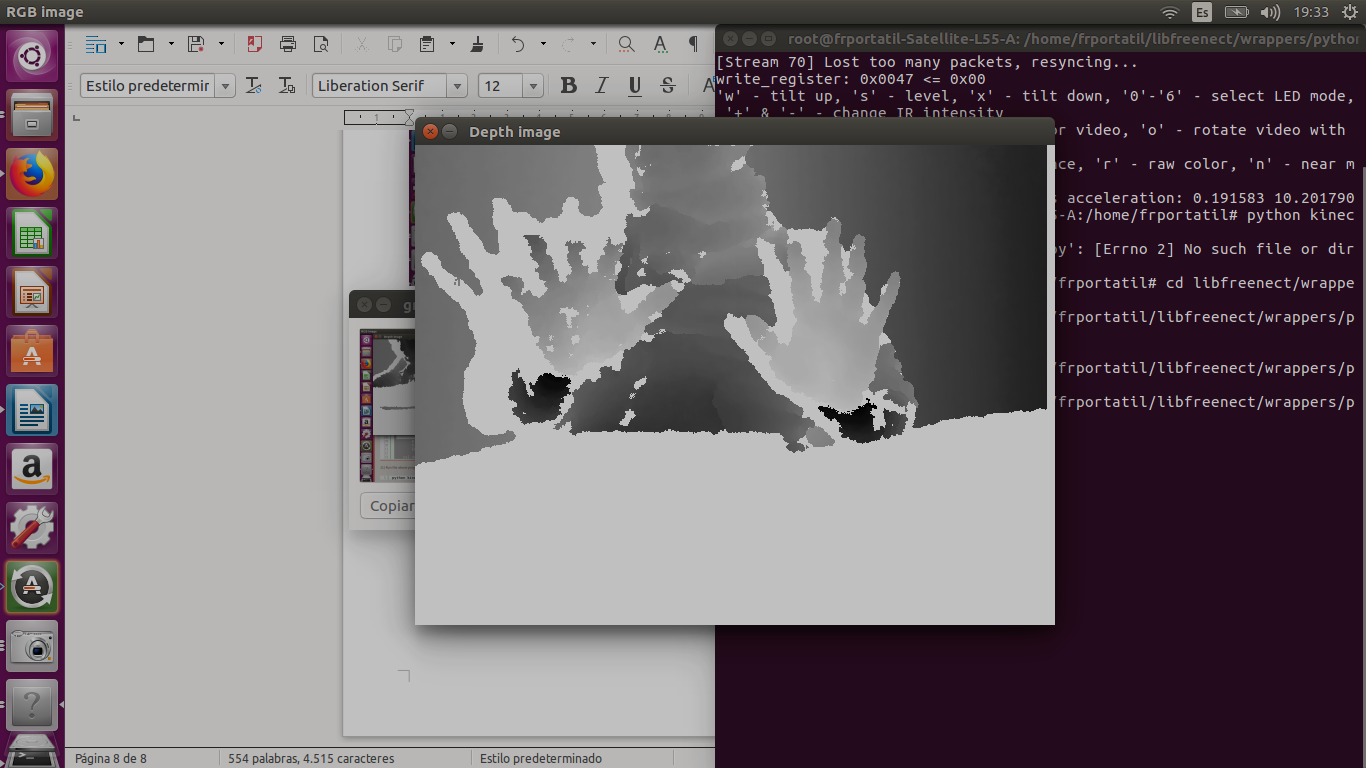
**editamos,copiamos el codigo del paso 10 del sitio y pegamos, salimos con ctrl** + x

libfreenect/wrappers/python/ nano kinect\_test.py

**ejecutamos con el siguiente comando**

sudo python kinect\_test.py

**debe aparecer la siguiente ventana**



**Open cv**

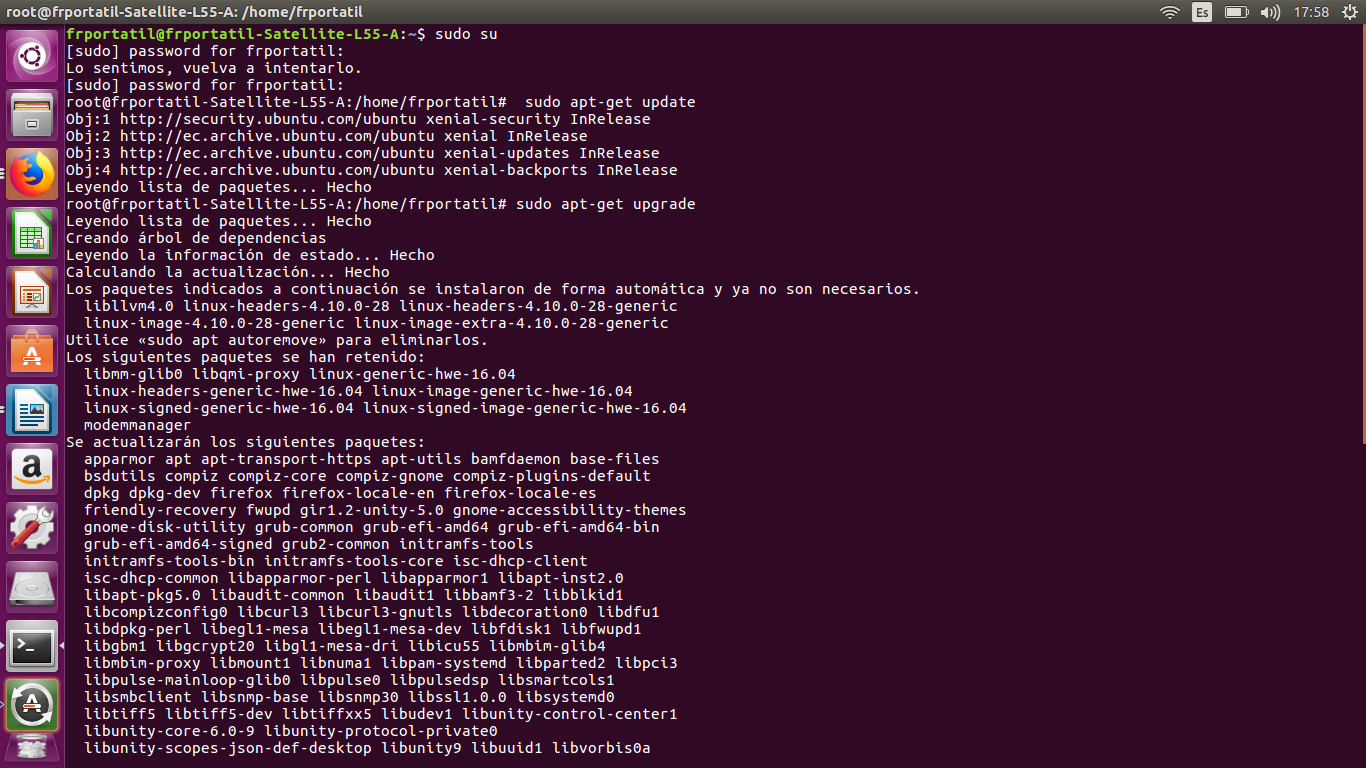
**Paso 1. Instalar OpenCV dependendencias en Ubuntu 16.04**

sudo apt-get update

sudo apt-get upgrade

aparece un mendsaje s /n, se debe presionar el s

la instalacion tiene un tiempo de 10 a 15 minutos

****

A continuación, instalemos algunas herramientas de desarrollador:

|  |
| --- |
| **sudo apt-get install build-essential cmake pkg-config** |

El paquete pkg-config ya está (muy probablemente) instalado en su sistema, pero asegúrese de incluirlo en el comando apt-get anterior, por si acaso. El programa cmake se usa para configurar automáticamente nuestra compilación OpenCV.

OpenCV es una biblioteca de procesamiento de imágenes y visión por computadora. Por lo tanto, OpenCV necesita poder cargar varios formatos de archivos de imagen desde el disco como JPEG, PNG, TIFF, etc. Para cargar estas imágenes desde el disco, OpenCV realmente llama a otras bibliotecas de E / S de imagen que realmente facilitan la carga y decodificación proceso. Instalamos los necesarios a continuación:

**sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev**

Bueno, ahora tenemos bibliotecas para cargar imágenes desde el disco, pero ¿y el video? Use los siguientes comandos para instalar los paquetes utilizados para procesar las transmisiones de video y acceder a las imágenes de las cámaras:

**sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev**

**sudo apt-get install libxvidcore-dev libx264-dev**

**OpenCV viene con un conjunto muy limitado de herramientas GUI. Estas herramientas GUI nos permiten mostrar una imagen en nuestra pantalla (cv2.imshow), esperar / registrar pulsaciones de teclas (cv2.waitKey), rastrear eventos del mouse (cv2.setMouseCallback) y crear elementos simples de la GUI, como controles deslizantes y barras de seguimiento. De nuevo, no debe esperar crear aplicaciones de GUI completas con OpenCV; estas son solo herramientas simples que le permiten depurar su código y crear aplicaciones muy simples.**

Internamente, el nombre del módulo que maneja las operaciones de la GUI de OpenCV es highgui. El módulo highgui se basa en la biblioteca GTK, que debe instalar utilizando el siguiente comando:

**sudo apt-get install libgtk-3-dev**

A continuación, instalamos bibliotecas que se utilizan para optimizar varias funcionalidades dentro de OpenCV, como operaciones de matriz:

**sudo apt-get install libatlas-base-dev gfortran**

Concluiremos el Paso # 1 instalando los encabezados de desarrollo y las bibliotecas de Python para Python 2.7 y Python 3.5 (de esa manera usted tiene ambos):

**sudo apt-get install python2.7-dev python3.5-dev**

Nota: Si no instala los encabezados de desarrollo Python y la biblioteca estática, se encontrará con problemas durante el Paso 4, donde ejecutaremos cmake para configurar nuestra compilación. Si estos encabezados no están instalados, entonces el comando cmake no podrá determinar automáticamente los valores correctos del intérprete de Python y las bibliotecas de Python. En resumen, el resultado de esta sección se verá "vacío" y no podrá construir los enlaces de Python. Cuando llegue al Paso 4, tómese el tiempo para comparar su salida del comando con el mío.

### Paso 2: Download the OpenCV source

At the time of this article’s publication, the most recent version of OpenCV is 3.1.0 , which we download a .zip of and unarchive using the following commands:

**cd ~**

**wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip**

**unzip opencv.zip**

La desrcarga se demora de 5 a 10 minutos

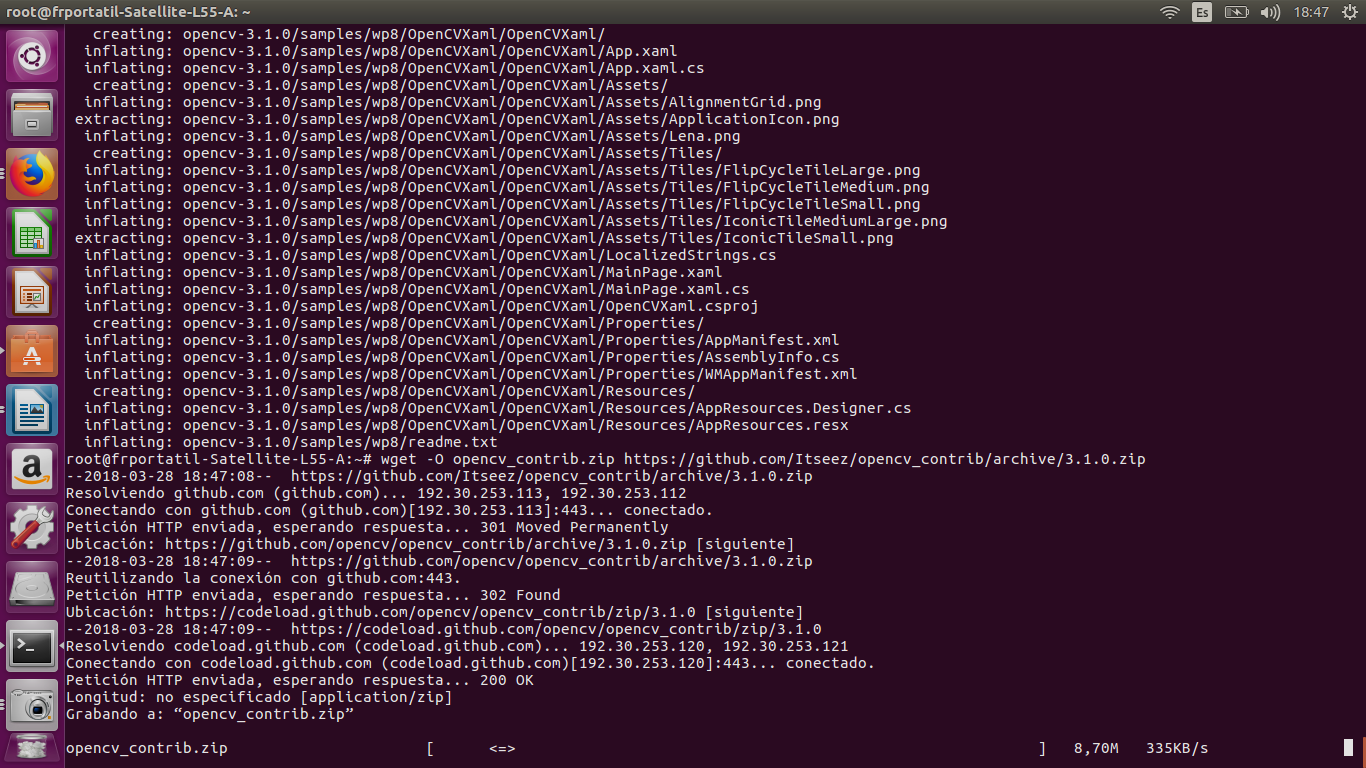


Cuando se lanzan nuevas versiones de OpenCV, puede consultar el OpenCV GitHub oficial y descargar la última versión cambiando el número de versión de .zip.

Sin embargo, aún no hemos terminado de descargar el código fuente; también necesitamos el repositorio opencv\_contrib:

wget -O opencv\_contrib.zip <https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip>

se demora en descargar 5 min



unzip opencv\_contrib.zip

¿Por qué nos molestamos en descargar también el repositorio contrib?

Bueno, queremos que la instalación completa de OpenCV 3 tenga acceso a las características (sin juego de palabras) como SIFT y SURF. En OpenCV 2.4, SIFT y SURF se incluyeron en la instalación predeterminada de OpenCV. Sin embargo, con el lanzamiento de OpenCV 3+, estos paquetes se han movido a contrib, que alberga ya sea (1) módulos que están actualmente en desarrollo o (2) módulos que están marcados como "no libres" (es decir, patentados). Puede obtener más información sobre el razonamiento detrás de la reestructuración de SIFT / SURF en esta publicación de blog.

Nota: Es posible que necesite expandir los comandos anteriores utilizando el botón "<=>" durante su copia y pegado. El .zip en 3.1.0.zip puede cortarse en ventanas de navegador más pequeñas. Para su comodidad, he incluido la URL completa tanto del archivo opencv como del archivo opencv\_contrib a continuación:

    https://github.com/Itseez/opencv/archive/3.1.0.zip

    https://github.com/Itseez/opencv\_contrib/archive/3.1.0.zip

También quiero mencionar que tanto las versiones de opencv como de opencv\_contrib deben ser las mismas (en este caso, 3.1.0). Si los números de las versiones no coinciden, fácilmente podría encontrarse con errores de tiempo de compilación (o peor, errores de tiempo de ejecución que son casi imposibles de depurar).

**Paso # 3: Configura tu entorno de Python - Python 2.7 o Python 3**

Instalar Python 2.7

Ahora estamos listos para comenzar a configurar nuestro entorno de desarrollo Python para la compilación. El primer paso es instalar pip, un administrador de paquetes de Python:

**cd ~**

**wget https://bootstrap.pypa.io/get-pip.py**

**sudo python get-pip.py**



He mencionado esto en cada uno de los tutoriales de instalación de OpenCV + Python que he hecho alguna vez, pero lo diré de nuevo aquí hoy: soy un gran fan de virtualenv y virtualenvwrapper. Estos paquetes de Python le permiten crear entornos de Python independientes e independientes para cada proyecto en el que esté trabajando.

En resumen, el uso de estos paquetes le permite resolver el "Proyecto X depende de la versión 1.x, pero el Proyecto Y necesita un dilema 4.x. Un efecto secundario fantástico del uso de entornos virtuales de Python es que puede mantener su sistema ordenado, ordenado, y libre de ecos parásitos.

Si bien es cierto que puede instalar OpenCV con enlaces de Python sin entornos virtuales de Python, le recomiendo que los utilice, ya que otros tutoriales de PyImageSearch aprovechan los entornos virtuales de Python. También asumiré que tiene tanto virtualenv como virtualenvwrapper instalados en el resto de esta guía.

Si desea una explicación completa y detallada de por qué los entornos virtuales de Python son una buena práctica, debe dar una lectura perfecta a esta excelente publicación de blog en RealPython. También proporciono algunos comentarios sobre por qué personalmente prefiero los entornos virtuales de Python en la primera mitad de este tutorial.

De nuevo, permítanme reiterar que es una práctica estándar en la comunidad de Python aprovechar los entornos virtuales de algún tipo, por lo que le sugiero que haga lo mismo:

**sudo pip install virtualenv virtualenvwrapper**

**sudo rm -rf ~/get-pip.py ~/.cache/pip**

Pasomos por alto estas instrucciones xq no se encuentra el archivo

Una vez que tenemos virtualenv y virtualenvwrapper instalados, necesitamos actualizar nuestro archivo ~ / .bashrc para incluir las siguientes líneas en la parte inferior del archivo:

**# virtualenv and virtualenvwrapper**

**export WORKON\_HOME=$HOME/.virtualenvs**

**source /usr/local/bin/virtualenvwrapper.sh**

**echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.bashrc**

**echo "export WORKON\_HOME=$HOME/.virtualenvs" >> ~/.bashrc**

**echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc**

**Después de editar nuestro archivo ~ / .bashrc, tenemos que volver a cargar los cambios:**

**source ~/.bashrc**

Nota: Llamar a la fuente en .bashrc solo tiene que hacerse una vez para nuestra sesión de shell actual. Cada vez que abrimos una nueva terminal, los contenidos de .bashrc se ejecutarán automáticamente (incluidas nuestras actualizaciones).

Ahora que hemos instalado virtualenv y virtualenvwrapper, el siguiente paso es crear realmente el entorno virtual de Python; lo hacemos utilizando el comando mkvirtualenv.

Pero antes de ejecutar este comando, debe hacer una elección: ¿Desea usar Python 2.7 o Python 3?

El resultado de su elección determinará qué comando ejecutará en la siguiente sección.

Creando su entorno virtual de Python

Si decide usar Python 2.7, use el siguiente comando para crear un entorno virtual de Python 2.7:

**mkvirtualenv cv -p python2**

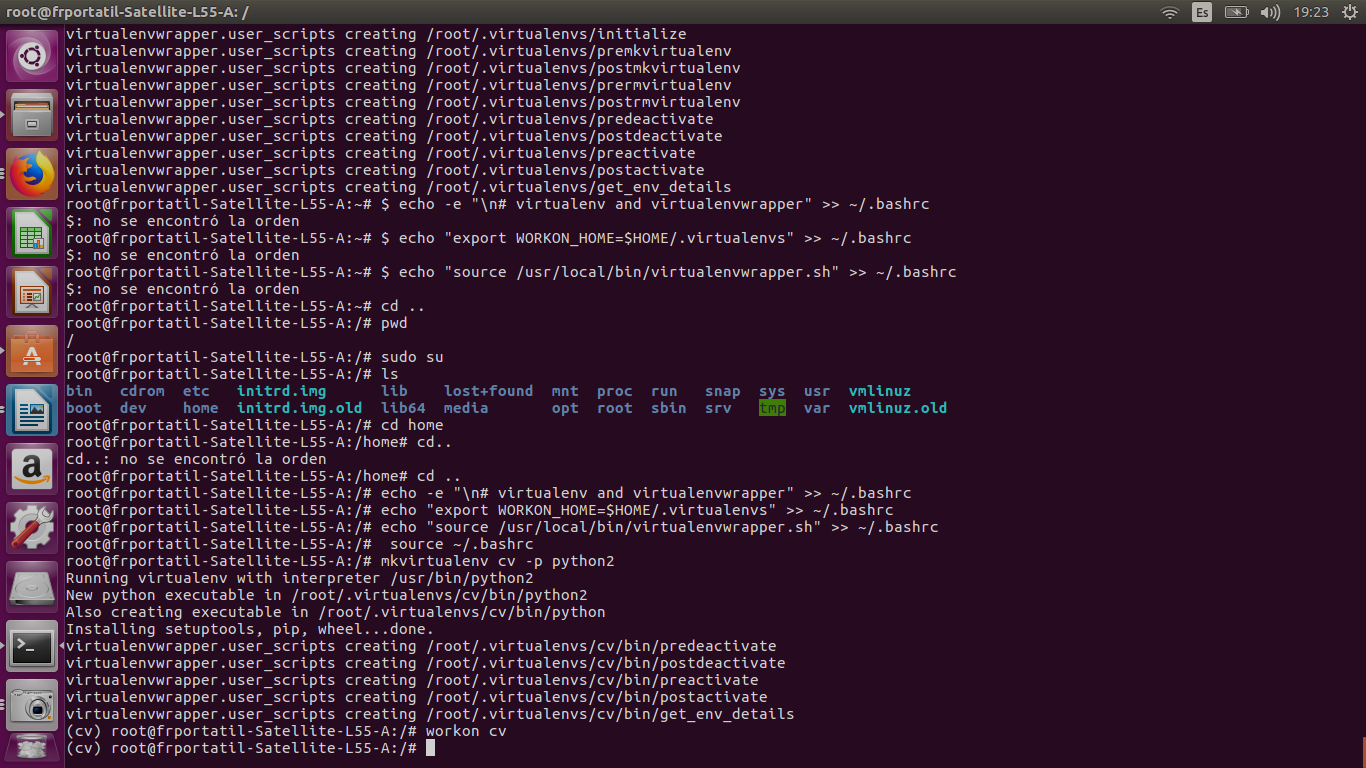
**Verificando que estás en el entorno virtual "cv"**

Si alguna vez reinicias tu sistema Ubuntu; cerrar sesión y volver a iniciar sesión; o abra una nueva terminal, necesitará usar el comando workon para volver a acceder a su entorno virtual cv. Un ejemplo del comando workon sigue:

**workon cv**

Para validar que se encuentra en el entorno virtual de cv, simplemente examine su línea de comando: si ve el texto (cv) precediendo su solicitud, entonces se encuentra en el entorno virtual de cv:

**aparece la siguiente ventana en el cual inicia con ( cv )**

****

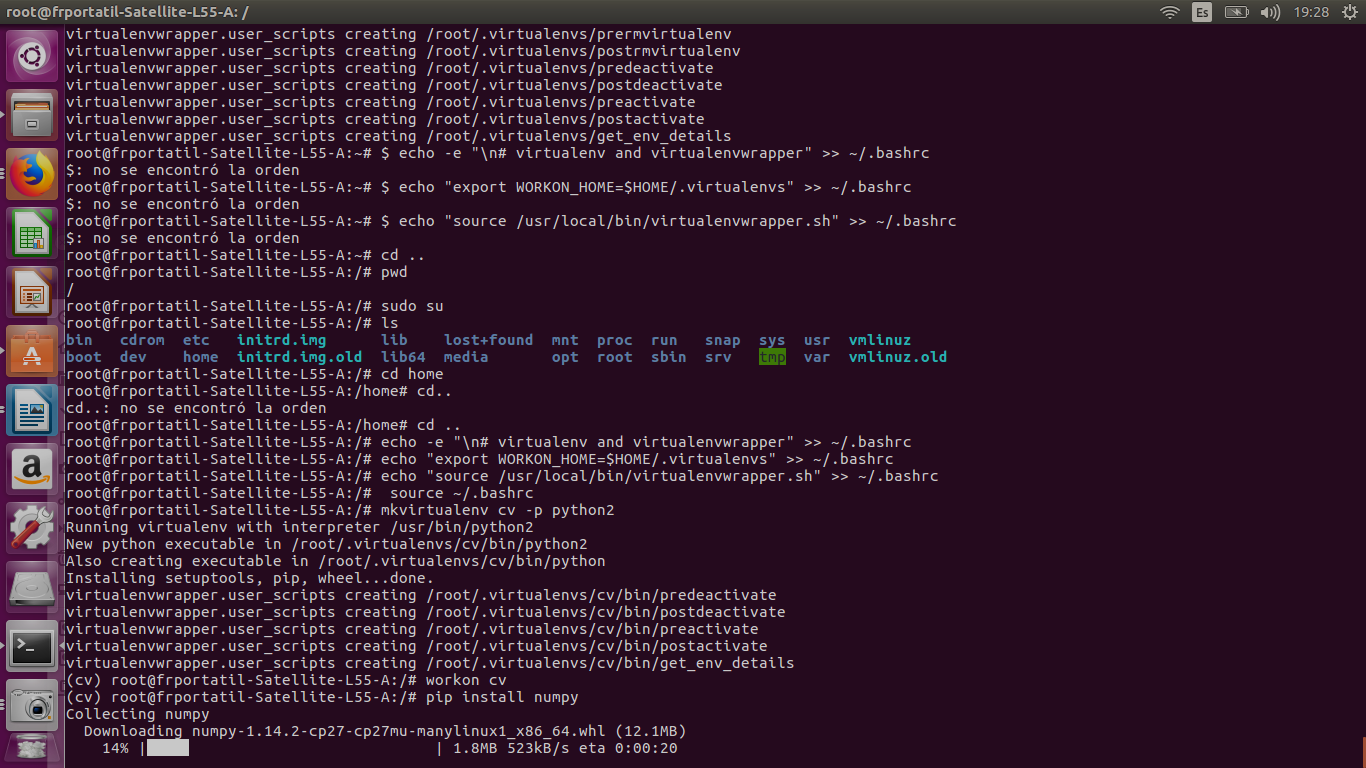
**nota:** De lo contrario, si no ve el texto cv, entonces no se encuentra en el entorno virtual cv:

**Instala NumPy en tu entorno virtual de Python**

El último paso antes de compilar OpenCV es instalar NumPy, un paquete de Python utilizado para el procesamiento numérico. Para instalar NumPy, asegúrese de estar en el entorno virtual cv (de lo contrario, NumPy se instalará en la versión del sistema de Python en lugar del entorno de cv). Desde allí ejecuta el siguiente comando:

**pip install numpy**

ventana de instalacion nuppy

****

**Paso # 4: Configurando y compilando OpenCV en Ubuntu 16.04**

**En este punto, todos nuestros prerrequisitos necesarios han sido instalados, ¡ahora estamos listos para compilar y OpenCV!**

Pero antes de hacerlo, verifique nuevamente que se encuentra en el entorno virtual cv examinando su mensaje (debe ver el texto (cv) que lo precede), y si no, use el comando workon:

**workon cv**

**debe ubicarse (cv ) en la linea de codigo**

**cd ~/opencv-3.1.0/**

mkdir build

cd build

cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr/local \

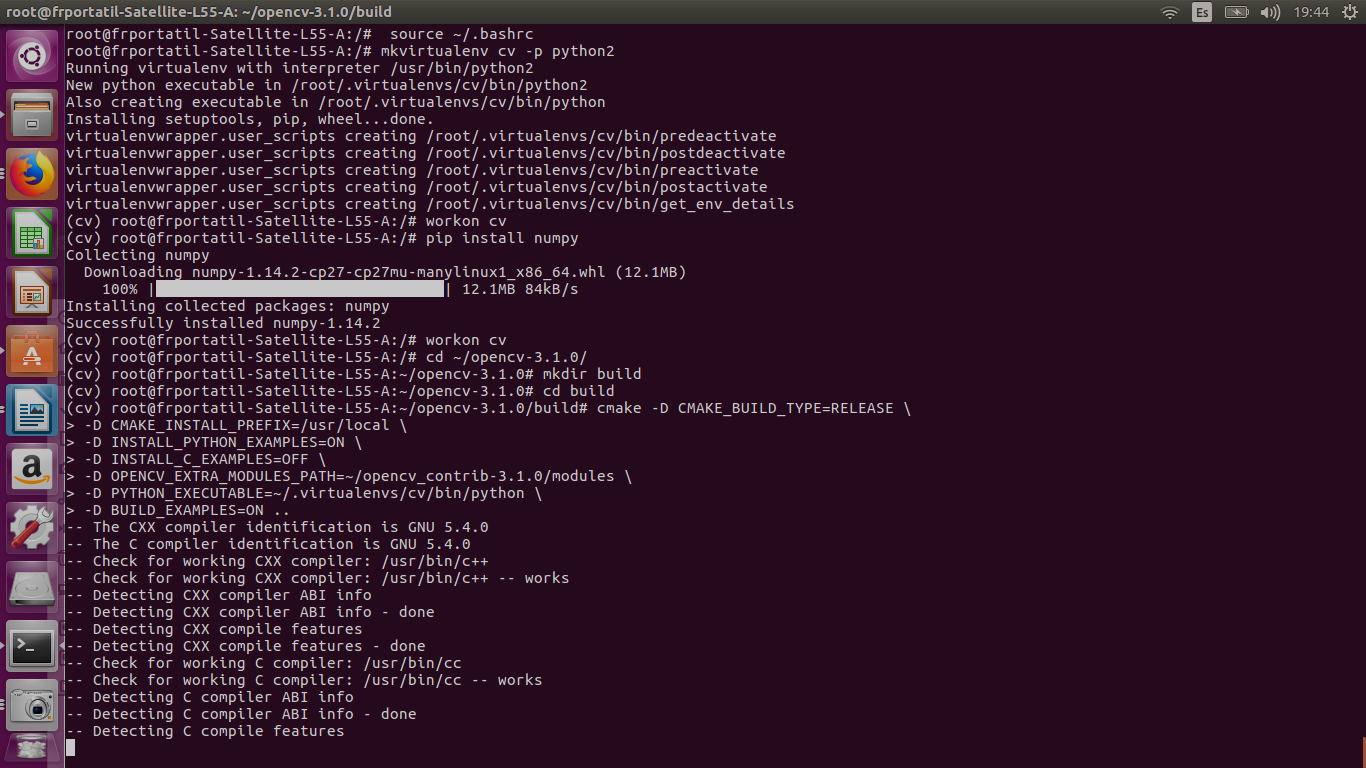
-D INSTALL\_PYTHON\_EXAMPLES=ON \

-D INSTALL\_C\_EXAMPLES=OFF \

-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib-3.1.0/modules \

-D PYTHON\_EXECUTABLE=~/.virtualenvs/cv/bin/python \

-D BUILD\_EXAMPLES=ON ..

****

**Los comandos anteriores cambian el directorio a ~ / opencv-3.1.0, que si ha seguido este tutorial es donde descargó y desarchivó los archivos .zip.**

Nota: Si está obteniendo un error relacionado con stdlib.h: No existe ningún archivo o directorio durante la fase cmake o make de este tutorial, también deberá incluir la siguiente opción en CMake: -D ENABLE\_PRECOMPILED\_HEADERS = OFF. En este caso, le sugiero que elimine su directorio de compilación, lo vuelva a crear y vuelva a ejecutar CMake con la opción anterior incluida. Esto resolverá el error stdlib.h. ¡Gracias a Carter Cherry y Marcin por señalar esto en la sección de comentarios!

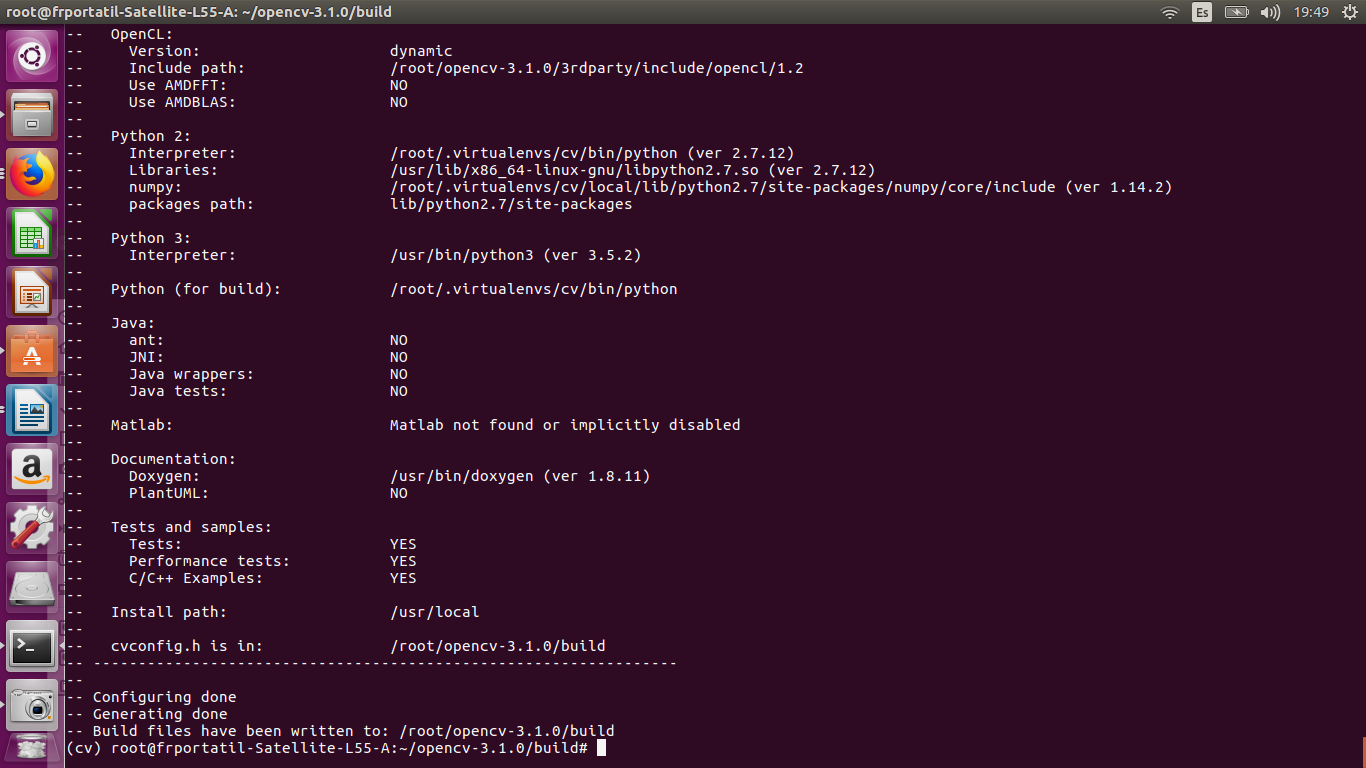
Dentro de este directorio creamos un subdirectorio llamado build y lo cambiamos. El directorio de compilación es donde se realizará la compilación real.

Finalmente, ejecutamos cmake para configurar nuestra compilación.

Antes de pasar a la compilación real de OpenCV, ¡asegúrese de examinar la salida de CMake!

Para hacer esto, desplácese hacia abajo en la sección titulada Python 2 y Python 3.

Si está compilando OpenCV en Ubuntu 16.04 con compatibilidad con Python 2.7, asegúrese de que la sección Python 2 incluya rutas válidas para la ruta de Intérprete, Bibliotecas, numpy y paquetes. Tu salida debería ser similar a la mía a continuación:



Nuevamente, observe cómo mi ruta de Intérprete, Bibliotecas, numpy y paquetes se han establecido correctamente.

Si no ve los entornos virtuales de cv en estas rutas de acceso variables, es casi seguro que no se encuentra en el entorno virtual de cv antes de ejecutar CMake.

Si ese es realmente el caso, simplemente acceda al entorno virtual cv llamando a workon cv y vuelva a ejecutar el comando CMake mencionado anteriormente.

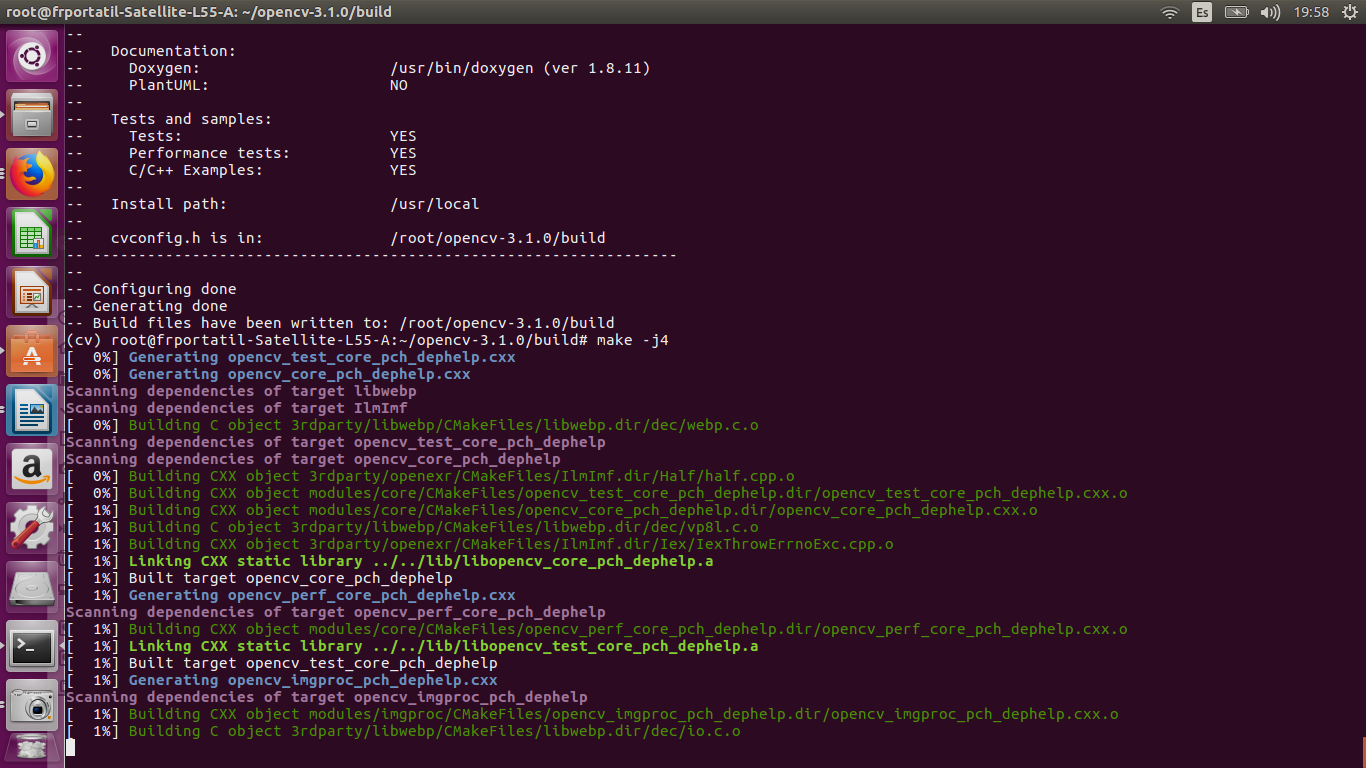
Suponiendo que su comando CMake salió sin ningún error, ahora puede compilar OpenCV:

**make -j4**

**El modificador -j controla el número de procesos que se utilizarán al compilar OpenCV; querrá establecer este valor en la cantidad de procesadores / núcleos en su máquina. En mi caso, tengo un procesador de cuatro núcleos, así que configuro -j4.**

El uso de procesos múltiples permite que OpenCV compile más rápido; sin embargo, hay momentos en los que se tocan las condiciones de carrera y las bombas de compilación salen. Si bien no se puede decir realmente si este es el caso sin una gran cantidad de experiencia previa compilando OpenCV, si se encuentra con un error, mi primera sugerencia sería ejecutar make clean para vaciar la compilación, seguido de la compilación usando solo un único núcleo:

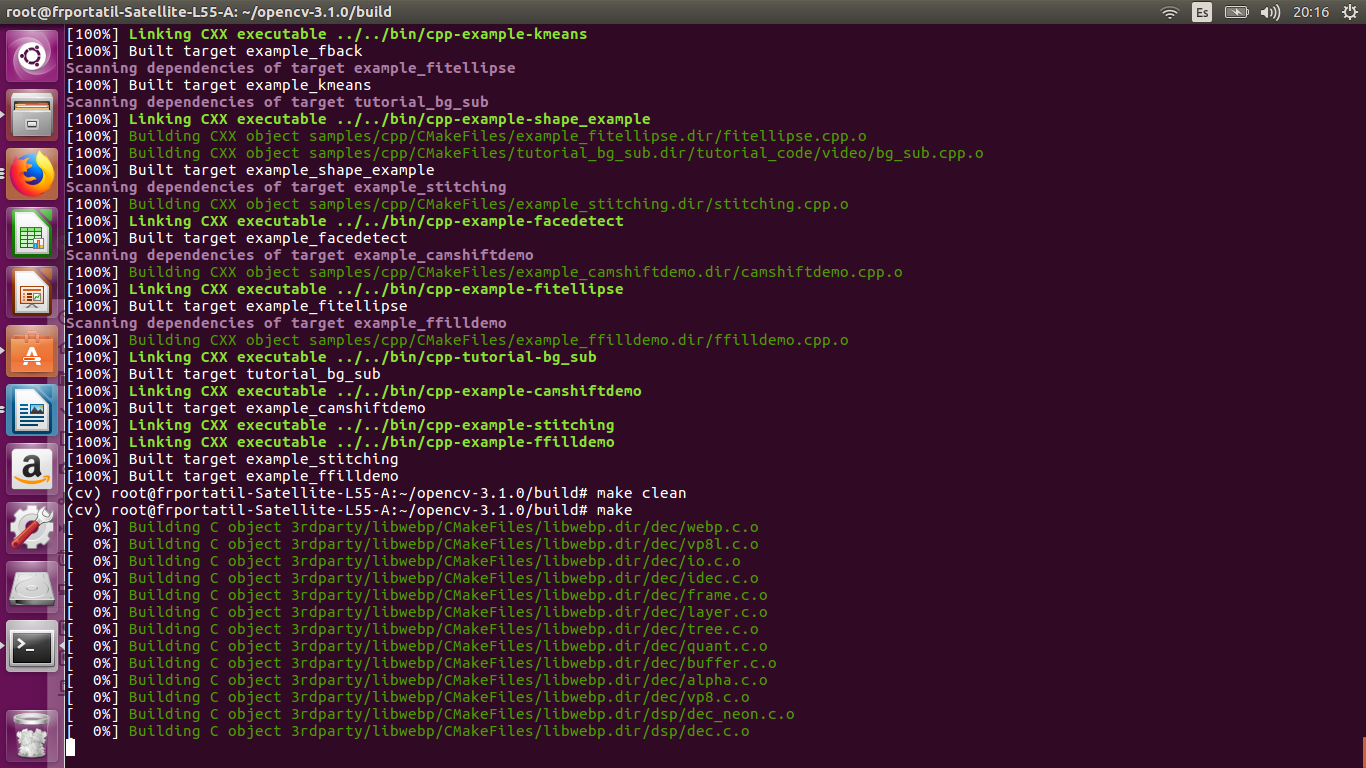
**Nota: Se tarda de 15 a 20 minutos**

****

**make clean**

**make**

aparece la siguiente pantalla de instalacion que tiene un tiempo de 30 a 45 minutos

****

El último paso es instalar OpenCV 3 en Ubuntu 16.04:

**sudo make install**

**sudo ldconfig**

### Paso #5: Finish your OpenCV install

**Estás llegando a la recta final, solo unos pasos más por recorrer y tu sistema Ubuntu 16.04 estará configurado con OpenCV 3.**

Para Python 2.7:

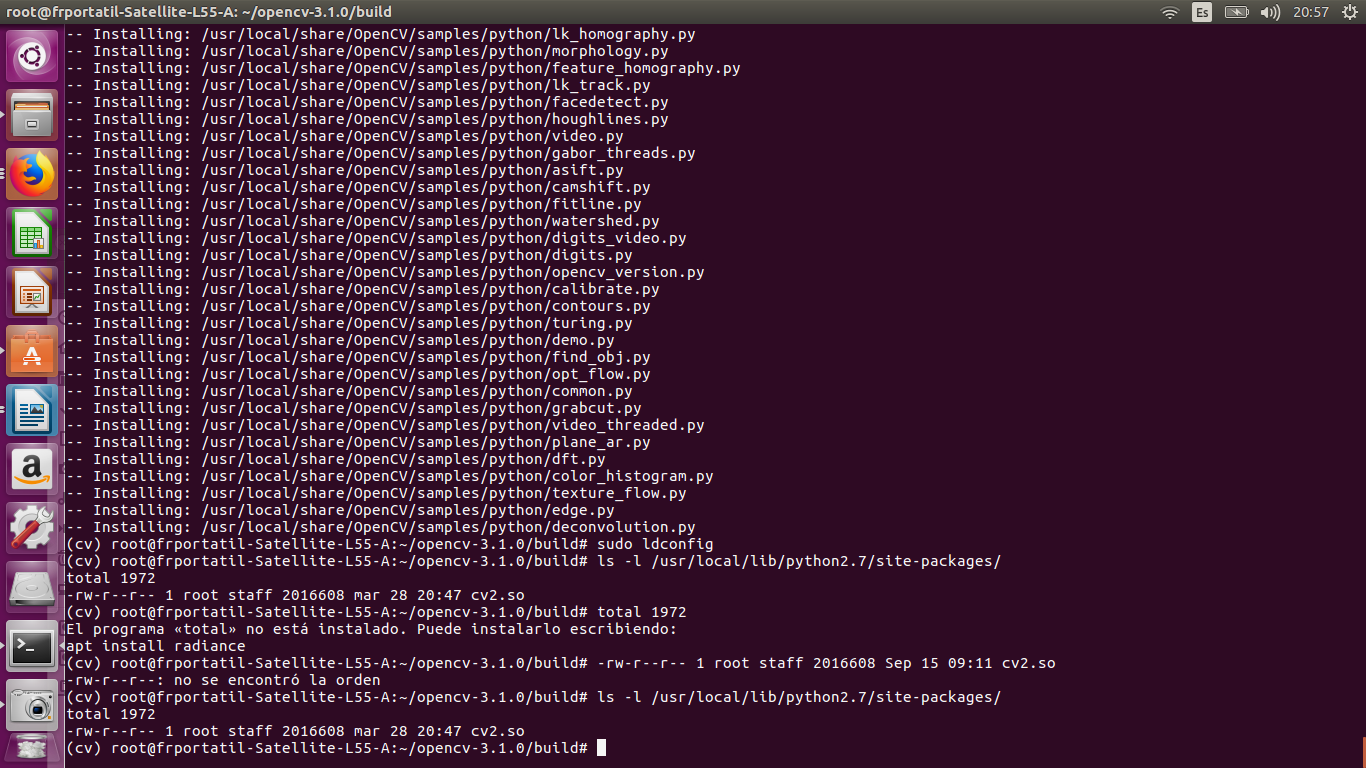
Después de ejecutar sudo make install, tus enlaces de Python 2.7 para OpenCV 3 ahora deberían estar ubicados en /usr/local/lib/python-2.7/site-packages/. Puede verificar esto usando el comando ls:

**ls -l /usr/local/lib/python2.7/site-packages/**

**Debe aparecr la siguiente pantalla**

total 1972

-rw-r--r-- 1 root staff 2016608 Sep 15 09:11 cv2.so

****

Nota: en algunos casos, puede encontrar que OpenCV se instaló en /usr/local/lib/python2.7/dist-packages en lugar de /usr/local/lib/python2.7/site-packages (note dist-packages versus site-packages). Si sus enlaces cv2.so no están en el directorio site-packages, asegúrese de marcar dist-pakages.

El último paso es sincronizar nuestras vinculaciones OpenCV cv2.so en nuestro entorno virtual cv para Python 2.7:

**cd ~/.virtualenvs/cv/lib/python2.7/site-packages/**

**ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so**

### Paso #6: Testing your OpenCV install

**¡Felicitaciones, ahora tiene OpenCV 3 instalado en su sistema Ubuntu 16.04!**

Para verificar que su instalación funciona:

    Abre una nueva terminal.

    Ejecute el comando workon para acceder al entorno virtual cv Python.

    Intenta importar las vinculaciones de Python + OpenCV.

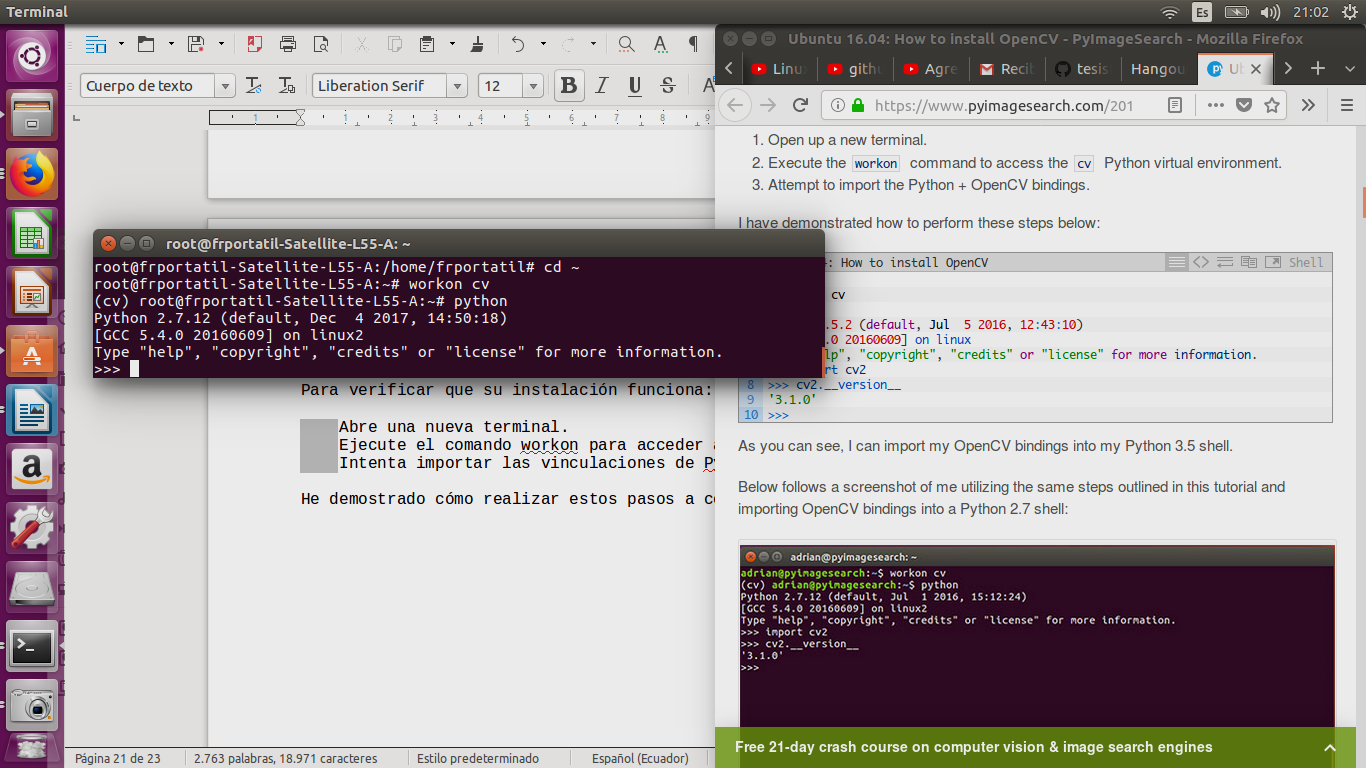
He demostrado cómo realizar estos pasos a continuación:

**cd ~**

workon cv

python

**Debe salir la siguiente pantalla**

****

**Python 2.7.12 (default, Dec 4 2017, 14:50:18)**

[GCC 5.4.0 20160609] on linux

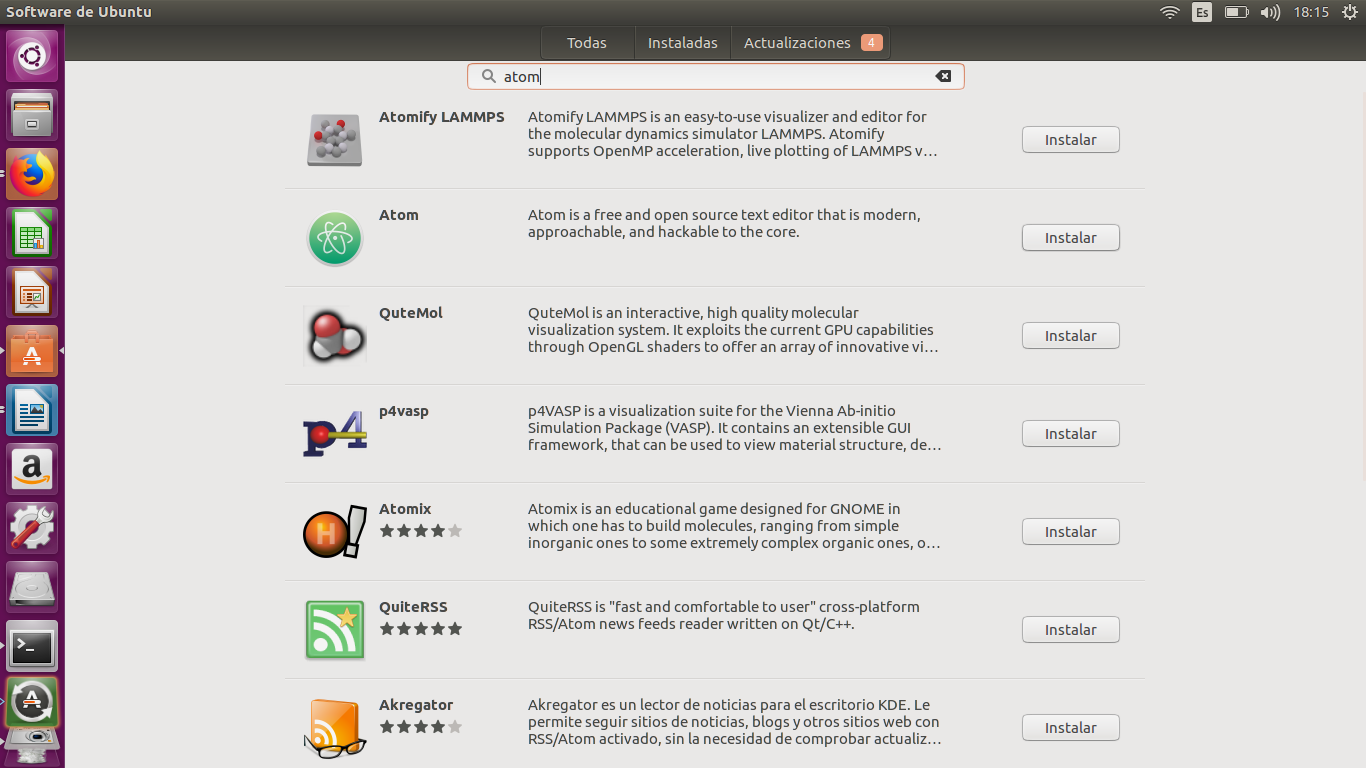
Type "help", "copyright", "credits" or "license" for more information.

>>> import cv2

>>> cv2.\_\_version\_\_

'3.1.0'

>>>

****

**Instalacion del Atom**

Atom es un editor de texto gratuito y de código abierto que es moderno,

accesible y pirateable hasta el núcleo.