```
sudo apt-get update

sudo apt-get upgrade

sudo apt-get dist-upgrade

#INSTALAR DRIVER DE NVIDIA DESDE software&updates

nvidia-smi    #Verificando que el driver este instalado

lspci | grep -i nvidia

python3

sudo apt-get install python3-pip python3-dev

sudo apt-get install build-essential cmake git unzip


sudo apt-get install pkg-config libopenblas-dev liblapack-dev


sudo apt-get install libatlas-base-dev gfortran


sudo apt-get install python3-numpy python3-scipy python3-matplotlib python3-yaml


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


sudo apt-get install libxvidcore-dev libx264-dev


sudo apt-get install libhdf5-serial-dev python3-h5py


sudo apt-get install graphviz


sudo pip3 install pydot-ng scikit-learn pillow

sudo apt-get install libjpeg8-dev libtiff5-dev

sudo apt-get install libgtk-3-dev

sudo apt install qtbase5-dev

pip3 install opencv-contrib-python
```

```
python3

        >>>import cv2

        >>> cv2.__version__
```

gcc --version

wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_10.0.130-1_amd64.deb

sudo dpkg -i cuda-repo-ubuntu1804_10.0.130-1_amd64.deb

sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub

sudo apt-get update && sudo apt full-upgrade

sudo apt-get install cuda-10-0

echo $PATH #agregar ruta al path

Step 8: Go to terminal and type:

```
nano ~/.bashrc
```

In the end of the file, add:

```
export PATH=/usr/local/cuda-9.1/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-9.1/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

ctrl+x then y to save and exit

```
source ~/.bashrc
sudo ldconfig
nvidia-smi
```

nano ~/.bashrc

export PATH=/usr/local/cuda-10.0/bin${PATH:+:${PATH}}

export LD_LIBRARY_PATH=/usr/local/cuda-10.0/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}

source ~/.bashrc

sudo ldconfig

Reboot the cpu

nvidia-smi

nvcc --version

nvcc   -V

lspci   -v

cat /proc/driver/nvidia/version

cuda-install-samples-10.0.sh ~/ExamplesCUDA

ir al directorio

make -j4

When done, go to `~/jccuda/NVIDIA_CUDA-8.0_Samples/bin/x86_64`
out some commands.

`deviceQuery` should produce something meaningful...

```
johnny@johnny-XPS-8700:~/jccuda/NVIDIA_CUDA-8.0_Samples/bin
./deviceQuery Starting...
```

Para instalar CuDNN

https://developer.nvidia.com/rdp/cudnn-archive

versión 7.0.5

Download the following:

cuDNN v7.1.2 Runtime Library for Ubuntu16.04 (Deb)

cuDNN v7.1.2 Developer Library for Ubuntu16.04 (Deb)

cuDNN v7.1.2 Code Samples and User Guide for Ubuntu16.04 (Deb)

Goto downloaded folder and in terminal perform following:

```
sudo dpkg -i libcudnn7_7.1.2.21-1+cuda9.1_amd64.deb
sudo dpkg -i libcudnn7-dev_7.1.2.21-1+cuda9.1_amd64.deb
sudo dpkg -i libcudnn7-doc_7.1.2.21-1+cuda9.1_amd64.deb
```

Verifying cuDNN installation:

```
cp -r /usr/src/cudnn_samples_v7/ $HOME
cd  $HOME/cudnn_samples_v7/mnistCUDNN
make clean && make
./mnistCUDNN
```

If cuDNN is properly installed and running on your Linux system, you will see a message similar to the following:

Test passed!

sudo apt-get install python3-venv

create virtual environment for tensorflow

python3 -m venv tf_env

source tf_env /bin/activate

#deactivate para cerrar el entorno

sudo pip3 install tensorflow-gpu==1.5

```
$ git clone https://github.com/fchollet/keras
$ cd keras
$ sudo python setup.py install
```

You can now try to run a Keras script, such as this MNIST example:

```
python examples/mnist_cnn.py
```

Check Installation of Frameworks

```
1   workon virtual-py2
2   python
3   import numpy
4   numpy.__version__
5   import theano
6   theano.__version__
7   import tensorflow
8   tensorflow.__version__
9   import keras
10  keras.__version__
11  import torch
12  torch.__version__
13  import cv2
14  cv2.__version__
```

```
apt-get update && apt-get install -y --allow-downgrades --no-install-recommends \
    libcudnn7=7.0.5.15-1+cuda9.0 \
    libcudnn7-dev=7.0.5.15-1+cuda9.0 && \
    rm -rf /var/lib/apt/lists/*
```