



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



Documentación Kinect

“Primeros Pasos”

Fecha: 26/04/2011

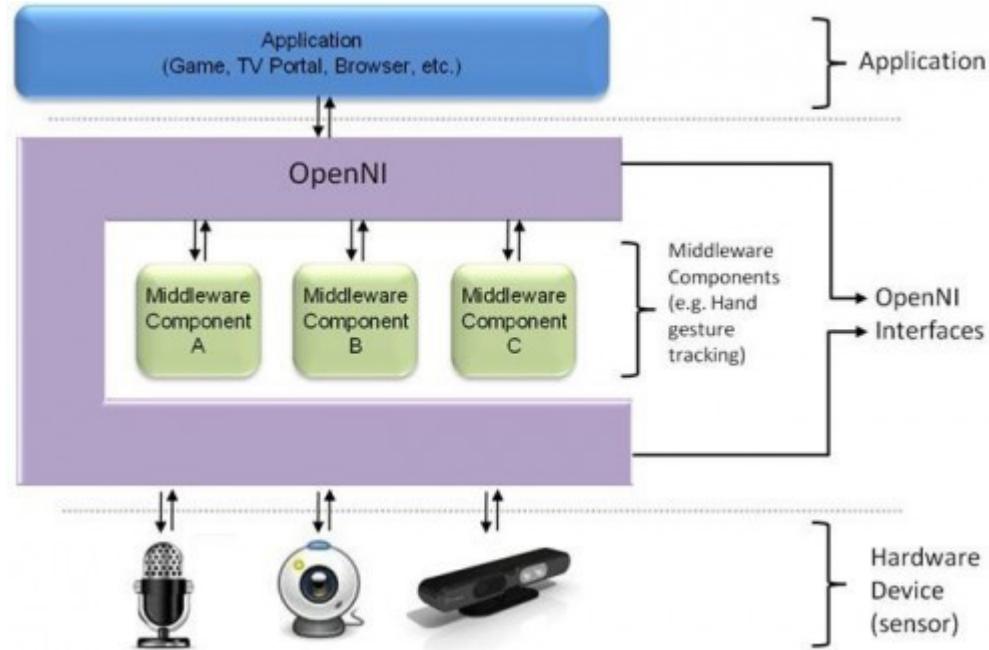
Versión: 1.0

Profesor: Marcos Zúñiga Barraza

Ayudante: Felipe López P.

Introducción

OpenNI permite comunicarse con los sensores de audio, video y sensor de profundidad de Kinect, mientras que proporciona una API que sirve de puente entre el hardware del equipo, NITE Middleware y las aplicaciones e interfaces del S.O. La idea es facilitar el desarrollo de aplicaciones que funcionen con interacción natural, como gestos y movimientos corporales.



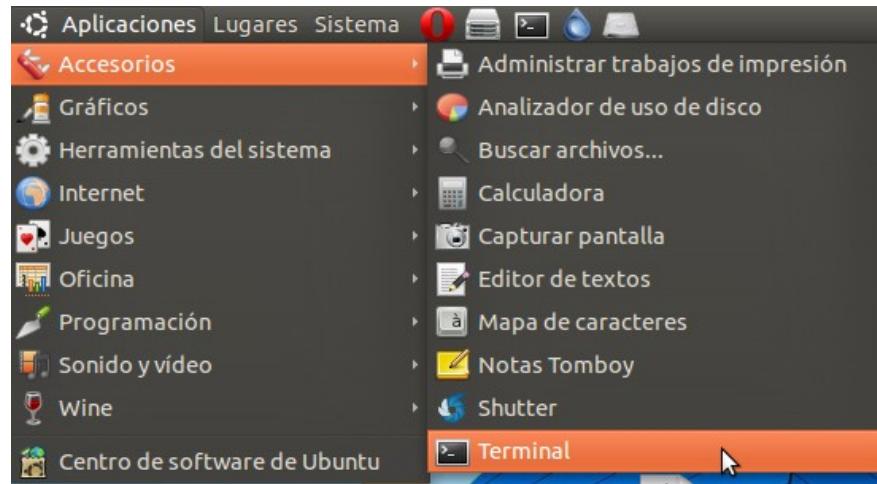
Actualmente OpenNI permite la captura de movimiento en tiempo real, el reconocimiento de gestos con las manos, el uso de comandos de voz y utiliza un analizador de escena que detecta y distingue las figuras en primer plano del fondo.

Instalación

En esta sección se mostrarán los pasos a seguir para instalar el driver oficial liberado para Kinect en diversos sistemas operativos. Las pruebas se realizaron en Ubuntu 10.10 y Windows XP / 7, sin embargo, además se muestran los pasos a seguir en Gentoo Linux.

Debian/Ubuntu 10.10

Para la instalación será necesaria una consola o terminal. Para abrir una vamos a Aplicaciones > Accesorios > Terminal.



En primer lugar, se instalan todos los paquetes necesarios para la instalación.

```
sudo apt-get install git-core cmake libglut3-dev pkg-config build-essential libxmu-dev libxi-dev libusb-1.0-0-dev doxygen graphviz
```

```
● felipe@felipe-ubuntu: ~
Archivo Editar Ver Buscar Terminal Ayuda
felipe@felipe-ubuntu:~$ sudo apt-get install git-core cmake libglut3-dev pkg-con
fig build-essential libxmu-dev libxi-dev libusb-1.0-0-dev doxygen graphviz
system.

Here's the installation process.
```

Se crea un directorio con nombre **kinect** en home:

mkdir ~/kinect

Dentro del directorio **kinect**, se descarga el repositorio de openNI.

cd ~/kinect

git clone https://github.com/OpenNI/OpenNI.git

Nota: la nueva versión beta de OpenNi trae un Wrapper para programar en .NET (C#), si quieres instalar ésta versión solo reemplaza la última línea por:

git clone https://github.com/OpenNI/OpenNI.git -b unstable

```
● felipe@felipe-ubuntu: ~/kinect
Archivo Editar Ver Buscar Terminal Ayuda
mktexlsr: Done. In Spanish so I have documented the
Building format(s) --refresh.
      This may take some time... done.
Procesando disparadores para tex-common ...
Running mktexlsr. This may take some time... done.
texlive-base is not ready, delaying updmap-sys call
Configurando texlive-base (2009-10) ...
Running mktexlsr. This may take some time... done.
Building format(s) --all --cnffile /etc/texmf/fmt.d/10texlive-base.cnf.
      This may take some time... done.
Procesando disparadores para tex-common ...
Running updmap-sys. This may take some time... done.
Building e-tex based formats --byhyphen /var/lib/texmf/tex/generic/config/language.def.
      This may take some time... done.
Configurando texlive-extra-utils (2009-9ubuntul) ...
Configurando texlive-luatex (2009-10) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
Procesando disparadores para tex-common ...
Running mktexlsr. This may take some time... done.
felipe@felipe-ubuntu:~$ mkdir ~/kinect
felipe@felipe-ubuntu:~$ cd ~/kinect
felipe@felipe-ubuntu:~/kinect$ git clone https://github.com/OpenNI/OpenNI.git
```

Los ejemplos necesitan ser compilados, para esto se realiza:

cd OpenNI/Platform/Linux-x86/Build

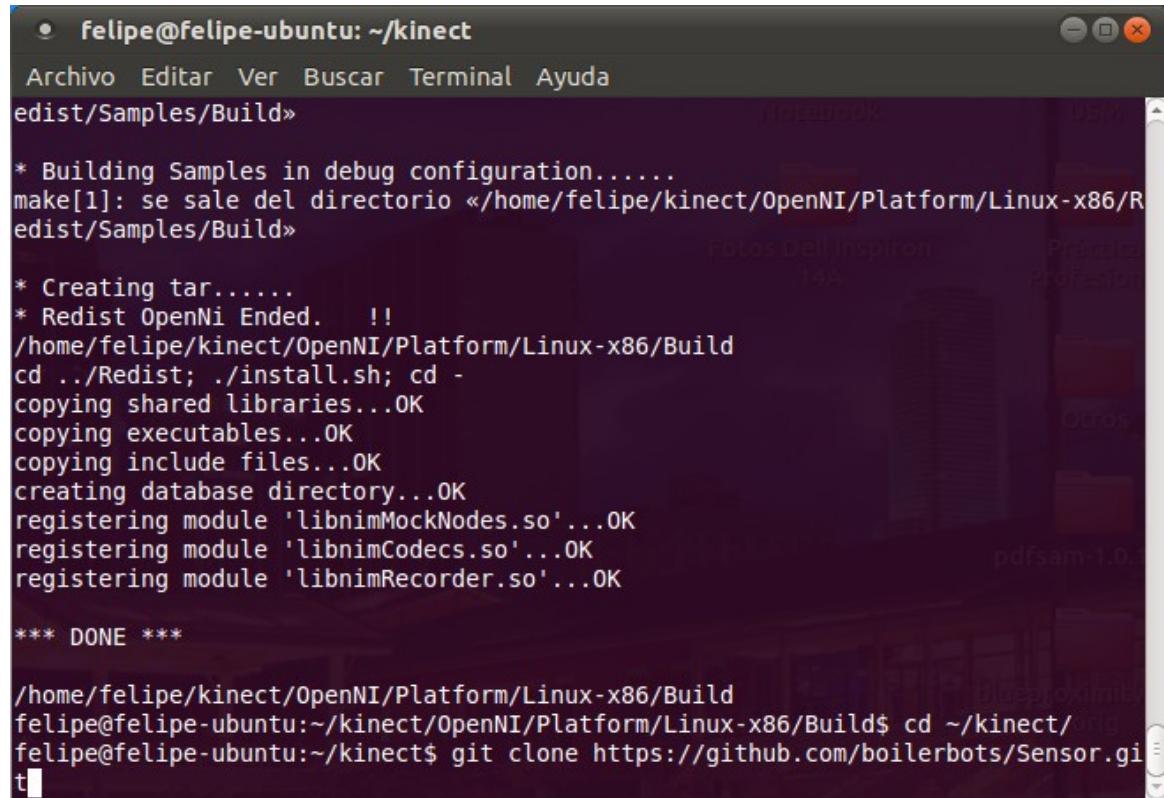
make && sudo make install

```
● felipe@felipe-ubuntu: ~/kinect/OpenNI/Platform/Linux-x86/Build
Archivo Editar Ver Buscar Terminal Ayuda
      This may take some time... done.
Procesando disparadores para tex-common ...
Running updmap-sys. This may take some time... done.
Building e-tex based formats --byhyphen /var/lib/texmf/tex/generic/config/language.def.
      This may take some time... done.
Configurando texlive-extra-utils (2009-9ubuntul) ...
Configurando texlive-luatex (2009-10) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
Procesando disparadores para tex-common ...
Running mktexlsr. This may take some time... done.
felipe@felipe-ubuntu:~$ mkdir ~/kinect
felipe@felipe-ubuntu:~$ cd ~/kinect
felipe@felipe-ubuntu:~/kinect$ git clone https://github.com/OpenNI/OpenNI.git
Initialized empty Git repository in /home/felipe/kinect/OpenNI/.git/
remote: Counting objects: 951, done.
remote: Compressing objects: 100% (555/555), done.
remote: Total 951 (delta 418), reused 784 (delta 305)
Receiving objects: 100% (951/951), 26.81 MiB | 963 KiB/s, done.
Resolving deltas: 100% (418/418), done.
felipe@felipe-ubuntu:~/kinect$ cd OpenNI/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/OpenNI/Platform/Linux-x86/Build$ make && sudo make
install
```

Se vuelve al directorio kinect y se descarga del repositorio git el sensor:

cd ~/kinect/

git clone https://github.com/boilerbots/Sensor.git



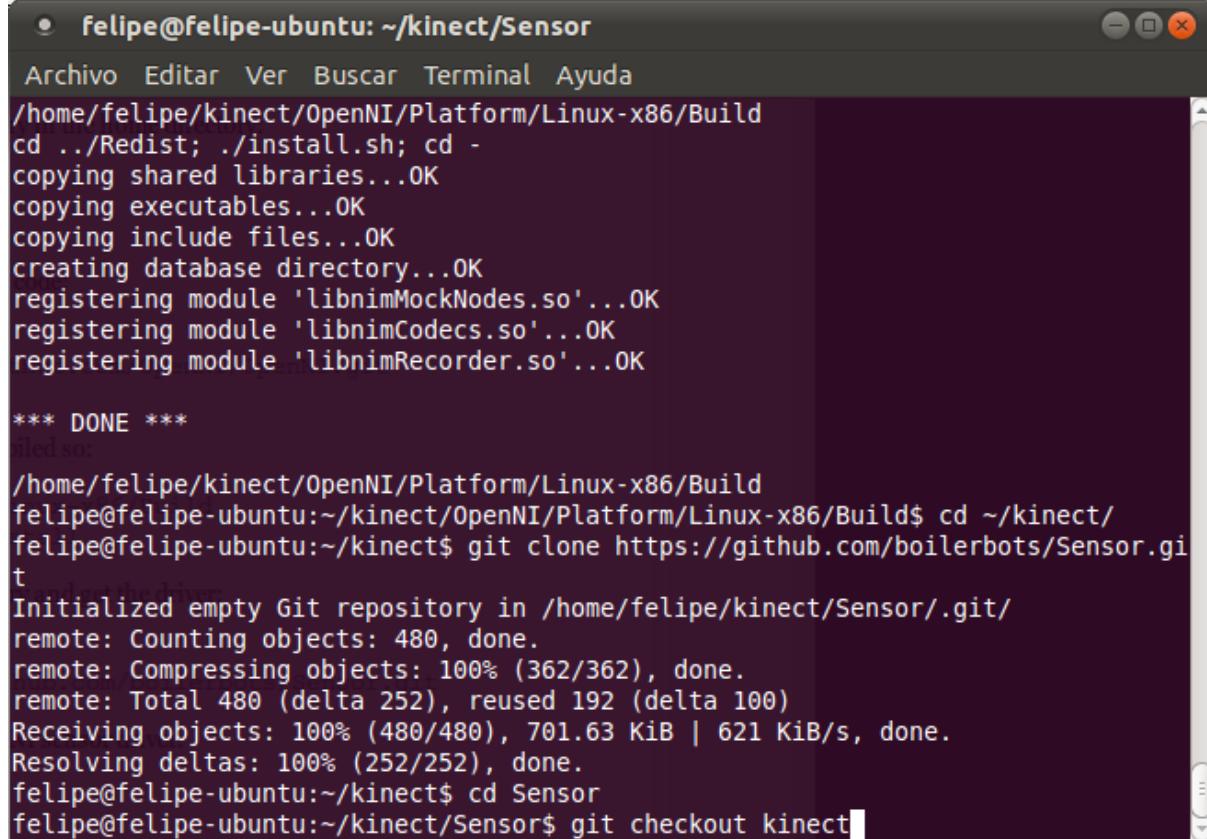
```
felipe@felipe-ubuntu: ~/kinect
Archivo Editar Ver Buscar Terminal Ayuda
edist/Samples/Build»
* Building Samples in debug configuration..... .
make[1]: se sale del directorio «/home/felipe/kinect/OpenNI/Platform/Linux-x86/R
edist/Samples/Build»
* Creating tar..... .
* Redist OpenNi Ended.  !!
/home/felipe/kinect/OpenNI/Platform/Linux-x86/Build
cd ..../Redist; ./install.sh; cd -
copying shared libraries...OK
copying executables...OK
copying include files...OK
creating database directory...OK
registering module 'libnimMockNodes.so'...OK
registering module 'libnimCodecs.so'...OK
registering module 'libnimRecorder.so'...OK
*** DONE ***
/home/felipe/kinect/OpenNI/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/OpenNI/Platform/Linux-x86/Build$ cd ~/kinect/
felipe@felipe-ubuntu:~/kinect$ git clone https://github.com/boilerbots/Sensor.git
```

Se accede a la carpeta Sensor:

cd Sensor

Y se salta a la rama de Kinect:

git checkout kinect



```
felipe@felipe-ubuntu: ~/kinect/Sensor
Archivo Editar Ver Buscar Terminal Ayuda
/home/felipe/kinect/OpenNI/Platform/Linux-x86/Build
cd ..../Redist; ./install.sh; cd -
copying shared libraries...OK
copying executables...OK
copying include files...OK
creating database directory...OK
registering module 'libnimMockNodes.so'...OK
registering module 'libnimCodecs.so'...OK
registering module 'libnimRecorder.so'...OK
*** DONE ***
/home/felipe/kinect/OpenNI/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/OpenNI/Platform/Linux-x86/Build$ cd ~/kinect/
felipe@felipe-ubuntu:~/kinect$ git clone https://github.com/boilerbots/Sensor.git
Initialized empty Git repository in /home/felipe/kinect/Sensor/.git/
remote: Counting objects: 480, done.
remote: Compressing objects: 100% (362/362), done.
remote: Total 480 (delta 252), reused 192 (delta 100)
Receiving objects: 100% (480/480), 701.63 KiB | 621 KiB/s, done.
Resolving deltas: 100% (252/252), done.
felipe@felipe-ubuntu:~/kinect$ cd Sensor
felipe@felipe-ubuntu:~/kinect/Sensor$ git checkout kinect
```

Vamos a la carpeta de las fuentes y compilamos:

cd Platform/Linux-x86/Build

make && sudo make install

```

felipe@felipe-ubuntu: ~/kinect/Sensor/Platform/Linux-x86/Build
Archivo Editar Ver Buscar Terminal Ayuda
creating database directory...OK
registering module 'libnimMockNodes.so'...OK
registering module 'libnimCodecs.so'...OK
registering module 'libnimRecorder.so'...OK

*** DONE ***

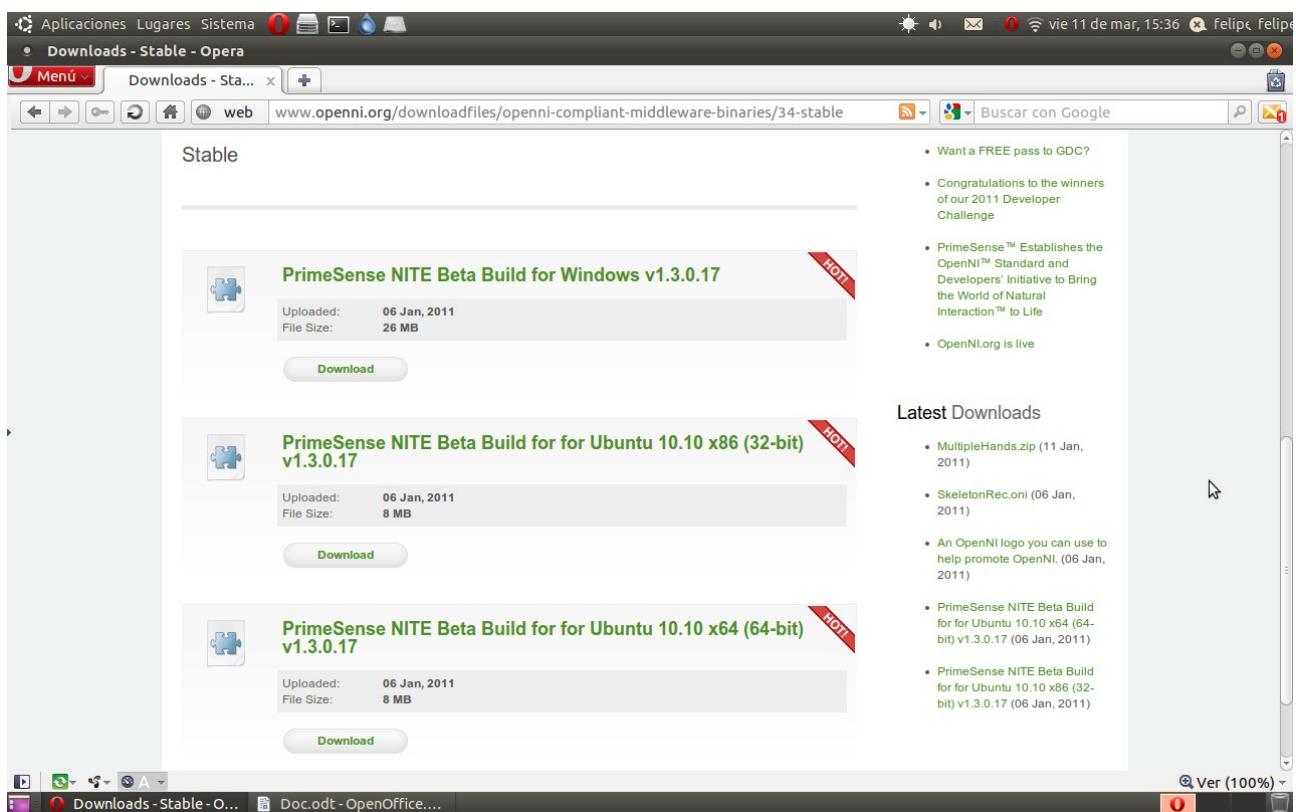
/home/felipe/kinect/OpenNI/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/OpenNI/Platform/Linux-x86/Build$ cd ~/kinect/
felipe@felipe-ubuntu:~/kinect$ git clone https://github.com/boilerbots/Sensor.git
Initialized empty Git repository in /home/felipe/kinect/Sensor/.git/
remote: Counting objects: 480, done.
remote: Compressing objects: 100% (362/362), done.
remote: Total 480 (delta 252), reused 192 (delta 100)
Receiving objects: 100% (480/480), 701.63 KiB | 621 KiB/s, done.
Resolving deltas: 100% (252/252), done.
felipe@felipe-ubuntu:~/kinect$ cd Sensor
felipe@felipe-ubuntu:~/kinect/Sensor$ git checkout kinect
Branch kinect set up to track remote branch kinect from origin.
Switched to a new branch 'kinect'
felipe@felipe-ubuntu:~/kinect/Sensor$ cd Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/Sensor/Platform/Linux-x86/Build$ make && sudo make install

```

Se vuelve a la carpeta kinect:

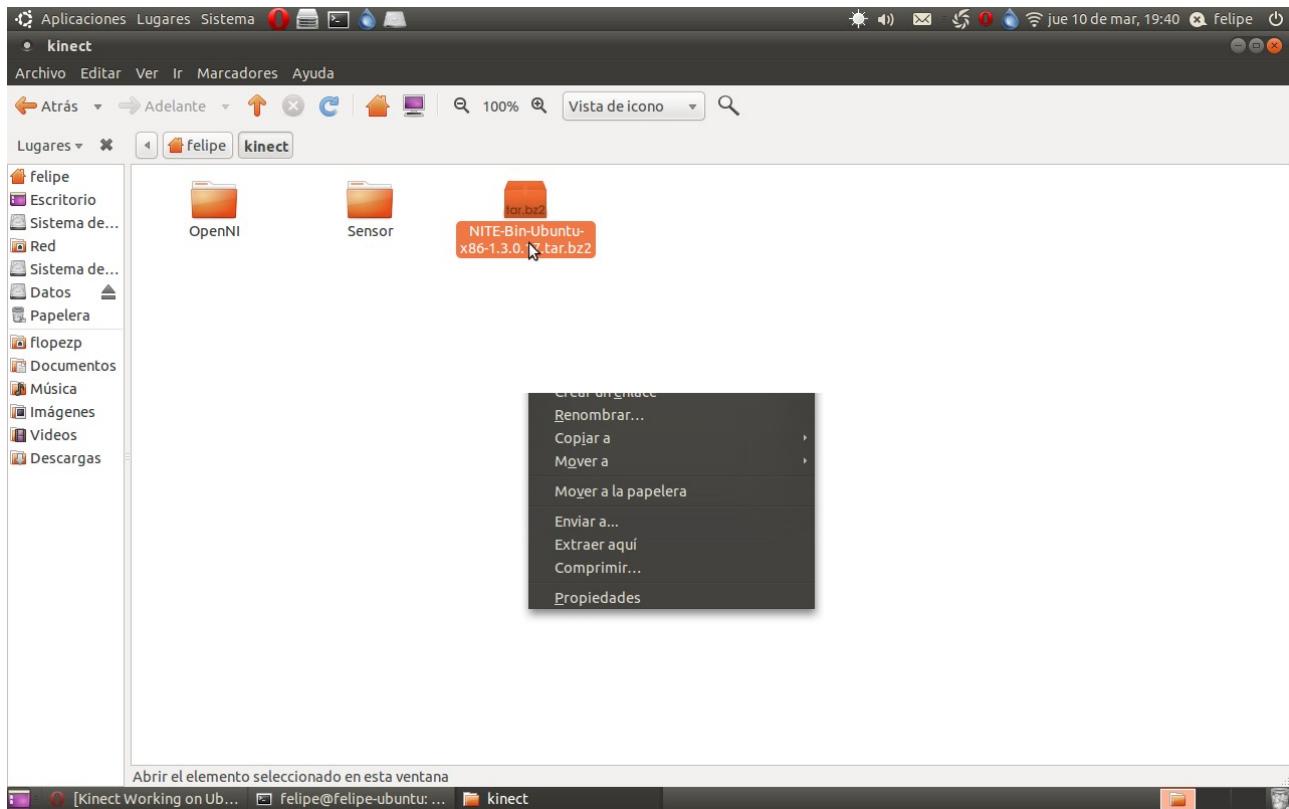
`cd ~/kinect/`

Ahora se tiene que descargar NITE desde [aquí](#).



Se escoge entre 32 o 64 bits según la arquitectura de nuestra distribución.

Una vez descargado, se descomprime en la carpeta kinect.



Se accede a la carpeta de las fuentes:

cd ~/kinect/Nite-1.3.0.17/Data

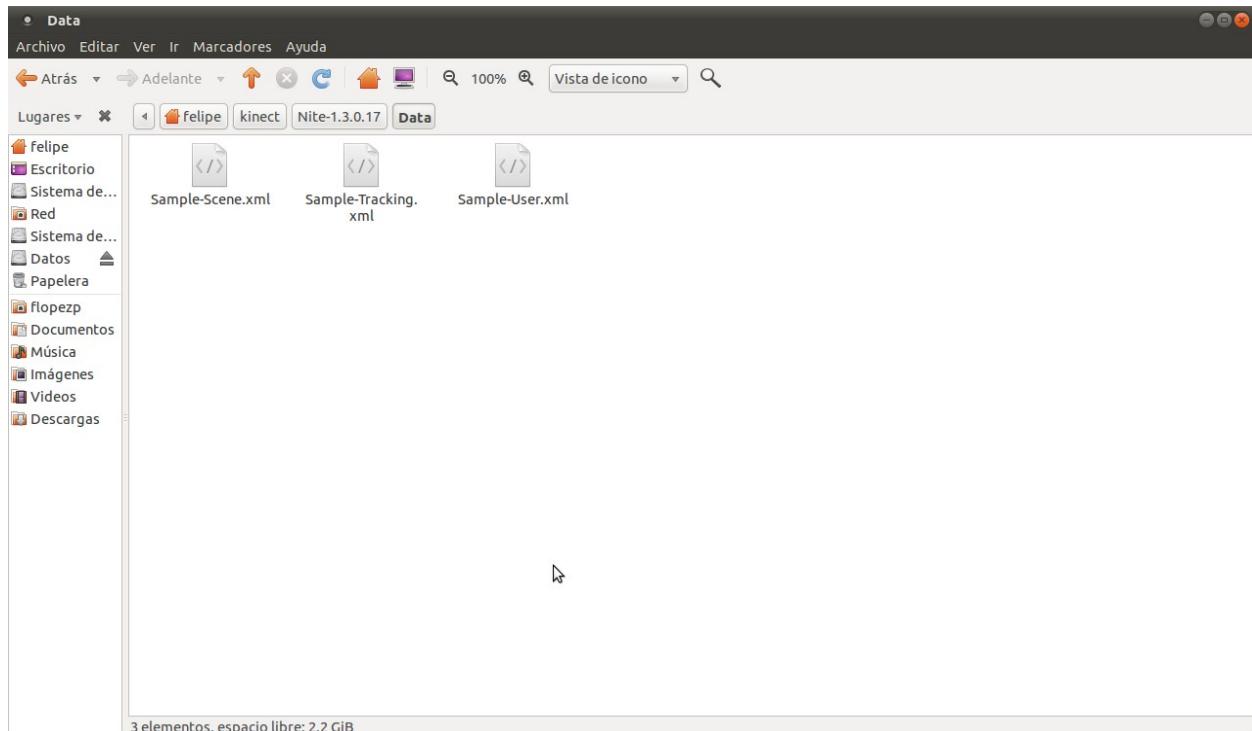
```
felipe@felipe-ubuntu: ~/kinect/Nite-1.3.0.17/Data
Archivo Editar Ver Buscar Terminal Ayuda
make[2]: No se hace nada para «all».
make[2]: se sale del directorio «/home/felipe/kinect/Sensor/Platform/Linux-x86/B
uild/XnDeviceFile»
make[1]: se sale del directorio «/home/felipe/kinect/Sensor/Platform/Linux-x86/B
uild»
/home/felipe/kinect/Sensor/Platform/Linux-x86/CreateRedist
/home/felipe/kinect/Sensor/Platform/Linux-x86/Build
cd .../Redist; ./install.sh; cd -
creating config dir /usr/etc/primesense...OK
copying shared libraries...OK
copying executables...OK
registering module 'libXnDeviceSensorV2.so' with OpenNI...OK
registering module 'libXnDeviceFile.so' with OpenNI...OK
copying server config file...OK
setting uid of server...OK
creating server logs dir...OK
installing usb rules...OK

*** DONE ***

/home/felipe/kinect/Sensor/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/Sensor/Platform/Linux-x86/Build$ cd ~/kinect/
felipe@felipe-ubuntu:~/kinect$ cd ~/kinect/Nite-1.3.0.17/Data
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17/Data$
```

(Atención a la versión, ya que variará el nombre de la carpeta)

En la carpeta se encuentran 3 archivos (**Sample-User.xml**, **Sample-Scene.xml** y **Sample-Tracking.xml**), con permisos solo de lectura, los cuales contienen información de la licencia.



Para poder modificarlos se le asignan permiso de escritura:

chmod a+w *

Se abre el fichero **Sample-User.xml**, puede ser con gedit, nano, o con el editor que prefieran. En este ejemplo se hizo mediante gedit.

Se deben remplazar las siguientes líneas:

```
<License vendor="PrimeSense" key="" />
```

A screenshot of the gedit text editor. The title bar says 'Sample-Scene.xml (~/kinect/Nite-1.3.0.17/Data) - gedit'. The editor interface includes a toolbar with file operations like Abrir, Guardar, Deshacer, and Herramientas. The main pane displays an XML configuration file named 'Sample-Scene.xml'. The XML code includes sections for 'OpenNI', 'Licenses', 'Log', 'Masks', 'Dumps', 'ProductionNodes', and 'Node'. A specific line of code is highlighted: '<License vendor="PrimeSense" key="insert key here"/>'. The status bar at the bottom shows 'XML', 'Ancho de la tabulación: 8', 'Ln 3, Col 69', and 'INS'.

Por:

```
<License vendor="PrimeSense" key="0KOIk2JeIBYCIPWVnMoRKn5cdY4=" />
```

```
<OpenNI>
    <Licenses>
        <License vendor="PrimeSense" key="0KOIk2JeIBYClPWVnMoRKn5cdY4="/>
    </Licenses>
    <Log writeToConsole="true" writeToFile="false">
        <!-- 0 - Verbose, 1 - Info, 2 - Warning, 3 - Error (default) -->
        <LogLevel value="3"/>
        <Masks>
            <Mask name="ALL" on="false"/>
        </Masks>
        <Dumps>
        </Dumps>
    </Log>
    <ProductionNodes>
        <Node type="Depth">
            <Configuration>
                <MapOutputMode xRes="320" yRes="240" FPS="30"/>
                <Mirror on="true"/>
            </Configuration>
        </Node>
        <Node type="Scene" />
    </ProductionNodes>
</OpenNI>
```

Ahora se busca la línea donde sale **MapOutputMode** y se reemplaza por:

```
<MapOutputMode xRes="640" yRes="480" FPS="30"/>
```

```
<OpenNI>
    <Licenses>
        <License vendor="PrimeSense" key="0KOIk2JeIBYClPWVnMoRKn5cdY4="/>
    </Licenses>
    <Log writeToConsole="true" writeToFile="false">
        <!-- 0 - Verbose, 1 - Info, 2 - Warning, 3 - Error (default) -->
        <LogLevel value="3"/>
        <Masks>
            <Mask name="ALL" on="false"/>
        </Masks>
        <Dumps>
        </Dumps>
    </Log>
    <ProductionNodes>
        <Node type="Depth">
            <Configuration>
                <MapOutputMode xRes="320" yRes="240" FPS="30"/>
                <Mirror on="true"/>
            </Configuration>
        </Node>
        <Node type="Scene" />
    </ProductionNodes>
</OpenNI>
```

```
<OpenNI>
    <Licenses>
        <License vendor="PrimeSense" key="0KOIk2JeIBYClPWVnMoRKn5cdY4="/>
    </Licenses>
    <Log writeToConsole="true" writeToFile="false">
        <!-- 0 - Verbose, 1 - Info, 2 - Warning, 3 - Error (default) -->
        <LogLevel value="3"/>
        <Masks>
            <Mask name="ALL" on="false"/>
        </Masks>
        <Dumps>
        </Dumps>
    </Log>
    <ProductionNodes>
        <Node type="Depth">
            <Configuration>
                <MapOutputMode xRes="640" yRes="480" FPS="30"/>
                <Mirror on="true"/>
            </Configuration>
        </Node>
        <Node type="Scene" />
    </ProductionNodes>
</OpenNI>
```

Con esto se aumenta la resolución de la cámara, que por defecto viene configurada en 320x240 píxeles.

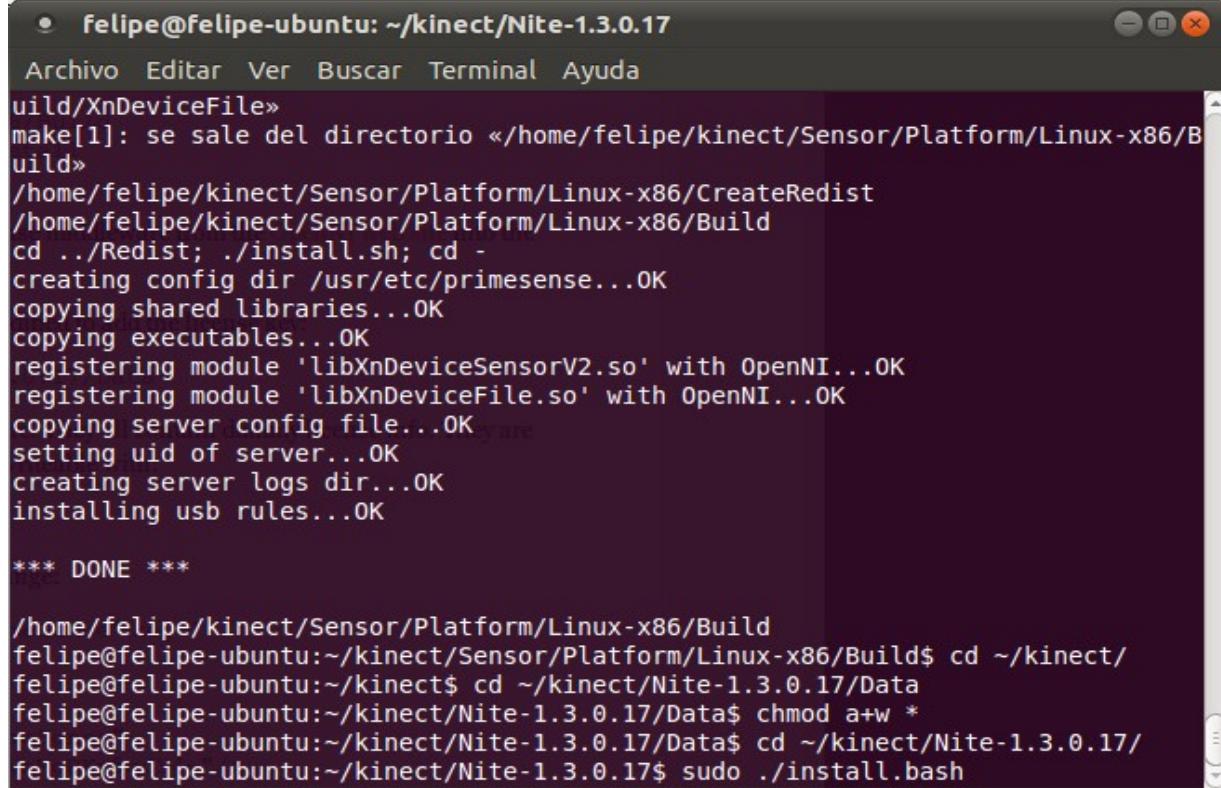
Se repiten estas modificaciones con los archivos **Sample-Scene.xml** y **Sample-Tracking.xml**.

Finalmente nos ubicamos en la carpeta donde está el script del instalador:

`cd ~/kinect/NITE/Nite-1.3.0.17/`

Y se ejecuta:

`sudo ./install.bash`



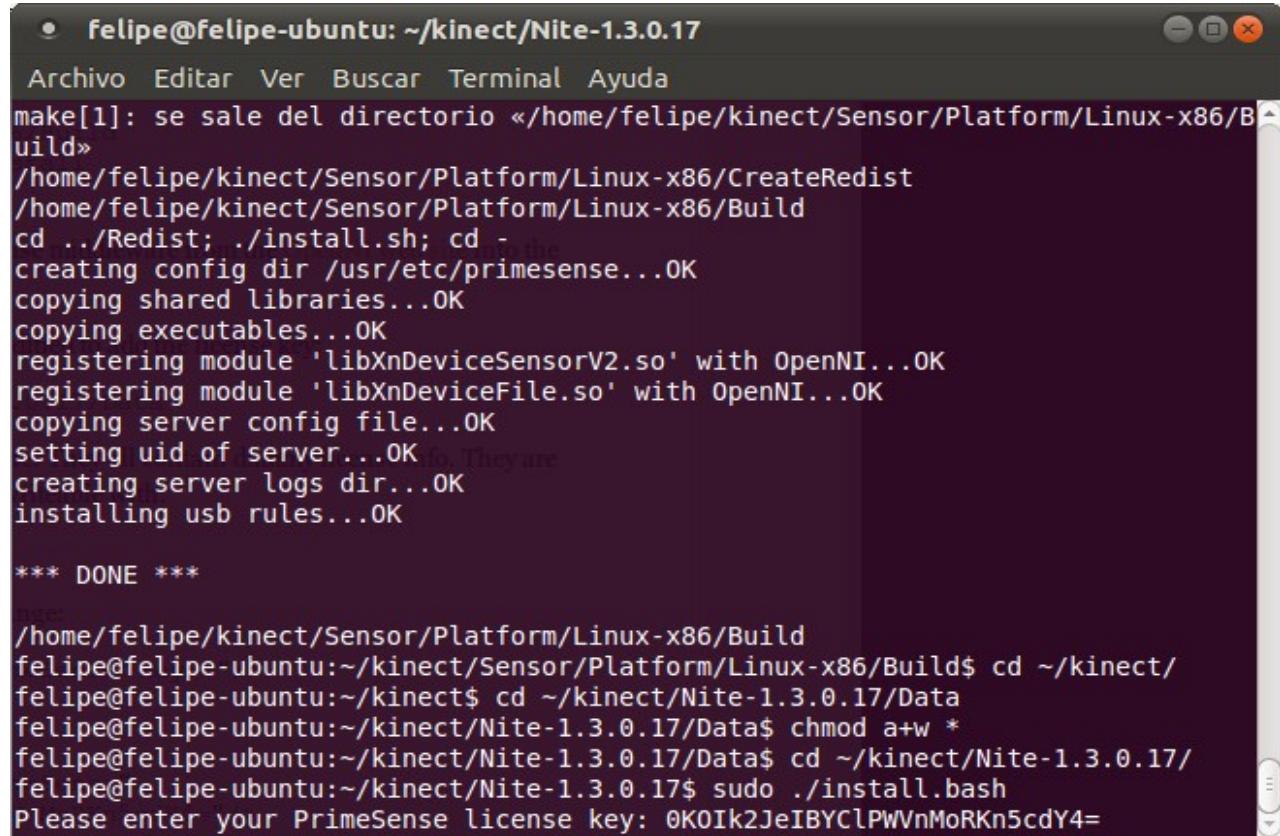
```
felipe@felipe-ubuntu: ~/kinect/Nite-1.3.0.17
Archivo Editar Ver Buscar Terminal Ayuda
uild/XnDeviceFile»
make[1]: se sale del directorio «/home/felipe/kinect/Sensor/Platform/Linux-x86/B
uild»
/home/felipe/kinect/Sensor/Platform/Linux-x86/CreateRedist
/home/felipe/kinect/Sensor/Platform/Linux-x86/Build
cd ../Redist; ./install.sh; cd -
creating config dir /usr/etc/primesense...OK
copying shared libraries...OK
copying executables...OK
registering module 'libXnDeviceSensorV2.so' with OpenNI...OK
registering module 'libXnDeviceFile.so' with OpenNI...OK
copying server config file...OK
setting uid of server...OK
creating server logs dir...OK
installing usb rules...OK

*** DONE ***

/home/felipe/kinect/Sensor/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/Sensor/Platform/Linux-x86/Build$ cd ~/kinect/
felipe@felipe-ubuntu:~/kinect$ cd ~/kinect/Nite-1.3.0.17/Data
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17/Data$ chmod a+w *
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17/Data$ cd ~/kinect/Nite-1.3.0.17/
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17$ sudo ./install.bash
```

En este paso nos pedirá el Key utilizado anteriormente:

0KOIk2JeIBYCIPWVnMoRKn5cdY4=



```
felipe@felipe-ubuntu: ~/kinect/Nite-1.3.0.17
Archivo Editar Ver Buscar Terminal Ayuda
make[1]: se sale del directorio «/home/felipe/kinect/Sensor/Platform/Linux-x86/B
uild»
/home/felipe/kinect/Sensor/Platform/Linux-x86/CreateRedist
/home/felipe/kinect/Sensor/Platform/Linux-x86/Build
cd ../Redist; ./install.sh; cd -
creating config dir /usr/etc/primesense...OK
copying shared libraries...OK
copying executables...OK
registering module 'libXnDeviceSensorV2.so' with OpenNI...OK
registering module 'libXnDeviceFile.so' with OpenNI...OK
copying server config file...OK
setting uid of server...OK
creating server logs dir...OK
installing usb rules...OK

*** DONE ***

/home/felipe/kinect/Sensor/Platform/Linux-x86/Build
felipe@felipe-ubuntu:~/kinect/Sensor/Platform/Linux-x86/Build$ cd ~/kinect/
felipe@felipe-ubuntu:~/kinect$ cd ~/kinect/Nite-1.3.0.17/Data
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17/Data$ chmod a+w *
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17/Data$ cd ~/kinect/Nite-1.3.0.17/
felipe@felipe-ubuntu:~/kinect/Nite-1.3.0.17$ sudo ./install.bash
Please enter your PrimeSense license key: 0KOIk2JeIBYCIPWVnMoRKn5cdY4=
```

Ahora ya se puede comenzar a probar los distintos ejemplos. Para esto, solo se debe conectar el Kinect al puerto USB y nos ubicamos en una de las siguientes carpetas dependiendo de los ejemplos que deseemos probar.

Ejemplos de prueba instalados por NITE:

cd ~/kinect/Nite-1.3.0.17/Samples/Bin

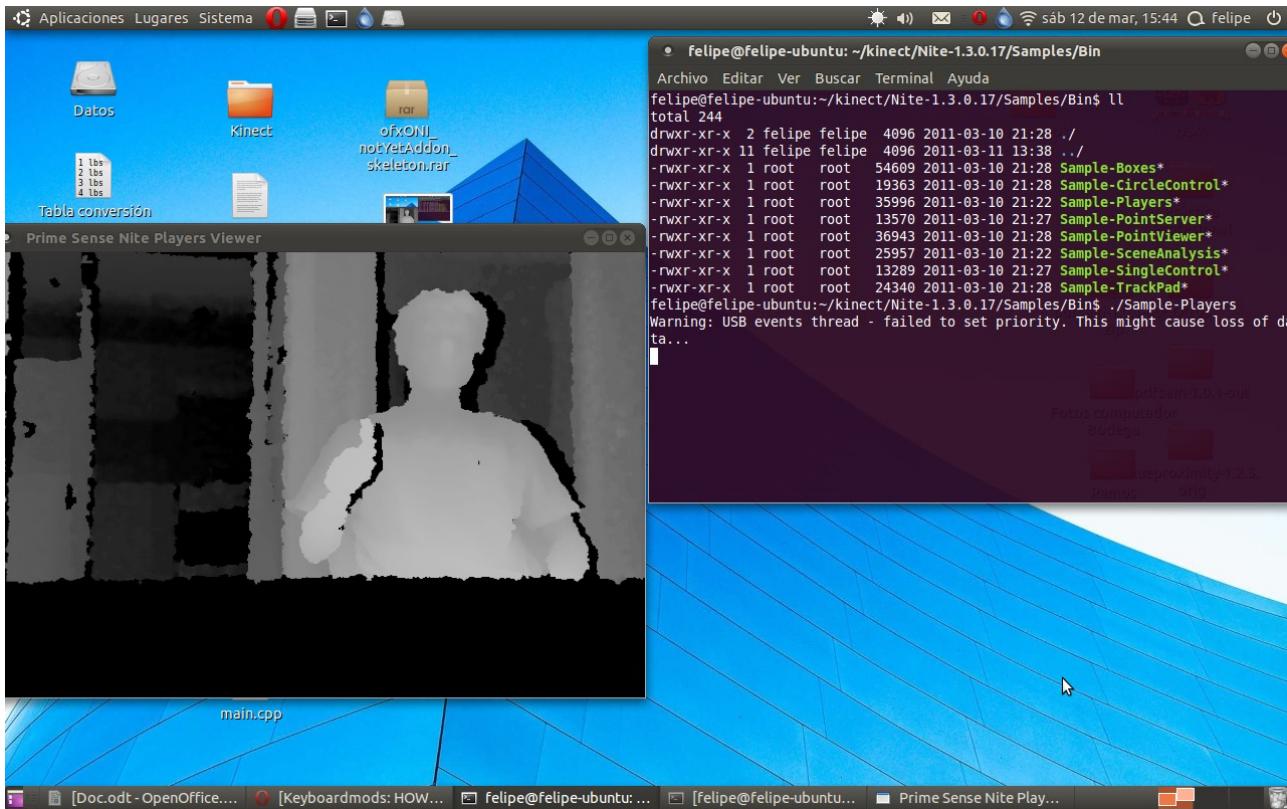
Ejemplos de prueba instalados por OpenNI:

cd ~/kinect/OpenNI/Platform/Linux-x86/Bin/Release

Nota: la posición del kinect y la distancia de separación en la que nos ubicemos es sumamente importante para el correcto funcionamiento de los ejemplos. La distancia recomendada para el funcionamiento óptimo es cercana a 1,5 metros.

Ejecución de Ejemplos

Para ejecutar los ejemplos, nos ubicamos en cualquiera de los dos directorios mencionados y se escribe el nombre del ejemplo a ejecutar anteponiendo ./



En el ejemplo se ejecutó: **./Sample-Players**

Como se observa aparece una advertencia:

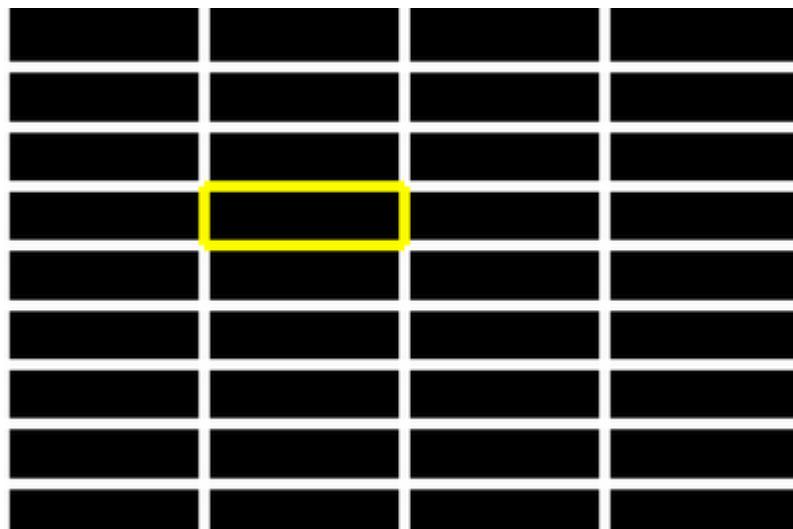
Warning: USB events thread - failed to set priority. This might cause loss of data...

Si deseamos que ésta no aparezca se debe ejecutar el ejemplo anteponiendo **sudo**.

A continuación se muestran algunos ejemplos.

Sample-TrackPad

Este ejemplo trackea la mano y muestra su posición relativa en una grilla desplegada en la pantalla.



Sample-Players

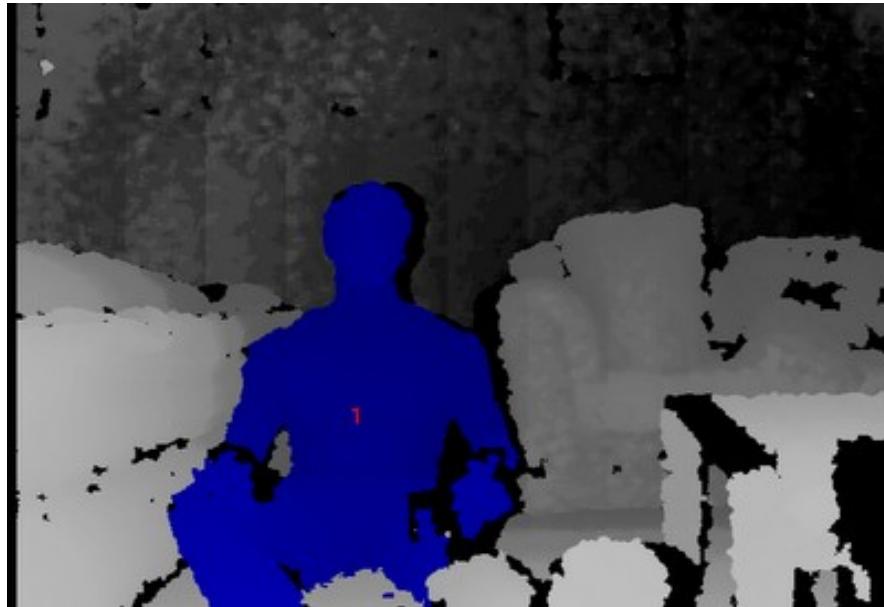
Este ejemplo muestra el trackeo de esqueleto. Luego de comenzar, muévete o camina alrededor hasta que tu cuerpo cambie a color azul.

Ejemplo: Primera persona: azul

Segunda persona: verde

Tercera persona: amarillo

Hasta este punto, en la ventana abierta se debería ver algo así:

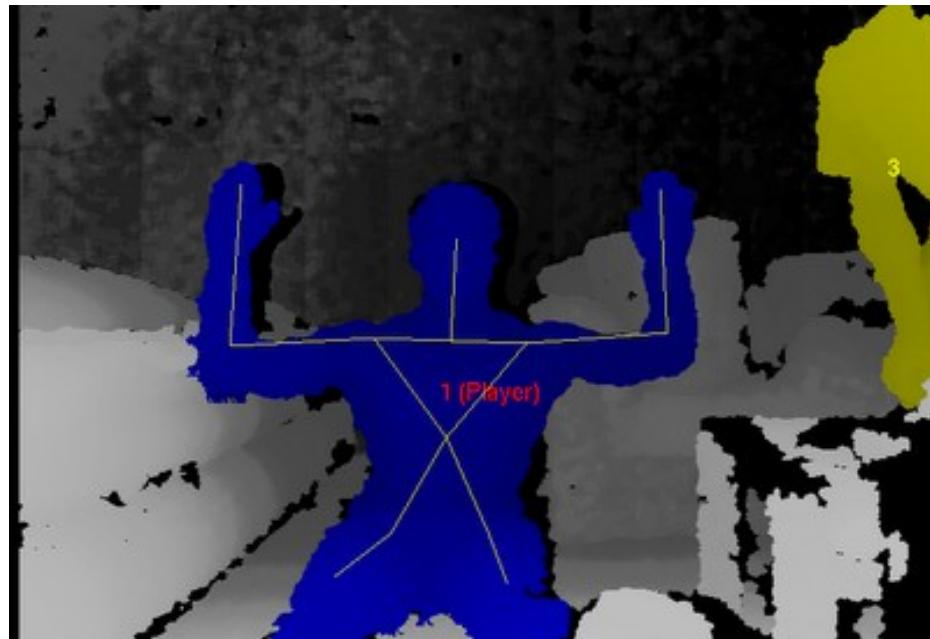


En la consola se debería ver algo por el estilo:

Look for pose

Found pose "Psi" for user 1

En este momento, se nos está pidiendo la pose por defecto utilizada para la calibración, la cual consiste en levantar las manos formando un ángulo de 90 grados con los hombros, como se muestra a continuación:



Con esta pose se calibra para posteriormente realizar el trackeo.

En este punto se debería observar algo parecido en la consola:

Calibration started

Calibration done [1] successfully

Writing 217.596 50 50 78.4388 64.6762

Matching for existing calibration

Read 217.596 50 50 78.4388 64.6762

Como se observa en la imagen, aparecen líneas sobre el cuerpo, es decir, apareció el skeletal tracking del ejemplo.

SamplePointViewer

En este ejemplo se observa el seguimiento de una mano. Para que aparezca el punto en la mano y comience el seguimiento se debe realizar un gesto llevando la mano hacia adelante. Ésta fue la manera más rápida experimentalmente en la cual apareciera el punto.



Esta aplicación viene por defecto para el seguimiento de una sola mano, sin embargo, esto es fácilmente modificable.

Para esto se debe editar el archivo /usr/etc/primesense/XnVHandGenerator/Nite.ini descomentando las dos primeras líneas que comienzan con ";" , quedando así:

```
[HandTrackerManager]  
AllowMultipleHands=1  
TrackAdditionalHands=1
```



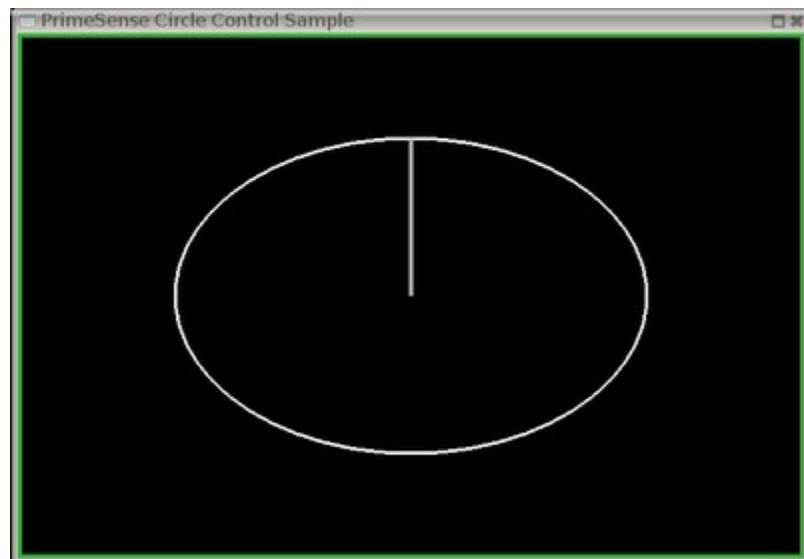
Sample-Boxes

Este ejemplo permite hacer click en una de tres cajas. El movimiento de la mano es trackeado de manera horizontal y dependiendo del contexto; arriba, abajo, izquierda, derecha.



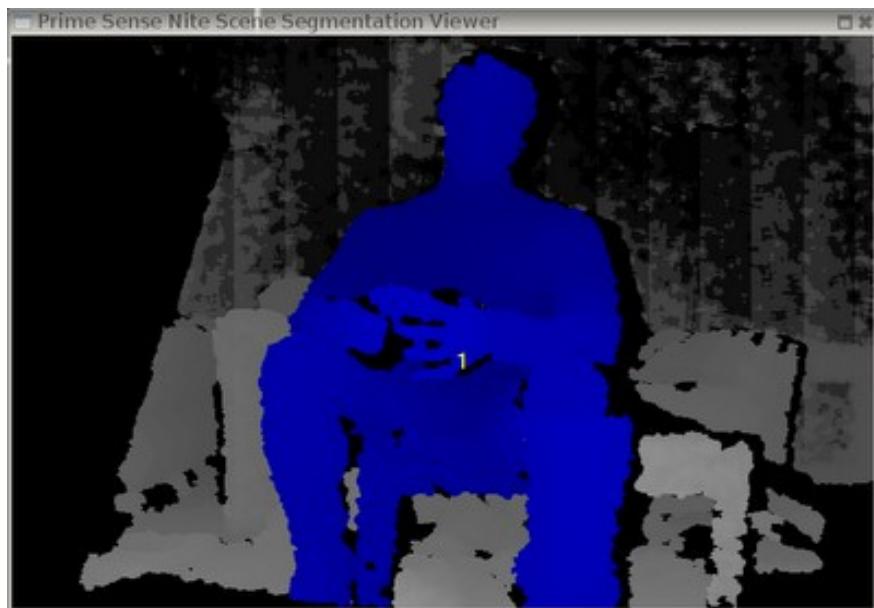
Sample-CircleControl

En este ejemplo se observa un círculo en pantalla el cual tiene una línea en posición 12 horas. Mueve las manos hasta que el borde de la ventana se ponga verde. Una vez que esto haya pasado puedes controlar la manecilla con el movimiento de las manos.



Sample-SceneAnalysis

Este es similar al ejemplo Sample-Players. Se ve la detección del usuario sin las líneas de trackeo en el cuerpo.

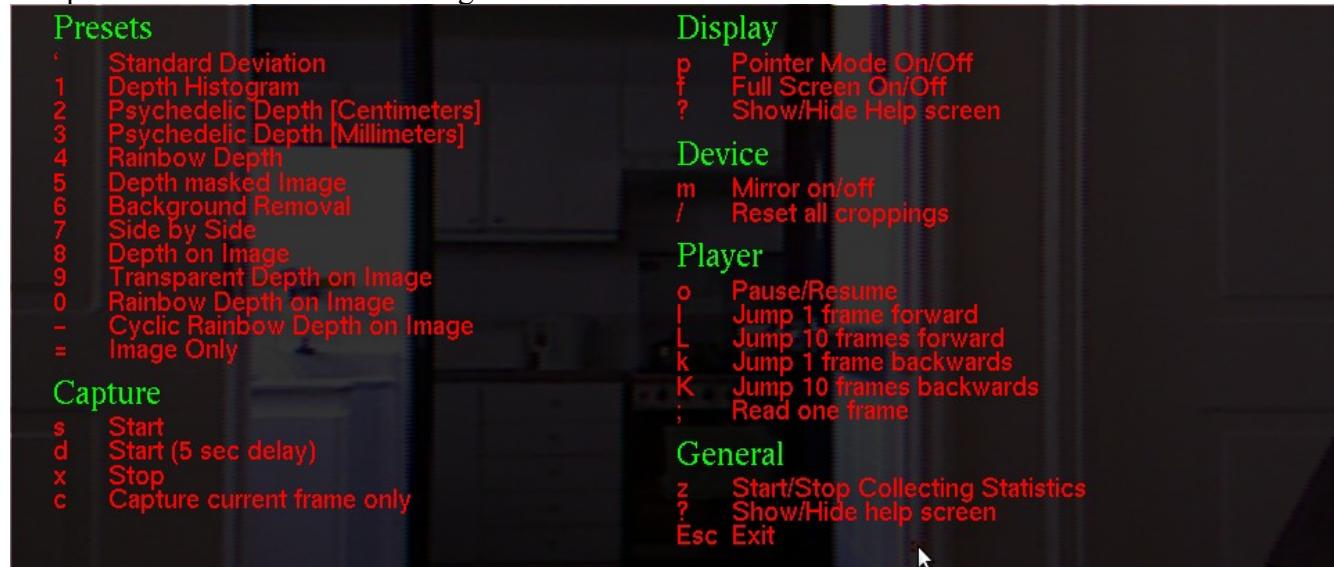


NiViewer

Este ejemplo es uno de los más completos, ya que nos da la opción de observar la cámara en un mapa de profundidad, imagen RGB o un mapa IR. Además, se muestran varias opciones de configuración.



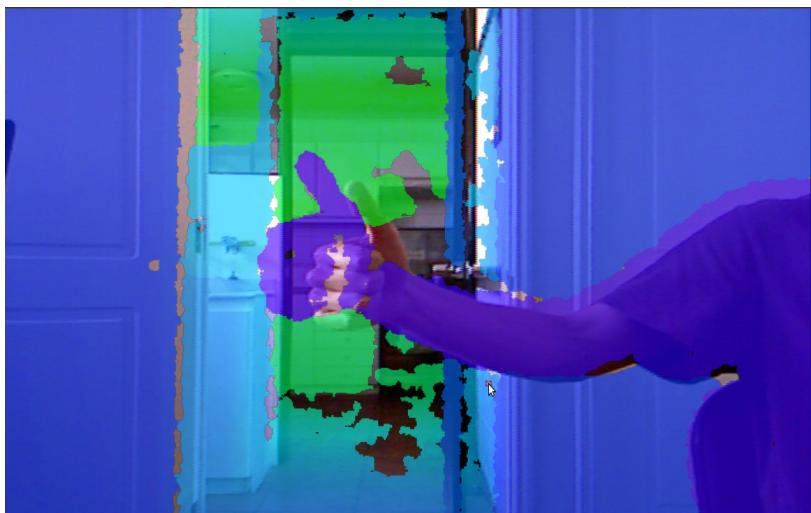
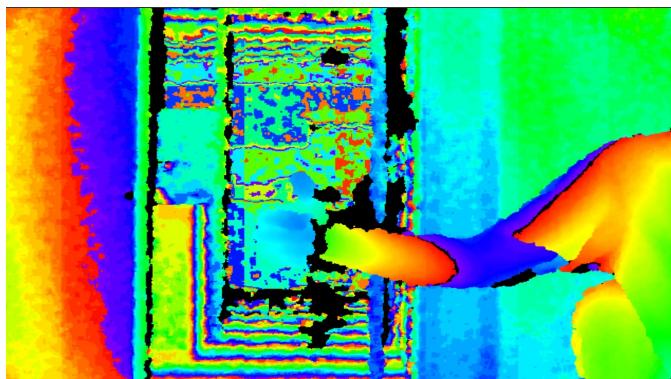
La aplicación se controla con las siguientes teclas:



Key	Description
1	Shows depth only, in histogram mode
2	Shows depth only, in psychedelic mode (centimeters)
3	Shows depth only, in psychedelic mode (millimeters)
4	Shows depth only, in rainbow mode
5	Shows depth masked image, meaning image pixels that don't have depth values are blacked out.
6	Background removal mode
7	Shows depth and image (or IR), side by side.
8	Shows depth on top of image (or IR)
9	Shows transparent depth on top of image (or IR)
0	Shows rainbow depth on top of image (or IR)
=	Shows image (or IR) only
'	Shows depth standard deviation
p	Toggles pointer mode on/off. When pointer mode is on, additional depth info is displayed regarding currently pointed pixel.
f	Toggles Full Screen / Window mode
?	Toggles help screen on/off
m	Toggles mirror on/off
/	Resets all cropping
s	Start recording
d	Start recording in 5 seconds
x	Stop recording
c	Capture current frame to files
z	Start/Stop collecting statistics about depth pixels
o	Pause/Play
i	Seek one frame forward (recordings only)
L	Seek 10 frames forward (recordings only)
k	Seek one frame backwards (recordings only)
K	Seek 10 frames backwards (recordings only)
;	Read one single frame and pause
Esc	Closes the application

Adicionalmente, también se puede utilizar el mouse para modificar la configuración.

Se muestran algunas fotos con las diferentes configuraciones ofrecidas por el ejemplo:



Windows XP / Windows 7

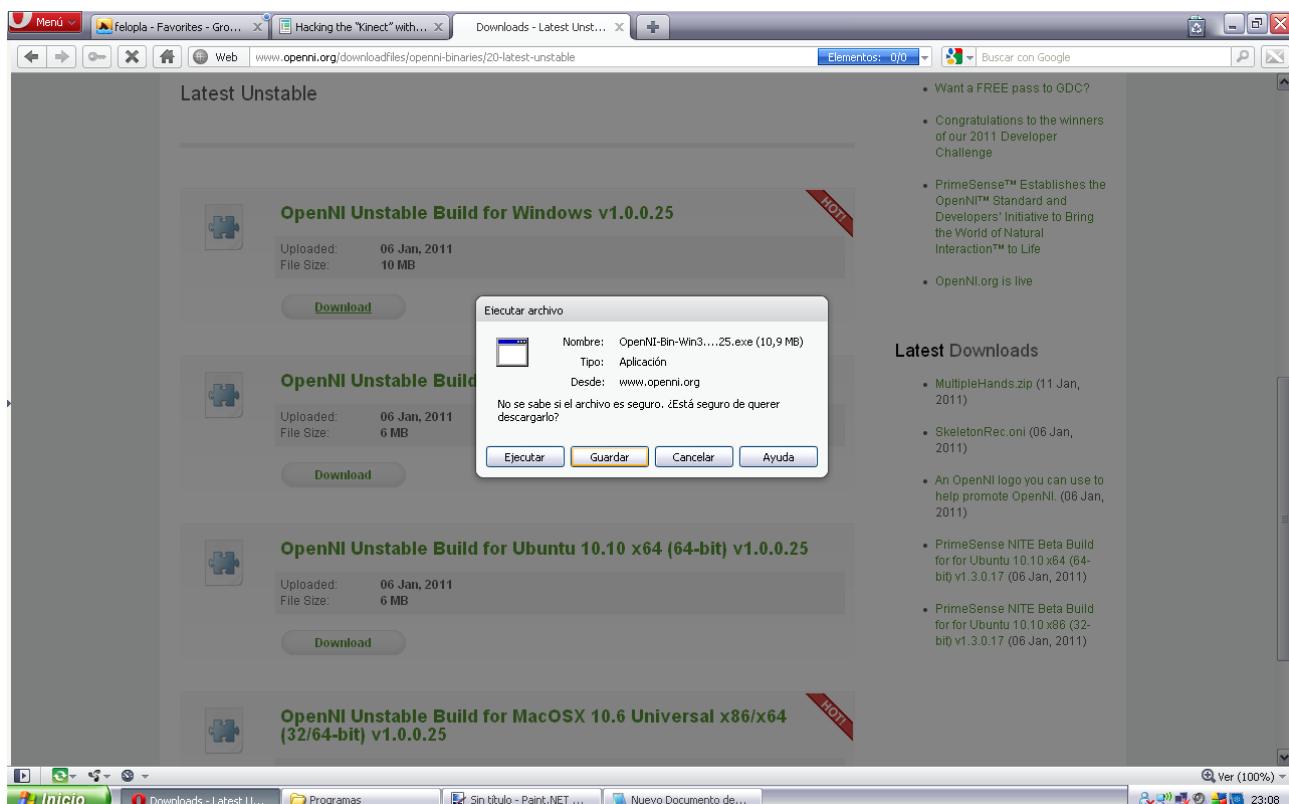
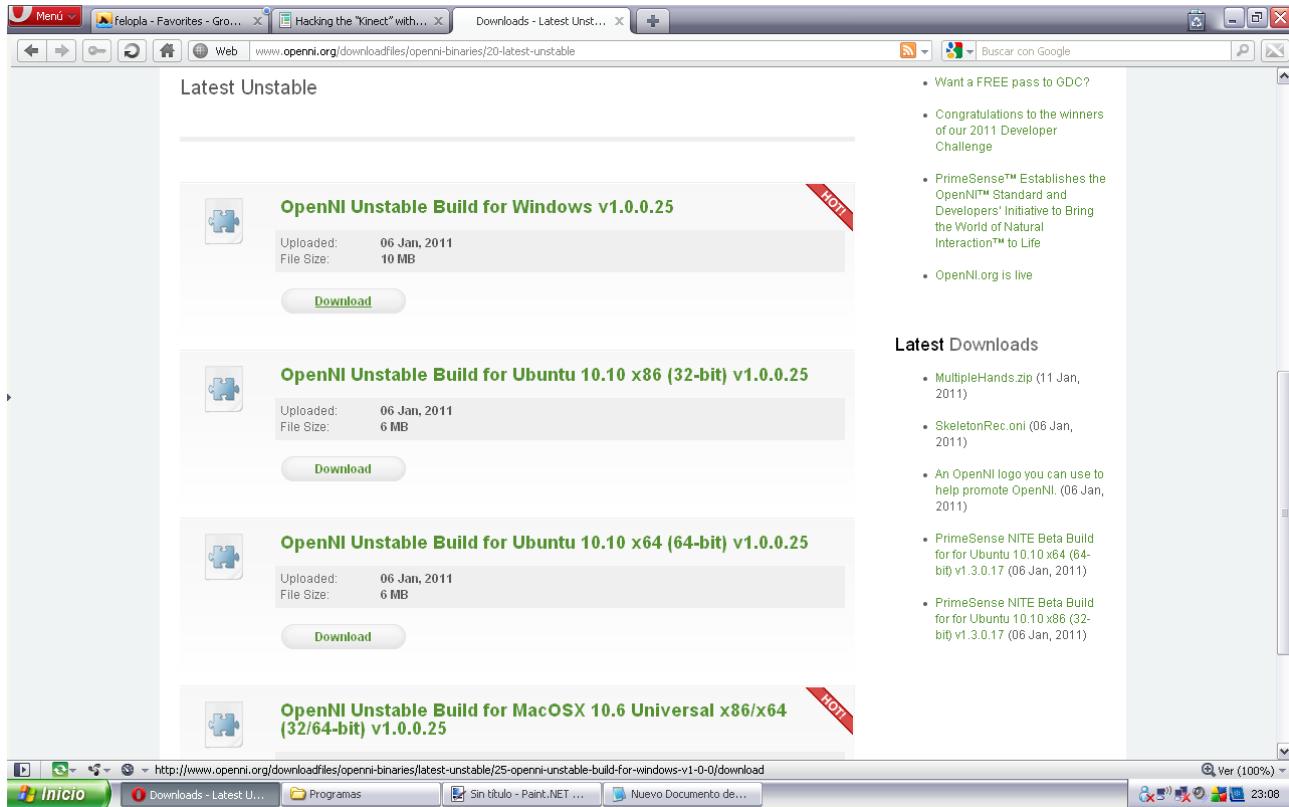
En Windows XP y Windows 7 el Kinect se instala de igual forma.

1. Se descargan los siguientes archivos:

Nota: toda la instalación se realizó con las últimas versiones (inestables), sin embargo, la instalación se realiza de igual forma para versiones estables. Es importante que si escogemos instalar la versión estable, los tres archivos deben tener la versión estable. Análogamente para la versión inestable. En teoría se pueden hacer combinaciones, sin embargo, hay un mayor grado de probabilidad a que finalizada la instalación el Kinect no funcione.

OpenNI:

<http://www.openni.org/downloadfiles/openni-binaries/20-latest-unstable>



Kinect drivers:

<https://github.com/avin2/SensorKinect>

avin2 is using GitHub to share code with you!

GitHub is more than just a place to share code. It's a place to keep tabs on your favorite developers and projects, easily contribute fixes and new features, and visualize what's going on inside your codebase!

Sign Up Now

FREE FOR OPEN SOURCE

avin2 / SensorKinect

Fuente Commits Red Solicitudes de pull (0) Gráficos

Cambiar de rama (2) Cambiar etiquetas (0) Lista de ramas

PrimeSensor Modules for OpenNI — Read more www.primesense.com

HTTP Git sólo lectura https://github.com/avin2/SensorKinect.git Esta URL tiene permisos de **Sólo lectura y Sólo lectura**

Download source, tagged packages and binaries.

Added binary installers for: Windows, Ubuntu 10.10 32/64 bit & Mac OSX 10.6 32/64 bit

commit 0124bd24667cec633ca1
ábol 80dbb74b4fea86678a83
padre 32278743fc9810b8312e

avin2 (author) January 06, 2011

SensorKinect /

Nombre	edad	mensaje	historial
Bin/	January 06, 2011	Added binary installers for: Windows, Ubuntu 10... [avin2]	

Ver (100%)

Downloads for avin2/SensorKinect

DESCARGAR CÓDIGO FUENTE

Download .tar.gz **Download zip** Rama: unstable

Sign Up Now

FREE FOR OPEN SOURCE

avin2 / SensorKinect

Fuente Commits Red Solicitudes de pull (0) Gráficos

Cambiar de rama (2) Cambiar etiquetas (0) Lista de ramas

PrimeSensor Modules for OpenNI — Read more www.primesense.com

HTTP GR sólo lectura https://github.com/avin2/SensorKinect.git Esta URL tiene permisos de **Sólo lectura y Sólo lectura**

Added binary installers for: Windows, Ubuntu 10.10 32/64 bit & Mac OSX 10.6 32/64 bit

commit 0124bd24667cec633ca1
ábol 80dbb74b4fea86678a83
padre 32278743fc9810b8312e

avin2 (author) January 06, 2011

SensorKinect /

Nombre	edad	mensaje	historial
Bin/	January 06, 2011	Added binary installers for: Windows, Ubuntu 10... [avin2]	

Ver (100%)

Downloads for avin2/SensorKinect

DESCARGAR CÓDIGO FUENTE

Download .tar.gz **Download zip** Rama: unstable

Sign Up Now

FREE FOR OPEN SOURCE

avin2 / SensorKinect

Fuente Commits Red Solicitudes de pull (0) Gráficos

Cambiar de rama (2) Cambiar etiquetas (0) Lista de ramas

PrimeSensor Modules for OpenNI — Read more www.primesense.com

HTTP Git sólo lectura https://github.com/avin2/SensorKinect.git Esta URL tiene permisos de **Sólo lectura y Sólo lectura**

Added binary installers for: Windows, Ubuntu 10.10 32/64 bit & Mac OSX 10.6 32/64 bit

commit 0124bd24667cec633ca1
ábol 80dbb74b4fea86678a83
padre 32278743fc9810b8312e

avin2 (author) January 06, 2011

SensorKinect /

Nombre	edad	mensaje	historial
Bin/	January 06, 2011	Added binary installers for: Windows, Ubuntu 10... [avin2]	

Ver (100%)

NITE:

<http://www.openni.org/downloadfiles/openni-compliant-middleware-binaries/33-latest-unstable>

Latest Unstable

PrimeSense NITE Unstable Build for Windows v1.3.0.18

Uploaded: 06 Jan, 2011
File Size: 10 MB

Download

PrimeSense NITE Unstable Build for Ubuntu 10.10 x86 (32-bit) v1.3.0.18

Uploaded: 06 Jan, 2011
File Size: 2 MB

Download

PrimeSense NITE Unstable Build for Ubuntu 10.10 x64 (64-bit) v1.3.0.18

Uploaded: 06 Jan, 2011
File Size: 2 MB

Download

PrimeSense NITE Unstable Build for Mac OSX 10.6 Universal v1.3.0.18

Latest Downloads

- Want a FREE pass to GDC?
- Congratulations to the winners of our 2011 Developer Challenge
- PrimeSense™ Establishes the OpenNI™ Standard and Developers' Initiative to Bring the World of Natural Interaction™ to Life
- OpenNI.org is live

Latest Downloads

- MultipleHands.zip (11 Jan, 2011)
- SkeletonRec.ori (06 Jan, 2011)
- An OpenNI logo you can use to help promote OpenNI. (06 Jan, 2011)
- PrimeSense NITE Beta Build for for Ubuntu 10.10 x64 (64-bit) v1.3.0.17 (06 Jan, 2011)
- PrimeSense NITE Beta Build for for Ubuntu 10.10 x86 (32-bit) v1.3.0.17 (06 Jan, 2011)

Inicio Downloads - Latest U... Programas Sin título - Paint.NET ... Nuevo Documento de...

Ver (100%) 23:09

Latest Unstable

PrimeSense NITE Unstable Build for Windows v1.3.0.18

Uploaded: 06 Jan, 2011
File Size: 10 MB

Download

PrimeSense NITE Unstable Build for Ubuntu 10.10 x86 (32-bit) v1.3.0.18

Uploaded: 06 Jan, 2011
File Size: 2 MB

Download

PrimeSense NITE Unstable Build for Ubuntu 10.10 x64 (64-bit) v1.3.0.18

Uploaded: 06 Jan, 2011
File Size: 2 MB

Download

PrimeSense NITE Unstable Build for Mac OSX 10.6 Universal v1.3.0.18

Latest Downloads

- Want a FREE pass to GDC?
- Congratulations to the winners of our 2011 Developer Challenge
- PrimeSense™ Establishes the OpenNI™ Standard and Developers' Initiative to Bring the World of Natural Interaction™ to Life
- OpenNI.org is live

Elección de archivo

Nombre: NITE-Bin-Win32-v...18.exe (10,3 MB)
Tipo: Aplicación
Desde: www.openni.org

No se sabe si el archivo es seguro. ¿Está seguro de querer descargarlo?

Ejecutar Guardar Cancelar Ayuda

Ver (100%) 23:09

Inicio Downloads - Latest U... Programas Sin título - Paint.NET ... Nuevo Documento de...

Una vez descargados los 3 archivos, se descomprime avin2-SensorKinect-0124bd2.

Programas

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Carpetas

Dirección C:\Documents and Settings\Felipe\Escritorio\Programas

Tareas de archivo y carpeta

- Cambiar nombre a esta carpeta
- Mover esta carpeta
- Copiar esta carpeta
- Publicar esta carpeta en Web
- Compartir esta carpeta
- Enviar por correo electrónico los archivos de esta carpeta
- Eliminar esta carpeta

Otros sitios

- Escritorio
- Mis documentos
- Documentos compartidos
- Mi PC
- Mis sitios de red

Detalles

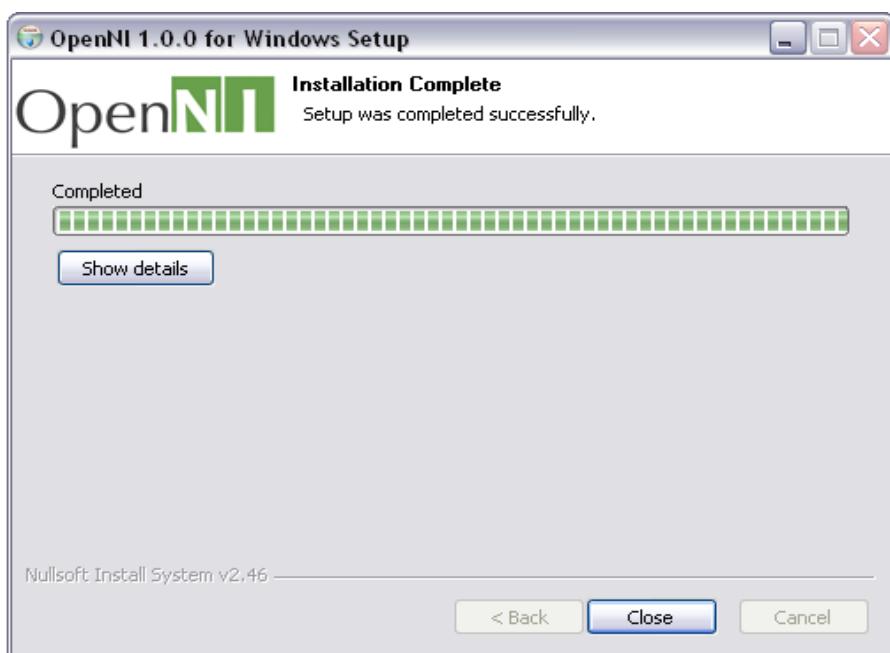
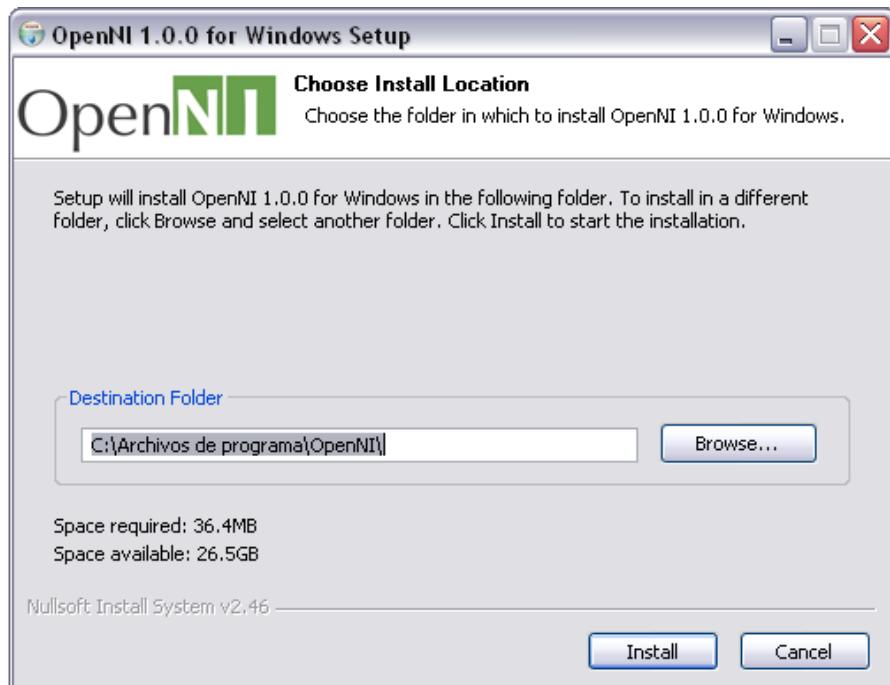
avin2-SensorKinect-0124bd2
Carpeta de archivos
Fecha de modificación: Jueves, 06 de Enero de 2011, 16:57

OpenNI-Bin-Win32-v1.0.0.25... 20.475 kB
avin2-SensorKinect-0124bd2.zip
NITE-Bin-Win32-v1.3.0.18.exe
avin2-SensorKinect-0124bd2

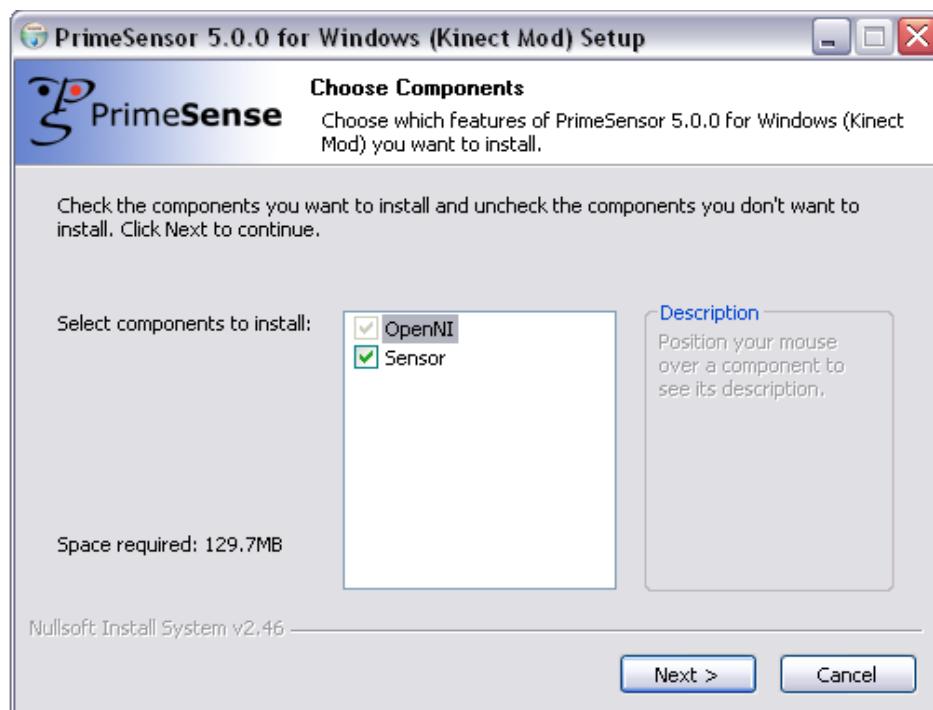
Ver (100%) 23:12

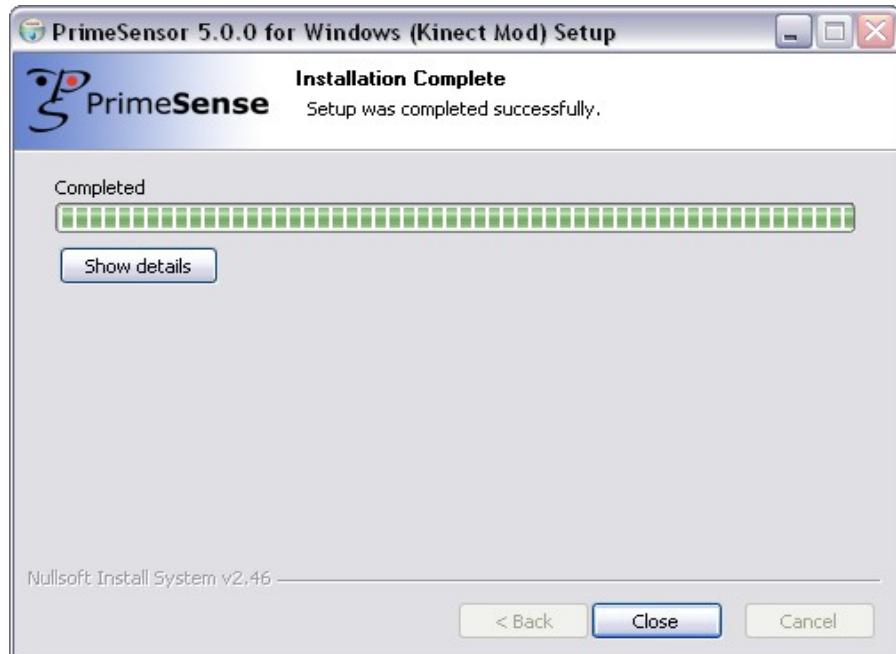
2. Se instalan los archivos recién descargados en el siguiente orden:

OpenNI-Bin-Win32-v1.0.0.25.exe

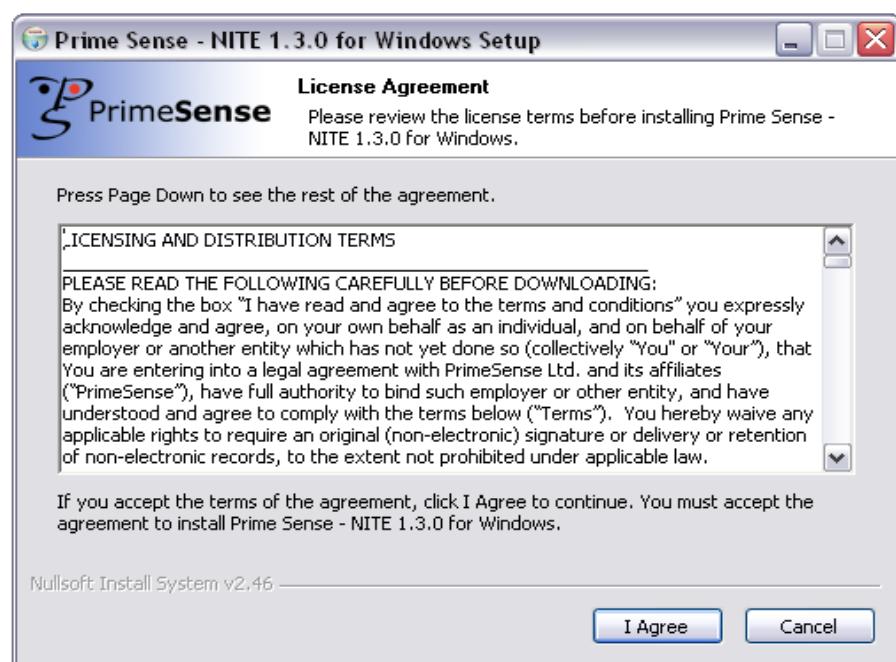


Para instalar el Sensor, debemos ir a la carpeta descomprimida **avin2-SensorKinect-0124bd2** y entrar a Bin. Ahora se ejecuta **SensorKinect-Win32-5.0.0.exe** y se instala normalmente.



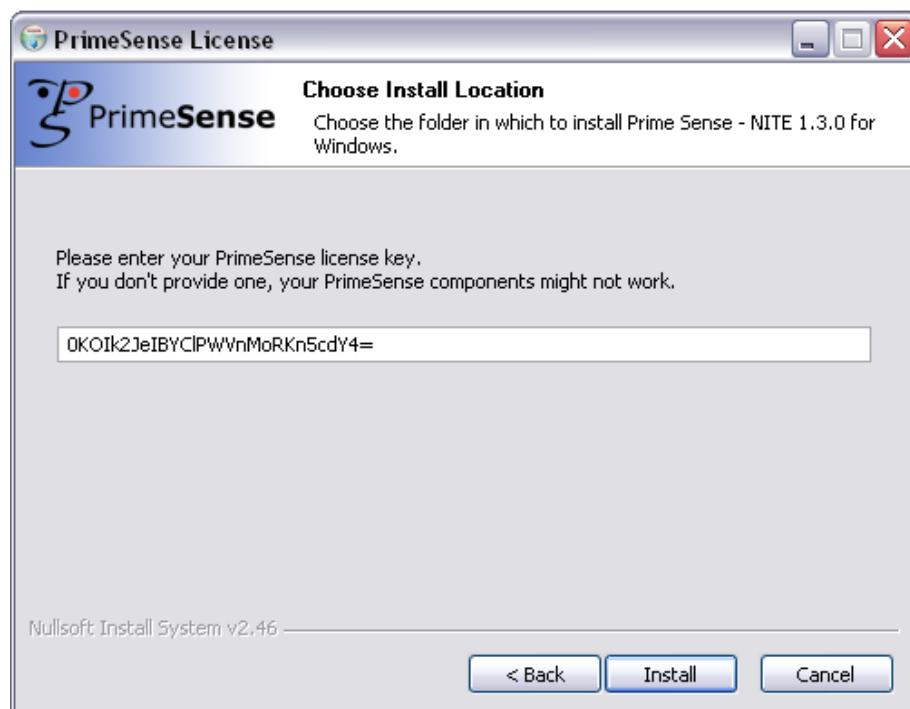


Por último se instala NITE-Bin-Win32-v1.3.0.18.exe.

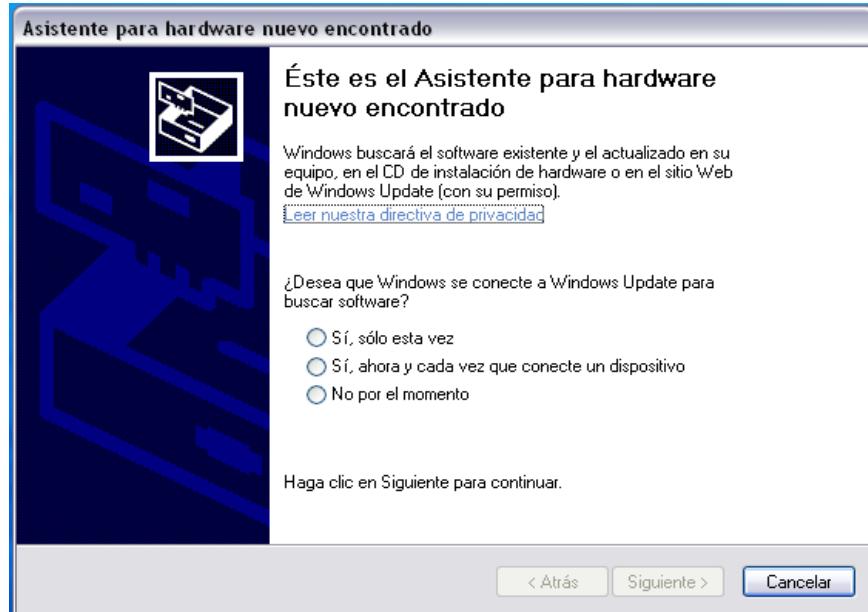


En esta parte el programa pide el número de licencia, por lo que se ingresa:

OKOIk2JeIBYCIPWVnMoRKn5cdY4=

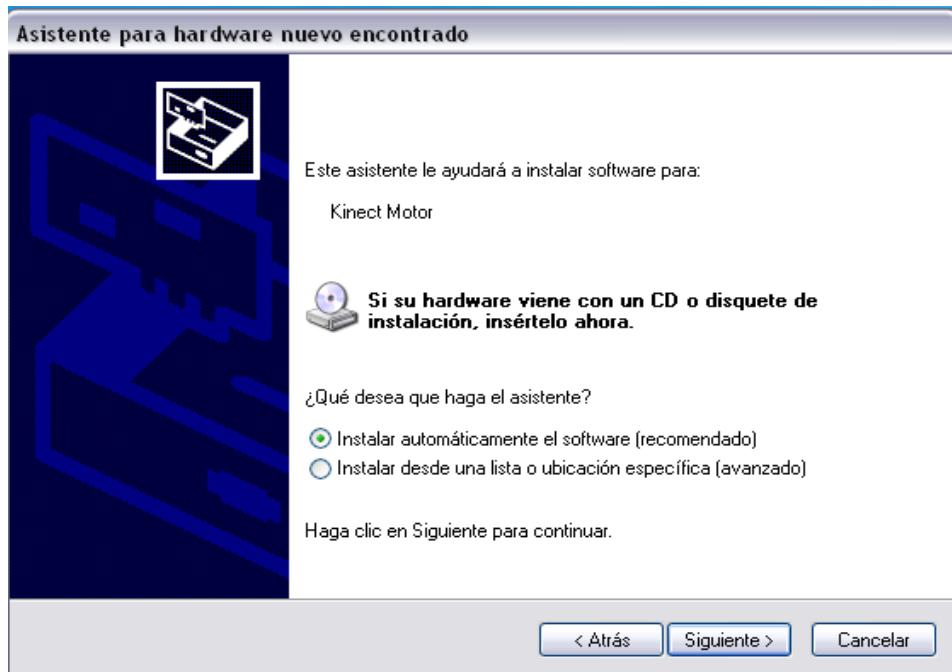


3. Ahora se conecta el Kinect. En este punto aparece automáticamente el Asistente para hardware encontrado, el cual nos indica que encontró el Motor, el Audio y la Cámara del Kinect.

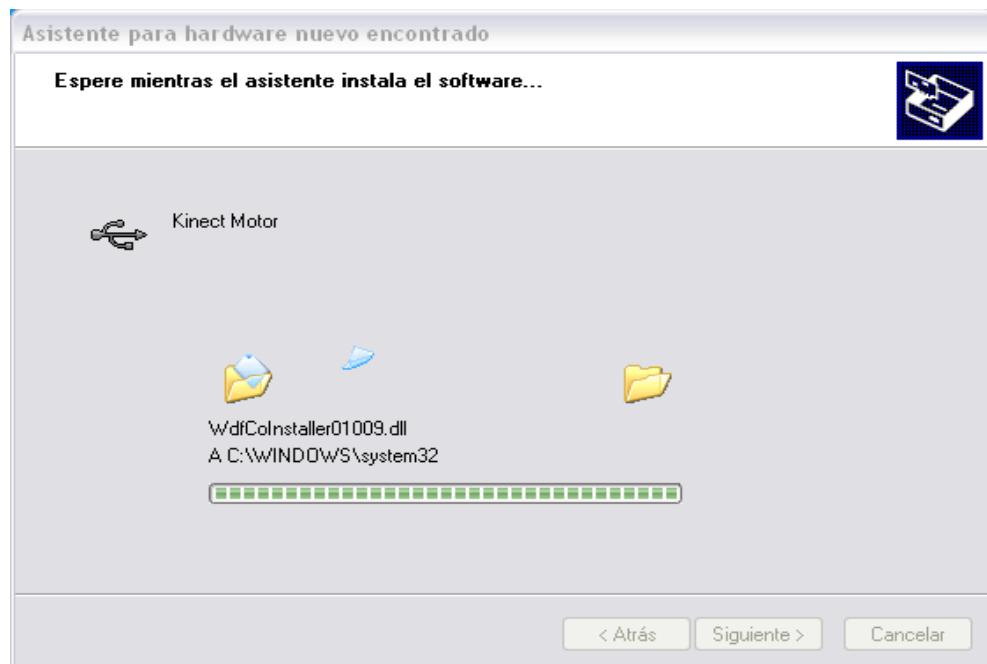


Se selecciona la primera opción: **Sí, sólo esta vez**

En la ventana posterior, se selecciona **Instalar automáticamente el software (recomendado)**



Con esto Windows instala correctamente el driver del motor del Kinect.

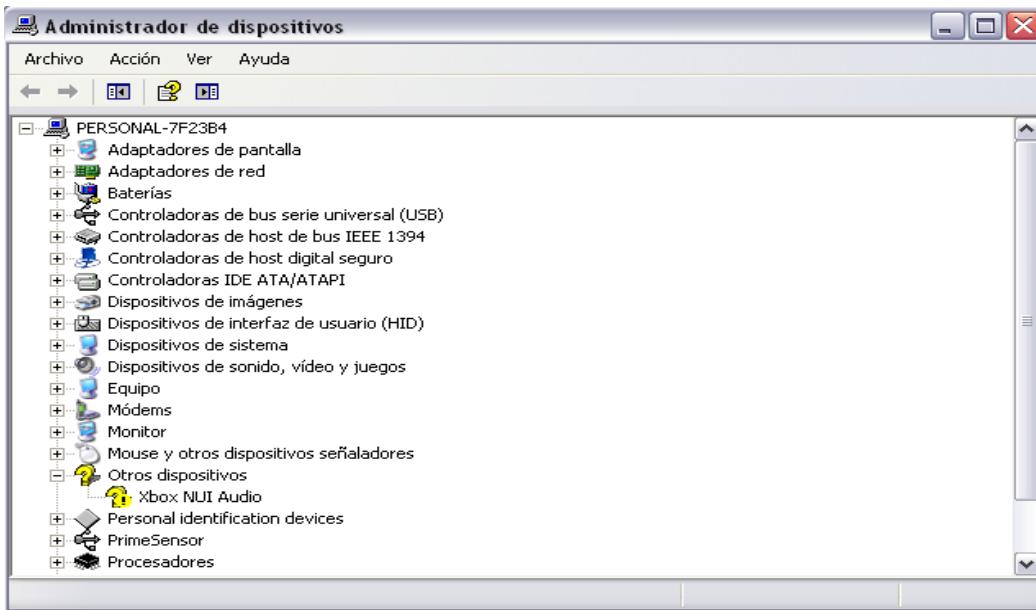




Luego de instalar el motor, aparece que encontró el Audio, sin embargo, Windows no instala este driver. Esto no es problema ya que no es necesario.

Finalmente aparece la cámara, la cual se instala análogamente al driver del motor.

Si se observa el administrador de dispositivos se debería ver algo por el estilo:



Observación: si por algún motivo Windows no reconoce los drivers, éstos se pueden instalar manualmente desde la carpeta **avin2-SensorKinect-0124bd2/Plataform/Win32/Driver**, instalando **dpinst-x86.exe** o **dpinst-amd64.exe** dependiendo si la arquitectura es 32 ó 64 bits.

4. El último paso antes de poder probar el Kinect es la modificación de 3 archivos XML (**Sample-Scene.xml**, **Sample-Tracking.xml** y **Sample-User.xml**). Éstos se encuentran en la siguiente dirección:

C:\Archivos de Programas\Prime Sense\NITE\DATA

Hay que realizar la **misma modificación para cada archivo**. Con el editor preferido se abre el archivo **Sample-Scene.xml** y se buscan las siguientes líneas:

```
<License vendor="PrimeSense" key="Insert Key"/>
```

```
<MapOutputMode xRes="320" yRes="280" FPS="30"/>
```

Las cuales se reemplazan correspondientemente por:

```
<License vendor="PrimeSense" key="0KOIk2JeIBYClPWVnMoRKn5cdY4="/>
```

```
<MapOutputMode xRes="640" yRes="480" FPS="30"/>
```

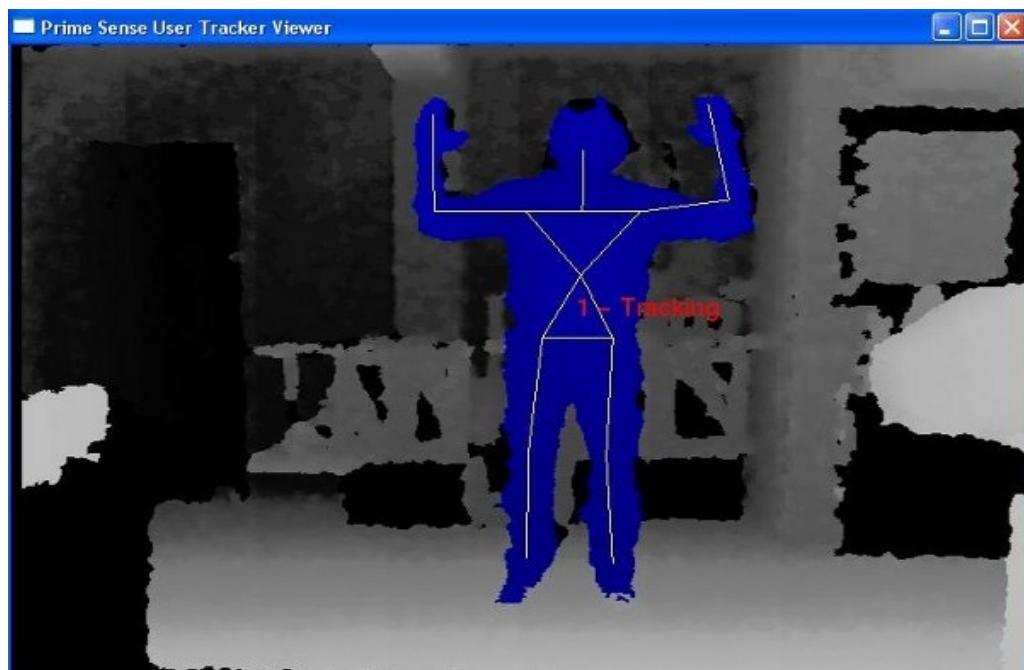
Luego se guarda el archivo y se repiten los mismo paso con los archivos **Sample-Tracking.xml** y **Sample-User.xml**.

Finalmente se pueden probar los ejemplos y comprobar que la instalación fue exitosa. Los ejemplos que OpenNI trae por defecto se encuentran en:

C:\Archivos de programa\OpenNI\Samples\Bin\Release\



También se pueden probar los ejemplos que trae NITE, los cuales se encuentran en C:\Archivos de programa\OpenNI\Samples\Bin\Release\



La posición mostrada en la imagen, es la posición de calibración que utiliza Kinect para posteriormente realizar tracking.

Observación: En algunos casos los ejemplos no funcionan o funcionan parcialmente cuando son ejecutados directamente, si esto te pasa prueba ejecutándolos directamente desde la consola de comandos de Windows.

Si estos test no funcionan significa que algo ha ido mal, así que deberás repetir los pasos anteriores.

Gentoo Linux

Se abre una consola y se siguen los siguientes pasos:

- (1) mkdir ~/kinect && cd ~/kinect
- (2) git clone <https://github.com/OpenNI/OpenNI.git>
- (3) cd OpenNI/Platform/Linux-x86/Build
- (4) make && sudo make install
- (5) cd ~/kinect/
- (6) git clone <https://github.com/boilerbots/Sensor.git>
- (7) cd Sensor
- (8) git checkout kinect
- (9) cd Platform/Linux-x86/Build
- (10) make && sudo make install
- (11) Descargar de la página [OpenNI](#) la última versión de NITE específica para tu plataforma.
- (12) Guardar el archivo y descomprimirlo en el directorio ~/kinect
- (13) cd ~/kinect/NITE/Nite-1.3.0.17/Data
- (14) Abrir Sample-User.xml y reemplazar las siguientes líneas donde corresponda:

```
<License vendor="PrimeSense" key="0KOIk2JeIBYClPWVnMoRKn5cdY4="/>
<MapOutputMode xRes="640" yRes="480" FPS="30"/>
```
- (15) Repetir el paso 14 para Sample-Scene.xml y Sample-Tracking.xml
- (16) niLicense PrimeSense 0KOIk2JeIBYClPWVnMoRKn5cdY4=
- (17) cd ~/kinect/NITE/Nite-1.3.0.17/
- (18) sudo ./install.bash
- (19) make && sudo make install # Este paso parece no ser estrictamente necesario.

Finalmente se pueden probar los ejemplos de igual forma como se explicó anteriormente para Ubuntu / Debian.

Documentación

La documentación oficial de OpenNi se puede encontrar en <http://openni.org/documentation>. Si ya se instaló OpenNi, este mismo archivo PDF (OpenNI_UserGuide.pdf) se encuentra en el siguiente directorio .../OpenNI/Documentation/.

Así mismo, en el directorio de Nite en el subdirectorio Documentation se encuentran dos archivos PDF:

- **NITE Controls 1.3 - Programmer's Guide.pdf**
- **NITE Algorithms 1.3.pdf**

Para estudiar los códigos de los ejemplos de OpenNI y Nite, se debe ir al directorio donde se instalaron los programas. Posteriormente entrar a .../OpenNI/Samples o .../Nite-1.3.0.17/Samples dependiendo de cuales ejemplos se quieran estudiar.

En la siguiente página se detalla la documentación de todos los módulos, en un estilo tipo JavaDoc:

<http://eris.liralab.it/chris/dox/html/namespacexn.html>

Además de la documentación presentada, se recomienda el grupo y wiki oficial:

<https://groups.google.com/group/openni-dev/>

<http://wiki.openni.org>

Acá se pueden ver o hacer preguntas, las cuales los usuarios más expertos van contestando en la medida de su conocimiento.

Finalmente, se dejan un par de páginas donde se puede encontrar información de OpenNi, Nite, OpenCV, Kinect y otros temas.

<http://asimmittal.net/blog/>

<http://www.codingbeta.com/?p=90>

http://groups.google.com/group/openni-dev/browse_thread/thread/1c4110041b603383

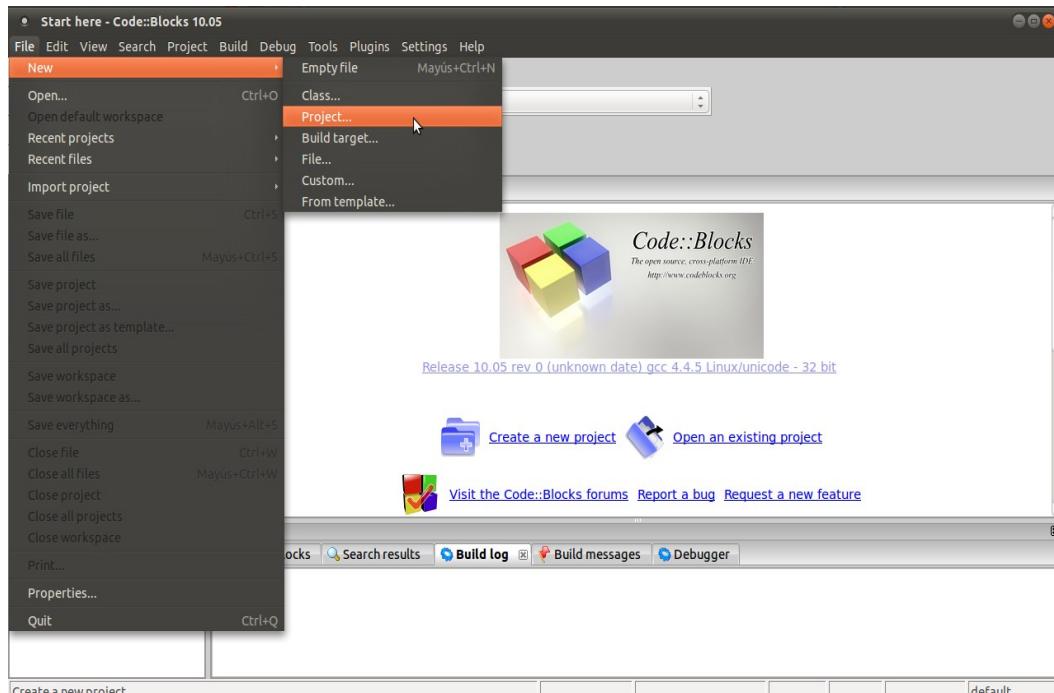
<http://www.openframeworks.cc/forum/>

Crear un proyecto nuevo con Code::Blocks

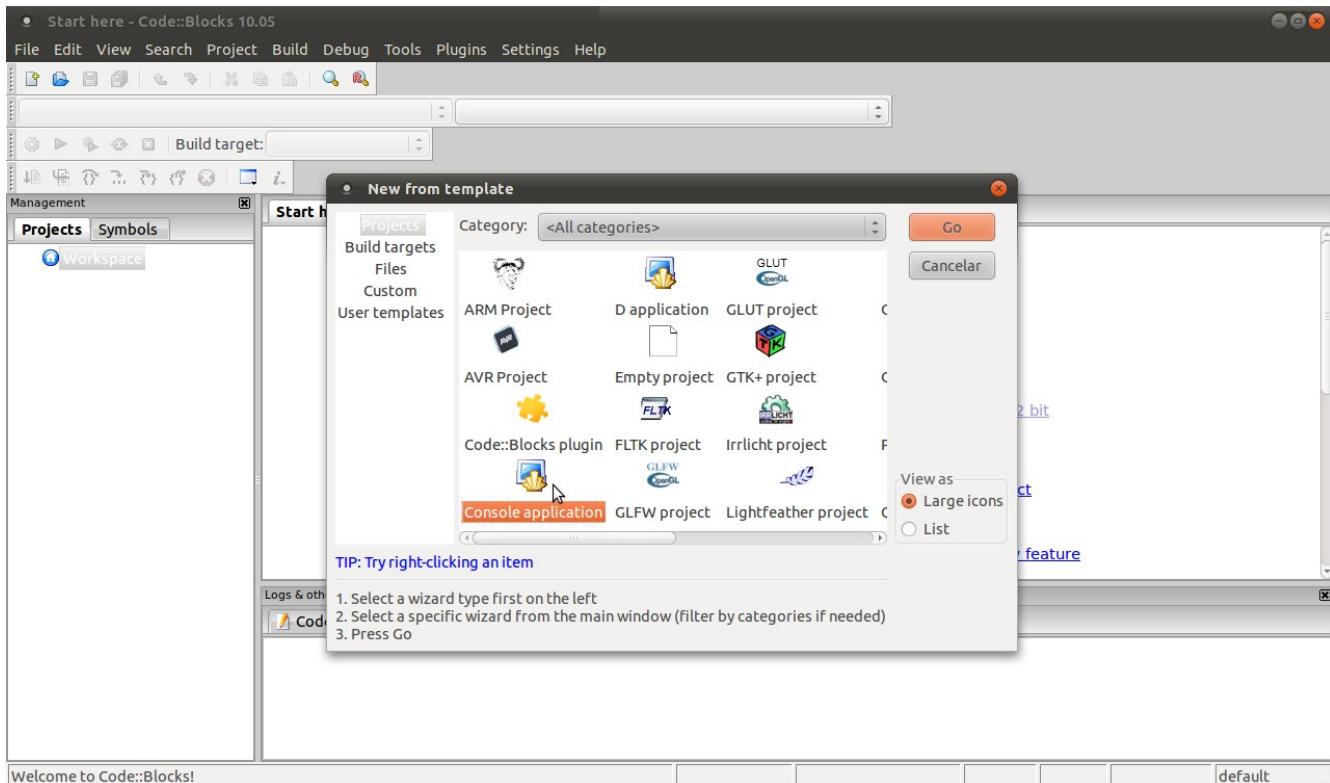
Para crear un proyecto utilizando OpenNI y/o Nite no es necesario un IDE, se puede utilizar cualquier editor de texto como **Gedit**, sin embargo, tareas como compilar y debuggear se simplifican enormemente. Por esto se señalan los pasos a seguir en Linux con el programa **Code::Blocks**.

La guía para realizar lo mismo en Windows se encuentra en la [documentación oficial](#), desde la página 18.

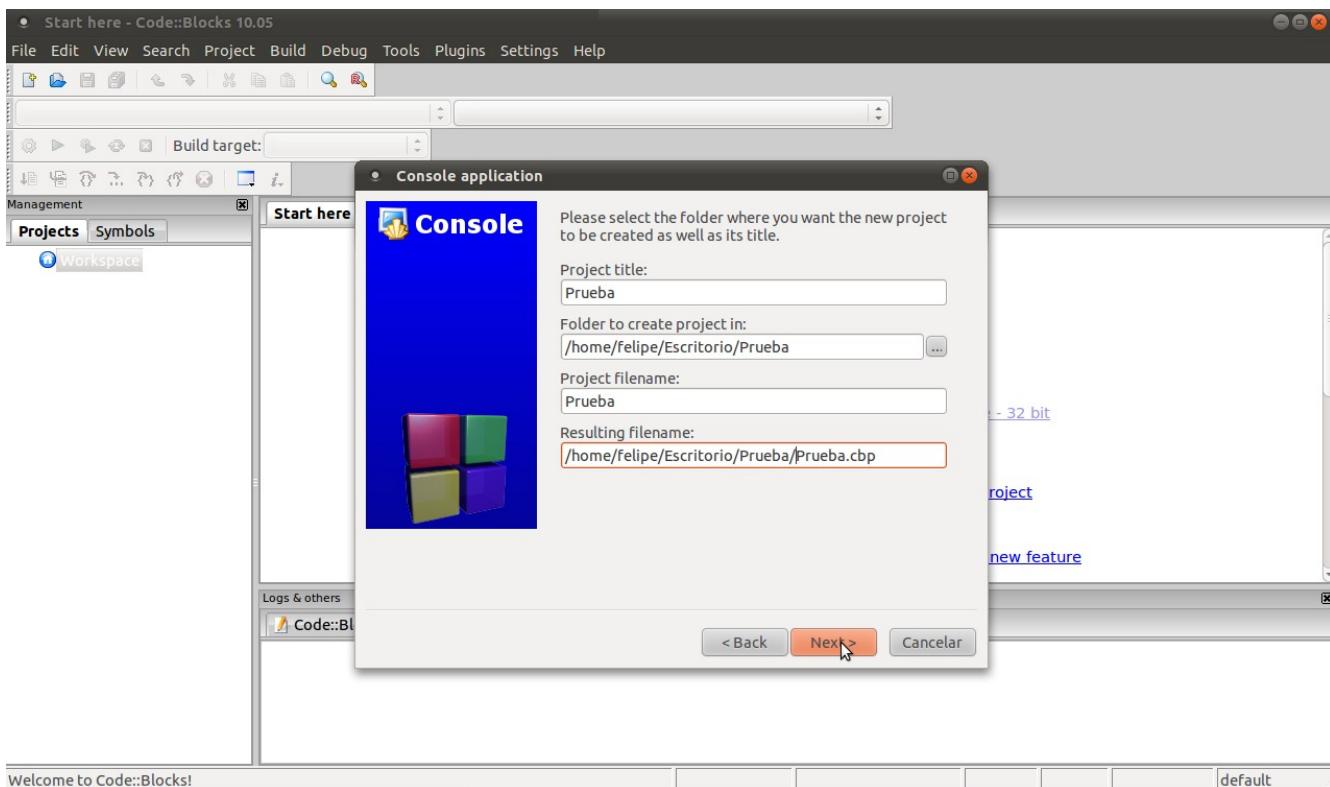
En primer lugar, se abre Code::Blocks. Se presiona **File → New → Project**



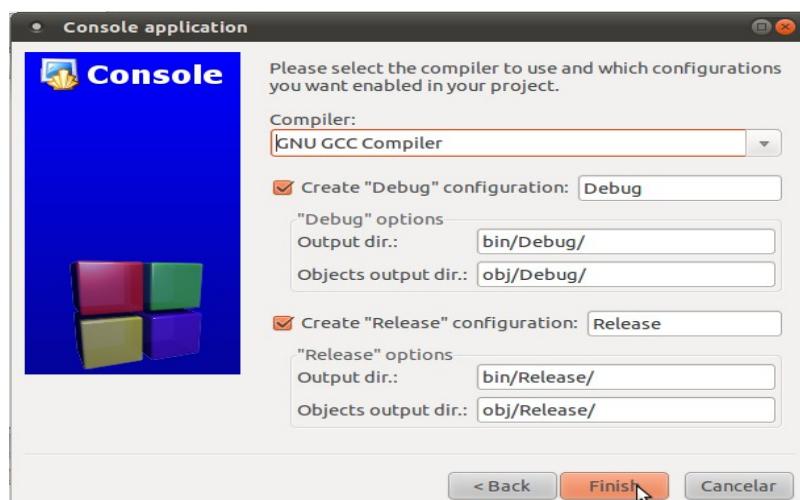
En la ventana que aparece, se escoge: **Console application**



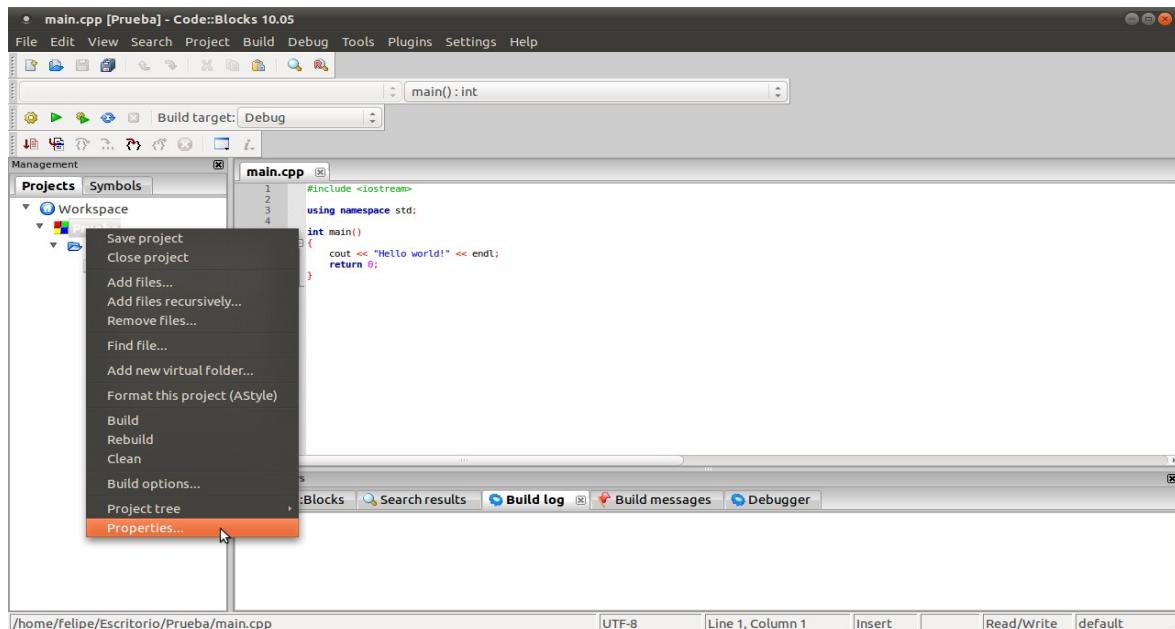
Se escoge el lenguaje a programar, C o C++ (en este caso se escoge C++), y se completa la información requerida en la ventana siguiente. Se muestra un ejemplo a continuación:



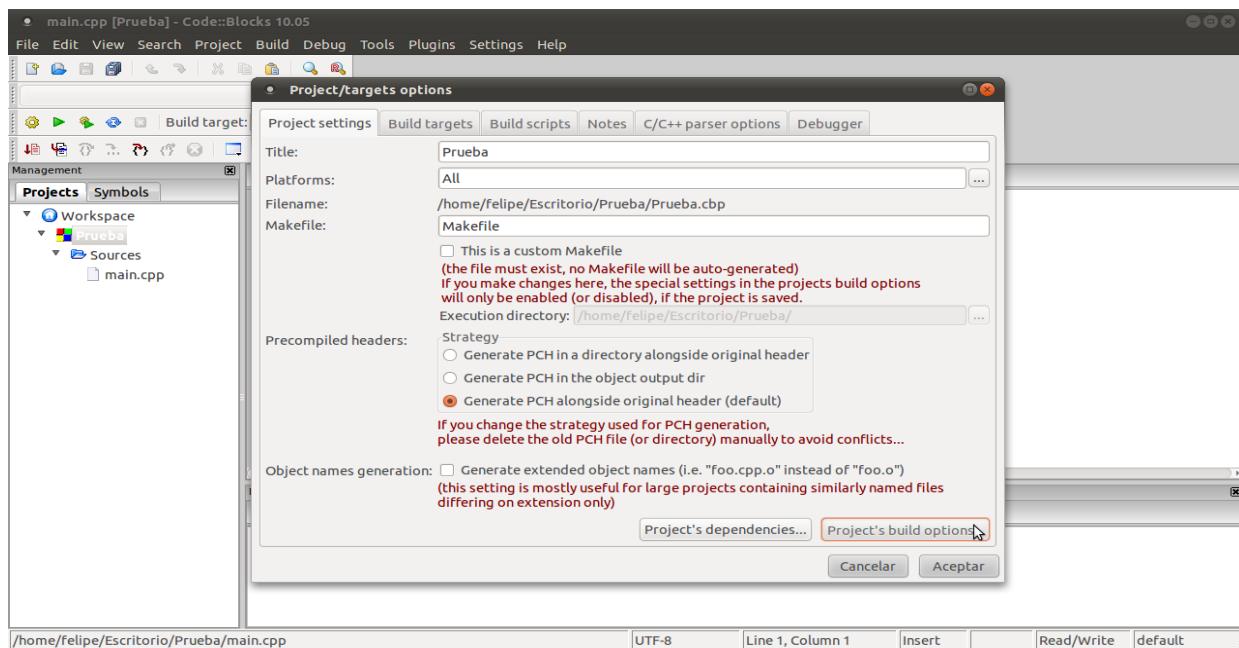
Posteriormente en la siguiente ventana presionamos Finish.



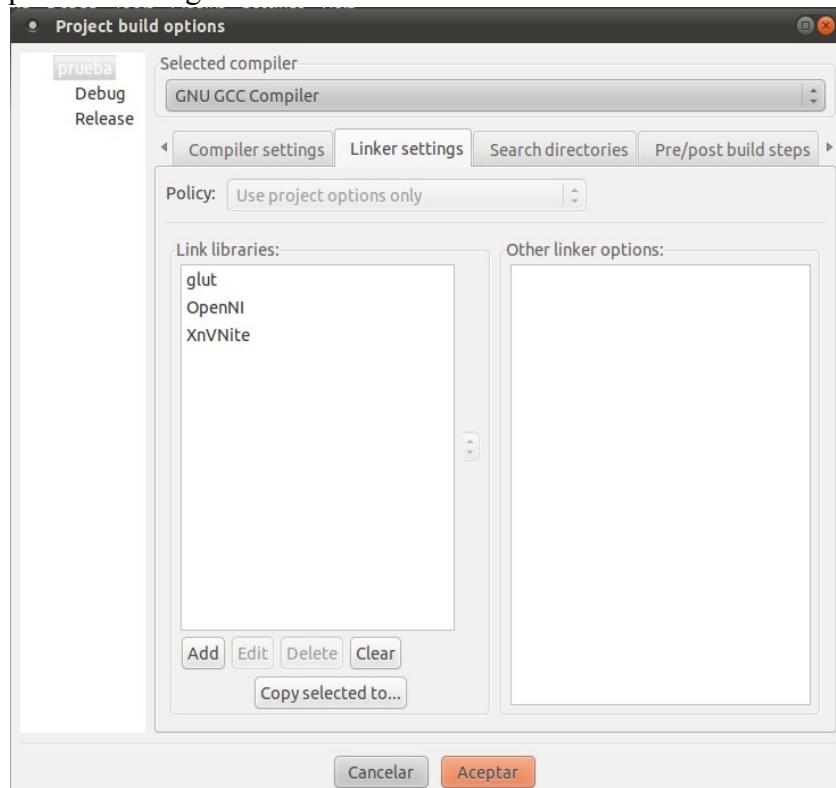
Ahora solo falta configurar las bibliotecas en el programa. Para esto presionamos el botón derecho sobre el nombre de nuestro proyecto, y seleccionamos **Properties**.



En la ventana que aparece se selecciona **Project's build options**.



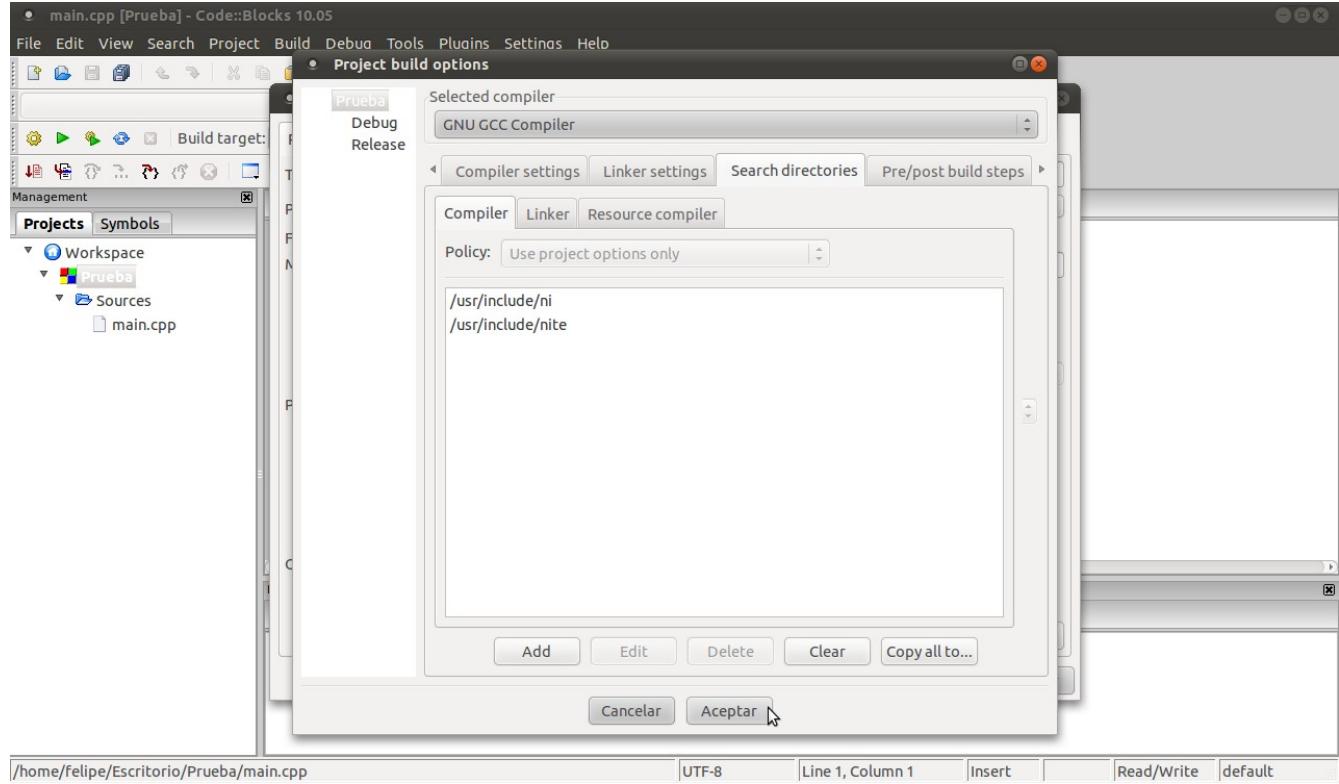
En las pestañas con opciones que aparecen, presionamos **Linker settings**, y agregamos: **glut**, **OpenNI** y **XnVNite**. Debe quedar de la siguiente forma:



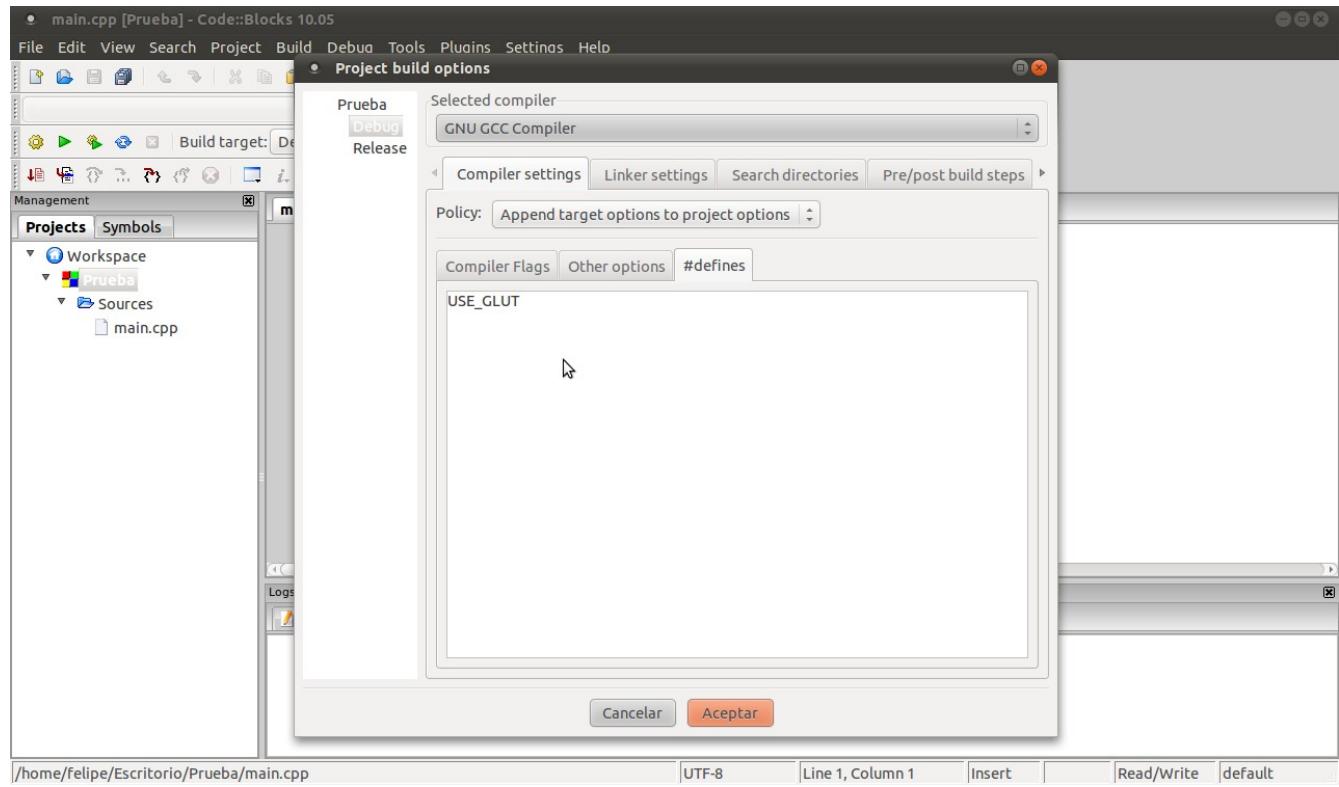
Se selecciona la pestaña **Search directories**, para indicarle al programa donde se encuentran las bibliotecas, y se añade:

/usr/include/ni

/usr/include/nite



Finalmente, algunos programas utilizan Glut, para especificar esto, seleccionamos la pestaña **Compiler Settings**, la sub pestaña **#defines**, y se escribe: **USE GLUT**



Con estos pasos, se configura **Code::Blocks** para que reconozca las clases de OpenNi y Nite, y se pueda compilar y debuggear sin problemas.

Documentación de los formatos de salida de la información entregada por el dispositivo

El driver entregado por Primesense permite obtener; depthMap, mapa IR y mapa RGB.

Un depthMap corresponde a una matriz escalonada de píxeles de profundidad. Esto quiere decir, que una “matriz 1D” (arreglo) representa una matriz 2D.

Cada píxel de profundidad es de 2 bytes y representa la distancia en milímetros de ese píxel al sensor. El número de píxeles está determinado por la resolución. Estos datos de profundidad se pueden obtener por medio del objeto llamado DepthMetaData.

Una imagen RGB se compone de Rojo, Verde y Azul. Cada uno de estos colores es de 8 bits, por lo que un píxel tiene 24 bits, que equivale a 3 bytes.

Por otro lado, una imagen del sensor infrarrojo, se compone de 16 bits en escala de grises.

Para saber cómo trabajar con OPENNI y OpenCV en conjunto, se recomienda leer el pequeño manual adjunto a continuación:

http://opencv.jp/opencv-2svn_org/opencv_user.pdf

VideoCapture provides retrieving the following Kinect data:

```
a.) data given from depth generator:  
    OPENNI_DEPTH_MAP           - depth values in mm (CV_16UC1)  
    OPENNI_POINT_CLOUD_MAP     - XYZ in meters (CV_32FC3)  
    OPENNI_DISPARITY_MAP       - disparity in pixels (CV_8UC1)  
    OPENNI_DISPARITY_MAP_32F   - disparity in pixels (CV_32FC1)  
    OPENNI_VALID_DEPTH_MASK   - mask of valid pixels (not occluded,  
                                not shaded etc.) (CV_8UC1)  
b.) data given from RGB image generator:  
    OPENNI_BGR_IMAGE          - color image (CV_8UC3)  
    OPENNI_GRAY_IMAGE         - gray image (CV_8UC1)
```

/** Defines the depth values type (16-bit values). */

typedef XnUInt16 XnDepthPixel;

/** Defines the value of a no-sample depth pixel. */

#define XN_DEPTH_NO_SAMPLE_VALUE ((XnDepthPixel)0)

/** Defines the image RGB24 pixel type. */

typedef struct XnRGB24Pixel

{

XnUInt8 nRed;

XnUInt8 nGreen;

XnUInt8 nBlue;

} XnRGB24Pixel;

/** Defines the image YUV422 double-pixel type. */

typedef struct XnYUV422DoublePixel

{

XnUInt8 nU;

```

XnUInt8 nY1;
XnUInt8 nV;
XnUInt8 nY2;
} XnYUV422DoublePixel;

/** Defines the image Grayscale8 pixel type. */
typedef XnUInt8 XnGrayscale8Pixel;

/** Defines the image Grayscale16 pixel type. */
typedef XnUInt16 XnGrayscale16Pixel;

/** Defines the IR values type (16-bit grayscale). */
typedef XnGrayscale16Pixel XnIRPixel;

/** Defines the label type */
typedef XnUInt16 XnLabel;

```

A continuación se muestra cómo realizar los pasos básicos en OpenNI, como por ejemplo; crear un nodo y leer datos, la configuración de un nodo, etc.

Funciones Básicas: Inicializar, crear nodo y lectura de data

El siguiente ejemplo ilustra una funcionalidad básica de OpenNi. Se inicializa un Context object, se crea un nodo y se leen datos de profundidad de éste.

```
XnStatus nRetVal = XN_STATUS_OK;
```

```
xn::Context context;
```

```
// Initialize context object
```

```
nRetVal = context.Init();
```

```
// Create a DepthGenerator node
```

```
xn::DepthGenerator depth;
```

```
nRetVal = depth.Create(context);
```

```
// Make it start generating data
```

```
nRetVal = context.StartGeneratingAll();
```

```
// Main loop
```

```
while (bShouldRun)
```

```
{
```

```
    // Wait for new data to be available
```

```
    nRetVal = context.WaitOneUpdateAll(depth);
```

```
    if (nRetVal != XN_STATUS_OK)
```

```
{
```

```
        printf("Failed updating data: %s\n",
```

```
        xnGetStatusString(nRetVal));
```

```

    continue;
}

// Take current depth map
const XnDepthPixel* pDepthMap = depth.GetDepthMap();
}

// Clean-up
context.Shutdown();

```

Configuración de un Nodo

En el siguiente código se crea un Depth generator (generador de un mapa de profundidad) , se configura la resolución en VGA a 30 FPS (Frames por segundo) y se da inicio:

```

// Create a DepthGenerator node
xn::DepthGenerator depth;
nRetVal = depth.Create(context);

XnMapOutputMode outputMode;
outputMode.nXRes = 640;
outputMode.nYRes = 480;
outputMode.nFPS = 30;
nRetVal = depth.SetMapOutputMode(outputMode);

// We're done configuring it. Make it start generating data
nRetVal = context.StartGeneratingAll();

// Main loop
while (bShouldRun)
{
    // Wait for new data to be available
    nRetVal = context.WaitOneUpdateAll(depth);
    if (nRetVal != XN_STATUS_OK)
    {
        printf("Failed updating data: %s\n",
xnGetStatusString(nRetVal));
        continue;
    }

    // Take current depth map
    const XnDepthPixel* pDepthMap = depth.GetDepthMap();
}

```

Ejemplo Mapa profundidad y modo VGA

El siguiente código crea un mapa de profundidad, chequea que el mapa generado esté configurado en modo VGA con 30 FPS, lee cada frame e imprime por la salida el cálculo del píxel central.

```
XnStatus nRetVal = XN_STATUS_OK;
```

```
Context context;
nRetVal = context.Init();
```

```
// Create a depth generator
```

```

DepthGenerator depth;
nRetVal = depth.Create(context);

// Set it to VGA maps at 30 FPS
XnMapOutputMode mapMode;
mapMode.nXRes = XN_VGA_X_RES;
mapMode.nYRes = XN_VGA_Y_RES;
mapMode.nFPS = 30;
nRetVal = depth.SetMapOutputMode(mapMode);

// Start generating
nRetVal = context.StartGeneratingAll();

// Calculate index of middle pixel
XnUInt32 nMiddleIndex =
    XN_VGA_X_RES * XN_VGA_Y_RES/2 + // start of middle line
    XN_VGA_X_RES/2;           // middle of this line

while (TRUE)
{
    // Update to next frame
    nRetVal = context.WaitOneUpdateAll(depth);

    const XnDepthPixel* pDepthMap = depth.GetDepthMap();
    printf("Middle pixel is %u millimeters away\n", pDepthMap[nMiddleIndex]);
}

// Clean up
context.Shutdown();

```

Trabajando con Skeleton (Probado en la versión inestable que trae Wrapper C#)

El siguiente ejemplo muestra cómo identificar a un usuario cuando éste es detectado, buscando una posición para realizar la calibración y así poder realizar seguimiento. Además se imprime la ubicación de la cabeza del usuario trackeado.

```

#define POSE_TO_USE "Psi"
xn::UserGenerator g_UserGenerator;

void XN_CALLBACK_TYPE
User_NewUser(xn::UserGenerator& generator, XnUserID nId, void* pCookie)
{
    printf("New User: %d\n", nId);
    g_UserGenerator.GetPoseDetectionCap().StartPoseDetection(POSE_TO_USE, nId);
}

void XN_CALLBACK_TYPE User_LostUser(xn::UserGenerator& generator, XnUserID nId, void* pCookie) {}
void XN_CALLBACK_TYPE Pose_Detected(xn::PoseDetectionCapability& pose, const XnChar* strPose,
                                    XnUserID nId, void* pCookie)
{
    printf("Pose %s for user %d\n", strPose, nId);
    g_UserGenerator.GetPoseDetectionCap().StopPoseDetection(nId);
    g_UserGenerator.GetSkeletonCap().RequestCalibration(nId, TRUE);
}

```

```

}

void XN_CALLBACK_TYPE Calibration_Start(xn::SkeletonCapability& capability, XnUserID nId,
                                         void* pCookie)
{
    printf("Starting calibration for user %d\n", nId);
}

void XN_CALLBACK_TYPE Calibration_End(xn::SkeletonCapability& capability, XnUserID nId,
                                         XnBool bSuccess, void* pCookie)
{
    if (bSuccess)
    {
        printf("User calibrated\n");
        g_UserGenerator.GetSkeletonCap().StartTracking(nId);
    }
    else
    {
        printf("Failed to calibrate user %d\n", nId);
        g_UserGenerator.GetPoseDetectionCap().StartPoseDetection(POSE_TO_USE, nId);
    }
}

void main()
{
    XnStatus nRetVal = XN_STATUS_OK;

    xn::Context context;
    nRetVal = context.Init();
    // Create the user generator
    nRetVal = g_UserGenerator.Create(context);

    XnCallbackHandle h1, h2, h3;
    g_UserGenerator.RegisterUserCallbacks(User_NewUser, User_LostUser, NULL, h1);
    g_UserGenerator.GetPoseDetectionCap().RegisterToPoseCallbacks(Pose_Detected, NULL, NULL, h2);
    g_UserGenerator.GetSkeletonCap().RegisterCalibrationCallbacks(Calibration_Start, Calibration_End, NULL,
                                                                h3);

    // Set the profile
    OpenNI User Guide Page | 26
    g_UserGenerator.GetSkeletonCap().SetSkeletonProfile(XN_SKEL_PROFILE_ALL);
    // Start generating
    nRetVal = context.StartGeneratingAll();

    while (TRUE)
    {
        // Update to next frame
        nRetVal = context.WaitAndUpdateAll();
        // Extract head position of each tracked user
        XnUserID aUsers[15];
        XnUInt16 nUsers = 15;
        g_UserGenerator.GetUsers(aUsers, nUsers);
        for (int i = 0; i < nUsers; ++i)
    }
}

```

```

{
    if (g_UserGenerator.GetSkeletonCap().IsTracking(aUsers[i]))
    {
        XnSkeletonJointPosition Head;
        g_UserGenerator.GetSkeletonCap().GetSkeletonJointPosition(aUsers[i], XN_SKEL_HEAD, Head);
        printf("%d: (%f,%f,%f) [%f]\n", aUsers[i],
            Head.position.X, Head.position.Y, Head.position.Z,
            Head.fConfidence);
    }
}
}

// Clean up
context.Shutdown();
}

```

Hand Point

El siguiente ejemplo muestra cómo detectar e identificar gestos hecho con la mano, para comenzar el seguimiento de ésta.

```

#define GESTURE_TO_USE "Click"

xn::GestureGenerator g_GestureGenerator;
xn::HandsGenerator g_HandsGenerator;

void XN_CALLBACK_TYPE Gesture_Recognized(xn::GestureGenerator& generator,
                                         const XnChar* strGesture, const XnPoint3D* pIDPosition, const XnPoint3D* pEndPosition, void* pCook ie)
{
    printf("Gesture recognized: %s\n", strGesture);
    g_GestureGenerator.RemoveGesture(strGesture);
    g_HandsGenerator.StartTracking(*pEndPosition);
}

void XN_CALLBACK_TYPE
Gesture_Process(xn::GestureGenerator& generator, const XnChar* strGesture, const XnPoint3D* pPosition,
                XnFloat fProgress, void* pCookie)
{
}

void XN_CALLBACK_TYPE Hand_Create(xn::HandsGenerator& generator, XnUserID nId, const XnPoint3D* pPosition, XnFloat fTime, void* pCookie)
{
    printf("New Hand: %d @ (%f,%f,%f)\n", nId, pPosition->X, pPosition->Y, pPosition->Z);
}

void XN_CALLBACK_TYPE Hand_Update(xn::HandsGenerator& generator, XnUserID nId, const XnPoint3D* pPosition, XnFloat fTime, void* pCookie)
{
}

void XN_CALLBACK_TYPE Hand_Destroy(xn::HandsGenerator& generator, XnUserID nId, XnFloat fTime,
                                   void* pCookie)
{
    printf("Lost Hand: %d\n", nId);
    g_GestureGenerator.AddGesture(GESTURE_TO_USE, NULL);
}

```

```
}

void main()
{
    XnStatus nRetVal = XN_STATUS_OK;

    Context context;
    nRetVal = context.Init();

    // Create the gesture and hands generators
    nRetVal = g_GestureGenerator.Create(context);
    nRetVal = g_HandsGenerator.Create(context);

    // Register to callbacks
    XnCallbackHandle h1, h2;
    g_GestureGenerator.RegisterGestureCallbacks(Gesture_Recognized, Gesture_Process, NULL, h1);
    g_HandsGenerator.RegisterHandCallbacks(Hand_Create, Hand_Update, Hand_Destroy, NULL, h2);

    // Start generating
    nRetVal = context.StartGeneratingAll();

    nRetVal = g_GestureGenerator.AddGesture(GESTURE_TO_USE);
    while (TRUE)
    {
        // Update to next frame
        nRetVal = context.WaitAndUpdateAll();
    }

    // Clean up
    context.Shutdown();
}
```

Análisis de Rendimiento

Las pruebas se realizaron en un Notebook HP Pavilion dv2725la, con sistema operativo Ubuntu Maverick (10.10).

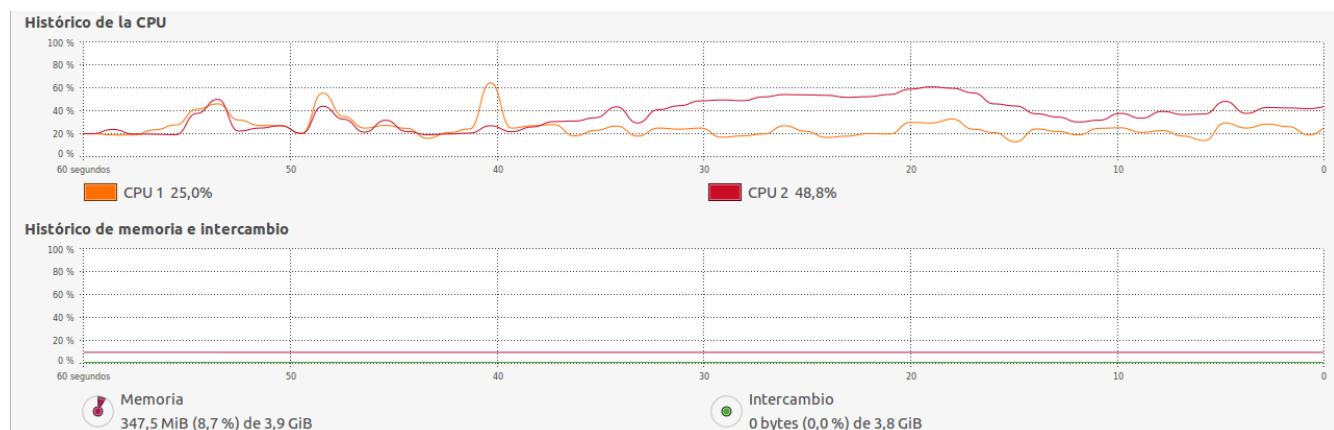
Procesador 1	Intel(R) Core(TM) 2 Duo CPU T5450 @ 1,66 GHz
Procesador 2	Intel(R) Core(TM) 2 Duo CPU T5450 @ 1,66 GHz
Memoria RAM	4 GiB
Memoria SWAP	4 GiB

Se testeó cuánto porcentaje de CPU consumían ciertas aplicaciones que trae por defecto OpenNI y Nite. A continuación se muestra los resultados:

Sample-Boxes

```
top - 18:47:04 up 5:29, 3 users, load average: 0.15, 0.20, 0.57
Tasks: 175 total, 3 running, 172 sleeping, 0 stopped, 0 zombie
Cpu(s): 14.2%us, 5.8%sy, 0.0%ni, 79.4%id, 0.5%wa, 0.0%hi, 0.2%si, 0.0%st
Mem: 4108288k total, 1652768k used, 2455520k free, 714200k buffers
Swap: 3999740k total, 0k used, 3999740k free, 580624k cached

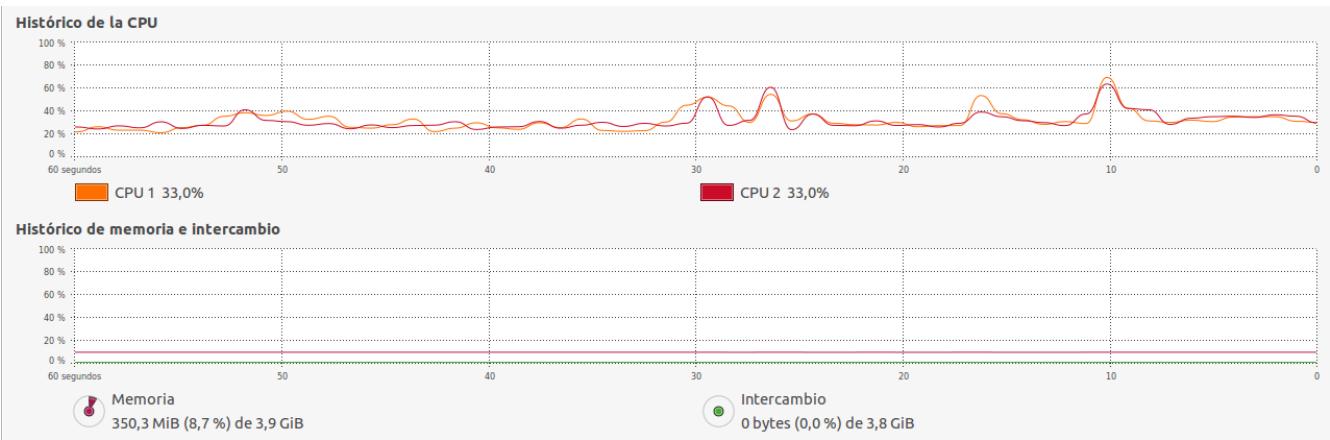
PID USER      PR  NI    VIRT   RES   SHR S %CPU %MEM     TIME+ COMMAND
6535 felipe    20   0 105m  38m  9328 S  25  1.0  0:16.00 Sample-Boxes
4950 root      20   0 78520  33m  13m R  15  0.8  3:48.87 Xorg
6448 felipe    20   0 86548  18m  15m R  13  0.5  0:41.84 gnome-system-mo
```



Sample-CircleControl

```
top - 18:51:12 up 5:34, 3 users, load average: 0.09, 0.16, 0.47
Tasks: 175 total, 3 running, 172 sleeping, 0 stopped, 0 zombie
Cpu(s): 19.1%us, 5.3%sy, 0.0%ni, 75.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 4108288k total, 1658200k used, 2450088k free, 714560k buffers
Swap: 3999740k total, 0k used, 3999740k free, 585084k cached

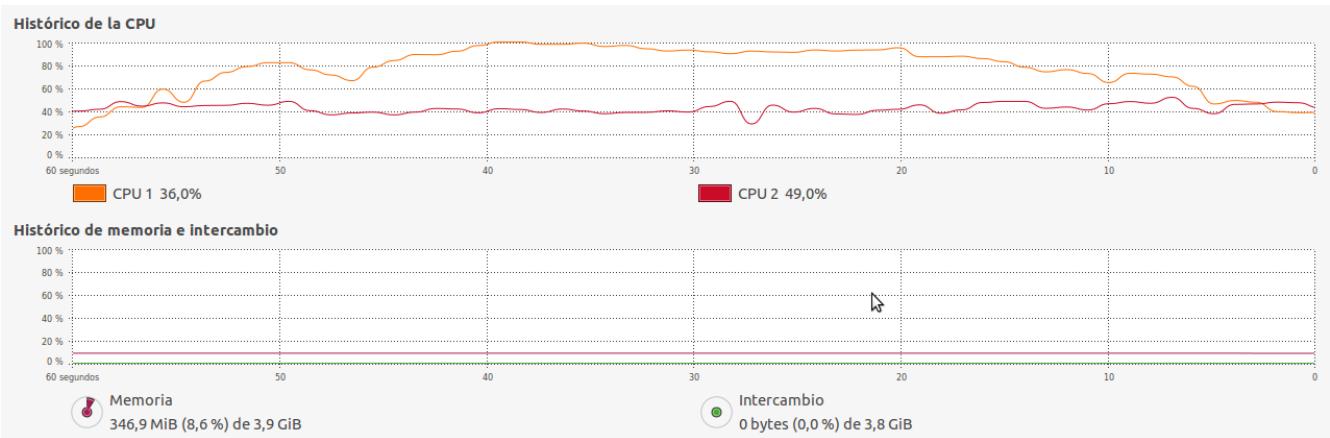
PID USER      PR  NI    VIRT   RES   SHR S %CPU %MEM     TIME+ COMMAND
6570 felipe    20   0 105m  38m  9228 R  31  1.0  0:42.52 Sample-CircleCo
6448 felipe    20   0 86564  18m  15m S  16  0.5  1:20.89 gnome-system-mo
4950 root      20   0 77080  33m  13m S  13  0.8  4:21.40 Xorg
```



Sample-PointViewer

```
Tasks: 174 total, 4 running, 170 sleeping, 0 stopped, 0 zombie
Cpu(s): 20.5%us, 2.6%sy, 0.0%ni, 76.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 4108288k total, 1628164k used, 2480124k free, 713540k buffers
Swap: 3999740k total, 0k used, 3999740k free, 560972k cached
```

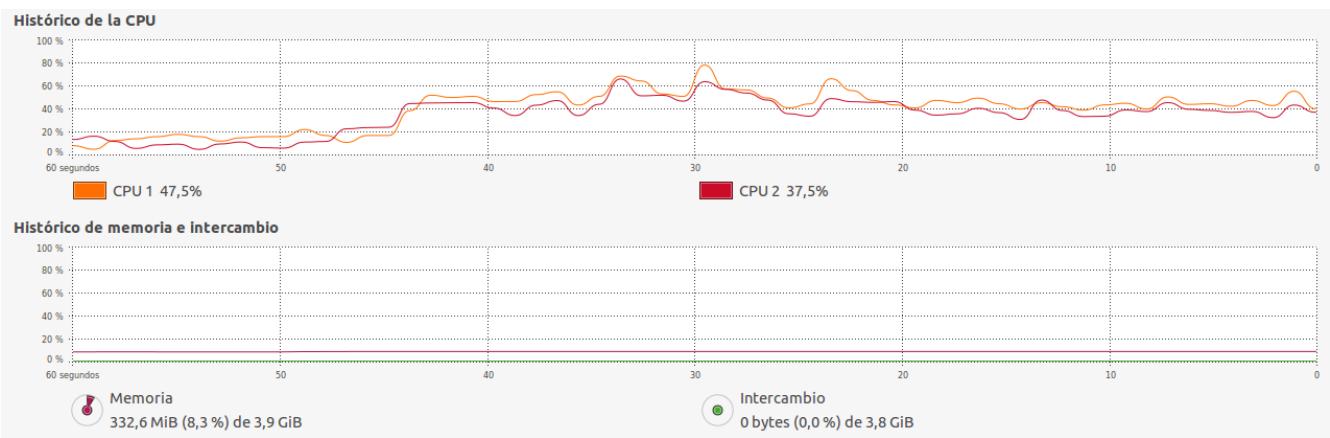
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6444	felipe	20	0	109m	40m	9480	S	52	1.0	0:46.15	Sample-PointVie
4950	root	20	0	70104	30m	12m	R	6	0.7	3:08.24	Xorg
5104	felipe	20	0	71388	24m	8396	S	2	0.6	0:22.02	compiz
758	root	20	0	0	0	0	S	0	0.0	0:09.52	iwlagn



Sample-NiSimpleViewer

```
top - 18:53:09 up 5:35, 3 users, load average: 0.05, 0.14, 0.43
Tasks: 175 total, 3 running, 172 sleeping, 0 stopped, 0 zombie
Cpu(s): 34.9%us, 9.9%sy, 0.0%ni, 55.0%id, 0.2%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 4108288k total, 1641568k used, 2466720k free, 714852k buffers
Swap: 3999740k total, 0k used, 3999740k free, 584992k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6653	felipe	20	0	83360	22m	11m	S	31	0.6	0:03.79	Sample-NiSimple
4950	root	20	0	76928	33m	13m	R	31	0.8	4:37.64	Xorg
6448	felipe	20	0	86196	18m	15m	S	14	0.5	1:38.72	gnome-system-mo



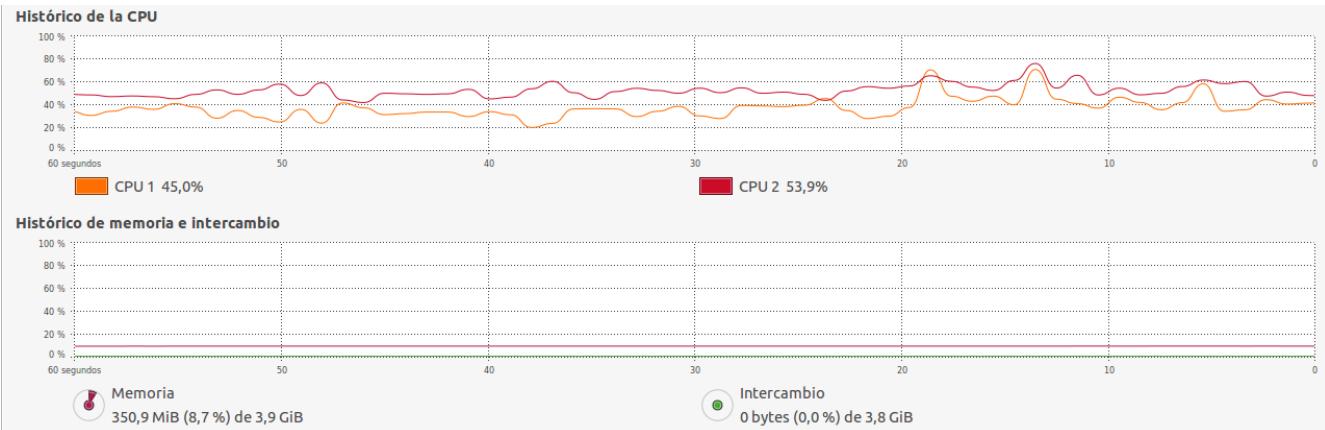
Sample-SceneAnalysis

```

top - 18:45:15 up 5:28, 3 users, load average: 0.20, 0.17, 0.61
Tasks: 175 total, 3 running, 172 sleeping, 0 stopped, 0 zombie
Cpu(s): 37.4%us, 4.9%sy, 0.0%ni, 57.4%id, 0.0%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 4108288k total, 1654588k used, 2453700k free, 713872k buffers
Swap: 3999740k total, 0k used, 3999740k free, 582256k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6486	felipe	20	0	109m	42m	9148	R	66	1.1	0:36.38	Sample-SceneAna
6448	felipe	20	0	86556	19m	15m	S	13	0.5	0:24.10	gnome-system-mo
4950	root	20	0	76280	32m	13m	S	12	0.8	3:32.88	Xorg
5104	felipe	20	0	71388	24m	8396	S	4	0.6	0:27.67	compiz



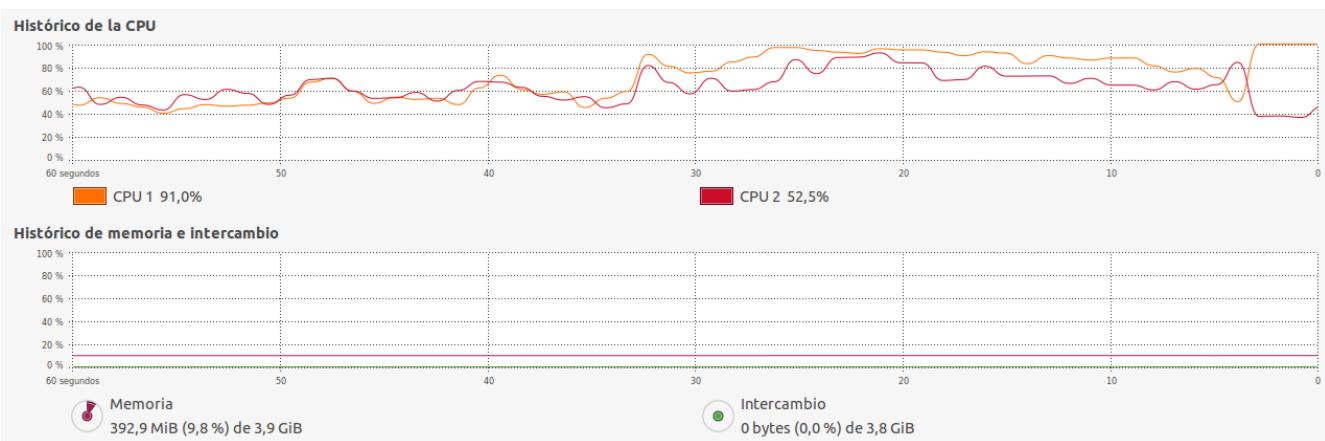
Sample-NiUserTracker

```

top - 18:55:01 up 5:37, 3 users, load average: 0.37, 0.20, 0.41
Tasks: 175 total, 2 running, 173 sleeping, 0 stopped, 0 zombie
Cpu(s): 47.0%us, 4.7%sy, 0.0%ni, 48.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 4108288k total, 1727232k used, 2381056k free, 715128k buffers
Swap: 3999740k total, 0k used, 3999740k free, 610528k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6710	felipe	20	0	186m	84m	14m	S	82	2.1	0:39.73	Sample-NiUserTr
6448	felipe	20	0	86208	18m	15m	S	14	0.5	1:56.27	gnome-system-mo
4950	root	20	0	77208	33m	13m	S	14	0.8	5:02.12	Xorg



Como se observa en los resultados obtenidos, lo que consume mayor porcentaje de CPU son los ejemplos donde se realiza trackeo, análisis de escena y se muestra el video por pantalla. Este resultado era totalmente esperable, debido a que para ambos casos se necesitan algoritmos más costosos. El mayor porcentaje de CPU utilizado fue para el ejemplo de trackeo, donde se utilizó un 82%. Además, se aprecia que cuando el video (mapa de profundidad, IR o imagen) no se muestra en pantalla, se reduce significativamente el consumo de CPU, como en los dos primeros casos donde se llegó a un máximo de 33%.

En todos los casos, el uso de memoria es bajo, siguiendo el mismo comportamiento recién descrito, donde influye la complejidad de los algoritmos a utilizar por el ejemplo.