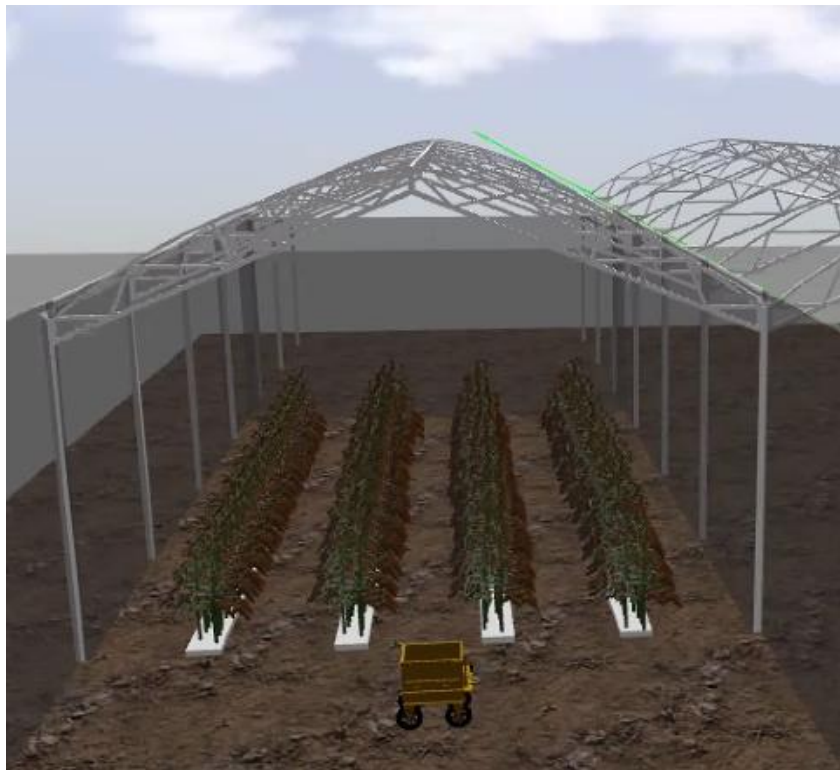


Simulación - UMA



Manual de comandos

Índice

Configuración del sistema

Instalación de software	4
Hector gazebo plugins	5
GPS navigation	10
Semestre i	15
UMA grover	20
Entorno	22
Actualización de paquetes	27

Habilitación de la simulación

Funciones de navegación del UMA	32
Navegación semiautónoma	33
Navegación autónoma	36

Configuración del sistema

Instalación de software	4
GPS navigation	5
Semestre i	10
UMA grover	15
Hector gazebo plugins	20
Entorno	22
Actualización de paquetes	27

Instalación de software

En esta sección se muestra la forma correcta de hacer la instalación de los paquetes de ROS asi como de Hector gazebo plugins para obtener un debido funcionamiento al momento de correr la simulación.

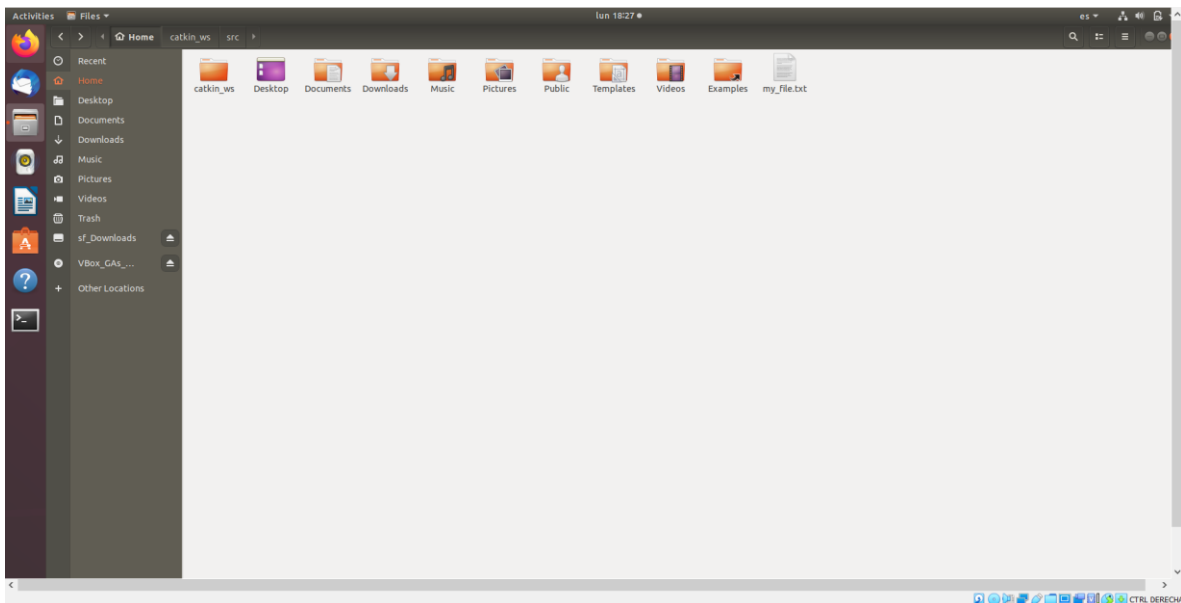
Es necesario observar que no se despliegue ningún error cuando se ejecute los comandos necesarios para realizar la clonación del repositorio para obtener los paquetes de ros gps navigation, semestre i y uma robot.

PREREQUISITO, si no ha instalado y configurado su ambiente de ROS, realizar el siguiente tutorial:
<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

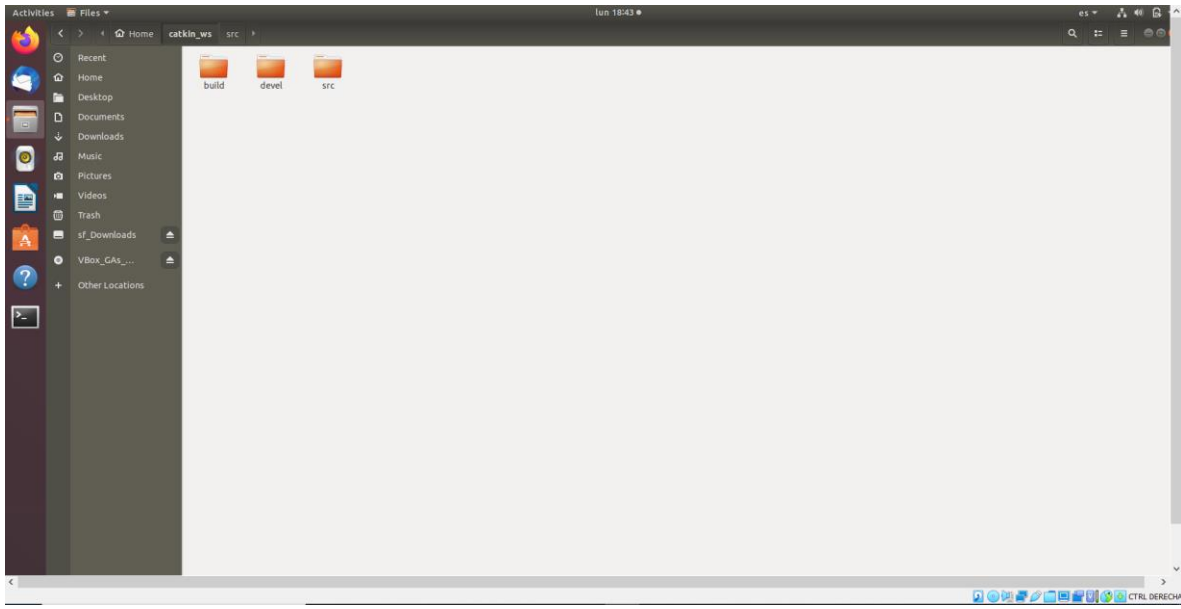
GPS navigation

Para clonar el paquete de ROS de gps_navigation del repositorio de GitHub groveragronics se deben seguir los siguientes pasos:

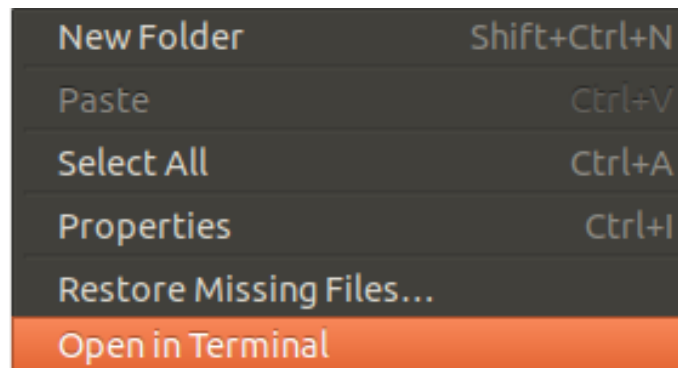
- Abrir Files.



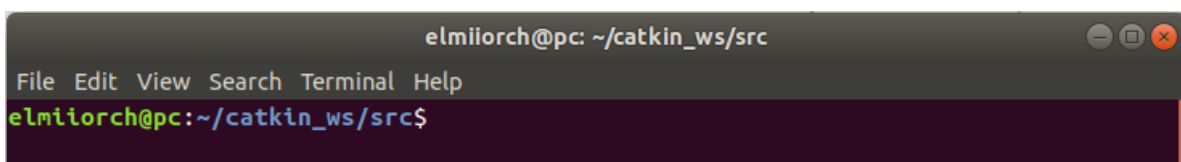
- Entrar a la carpeta catkin_ws.



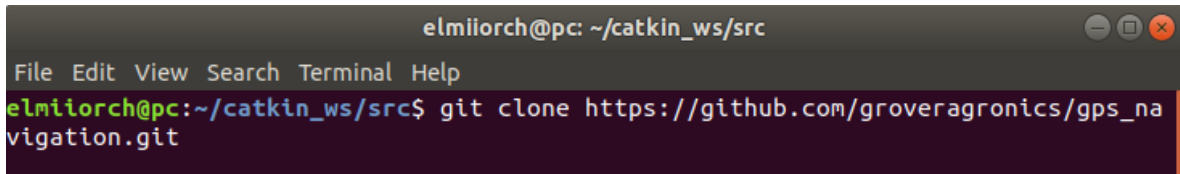
- Entrar a la carpeta src, dar clic derecho y seleccionar la opción Open in Terminal.



Al seleccionar esa opción se desplegará un terminal como esta:

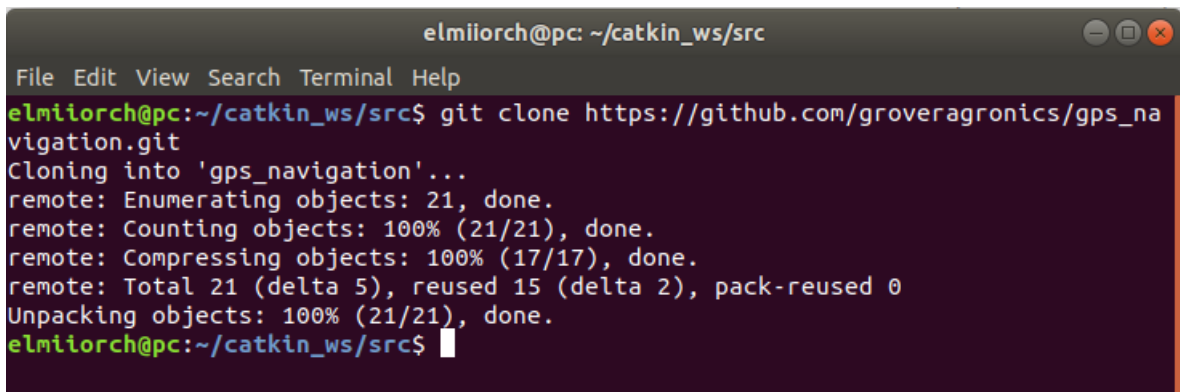


- Sobre esa terminal escribe y ejecutar el siguiente comando:
\$ git clone
https://github.com/groveragronics/gps_navigation.git



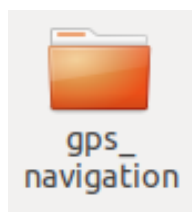
```
elmilorch@pc: ~/catkin_ws/src
File Edit View Search Terminal Help
elmilorch@pc:~/catkin_ws/src$ git clone https://github.com/groveragronics/gps_navigation.git
```

Aparecerá algo como esto, indicando que se clono correctamente el paquete de ROS con todos tus archivos correspondientes.

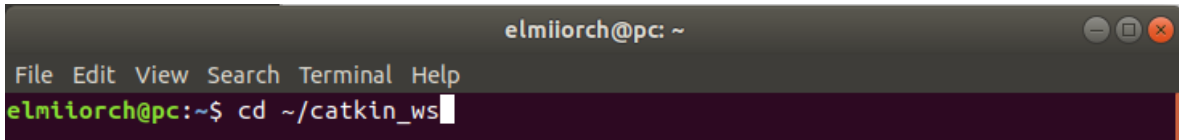


```
elmilorch@pc: ~/catkin_ws/src
File Edit View Search Terminal Help
elmilorch@pc:~/catkin_ws/src$ git clone https://github.com/groveragronics/gps_navigation.git
Cloning into 'gps_navigation'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 21 (delta 5), reused 15 (delta 2), pack-reused 0
Unpacking objects: 100% (21/21), done.
elmilorch@pc:~/catkin_ws/src$
```

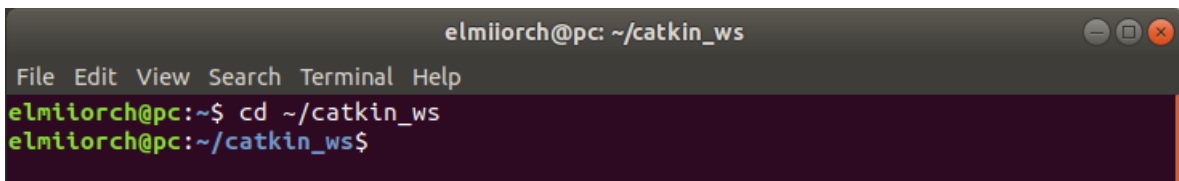
Si entramos a la carpeta de src podremos notar que tenemos una nueva carpeta llamada gps_navigation, comprobando que nuestro proceso fue exitoso.



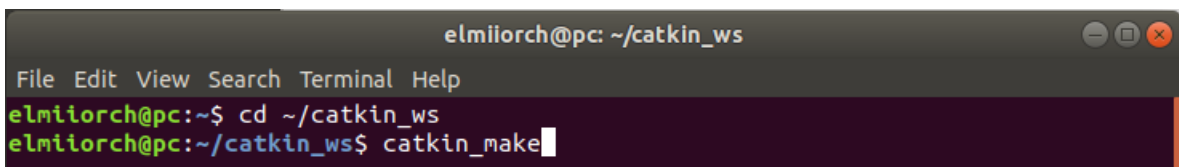
- Ahora debemos escribir y ejecutar el siguiente comando:
`$ cd ~/catkin_ws`

A terminal window titled 'elmiiorch@pc: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'elmiiorch@pc:~\$' and the command 'cd ~/catkin_ws' is being typed, with a cursor at the end of the line.

La función de este comando es llevarnos a la dirección de la carpeta catkin_ws.

A terminal window titled 'elmiiorch@pc: ~/catkin_ws' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'elmiiorch@pc:~\$' and the command 'cd ~/catkin_ws' has been executed. The prompt has changed to 'elmiiorch@pc:~/catkin_ws\$'.

- Sobre la misma terminal ejecute lo siguiente:
`$ catkin_make`

A terminal window titled 'elmiiorch@pc: ~/catkin_ws' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'elmiiorch@pc:~/catkin_ws\$' and the command 'catkin_make' is being typed, with a cursor at the end of the line.

Ahora procederemos a construir el espacio de trabajo con catkin_make.


```

elmiorch@pc: ~/catkin_ws
File Edit View Search Terminal Help
-- ==> add_subdirectory(two_wheels_description)
-- +++ processing catkin package: 'beginner_tutorials'
-- ==> add_subdirectory(beginner_tutorials)
-- +++ processing catkin package: 'semanai-robotica-espacial'
-- ==> add_subdirectory(semanai-robotica-espacial)
-- +++ processing catkin package: 'semestre_i'
-- ==> add_subdirectory(semestre_i)
-- +++ processing catkin package: 'gps_navigation'
-- ==> add_subdirectory(gps_navigation)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'myhector_slam_config'
-- ==> add_subdirectory(myhector_slam_config)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'uma_grover'
-- ==> add_subdirectory(uma_robot)
-- +++ processing catkin package: 'two_wheels_navigation'
-- ==> add_subdirectory(two_wheels_navigation)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/elmiorch/catkin_ws/build
####
#### Running command: "make -j4 -l4" in "/home/elmiorch/catkin_ws/build"
####
elmiorch@pc:~/catkin_ws$

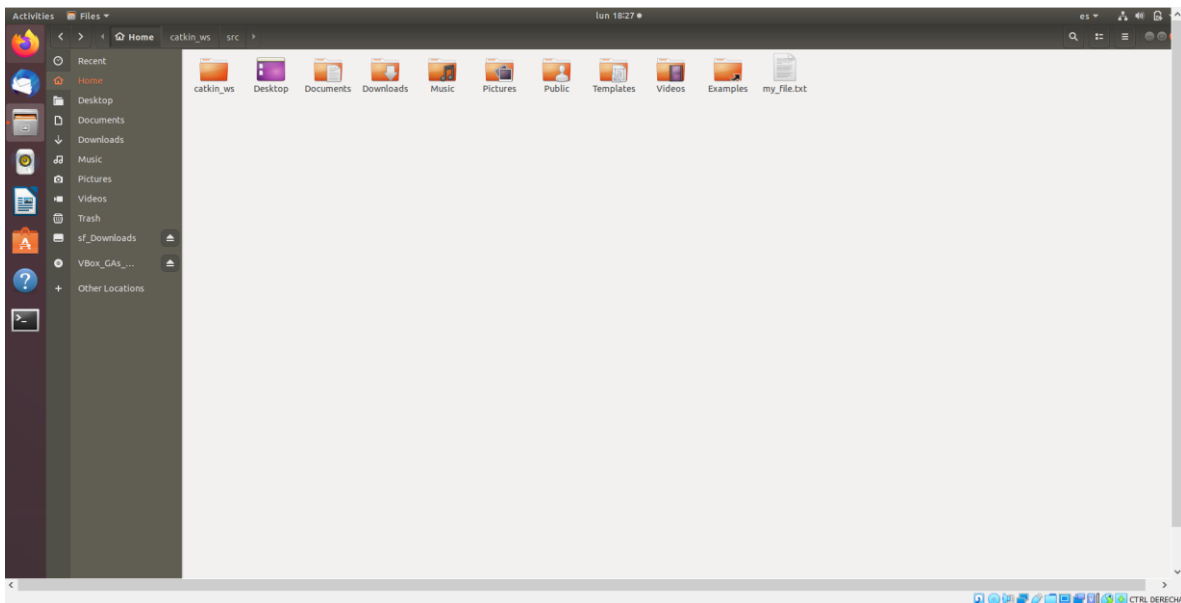
```

Acabando de realizar este proceso tendremos instalado correctamente nuestro paquete de ROS de gps_navigation.

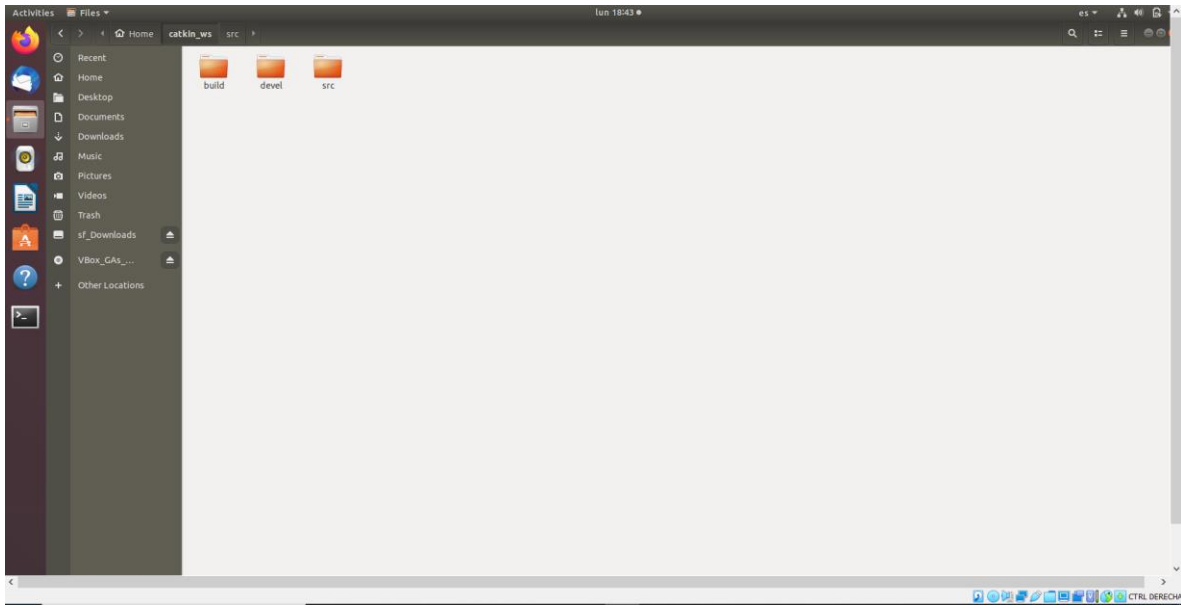
Semestre i

Para clonar el paquete de ROS de semestre i del repositorio de Git Hub groveragronics se deben seguir los siguientes pasos:

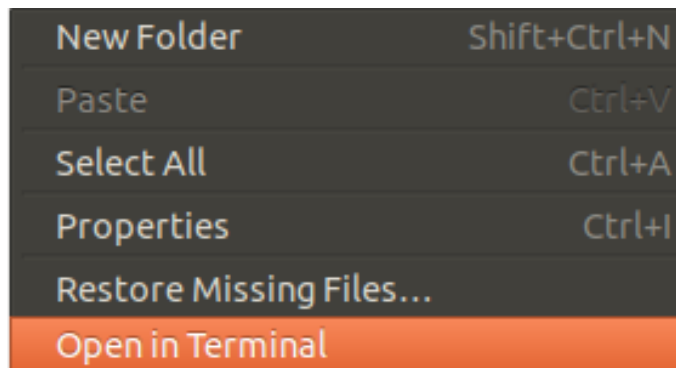
- Abrir Files.



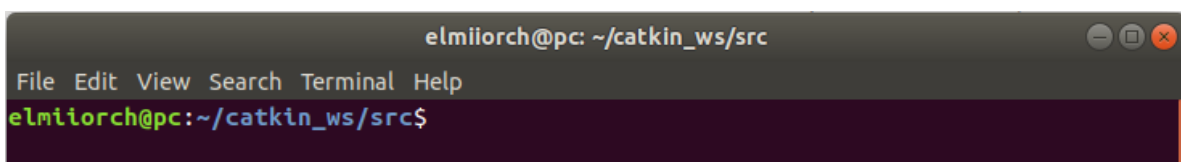
- Entrar a la carpeta catkin_ws.



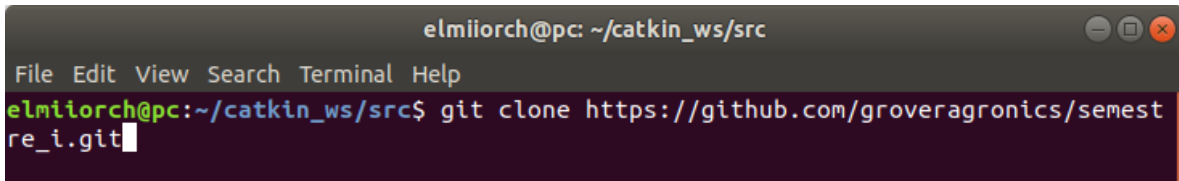
- Entrar a la carpeta src, dar clic derecho y seleccionar la opción Open in Terminal.



Al seleccionar esa opción se desplegará un terminal como esta:

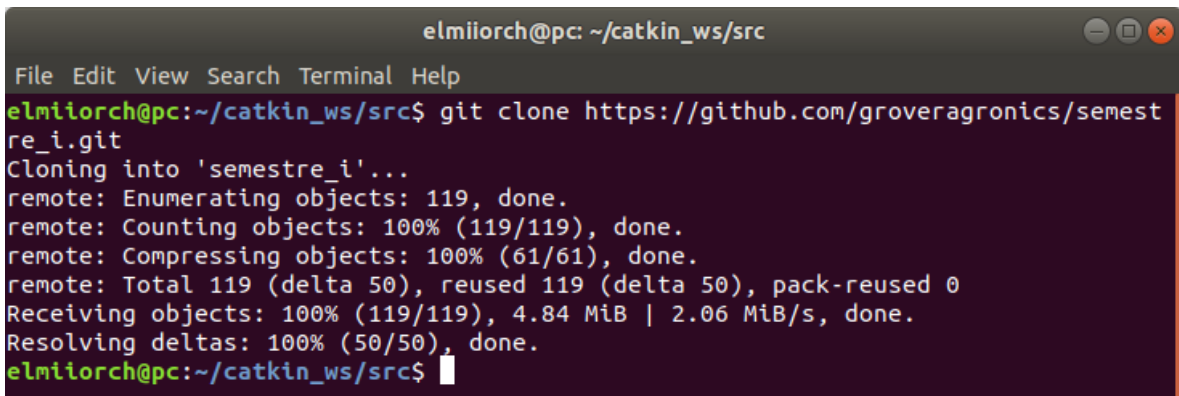


- Sobre esa terminal escribe y ejecuta el siguiente comando:
\$ git clone
https://github.com/groveragronics/semestre_i.git



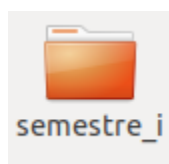
```
elmiliorch@pc: ~/catkin_ws/src
File Edit View Search Terminal Help
elmiliorch@pc:~/catkin_ws/src$ git clone https://github.com/groveragronics/semestre_i.git
```

Aparecerá algo como esto, indicando que se clono correctamente el paquete de ROS con todos tus archivos correspondientes.

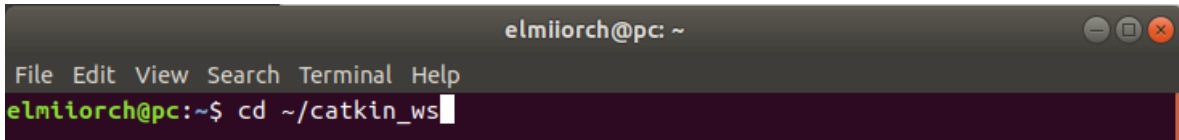


```
elmiliorch@pc: ~/catkin_ws/src
File Edit View Search Terminal Help
elmiliorch@pc:~/catkin_ws/src$ git clone https://github.com/groveragronics/semestre_i.git
Cloning into 'semestre_i'...
remote: Enumerating objects: 119, done.
remote: Counting objects: 100% (119/119), done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 119 (delta 50), reused 119 (delta 50), pack-reused 0
Receiving objects: 100% (119/119), 4.84 MiB | 2.06 MiB/s, done.
Resolving deltas: 100% (50/50), done.
elmiliorch@pc:~/catkin_ws/src$
```

Si entramos a la carpeta de src podremos notar que tenemos una nueva carpeta llamada semestre_i, comprobando que nuestro proceso fue exitoso.

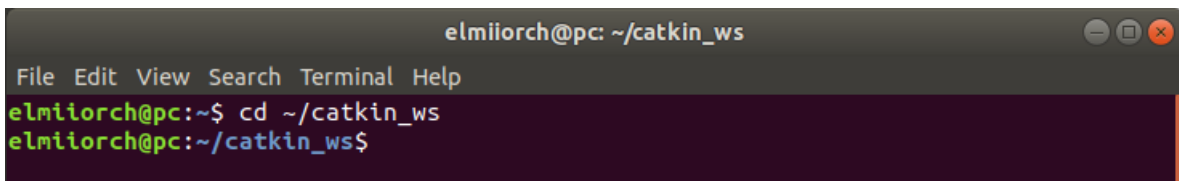


- Ahora debemos escribir y ejecutar el siguiente comando:
`$ cd ~/catkin_ws`

A terminal window titled 'elmiorch@pc: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command prompt shows 'elmiorch@pc:~\$' followed by 'cd ~/catkin_ws' with a cursor at the end.

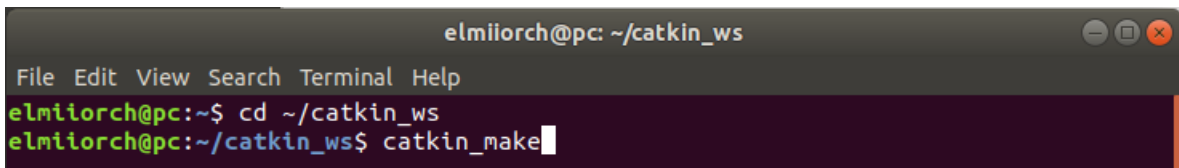
```
elmiorch@pc:~$ cd ~/catkin_ws
```

La función de este comando es llevarnos a la dirección de la carpeta catkin_ws.

A terminal window titled 'elmiorch@pc: ~/catkin_ws' with a menu bar (File, Edit, View, Search, Terminal, Help). The command prompt shows 'elmiorch@pc:~\$ cd ~/catkin_ws' followed by a new line with 'elmiorch@pc:~/catkin_ws\$'.

```
elmiorch@pc:~$ cd ~/catkin_ws
elmiorch@pc:~/catkin_ws$
```

- Sobre la misma terminal ejecute lo siguiente:
`$ catkin_make`

A terminal window titled 'elmiorch@pc: ~/catkin_ws' with a menu bar (File, Edit, View, Search, Terminal, Help). The command prompt shows 'elmiorch@pc:~\$ cd ~/catkin_ws' followed by a new line with 'elmiorch@pc:~/catkin_ws\$ catkin_make' and a cursor at the end.

```
elmiorch@pc:~$ cd ~/catkin_ws
elmiorch@pc:~/catkin_ws$ catkin_make
```

Ahora procederemos a construir el espacio de trabajo con catkin_make.

```

elmiorch@pc: ~/catkin_ws
File Edit View Search Terminal Help
-- ==> add_subdirectory(two_wheels_description)
-- +++ processing catkin package: 'beginner_tutorials'
-- ==> add_subdirectory(beginner_tutorials)
-- +++ processing catkin package: 'semanai-robotica-espacial'
-- ==> add_subdirectory(semanai-robotica-espacial)
-- +++ processing catkin package: 'semestre_i'
-- ==> add_subdirectory(semestre_i)
-- +++ processing catkin package: 'gps_navigation'
-- ==> add_subdirectory(gps_navigation)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'myhector_slam_config'
-- ==> add_subdirectory(myhector_slam_config)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'uma_grover'
-- ==> add_subdirectory(uma_robot)
-- +++ processing catkin package: 'two_wheels_navigation'
-- ==> add_subdirectory(two_wheels_navigation)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/elmiorch/catkin_ws/build
####
#### Running command: "make -j4 -l4" in "/home/elmiorch/catkin_ws/build"
####
elmiorch@pc:~/catkin_ws$

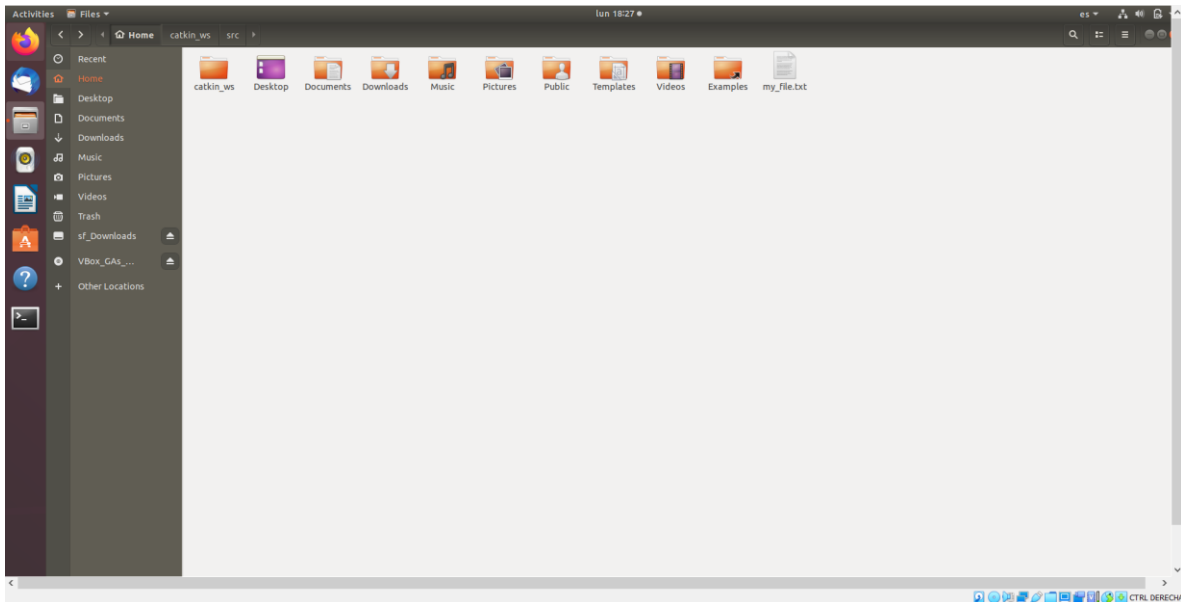
```

Acabando de realizar este proceso tendremos instalado correctamente nuestro paquete de ROS de semestre_i.

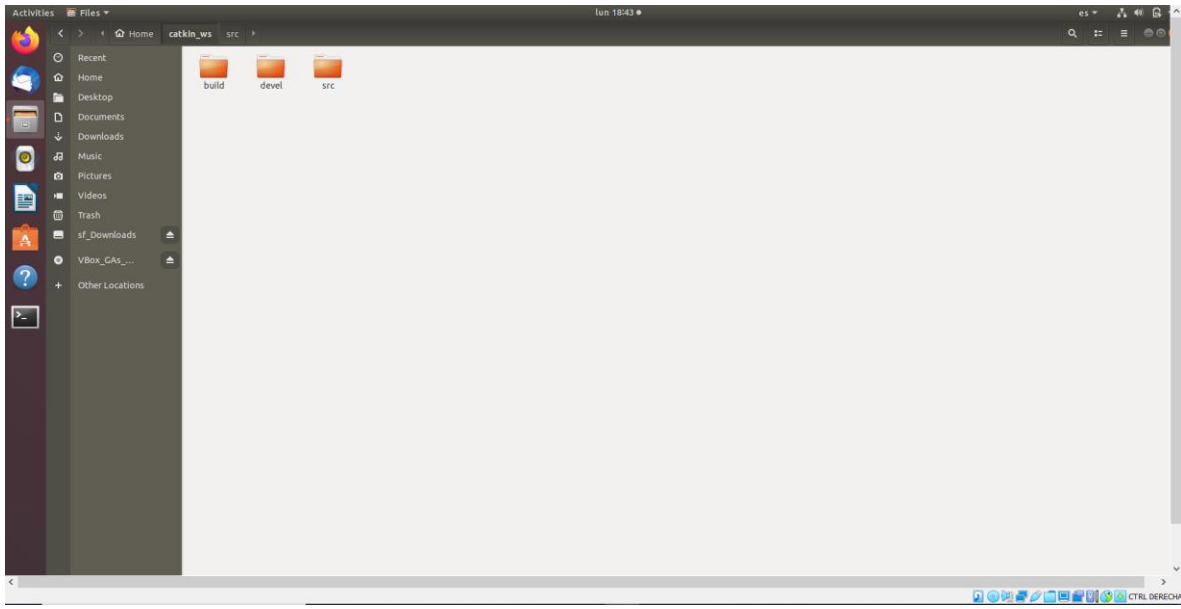
UMA grover

Para clonar el paquete de ROS de uma grover del repositorio de Git Hub groveragronics se deben seguir los siguientes pasos:

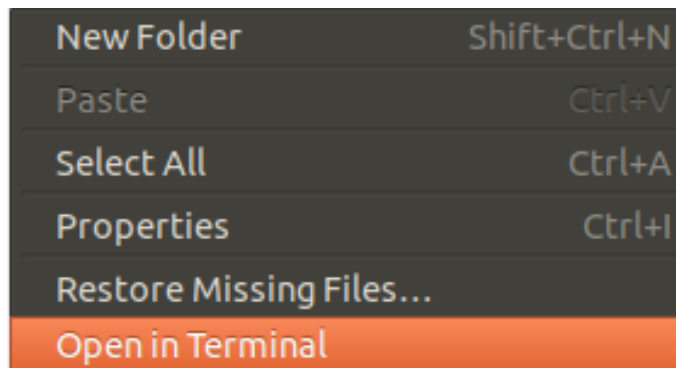
- Abrir Files.



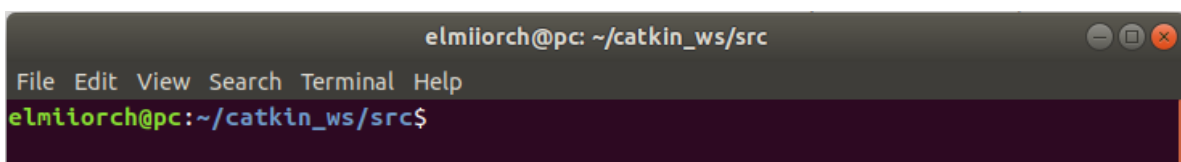
- Entrar a la carpeta catkin_ws.



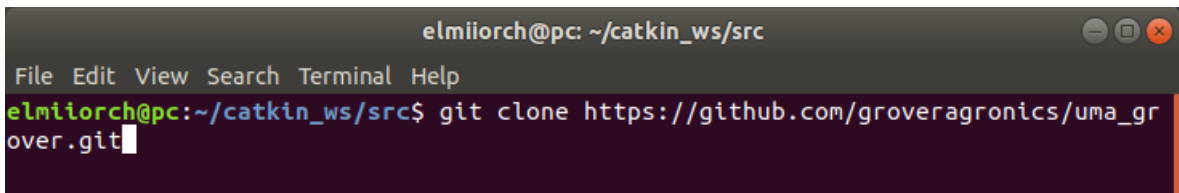
- Entrar a la carpeta src, dar clic derecho y seleccionar la opción Open in Terminal.



Al seleccionar esa opción se desplegará un terminal como esta:

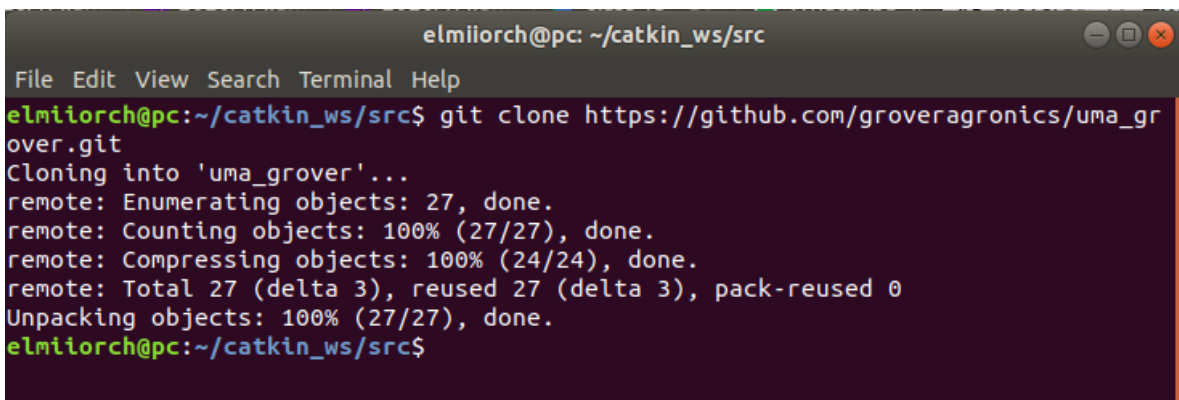


- Sobre esa terminal escribe y ejecutar el siguiente comando:
\$ git clone
https://github.com/groveragronics/uma_grover.git

A terminal window titled 'elmiiorch@pc: ~/catkin_ws/src' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'git clone https://github.com/groveragronics/uma_grover.git' is partially entered at the prompt.

```
elmiiorch@pc: ~/catkin_ws/src
File Edit View Search Terminal Help
elmiiorch@pc:~/catkin_ws/src$ git clone https://github.com/groveragronics/uma_grover.git
```

Aparecerá algo como esto, indicando que se clono correctamente el paquete de ROS con todos tus archivos correspondientes.

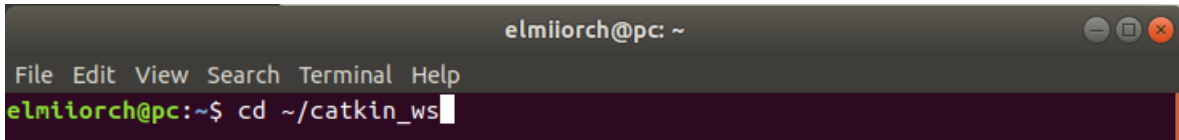
A terminal window titled 'elmiiorch@pc: ~/catkin_ws/src' showing the output of the git clone command. The output indicates successful cloning and unpacking of the repository.

```
elmiiorch@pc: ~/catkin_ws/src
File Edit View Search Terminal Help
elmiiorch@pc:~/catkin_ws/src$ git clone https://github.com/groveragronics/uma_grover.git
Cloning into 'uma_grover'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 27 (delta 3), reused 27 (delta 3), pack-reused 0
Unpacking objects: 100% (27/27), done.
elmiiorch@pc:~/catkin_ws/src$
```

Si entramos a la carpeta de src podremos notar que tenemos una nueva carpeta llamada uma_grover, comprobando que nuestro proceso fue exitoso.

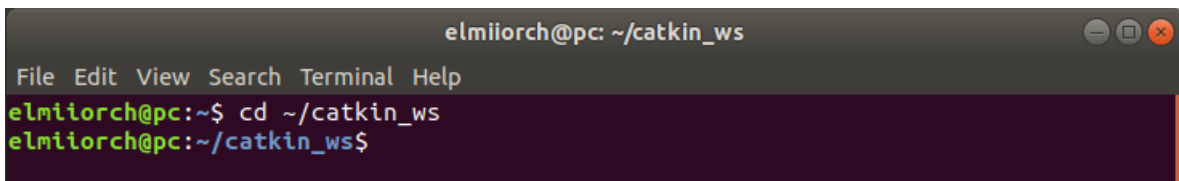


- Ahora debemos escribir y ejecutar el siguiente comando:
`$ cd ~/catkin_ws`

A terminal window titled 'elmiiorch@pc: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command prompt shows 'elmiiorch@pc:~\$' followed by 'cd ~/catkin_ws' with a cursor at the end.

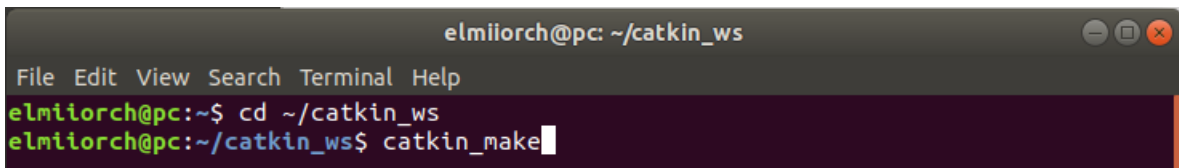
```
elmiiorch@pc:~$ cd ~/catkin_ws
```

La función de este comando es llevarnos a la dirección de la carpeta catkin_ws.

A terminal window titled 'elmiiorch@pc: ~/catkin_ws' with a menu bar (File, Edit, View, Search, Terminal, Help). The command prompt shows 'elmiiorch@pc:~\$ cd ~/catkin_ws' followed by a new line with 'elmiiorch@pc:~/catkin_ws\$'.

```
elmiiorch@pc:~$ cd ~/catkin_ws
elmiiorch@pc:~/catkin_ws$
```

- Sobre la misma terminal ejecute lo siguiente:
`$ catkin_make`

A terminal window titled 'elmiiorch@pc: ~/catkin_ws' with a menu bar (File, Edit, View, Search, Terminal, Help). The command prompt shows 'elmiiorch@pc:~\$ cd ~/catkin_ws' followed by a new line with 'elmiiorch@pc:~/catkin_ws\$ catkin_make' and a cursor at the end.

```
elmiiorch@pc:~$ cd ~/catkin_ws
elmiiorch@pc:~/catkin_ws$ catkin_make
```

Ahora procederemos a construir el espacio de trabajo con catkin_make.

```

elmiorch@pc: ~/catkin_ws
File Edit View Search Terminal Help
-- ==> add_subdirectory(two_wheels_description)
-- +++ processing catkin package: 'beginner_tutorials'
-- ==> add_subdirectory(beginner_tutorials)
-- +++ processing catkin package: 'semanai-robotica-espacial'
-- ==> add_subdirectory(semanai-robotica-espacial)
-- +++ processing catkin package: 'semestre_i'
-- ==> add_subdirectory(semestre_i)
-- +++ processing catkin package: 'gps_navigation'
-- ==> add_subdirectory(gps_navigation)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'myhector_slam_config'
-- ==> add_subdirectory(myhector_slam_config)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'uma_grover'
-- ==> add_subdirectory(uma_robot)
-- +++ processing catkin package: 'two_wheels_navigation'
-- ==> add_subdirectory(two_wheels_navigation)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/elmiorch/catkin_ws/build
####
#### Running command: "make -j4 -l4" in "/home/elmiorch/catkin_ws/build"
####
elmiorch@pc:~/catkin_ws$

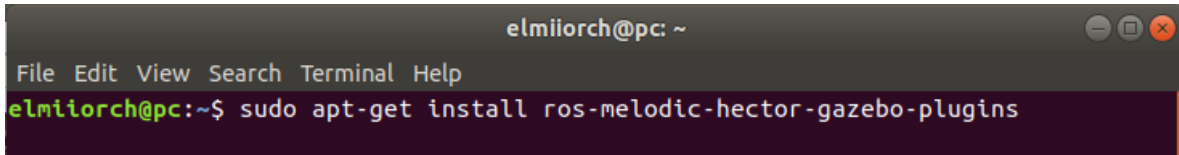
```

Acabando de realizar este proceso tendremos instalado correctamente nuestro paquete de ROS de uma_grover.

Hector gazebo plugins

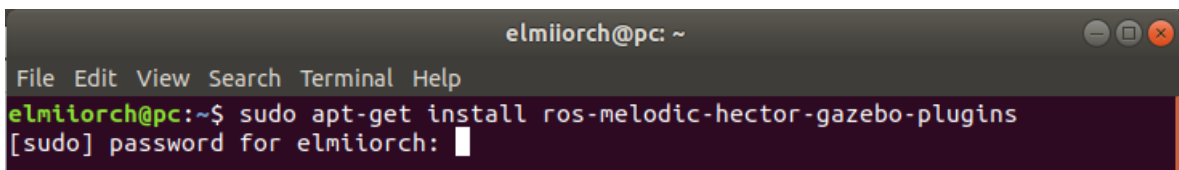
Para instalar instalar hector gazebo plugins en ROS se debe de realizar los siguientes pasos:

- Abrir una terminal correr el siguiente comando:
\$ sudo apt-get install ros-melodic-hector-gazebo-plugins

A terminal window titled 'elmiorch@pc: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'elmiorch@pc:~\$ sudo apt-get install ros-melodic-hector-gazebo-plugins' is entered and highlighted in green.

```
elmiorch@pc: ~  
File Edit View Search Terminal Help  
elmiorch@pc:~$ sudo apt-get install ros-melodic-hector-gazebo-plugins
```

Te pedirá que ingreses tu contraseña de usuaria.

A terminal window titled 'elmiorch@pc: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'elmiorch@pc:~\$ sudo apt-get install ros-melodic-hector-gazebo-plugins' is entered. Below it, the prompt '[sudo] password for elmiorch:' is shown with a cursor.

```
elmiorch@pc: ~  
File Edit View Search Terminal Help  
elmiorch@pc:~$ sudo apt-get install ros-melodic-hector-gazebo-plugins  
[sudo] password for elmiorch: █
```

Una vez puesta la contraseña tendrás que presionar tu tecla enter para continuar con la instalación.

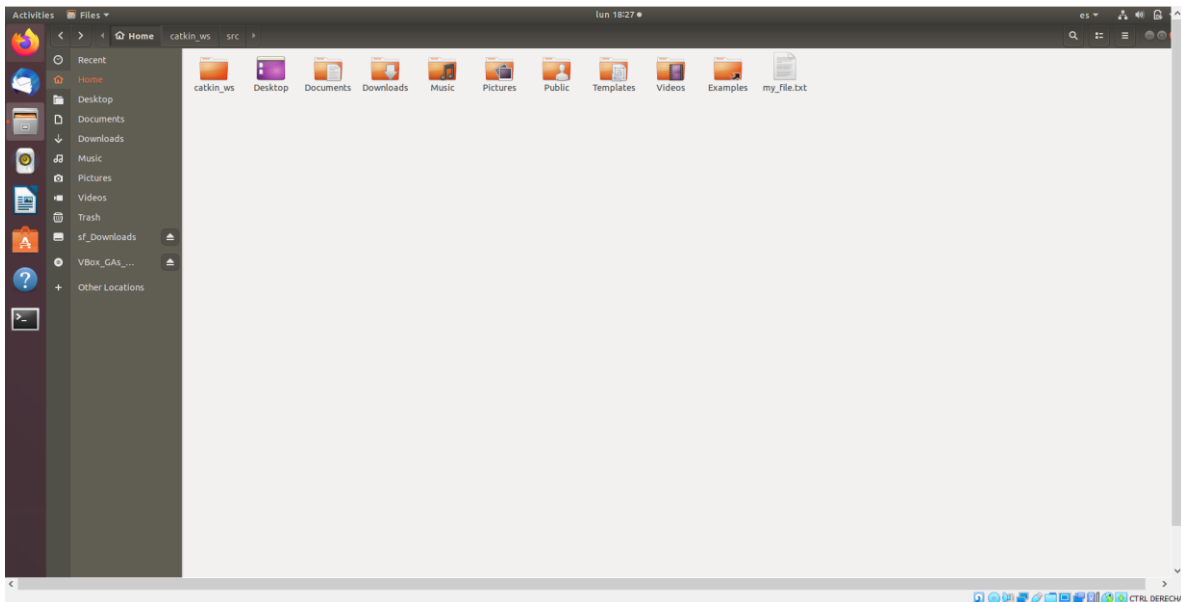
```
elmiiorch@pc: ~  
File Edit View Search Terminal Help  
elmiiorch@pc:~$ sudo apt-get install ros-melodic-hector-gazebo-plugins  
[sudo] password for elmiiorch:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  efibootmgr libfwup1  
Use 'sudo apt autoremove' to remove them.  
The following packages will be upgraded:  
  ros-melodic-hector-gazebo-plugins  
1 upgraded, 0 newly installed, 0 to remove and 156 not upgraded.  
Need to get 370 kB of archives.  
After this operation, 0 B of additional disk space will be used.  
Get:1 http://packages.ros.org/ros/ubuntu bionic/main amd64 ros-melodic-hector-ga  
zebo-plugins amd64 0.5.1-0bionic.20200501.202040 [370 kB]  
Fetched 370 kB in 1s (267 kB/s)  
(Reading database ... 256483 files and directories currently installed.)  
Preparing to unpack .../ros-melodic-hector-gazebo-plugins_0.5.1-0bionic.20200501  
.202040_amd64.deb ...  
Unpacking ros-melodic-hector-gazebo-plugins (0.5.1-0bionic.20200501.202040) over  
(0.5.1-0bionic.20200320.144204) ...  
Setting up ros-melodic-hector-gazebo-plugins (0.5.1-0bionic.20200501.202040) ...  
elmiiorch@pc:~$
```

Ya has instalado lo necesario para habilitar el GPS e IMU en la simulación de navegación autónoma de la sección de Funciones de navegación del UMA.

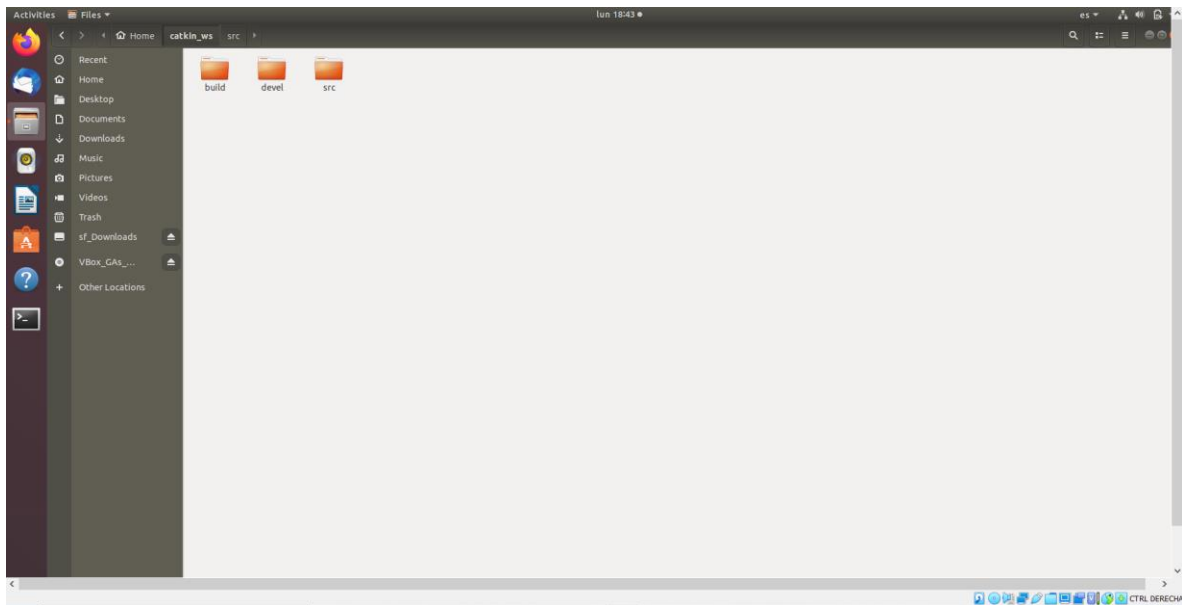
Entorno

Para dejar el entorno de la simulación en óptimas condiciones se debe de realizar los siguientes pasos:

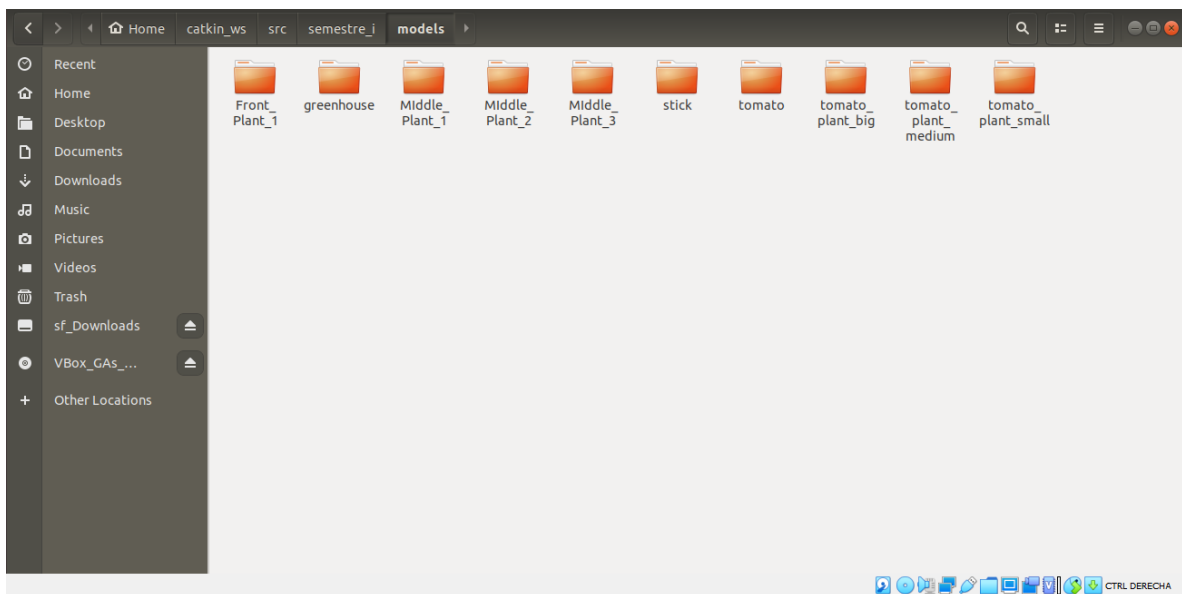
- Abrir Files.



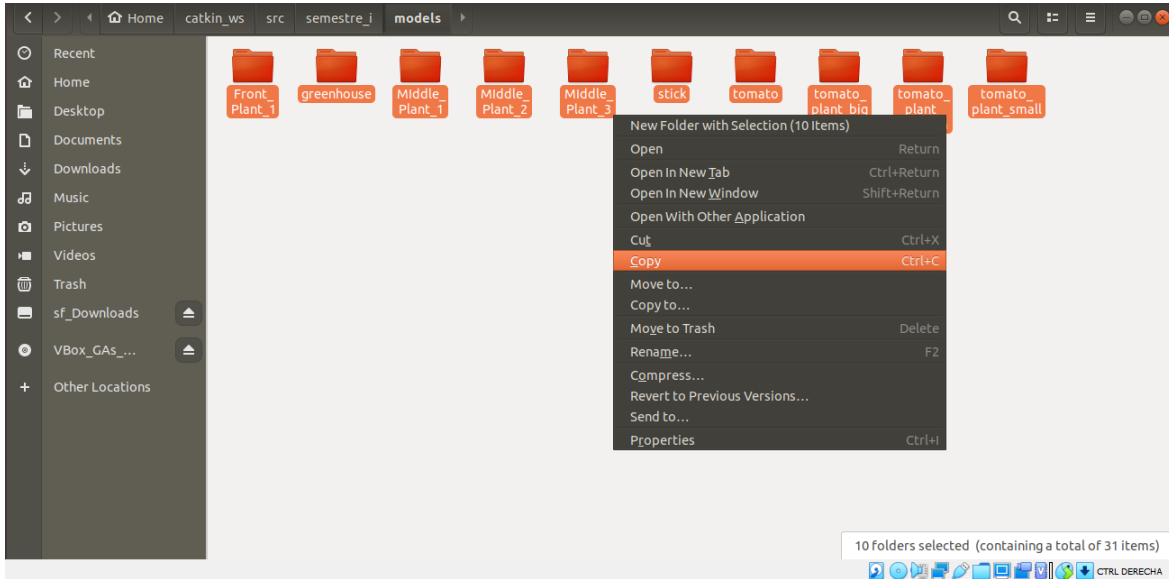
- Entrar a la carpeta catkin_ws.



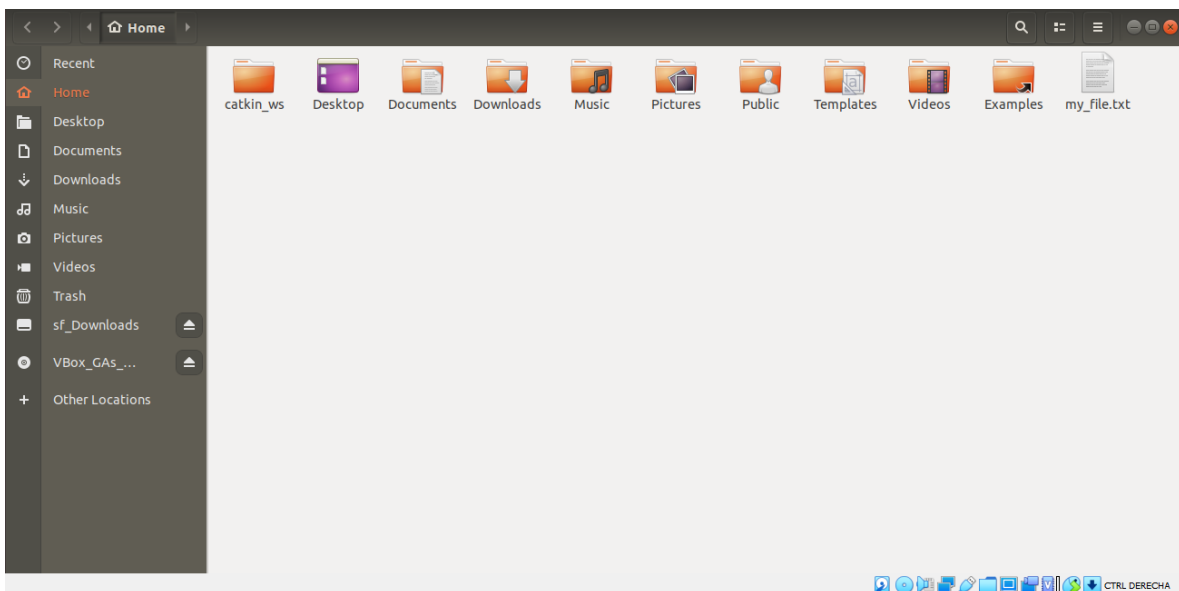
- Entrar a la carpeta src, después a la carpeta semestre_i y por último a la carpeta models.



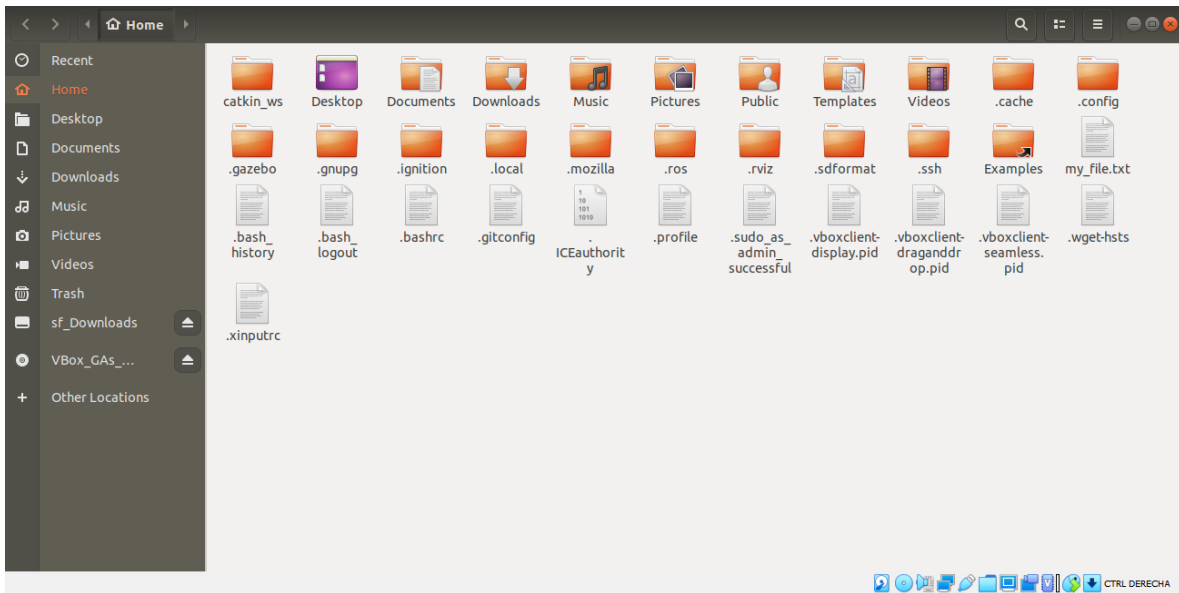
- Una vez dentro de la carpeta models se debe copiar su contenido.



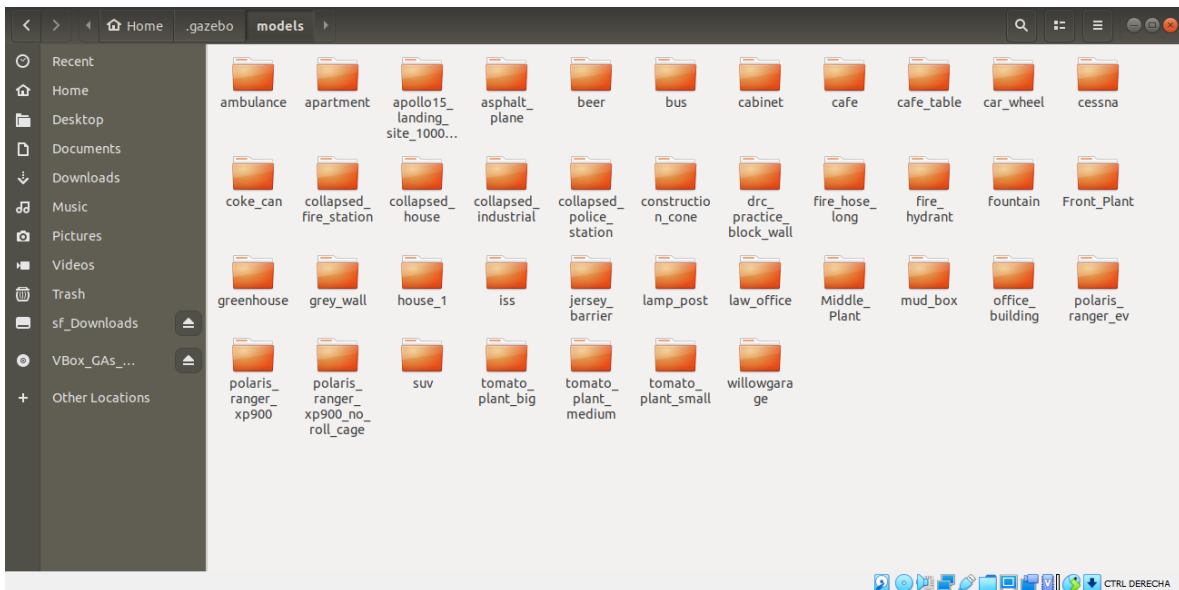
- Ahora abrimos files.



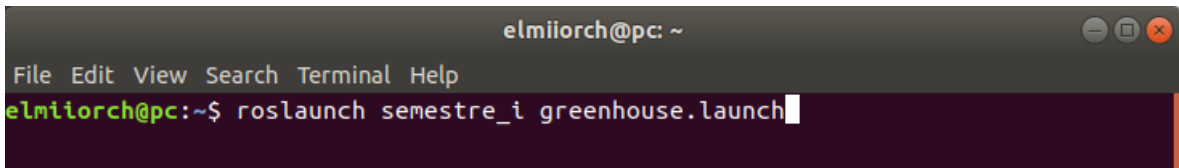
- Para activar las carpetas ocultas y mostrar la carpeta `.gazebo` se utiliza `ctrl + h`.



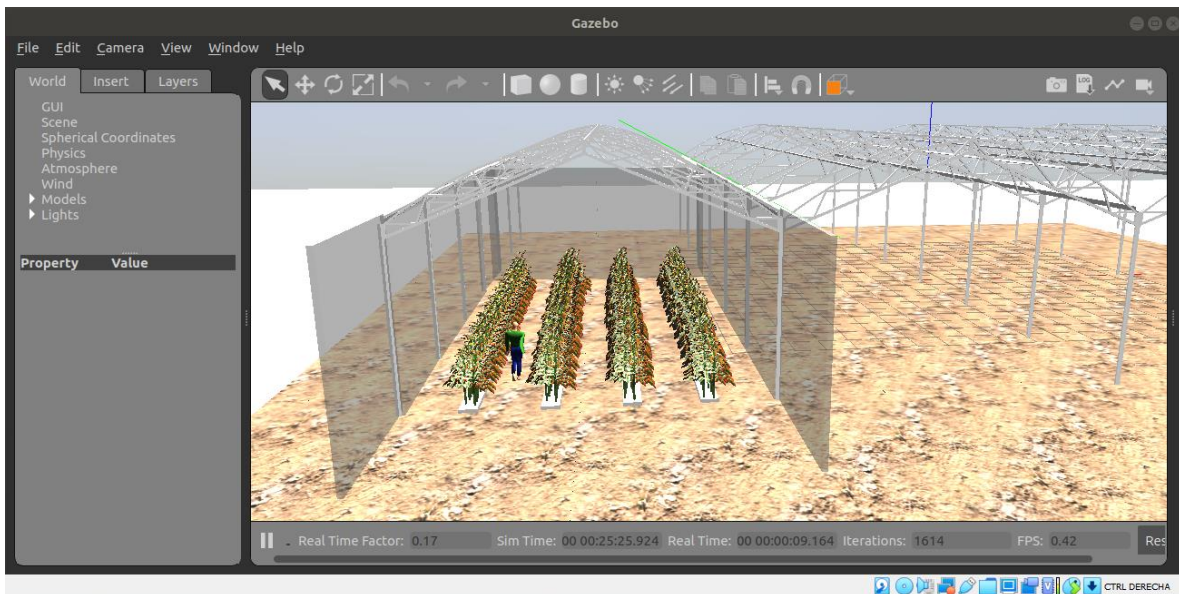
- Acceder a la carpeta `.gazebo` y después entrar a la carpeta `models`, una vez dentro se debe pegar el contenido que copiamos previamente de la carpeta `models` de `semestre_i`.



- Abrimos una nueva terminal para comprobar el correcto funcionamiento de la simulación, ahora bien se deberá escribir y ejecutar el siguiente comando:
\$ roslaunch semestre_i greenhouse.launch



Al ejecutar el comando se deberá de desplegar la siguiente ventana de Gazebo.

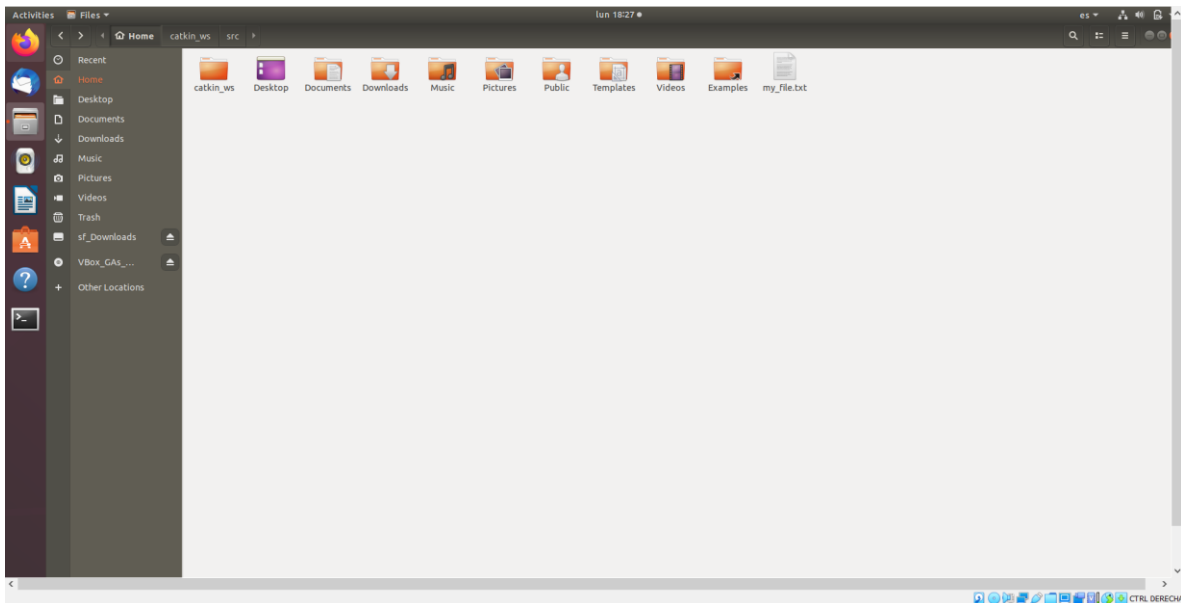


- En hora buena, ya has concluido con la configuración de tu sistema, ahora podrás continuar con la siguiente sección de este manual la cual es la habilitación del sistema.

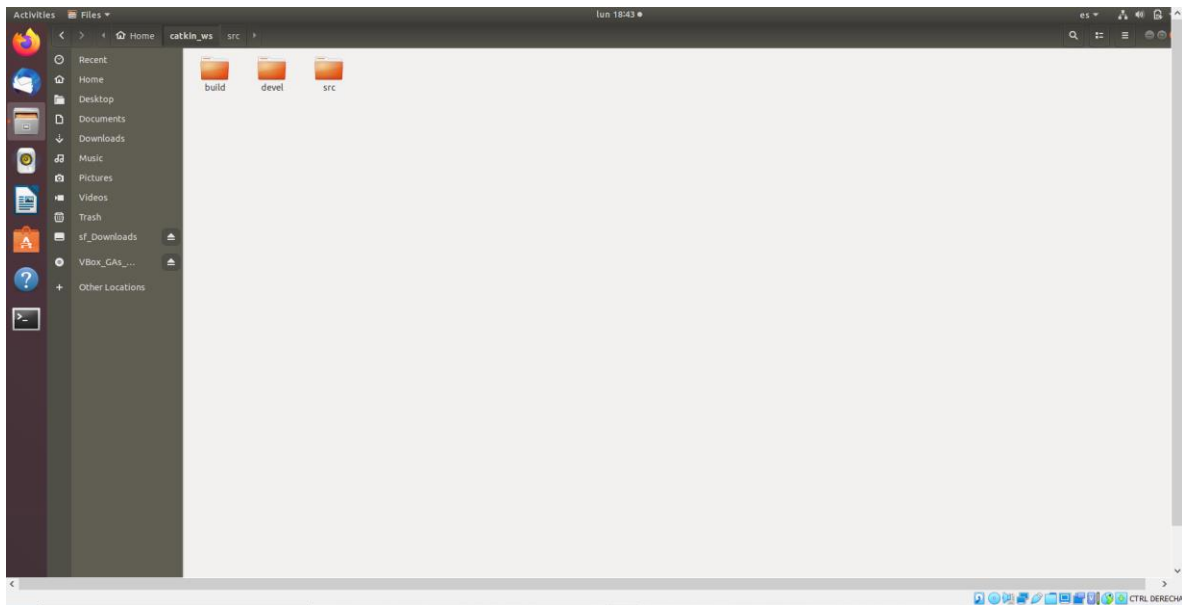
Actualización de paquetes

Para actualizar los archivos que contiene algún paquete de ROS se deben seguir los siguientes pasos, tomaremos como ejemplo la carpeta `grover`:

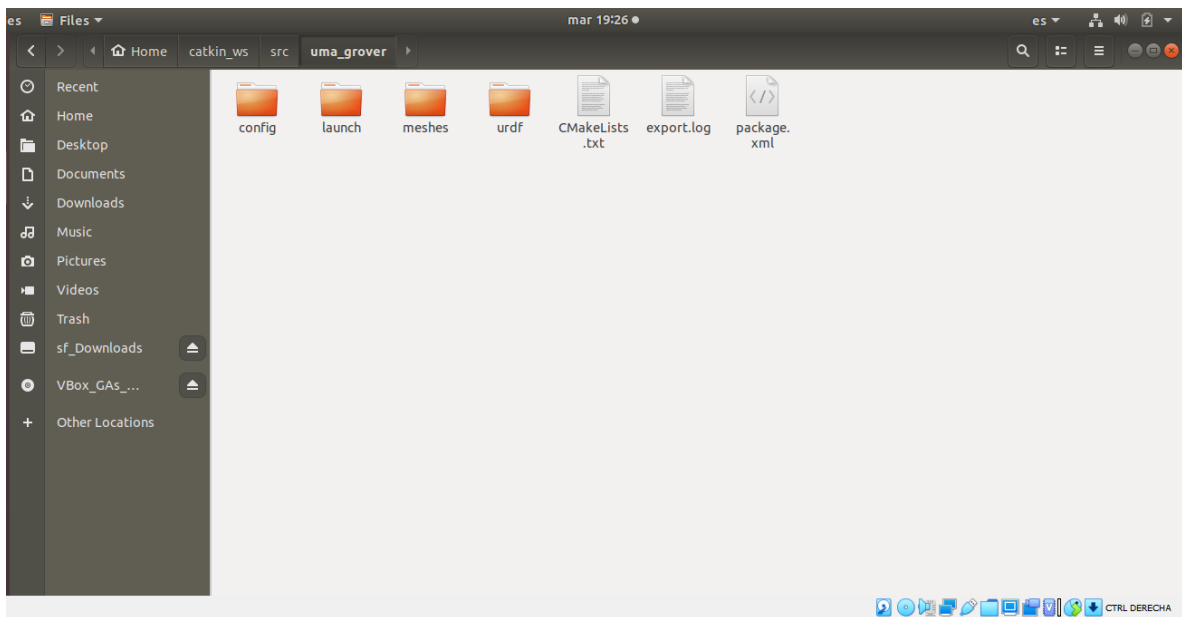
- Abrir Files.



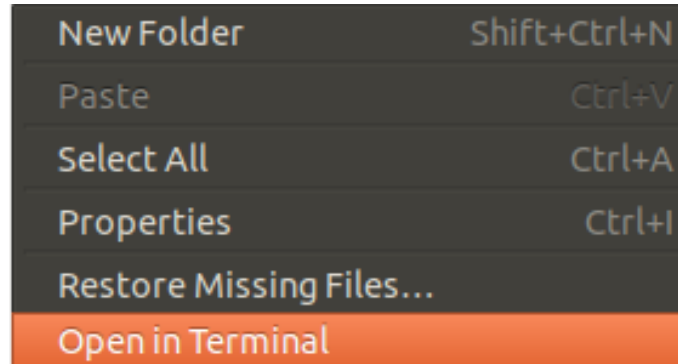
- Entrar a la carpeta `catkin_ws`.



- Entrar a la carpeta src y después a la carpeta uma_grover.



- Una vez estando aquí dar clic derecho y seleccionar la opción Open in Terminal.



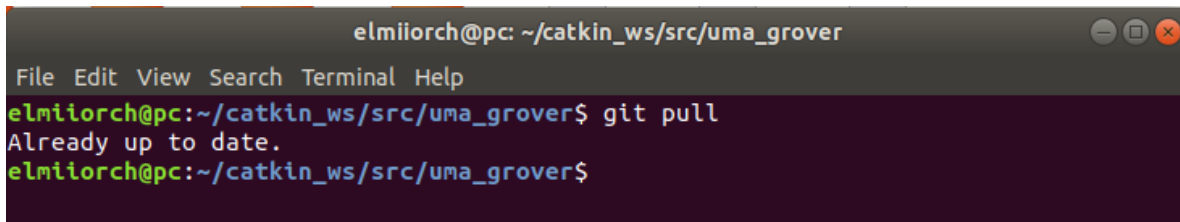
Al seleccionar esa opción se desplegará un terminal como esta:

```
elmiiorch@pc: ~/catkin_ws/src/uma_grover
File Edit View Search Terminal Help
elmiiorch@pc:~/catkin_ws/src/uma_grover$
```

- Sobre esa terminal escribe y ejecute el siguiente comando:
\$ git pull

```
elmiiorch@pc: ~/catkin_ws/src/uma_grover
File Edit View Search Terminal Help
elmiiorch@pc:~/catkin_ws/src/uma_grover$ git pull
```

Si su paquete de ROS ya cuenta con la versión más actualizada le aparecerá un mensaje como ese, en dado caso que no se esté actualizado se verán reflejados los cambios en la terminal.

A screenshot of a terminal window titled 'elmiorch@pc: ~/catkin_ws/src/uma_grover'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command 'git pull' being executed, which returns the message 'Already up to date.' The prompt 'elmiorch@pc:~/catkin_ws/src/uma_grover\$' is visible at the bottom.

```
elmiorch@pc: ~/catkin_ws/src/uma_grover
File Edit View Search Terminal Help
elmiorch@pc:~/catkin_ws/src/uma_grover$ git pull
Already up to date.
elmiorch@pc:~/catkin_ws/src/uma_grover$
```

Este proceso se puede realizar para cualquiera de los tres paquetes de ROS que acabamos de clonar, lo único que cambiaría es la carpeta sobre la que desplegamos la terminal.

Habilitación de la simulación

Funciones de navegación del UMA	32
Navegación semiautónoma	33
Navegación autónoma	36

Funciones de navegacion del UMA

El UMA es un robot totalmente capaz pensado para que el agricultor lo utilice al momento de la cosecha principalmente en invernaderos, ya que normalmente la persona arrastra un carrito generando un gran esfuerzo físico. Contando con dos distintas modalidades para la solución de la problemática: semiautónomo y autónomo.

Navegación semiautónoma

Con esta modalidad el UMA te seguirá, dando la ayuda necesaria para que el usuario coloque lo que se ocupe dentro de este.

Se basa en el seguimiento de color, donde en la simulación se utiliza el color azul ya que es el color que tiene el pantalón de la persona que esta caminando, entonces el robot lo estará siguiendo y dependiendo del radio de su medición corresponderá a su velocidad. Además, esta implementada la función de evasión de obstáculo por si se presenta un objeto de improviso en el camino, el robot podrá evadirlo y seguir a su objetivo.

Se deberá escribir dos comandos para poder hacer uso de esta modalidad:

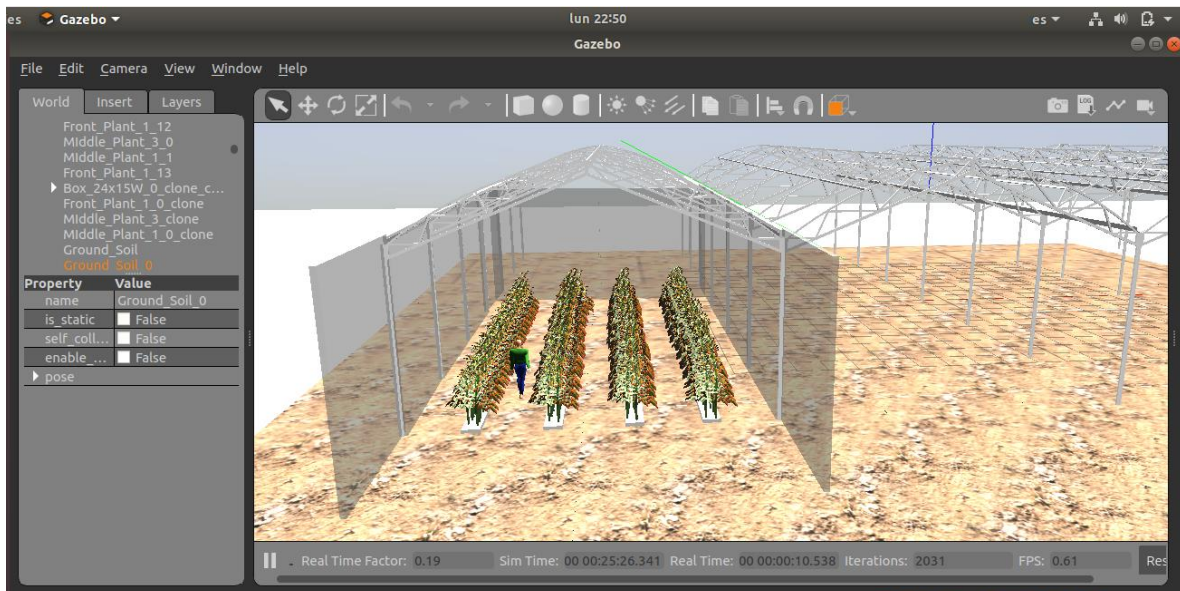
- En la primer terminal, escribir lo siguiente:
\$ roslaunch semestre_i greenhouse.launch

```

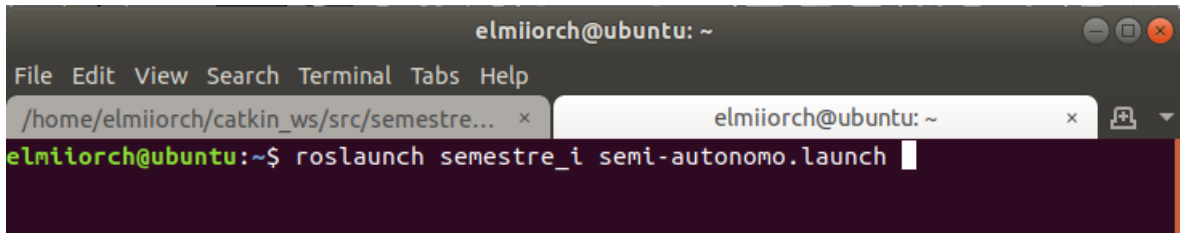
elmliorch@ubuntu: ~
File Edit View Search Terminal Help
elmliorch@ubuntu:~$ roslaunch semestre_i greenhouse.launch

```

Al ejecutar este comando se despliega el entorno virtual que emula un invernadero de plantas de jitomates por el cual estará caminando un agricultor.



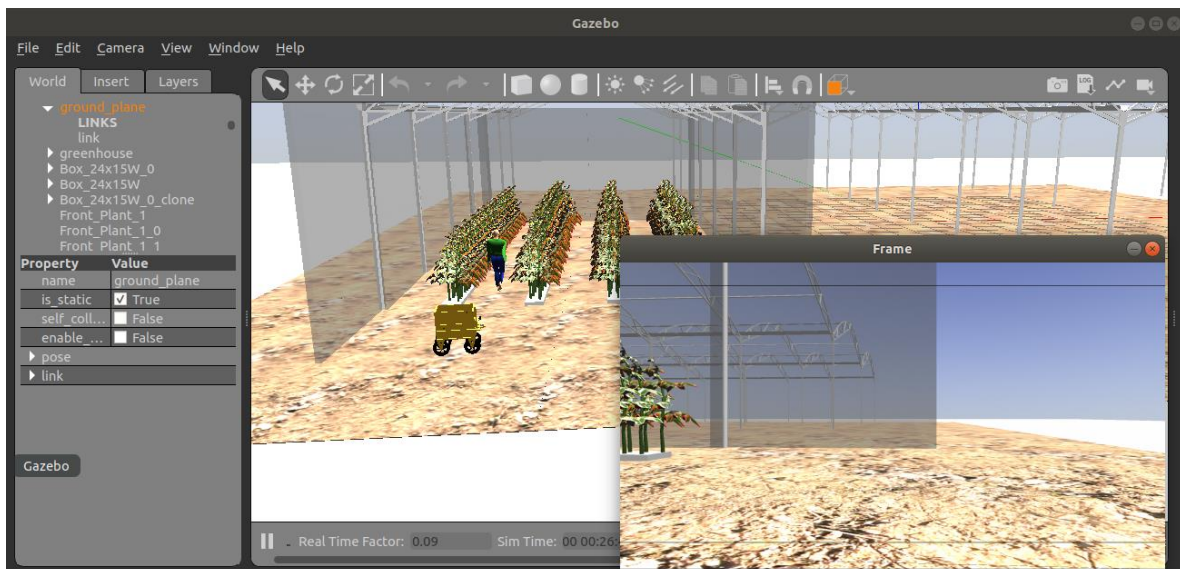
- En la segunda terminal, escribir lo siguiente:
\$ roslaunch semestre_i semi-autonomo.launch



```

elmiiorch@ubuntu: ~
File Edit View Search Terminal Tabs Help
/home/elmiiorch/catkin_ws/src/semestre... x elmiiorch@ubuntu: ~ x
elmiiorch@ubuntu:~$ roslaunch semestre_i semi-autonomo.launch
  
```

Una vez ejecutado este comando se iniciará la función de navegación.



- Ahora podrá emplear el UMA, el cual seguirá al usuario hacia donde el este se dirija, también evadirá obstáculos si se interfieren en su trayectoria y usted podrá ver lo que ve la cámara en la pantalla de su computador.

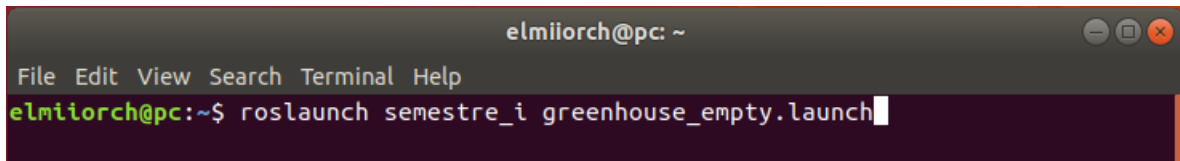
Navegación autónoma

Con esta modalidad el UMA realizar un recorrido programado por el interior de un invernadero, dando la ayuda necesaria para que el usuario para que el robot se presente en los puntos de control.

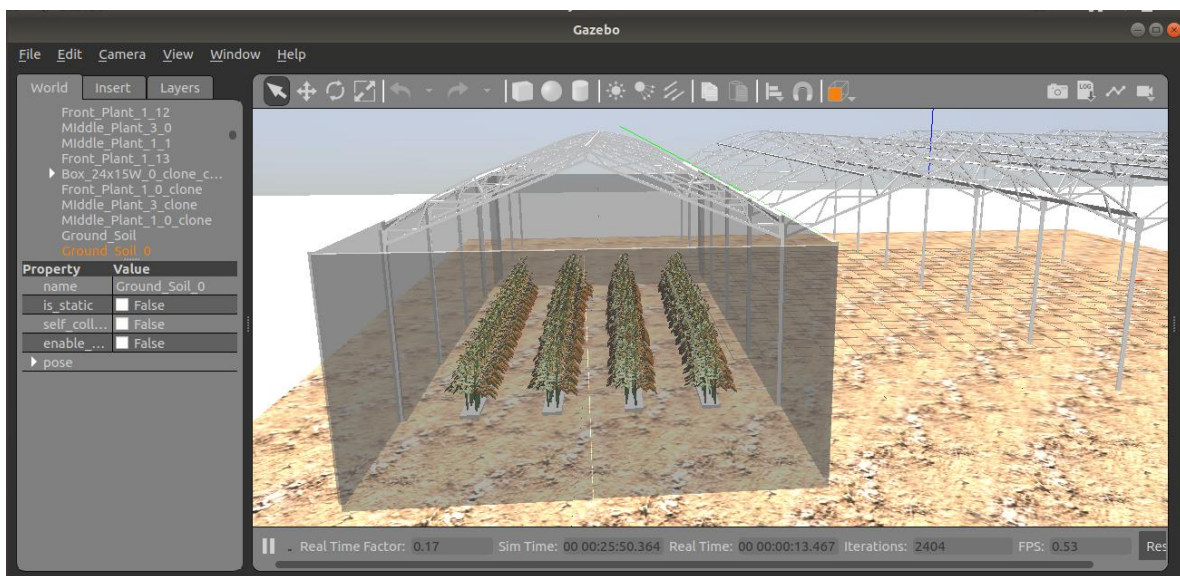
Se funcionalidad se basa en el seguimiento de coordenadas asignadas por el operador, donde en la simulación se hace uso de un RTK en cual proporciona la información necesaria para conocer la posición exacta del UMA, entonces el robot calculara una trayectoria para llegar su meta. Aunado a esto se implementa la función de evasión de obstáculo por si se presenta un objeto de improviso en el camino, el robot podrá evadirlo y llegar a su objetivo.

Se deberá escribir cuatro comandos para poder hacer uso de esta modalidad:

- En la primer terminal, escribir lo siguiente:
\$ roslaunch semestre_i greenhouse_empty.launch



Al correr este comando se desplegará el invernadero en un entorno cerrado con paredes traslucidas.



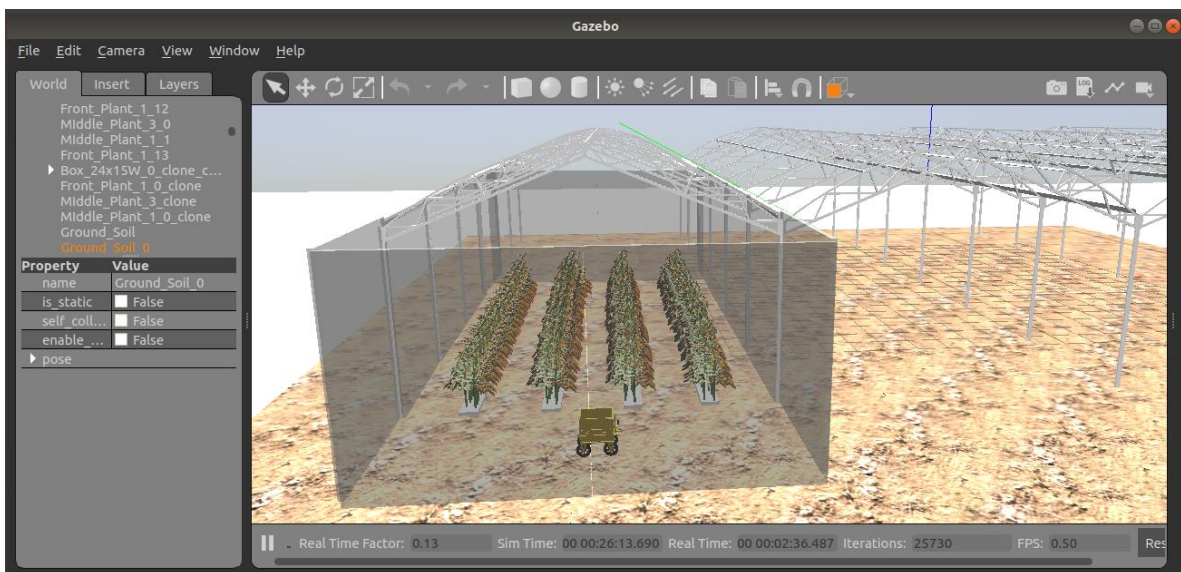
- En la segunda terminal, escribir lo siguiente:
\$ roslaunch semestre_i uma2.launch


```

elmiiorch@pc: ~
File Edit View Search Terminal Tabs Help
/home/elmiiorch/catkin_ws/src/semestre... x elmiiorch@pc: ~ x
elmiiorch@pc:~$ roslaunch semestre_i uma2.launch

```

Cuando utilizamos este comando se manda a llamar el UMA con las funciones de un RTK y un IMU.



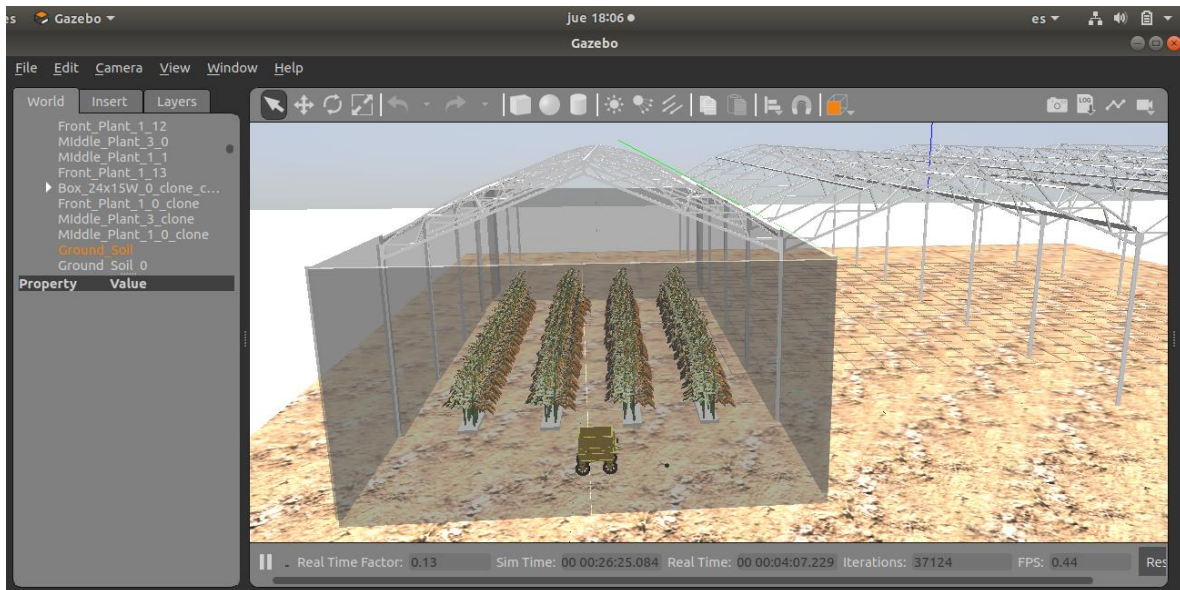
- En la tercera terminal, escribir lo siguiente:
\$ roslaunch gps_navigation gps_goal.launch

```

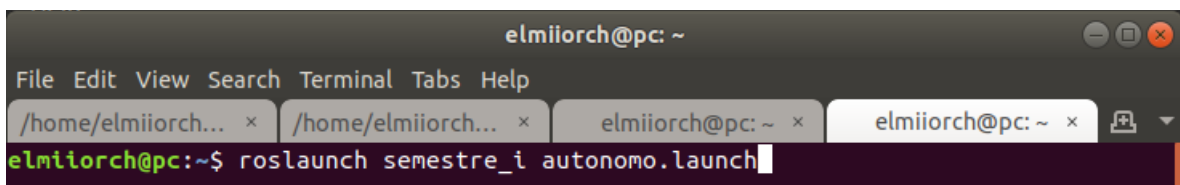
elmiiorch@pc: ~
File Edit View Search Terminal Tabs Help
/home/elmiiorch/catkin_... x /home/elmiiorch/catkin_... x elmiiorch@pc: ~ x
elmiiorch@pc:~$ roslaunch gps_navigation gps_goal.launch

```

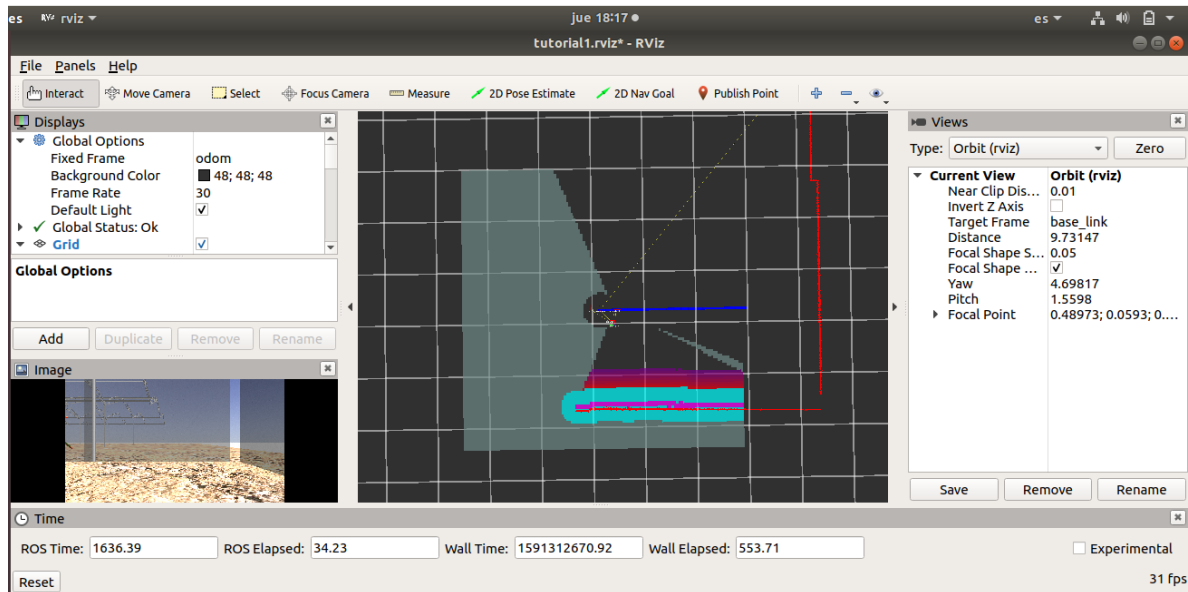
Al momento de usar dicho comando se manda a llamar un GPS, el cual será utilizado para que el usuario asigne coordenadas específicas para la navegación.



- En la cuarta terminal, escribir lo siguiente:
\$ roslaunch semestre_i autonomo.launch



Al ejecutar este comando se habilita la función de navegación autónoma de RTK con evasión de obstáculos.



- Ahora podrá emplear el UMA, el cual llegará a cualquier punto de control dentro del invernadero asignado por el operador, a su vez evadirá obstáculos si llegan a interferir en su trayectoria.