

# *"Entorno de simulación para el entrenamiento mediante Reinforcement Learning del vuelo autónomo de un cuadricóptero"*

## Requisitos

- Ubuntu 16.04 con OpenGL 3.3 o superior.
- Git
- Python

## Instalación del framework

### 1. Instalación de Unreal Engine 4

Fuente: [https://github.com/danijc8196/MyRosAirSim/blob/tfg\\_definitive/docs/build\\_linux.md](https://github.com/danijc8196/MyRosAirSim/blob/tfg_definitive/docs/build_linux.md)

```
> cd ~/
> git clone -b 4.17 https://github.com/EpicGames/UnrealEngine.git
> cd UnrealEngine
> git checkout af96417313a908b20621a443175ba91683c238c8
> ./Setup.sh
> ./GenerateProjectFiles.sh
> make
```

### 2. Instalación de AirSim

Fuente: [https://github.com/danijc8196/MyRosAirSim/blob/tfg\\_definitive/docs/build\\_linux.md](https://github.com/danijc8196/MyRosAirSim/blob/tfg_definitive/docs/build_linux.md)

```
> cd ~/
> git clone https://github.com/danijc8196/MyRosAirSim.git
> cd AirSim
> ./setup.sh
> ./build.sh
```

### 3. Instalación de ROS (Kinetic)

Fuente: <http://wiki.ros.org/kinetic/Installation/Ubuntu>

```
> sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'
> sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
421C365BD9FF1F717815A3895523BAEEB01FA116
> sudo apt-get update
```

```
> sudo apt-get install ros-kinetic-desktop-full
> sudo rosdep init
> rosdep update
> echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
> source ~/.bashrc
> sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
> mkdir -p ~/ROS_WS/src # ROS_WS será el workspace de ROS
> cd ~/ROS_WS/
> catkin_make
```

#### 4. Instalación de PX4 (versión 1.6.4 modificada)

Fuente (requisitos): [http://dev.px4.io/en/setup/dev\\_env\\_linux.html](http://dev.px4.io/en/setup/dev_env_linux.html)

Fuente (autopilot): [https://github.com/danijc8196/MyRosAirSim/blob/tfg\\_definitive/docs/px4\\_build.md](https://github.com/danijc8196/MyRosAirSim/blob/tfg_definitive/docs/px4_build.md)

Descargar el archivo ubuntu\_sim\_nutt.sh de

[https://raw.githubusercontent.com/PX4/Devguide/master/build\\_scripts/ubuntu\\_sim\\_nutt.sh](https://raw.githubusercontent.com/PX4/Devguide/master/build_scripts/ubuntu_sim_nutt.sh)

##### # Instalación de los requisitos del autopilot

```
> source ubuntu_sim_nutt.sh
> sudo add-apt-repository ppa:george-edison55/cmake-3.x -y
> sudo apt-get update
> sudo apt-get install python-argparse git-core wget zip \
python-empy cmake build-essential genromfs -y
> sudo apt-get install python-serial openocd \
flex bison libncurses5-dev autoconf texinfo libftdi-dev libtool zlib1g-dev -y
> sudo apt-get install python-pip python-jinja2 -y
```

##### # Instalación del autopilot como paquete ROS

```
> cd ~/ROS_WS/src
> git clone https://github.com/danijc8196/MyPX4.git
> cd MyPX4
> make posix_sitl_default
> cd ~/ROS_WS
> catkin_make
```

#### 5. Instalación Gym

Fuente: <https://github.com/openai/gym>

```
> pip install gym
> cd ~/ROS_WS/src
> git clone https://github.com/danijc8196/gym_training.git
```

#### 6. Últimos pasos

- Configuración AirSim

Sobreescribir el archivo **Documents/AirSim/settings.json** con el siguiente contenido:

```
{
  "SeeDocsAt": "https://github.com/Microsoft/AirSim/blob/master/docs/settings.md",
  "SettingsVersion": 1.0,
  "SimMode": "Multirotor",
  "DefaultVehicleConfig": "PX4",

  "PX4": {
    "UseSerial": false,
    "LocalHostIp": "127.0.0.1",
    "UdpIp": "127.0.0.1",
    "UdpPort": 14560,
    "SitlIp": "127.0.0.1",
    "SitlPort": 14556,
    "QgcHostIp": "127.0.0.1",
    "QgcPort": 14550
  }
}
```

- Enlaces

Crear un link de la librería AirSimClient en el paquete ROS de Gym Training:

```
> ln -s ~/MyRosAirSim/rosbridge/airsim_bridge/scripts/AirSimClient.py ~/ROS_WS/src/gym_training/src
```

Crear un link del paquete ROS image\_publisher a un workspace ROS para poder ejecutarlo:

```
> ln -s ~/MyRosAirSim/image_publisher/ ~/ROS_WS/src/
```

## Iniciar simulación

Abrir 2 terminales y arrancar el entorno de simulación y el controlador de vuelo:

### Terminal 1: AirSim

```
> cd ~/
> ./UnrealEngine-4.18/Engine/Binaries/Linux/UE4Editor
```

### Terminal 2: PX4

```
roslaunch px4 customPX4.launch
```

Esperar a que el Unreal Engine cargue, y luego presionar el botón "Play" (ALT+P)

Una vez comience la simulación, en otro terminal se ejecuta el nodo que publica las imágenes de AirSim (opcional)

### Terminal 3: image\_publisher (opcional)

```
roslaunch image_publisher publisher.launch
```

Por último, ejecutar el paquete Gym Training (nodo Offboard + nodo Training)

### Terminal 4: gym\_training

```
roslaunch gym_training gym.launch
```

## Repositorios framework:

RotoLearning: <https://github.com/danijc8196/RotoLearning>

MyRosAirSim: <https://github.com/danijc8196/MyRosAirSim>

MyPX4: <https://github.com/danijc8196/MyPX4>

Gym Training: [https://github.com/danijc8196/gym\\_training](https://github.com/danijc8196/gym_training)