DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

“DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE PILOTO INTELIGENTE PARA UN VEHÍCULO AÉREO NO TRIPULADO QUE PERMITA EVADIR OBSTÁCULOS, PARA LA GENERACIÓN DE UN ORTOMOSAICO, CON EL FIN DE MONITOREAR ÁREAS DE DIFÍCIL ACCESO.

CRISTHIAN JESUS CAMPOS CAMPOS
ERICK ALEXANDER NOBOA CASTRO

TUTOR: ING. CÓRDOVA CRUZATTY, ANDREA CONCEPCIÓN

INTRODUCCIÓN

En este proyecto se desarrolla un sistema de piloto inteligente, que tenga la capacidad de volar autónomamente y evadir obstáculos empleado en un vehículo aéreo no tripulado del tipo quadrotor. En este caso una de las características más relevantes es el uso del sistema operativo robótico (R.O.S), el cual estará encargado de mantener estabilizada la aeronave, además de la detección y evasión de obstáculos.

ROS

ANTECEDENTES

Los UAV´s por sus siglas en inglés (Vehículos aéreos no tripulados) han sido motivo de un sin número de investigaciones dada su versatilidad, con aplicaciones tanto civiles como militares. Como por ejemplo el monitoreo, vigilancia o seguridad en zonas de interés o de difícil acceso.

Los Quadrotores y en sí la navegación con este tipo de vehículos se pueden usar en operaciones de evaluación de riesgos de trabajo para prevenir accidentes laborales aplicado a la industria (Toriz, Raygoza, & Martínez, 2017). las cuales emplean reconstrucción de mapas 3D para la identificación y reconocimientos previos de riesgos.

ANTECEDENTES

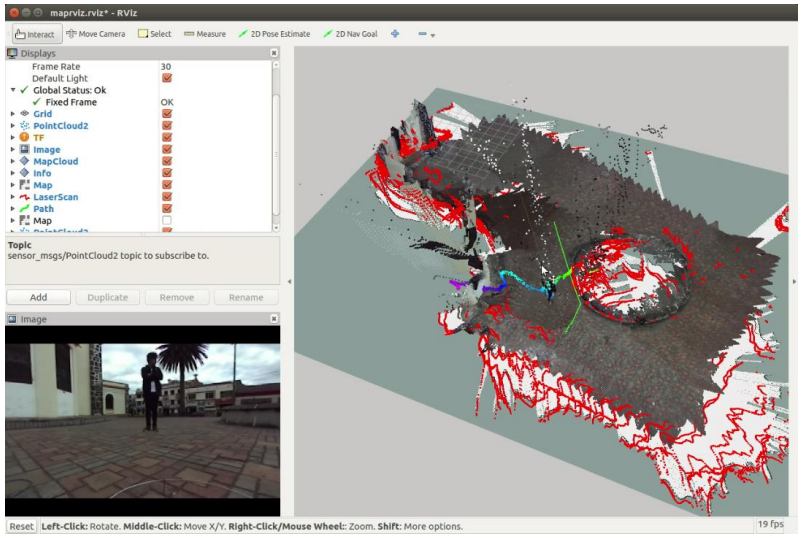
Ecuador es un país que, debido a su variabilidad topográfica, posee diferentes escenarios de los que se requiere un monitoreo, como: quebradas, acantilados, montañas, bosques y hasta la frontera ecuatoriana. Actualmente se realiza estas actividades de monitoreo con el uso de equipo y personal especializado, sin embargo existen varios proyectos de investigación para la supervisión de estas áreas, pero con vuelos manuales.

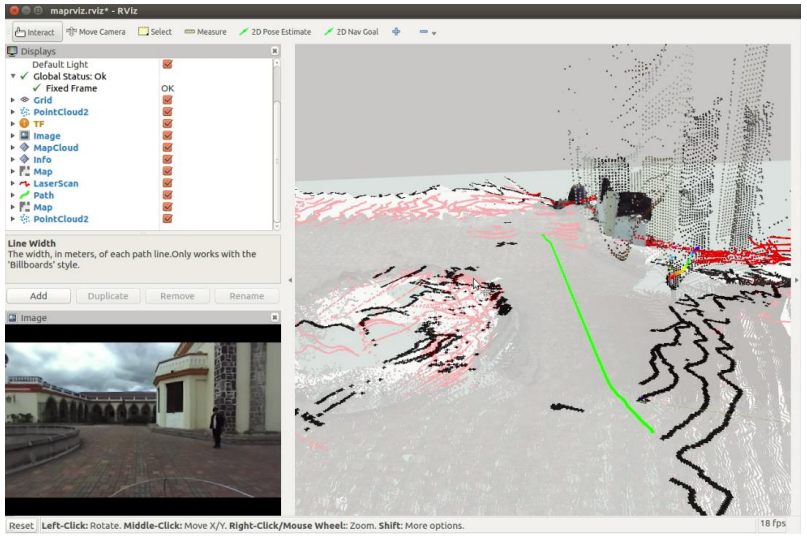
JUSTIFICACIÓN

De acuerdo con el ministro del Interior del Ecuador, solo en el año 2016, la Escuadra de Rescate del Grupo GIR ejecutó 24 operaciones relevantes a nivel nacional, en alta, media y baja montaña y rescates en zonas verticales (pendientes o quebradas). Por Estas razones resulta importante la implementación de un sistema autómata que permita la navegación y la adquisición de orto-imágenes que facilite el monitoreo previo de áreas de difícil acceso.

PROPUESTA

Un sistema de piloto inteligente con la capacidad de detectar y evadir obstáculos para un UAV, que permitirá navegar y generar un Ortomosaico mediante el uso de software computacional para el monitoreo de zonas de difícil acceso.

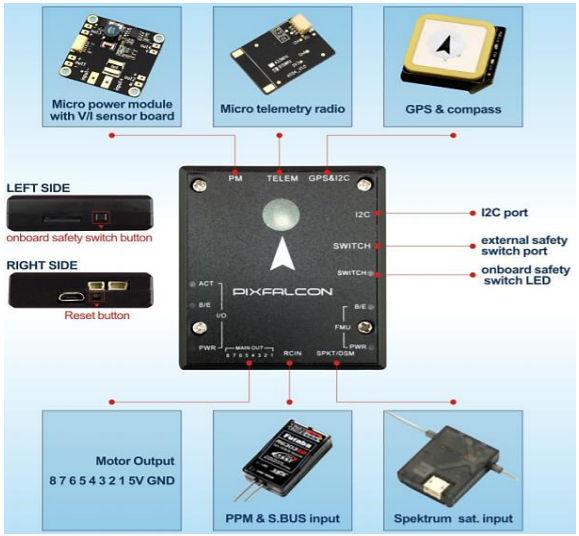




OBJETIVO GENERAL

Desarrollar e implementar un sistema de piloto inteligente para un vehículo aéreo no tripulado, con capacidad de evadir obstáculos, para la generación de un Ortomosaico con fines de monitoreo de áreas de difícil acceso.

DESCRIPCIÓN Y SELECCIÓN DE COMPONENTES



DESCRIPCIÓN Y SELECCIÓN DE COMPONENTES

- **Quadrotor**

Es un vehículo aéreo no tripulado que puede despegar y desplazarse gracias a sus cuatro rotores coplanares, cuyo movimiento depende de la variación de potencia por parte de cada uno de ellos

El cuadricóptero flota o ajusta su altura mediante la aplicación del mismo empuje a los cuatro rotores

El cuadricóptero ajusta su orientación mediante la aplicación de más empuje a los rotores que giran en una misma dirección

El cuadricóptero ajusta su giro aplicando más empuje a un rotor y disminuyendo a su opuesto

SELECCIÓN DE COMPONENTES

La tabla indica los parámetros de selección del quadrotor, para este proyecto

N°	Métrica	Import.	Phanton 4 pro	DJI Mavic Pro	Xplus One
1	Tiempo de Vuelo por Batería	4	4	4	3
2	Peso	3	2	3	3
5	Medidas	2	3	3	4
6	Radio de señal (Rango)	3	3	2	4
7	Factor de seguridad	3	4	4	4
8	Ruido Producido	2	1	1	1
10	Velocidad de desplazamiento	5	3	2	5
11	error de seguimiento de ruta	5	3	4	4
	Total		75	71	86

SELECCIÓN DE COMPONENTES

La tabla indica los parámetros de selección del quadrotor, para este proyecto

N°	Métrica	Imp.	LeddarTech-Vu8	Estéreo Visión-ZED	Kinect
1	robustez al ambiente*	4	4	5	3
2	Peso	3	3	5	4
3	Distancia Detección de obstáculos	5	3	4	1
4	Velocidad de adquisición	5	3	4	5
5	Ángulo de detección	3	1	5	3
	TOTAL		58	90	63

SELECCIÓN DE COMPONENTES

- **Quadrotor: Xplus One**

La aeronave Xplus One es la que presenta las mejores características de velocidad de desplazamiento (hasta 80 km/h), mayor radio de alcance de la señal de control (4 km). debido a sus medidas adecuadas y su peso (1300 gr) ha sido seleccionado como adecuado para el desarrollo del proyecto de investigación. Además de que presenta una generosa capacidad de carga.

SELECCIÓN DE COMPONENTES

- **Sensor de detección de obstáculos: Estéreo visión - ZED**

El sensor de estéreo Visión ZED presenta una mayor capacidad de vista a larga distancia (20 m), es decir, permite visualizar un obstáculo que se aproxima a una mayor distancia, brindándole un mayor tiempo al procesador para poder tomar cualquier acción evasiva. Por otra parte, su bajo consumo, peso (160 gr) y ángulo de detección (80°) lo convierten en la opción ideal para el desarrollo de la aplicación.

DESCRIPCIÓN DE COMPONENTES

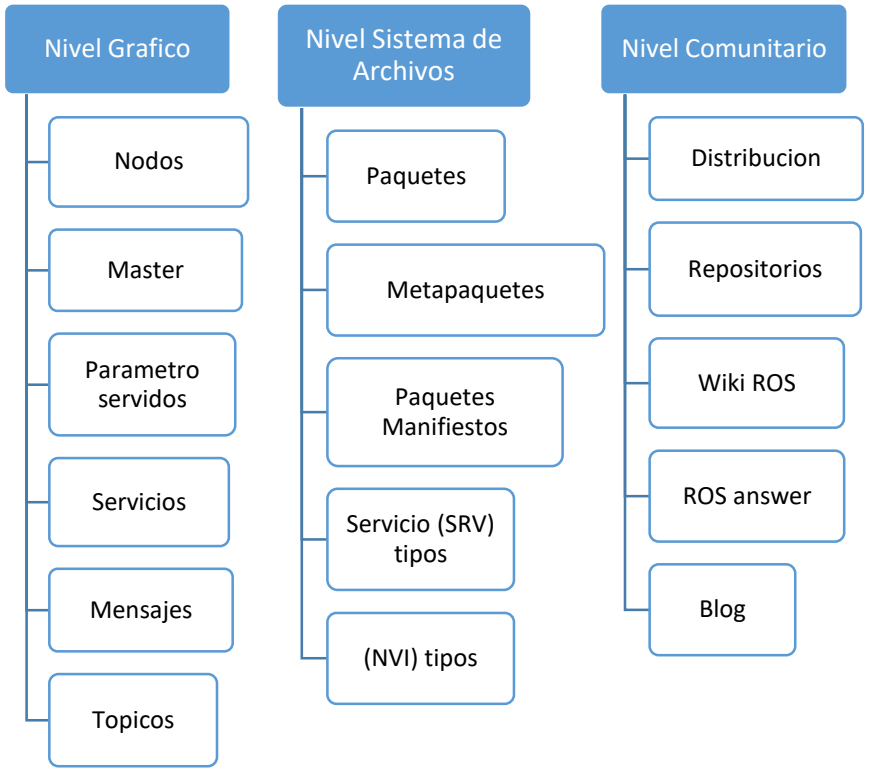
- **ROS(Robotic Operating system)**

Es un sistema operativo de código abierto que permite desarrollar aplicaciones para robots o plataformas de investigación" (ROS.org, 2018). Actualmente este sistema es uno de los más utilizados para programar sobre cualquier plataforma robótica, debido a los servicios estándar provee, como: abstracción de hardware, librerías y herramientas, control de dispositivos de bajo nivel, implementación de funciones utilizadas normalmente, gestión de paquetes, entre otros.

ROS

DESCRIPCIÓN DE COMPONENTES

- **ARQUITECTURA DE ROS**



ARQUITECTURA DE ROS

NIVEL GRÁFICO

Nodos: Se pueden definir como los procesos ejecutables encargados de realizar el computo en un sistema de control robótico, generalmente existen varios nodos en una misma red, donde cada uno se encarga de ejecutar una tarea en particular.

Master: es el proceso central encargado de mantener un registro de todo el grafo computacional, permitiendo la comunicación entre nodos.

Servidor de parámetros: permite que los datos sean almacenados de manera segura en un lugar central, y este forma parte del Master.

ARQUITECTURA DE ROS

NIVEL GRÁFICO

Servicio: Están conformados por 2 partes: un mensaje que es el encargado de preguntar y otro mensaje que es la respuesta. Los servicios se definen mediante archivos SRV, los cuales se compilan en el código fuente de una biblioteca de ROS.

Mensaje: un mensaje es una estructura de datos simple, que puede ser del tipo primitivo (entero, flotante, boolean, etc.) o un arreglo de estos, además es la manera de comunicación entre nodos ya que estos se comunican enviándose mensajes.

Tópico: los mensajes que se envían son generalmente en base a un sistema de publicación o suscripción.

Bags: Son un formato para guardar y reproducir datos de mensajes de ROS

ARQUITECTURA DE ROS

NIVEL SISTEMA DE ARCHIVOS

Paquetes: se pueden considerar como la unidad de organización de ROS. Este contiene procesos (nodos), librerías de desarrollo, ficheros de configuración.

Metapaquetes: son paquetes que únicamente sirven para representar un conjunto de paquetes que se encuentren relacionados

Paquete manifiesto: (paquete.xml) su función principal es brindar metadatos sobre un paquete, como lo son: nombre, versión, descripción, dependencias y otra información propia del meta como paquetes exportados. (Tumbaco M. & Quimbita Z., 2014).

(NVI) tipos: definen la estructura de los datos que son enviados por la mensajería de ROS**.**

Servicio (SRV) tipos: las descripciones de servicio que se encuentran almacenados en un paquete definen la solicitud y la estructura de los datos de respuesta, propios de los servicios ROS.

ARQUITECTURA DE ROS

NIVEL COMUITARIO

ROS Distribuciones: Es una colección de paquetes que se pueden instalar, las cuales tienen un propósito similar a las distribuciones Linux, es decir, permiten una fácil instalación de una colección de softwares.

Repositorios: Ya que se trata de un lugar donde se recopila software de código libre, generalmente asociadas en stack´s (pilas). De esta manera es posible acceder a ellas y descargarlas para su uso. El lugar más usado es Github

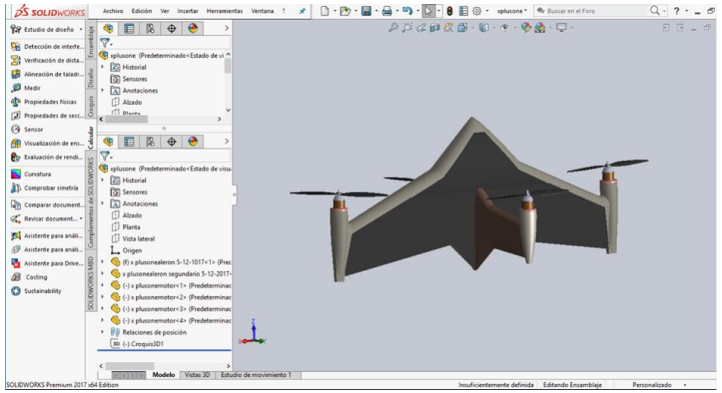
Wiki ROS: se trata del principal foro de información sobre ROS en donde se encuentra mucha información relevante. Cualquier persona mediante una cuenta puede contribuir con documentación, o realizar correcciones o actualizaciones de trabajos y más.

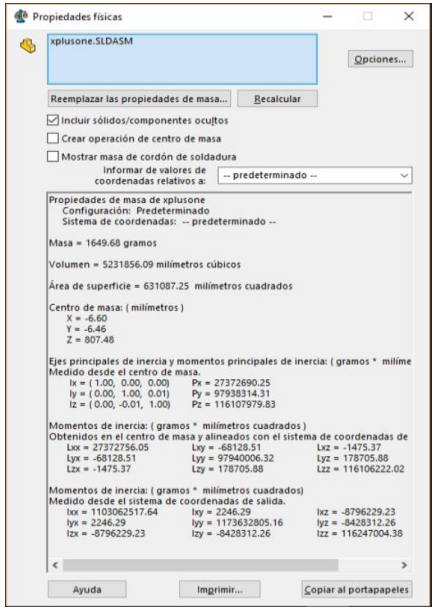
ROS answers: Es un sitio de preguntas y respuestas sobre temas relacionados a ROS

MODELO MATEMÁTICO

PARÁMETROS DEL SOLIDO RÍGIDO

los parámetros se obtienen a partir de un modelo CAD del quadrotor realizado en el programa SolidWorks y otros datos se comprueban experimentalmente.





MODELO MATEMÁTICO

MOMENTOS DE INERCIA

Los valores identificados para el momento de inercia del cuerpo son:

$$I = \begin{bmatrix} 1{,}1030 & 0 & 0 \\ 0 & 1{,}1736 & 0 \\ 0 & 0 & 0{,}11627 \end{bmatrix} kg\,.m^2$$

MODELO MATEMÁTICO

RESUMEN DE PARÁMETROS

	Parámetros del Rotor		
Ganancia Estática $K \quad \left[\frac{rpm}{\%}\right]$	Const. / tiempo $\tau \quad [seg]$	Ganancia de empuje $k_t \ [N\frac{s^2}{rad}]$	Ganancia de arrastre $k_d \ [N\frac{m.s^2}{rad}]$
90	0,05	$4{,}94 * 10^{-6}$	$3{,}05 * 10^{-6}$
	Parámetros del solido Rígido		
Inercia del cuerpo $[Kg\ m^2]$ $I_x = 1{,}1030$ $I_y = 1{,}1736$ $I_z = 0{,}11627$			Masa $m \ [Kg]$ $1650\ gr$

MODELO MATEMÁTICO

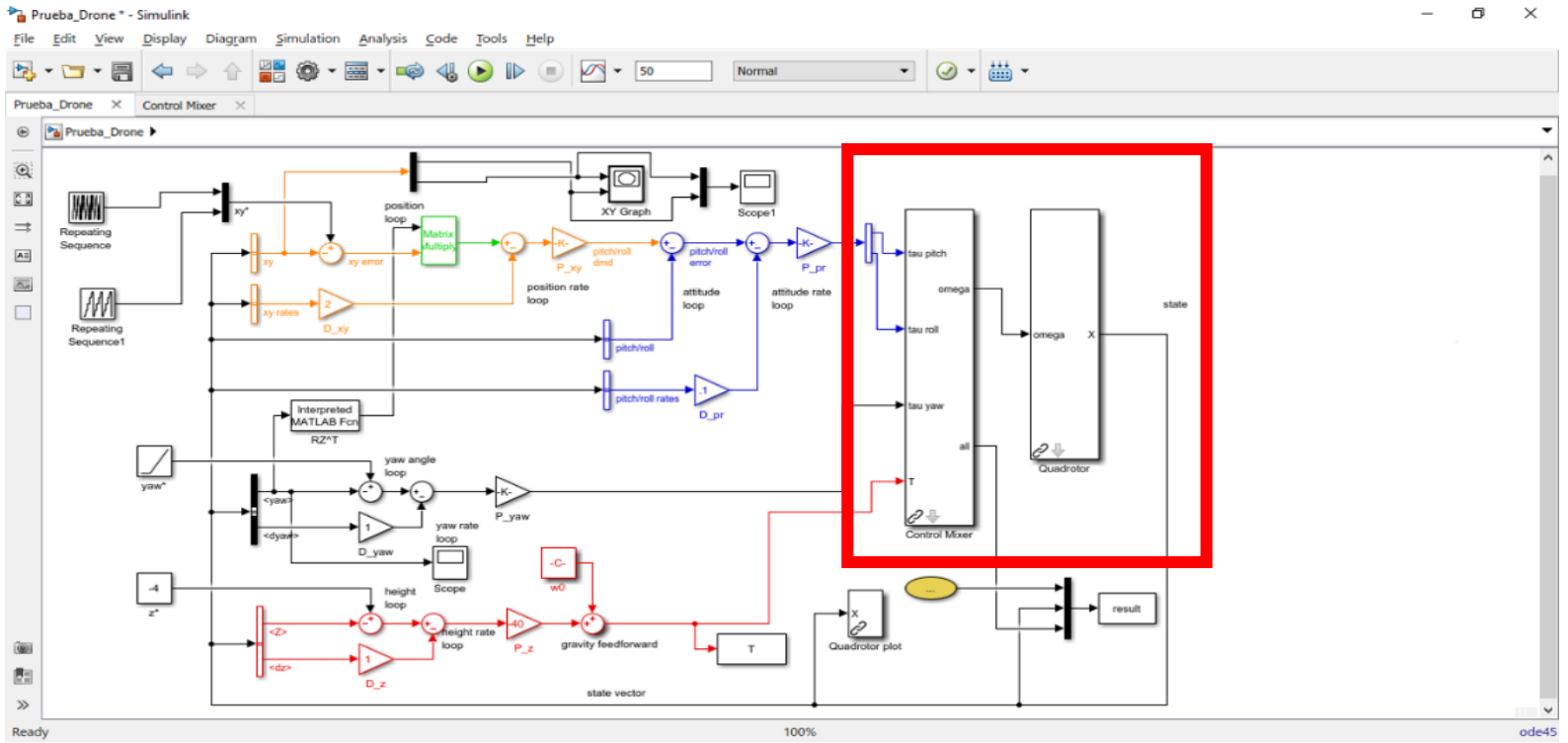
SIMULACIÓN

Toda la simulación se ha realizado usando el paquete Simulink de MATLAB, el cual tiene bloques dedicados en los cuales se puede incluir la dinámica rotacional del quadrotor. El Toolbox utilizado es el de Robotics de Peter Corke (Corke, 2018)

El bloque de Simulink “Quadrotor Dinamic´s”, es empleado para implementar la cinemática y dinámica estudiadas anteriormente. Este bloque consta de una función “S”, la cual representa un sistema dinámico. Este bloque cuenta con una sintaxis específica y el usuario debe definir sus propios parámetros.

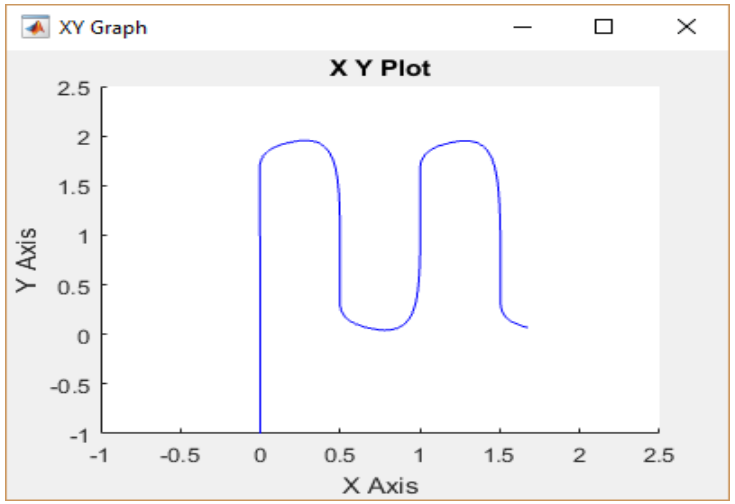
MODELO MATEMÁTICO

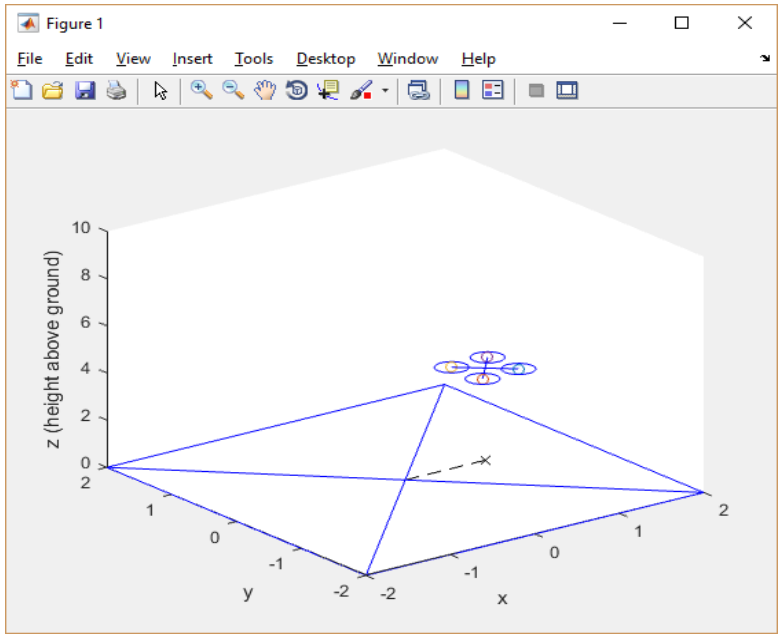
SIMULACIÓN



MODELO MATEMÁTICO

RESULTADOS





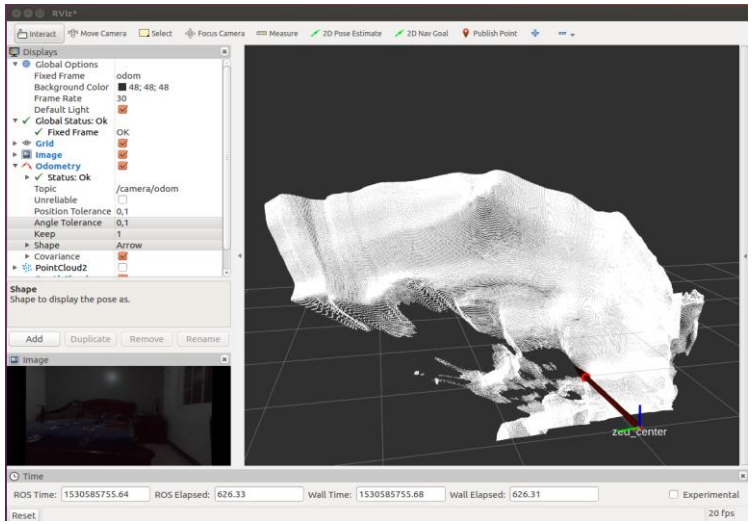
CONEXIÓN

El quadrotor Xplus One incorpora en su interior un módulo de control Pixfalcon, el cual permite comunicar con una PC a través de su módulo de telemetría. Este módulo de telemetría es de la marca Holybro y trabaja en una banda de 915 MHz.

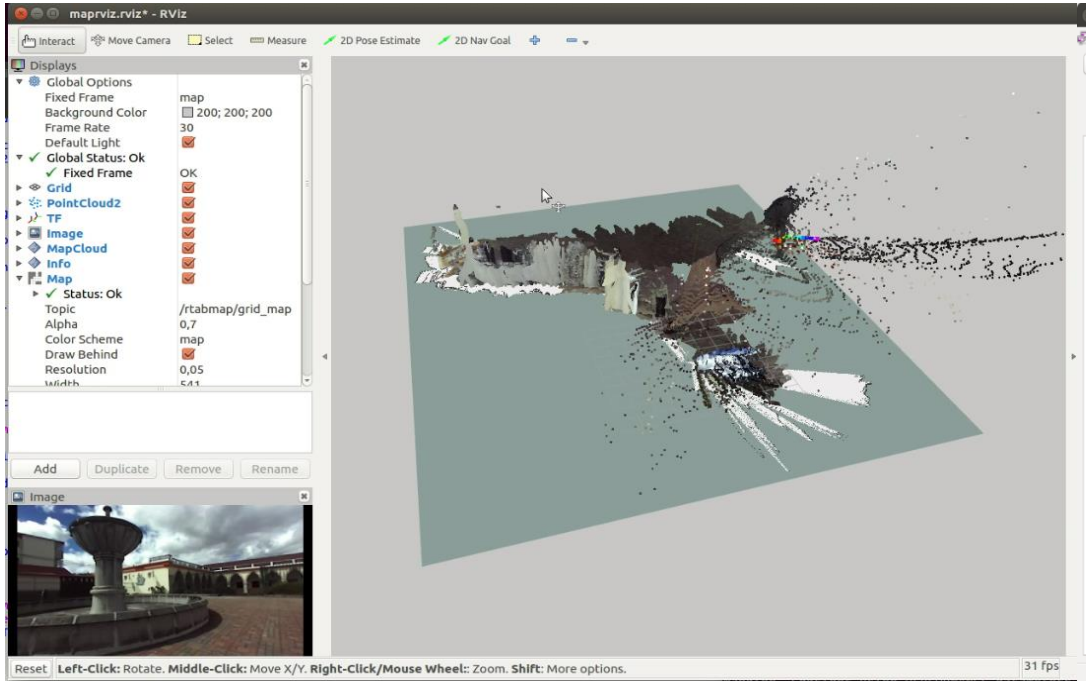
MAPEADO

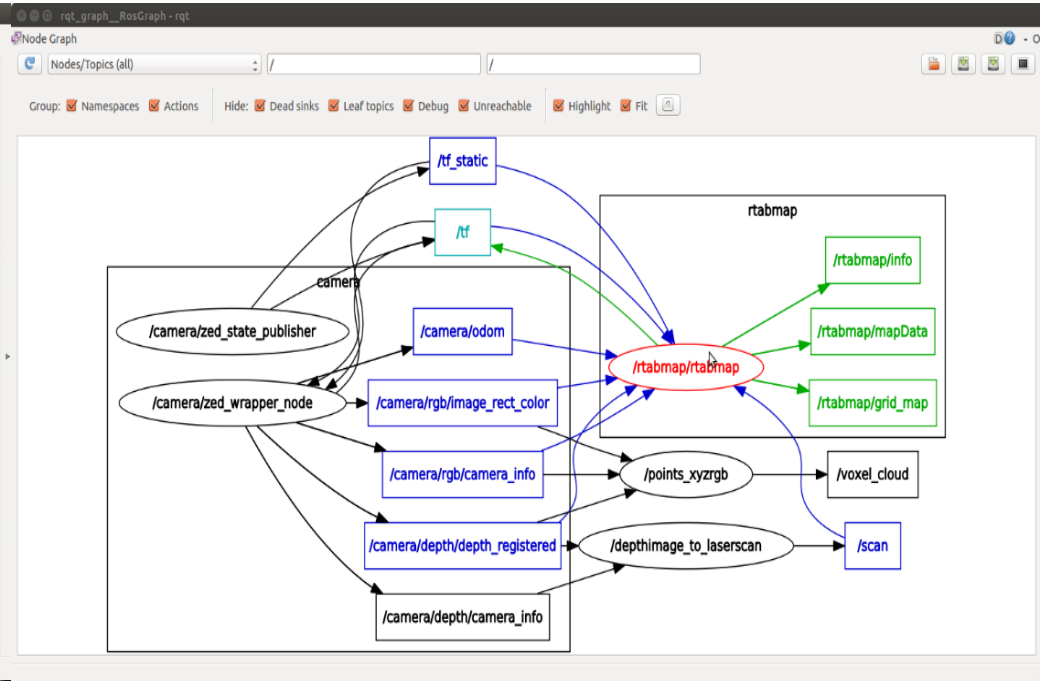
Es necesario crear un mapa para que el quadrotor sea capaz de localizarse y navegar en el entorno que se le presente

En este proyecto se utiliza como sensor la cámara estereoscópica ZED la cual responde muy bien ante este tipo de aplicaciones ya que además de generar una nube de puntos y un mapa de profundidad ofrece la ventaja de extraer información de la odometría visual de esta.



MAPEADO


mapviz.rviz* - RViz
Interact
Move Camera
Select
Measure
2D Pose Estimate
2D Nav Goal
Displays
Global Options
Fixed Frame map
Background Color 200; 200; 200
Frame Rate 30
Default Light
Global Status: Ok
Fixed Frame OK
Grid
PointCloud2
TF
Image
MapCloud
Info
Map
Status: Ok
Topic /rtabmap/grid_map
Alpha 0,7
Color Scheme map
Draw Behind
Resolution 0,05
Add
Duplicate
Remove
Rename
Image
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps


rqt_graph__RosGraph - rqt
Node Graph
Nodes/Topics (all)
Group: Namespaces Actions
Hide: Dead sinks Leaf topics Debug Unreachable
Highlight Fit
/tf_static
/tf
rtabmap
/rtabmap/info
/rtabmap/mapData
/rtabmap/grid_map
camera
/camera/zed_state_publisher
/camera/odom
/rtabmap/rtabmap
/camera/zed_wrapper_node
/camera/rgb/image_rect_color
/camera/rgb/camera_info
/points_xyzrgb
/voxel_cloud
/camera/depth/depth_registered
/depthimage_to_laserscan
/scan
/camera/depth/camera_info

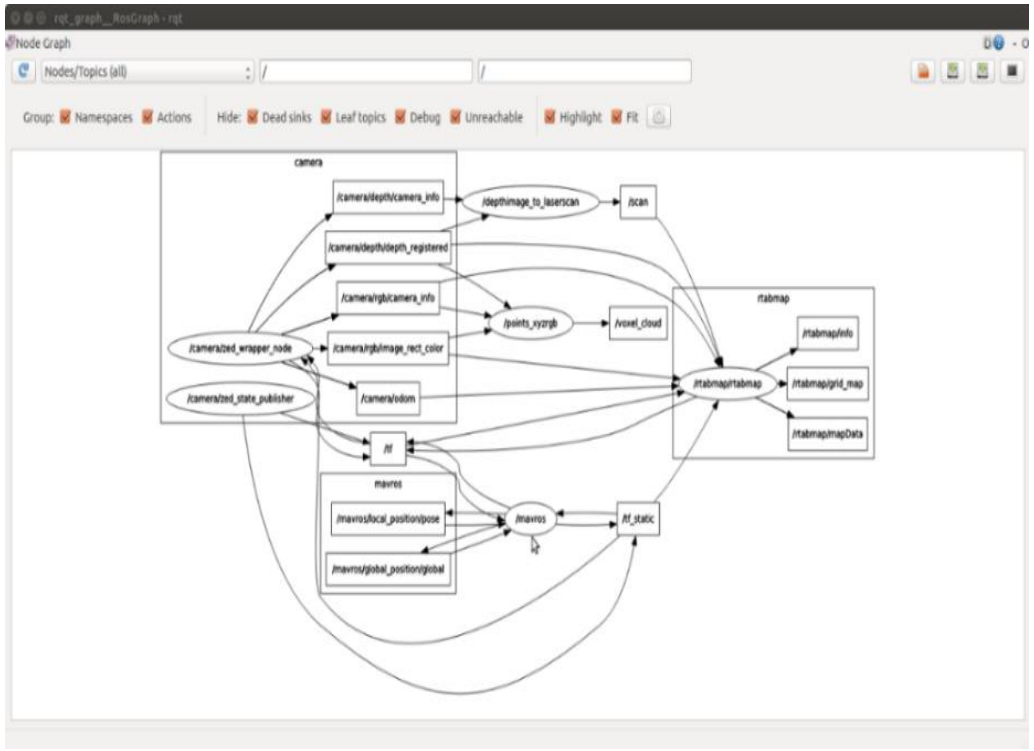
LOCALIZACIÓN

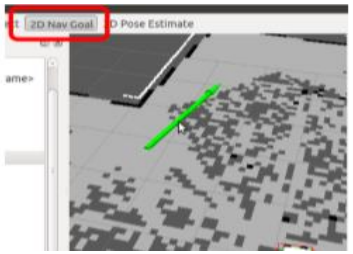
Para que el quadrotor pueda moverse de manera autónoma es indispensable que antes tenga la capacidad de encontrarse dentro del mapa anteriormente generado, a este proceso se le conoce como localización. Este proceso se lleva a cabo dentro del paquete de Move_base el cual realiza una comparación entre los puntos fuertes de las imágenes captadas por la cámara ZED y el mapa global anteriormente creado

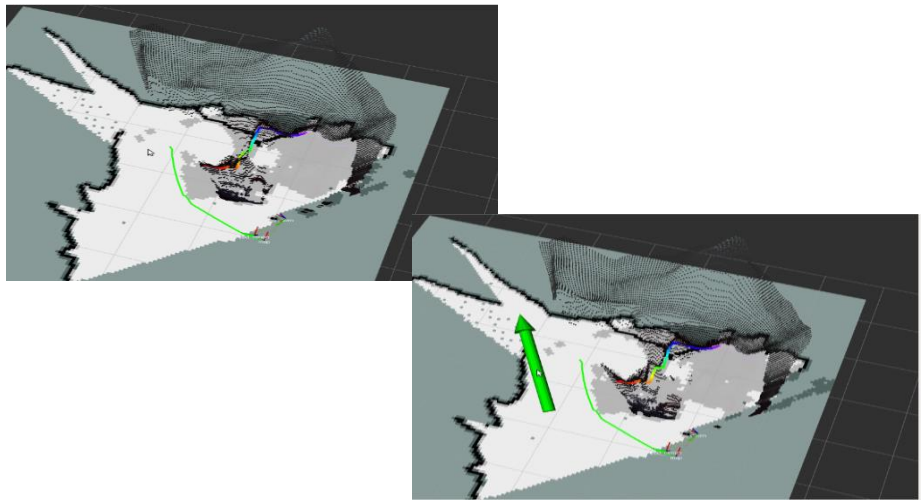
NAVEGACIÓN

El último paso en esta investigación es poder realizar una navegación autónoma en donde el quadrotor pueda detectar y evadir los obstáculos que se le presenten en el entorno. Para poder llevar a cabo este paso se hace uso de la pila de navegación, la cual como ya se explicó anteriormente se necesita información de la odometría del quadrotor, los datos de los sensores y un objetivo a donde el quadrotor debe desplazarse, para que luego se envíen mensajes de velocidad a una base móvil. El tipo de mensaje que se publica para el control del desplazamiento es del tipo “twist” y generalmente se publica en el tópico “/cmd_vel”

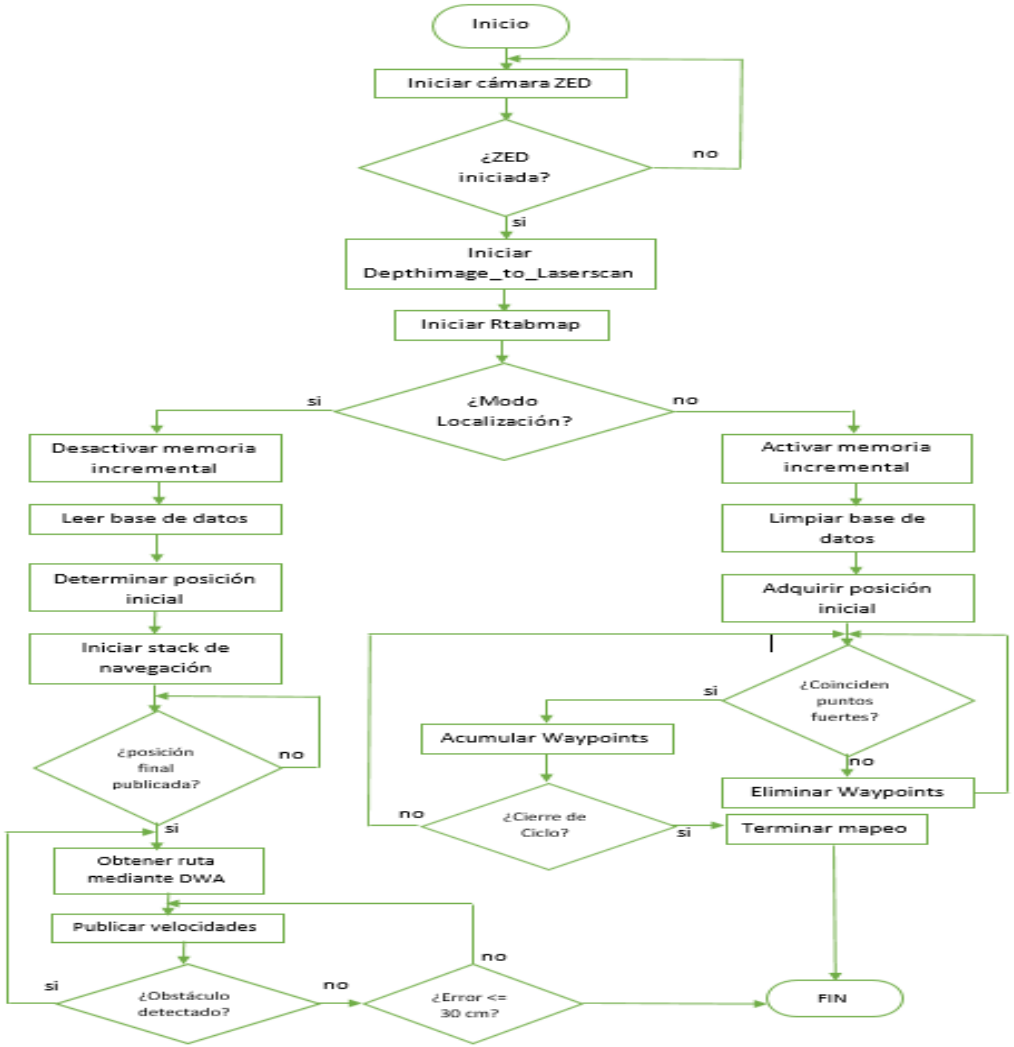
NAVEGACIÓN


rqt_graph__RosGraph - rqt
Node Graph
Nodes/Topics (all)
Group: Namespaces Actions
Hide: Dead sinks Leaf topics Debug Unreachable
Highlight Fit
camera
/camera/depth/camera_info
/depthimage_to_laserscan
/scan
/camera/depth/depth_registered
/camera/rgb/camera_info
/points_xyzrgb
/voxel_cloud
/camera/zed_wrapper_node
/camera/rgb/image_rect_color
/camera/zed_state_publisher
/camera/odom
/tf
rtabmap
/rtabmap/info
/rtabmap/rtabmap
/rtabmap/grid_map
/rtabmap/mapData
mavros
/mavros/local_position/pose
/mavros
/tf_static
/mavros/global_position/global

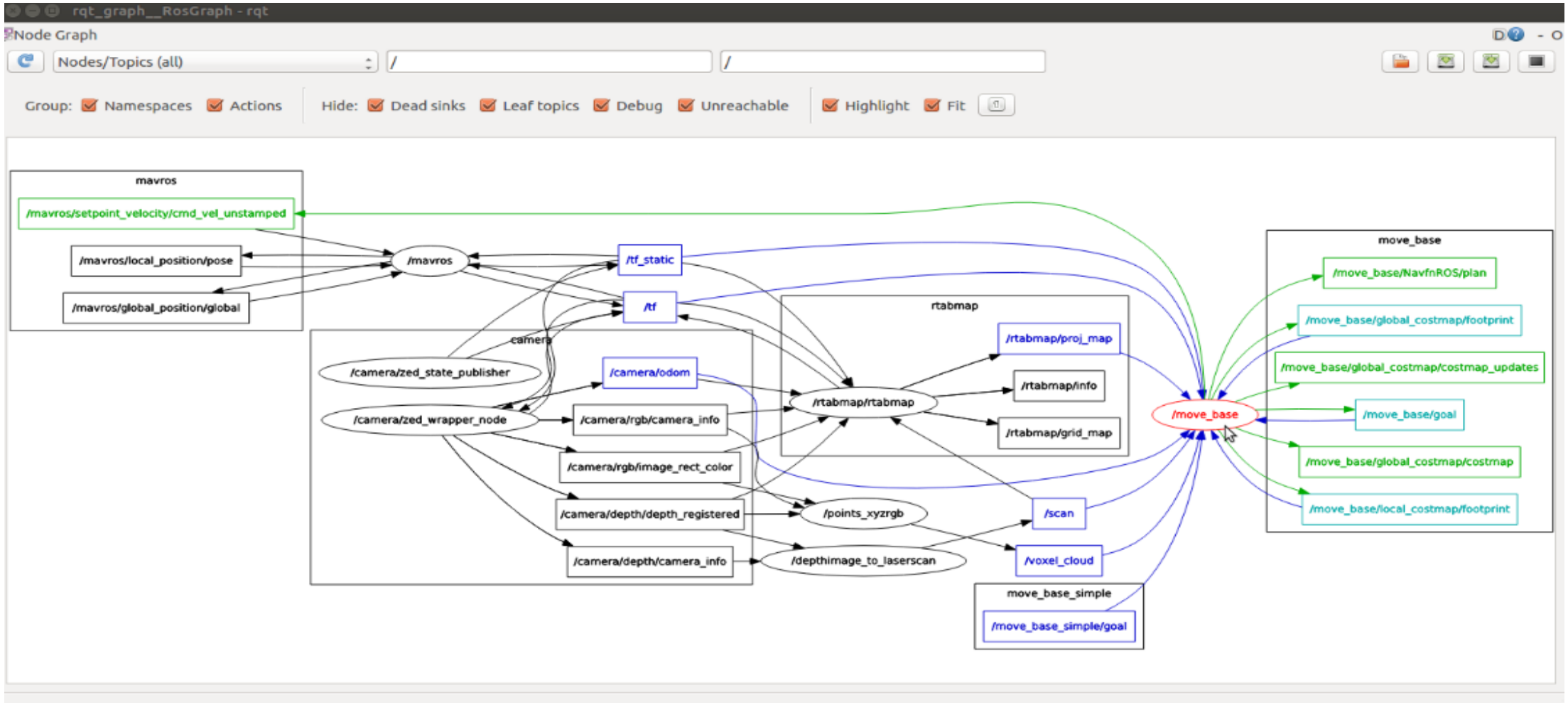

2D Nav Goal
Pose Estimate



IMPLEMENTACIÓN DEL ALGORITMO

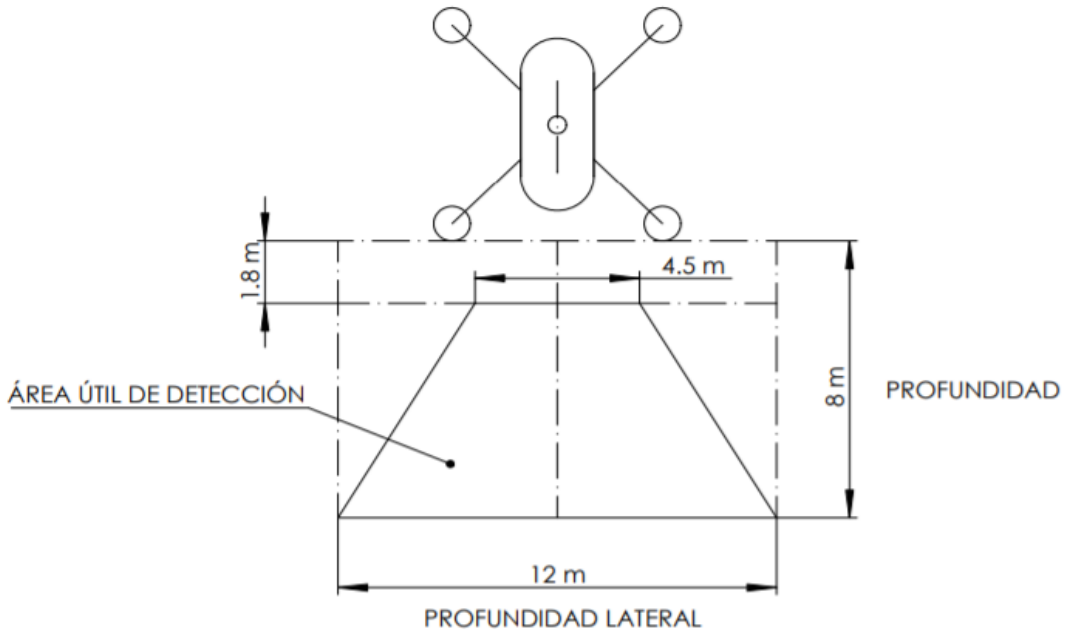


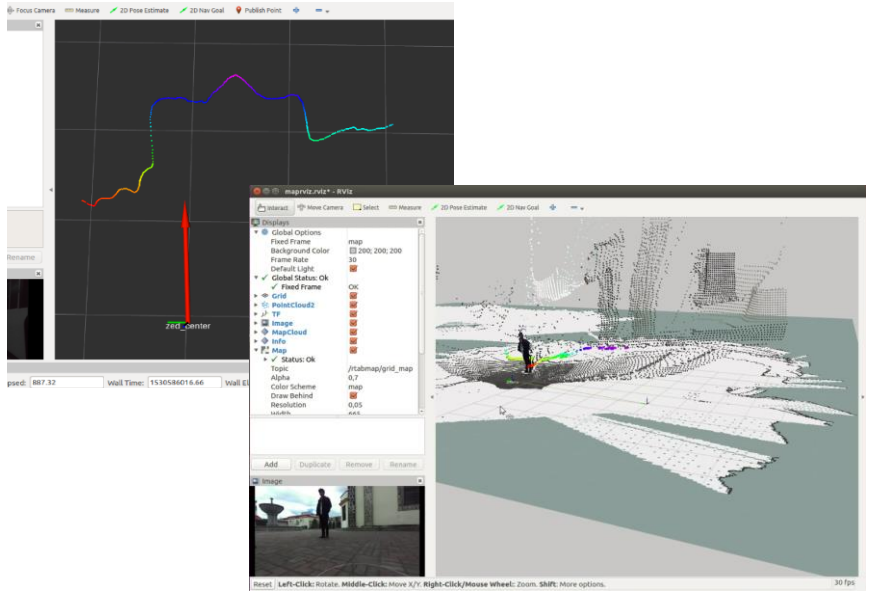
IMPLEMENTACIÓN DEL ALGORITMO



PRUEBAS Y RESULTADOS

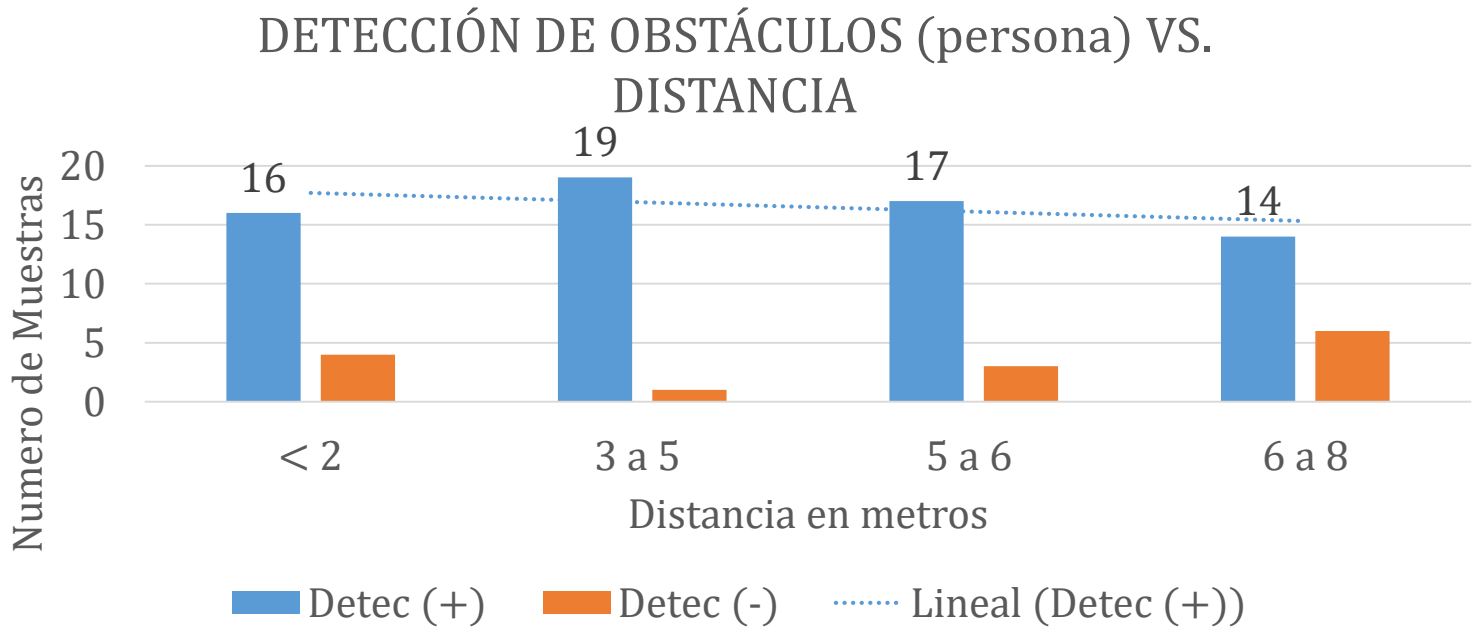
- **Área de detección (obstáculos) útil**





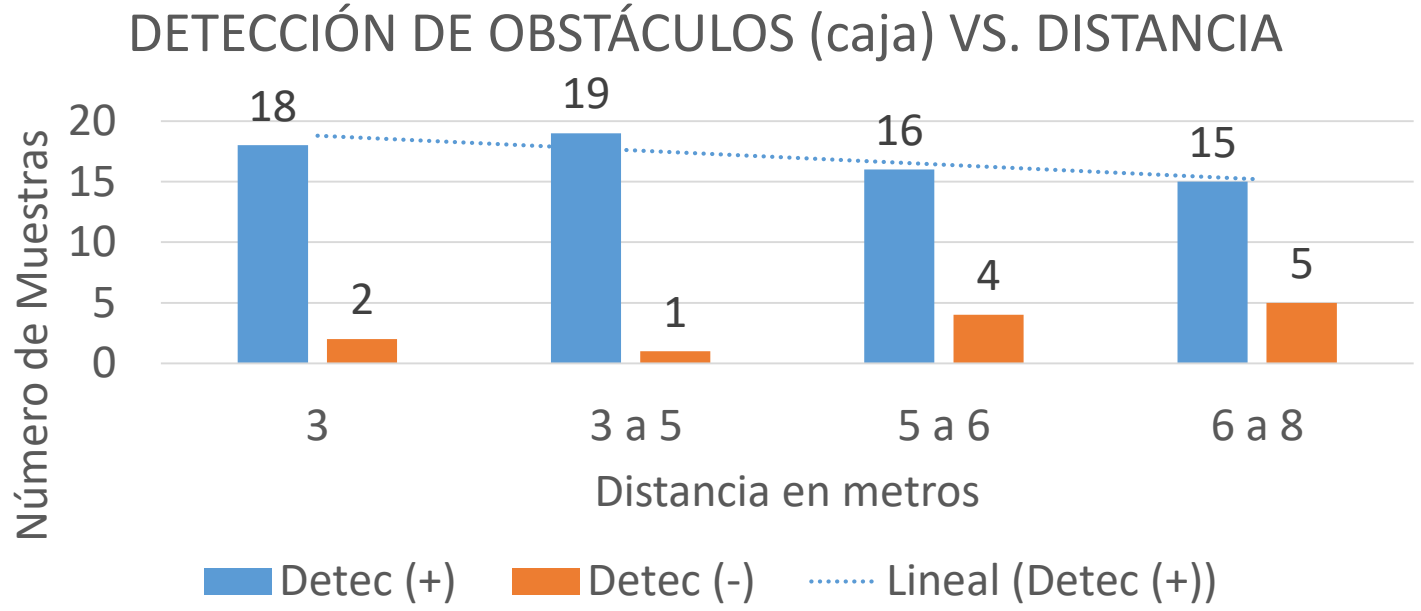
PRUEBAS Y RESULTADOS

- **Evaluación del algoritmo de detección de obstáculos**



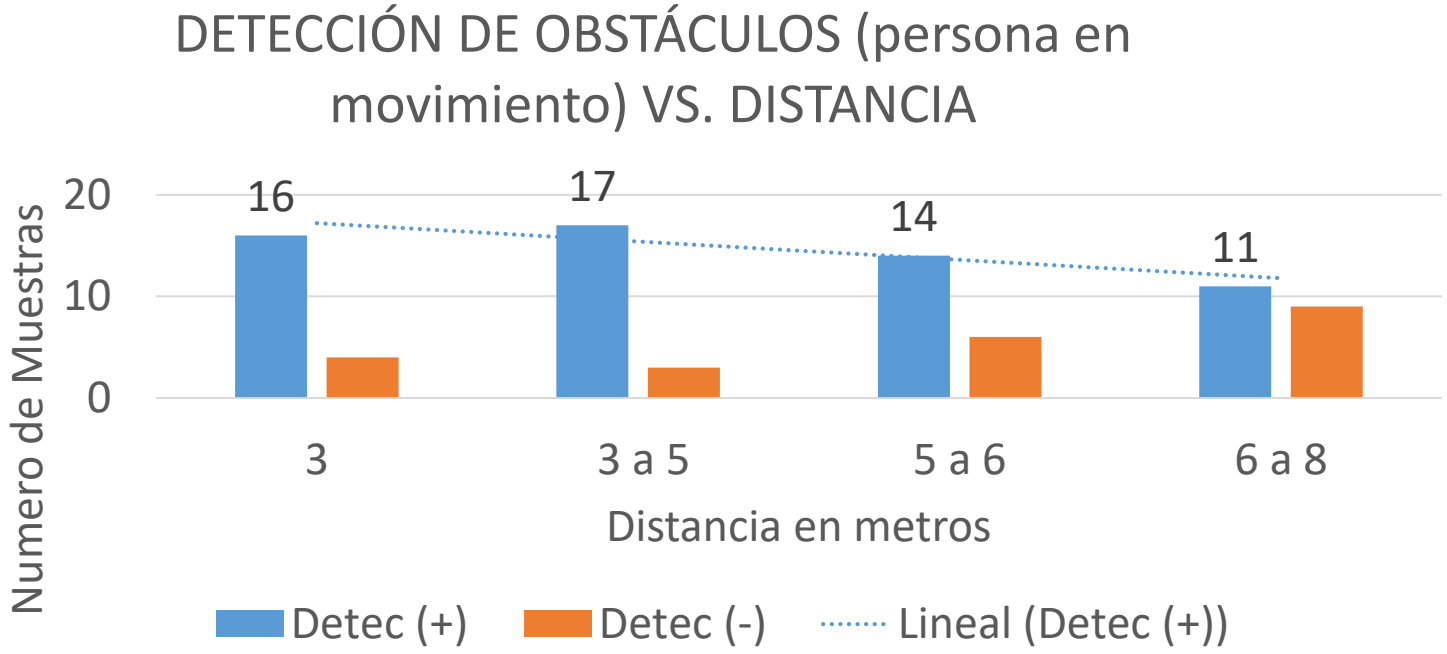
PRUEBAS Y RESULTADOS

- **Evaluación del algoritmo de detección de obstáculos**



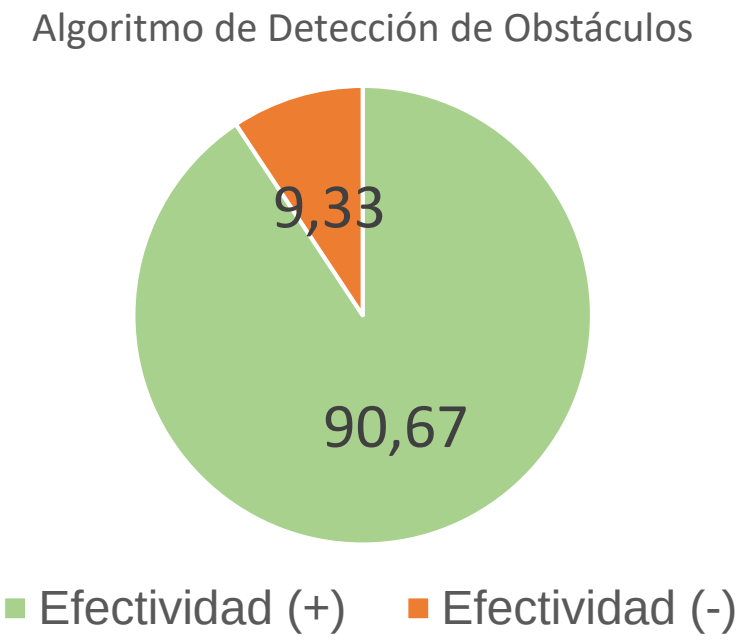
PRUEBAS Y RESULTADOS

- **Evaluación del algoritmo de detección de obstáculos**



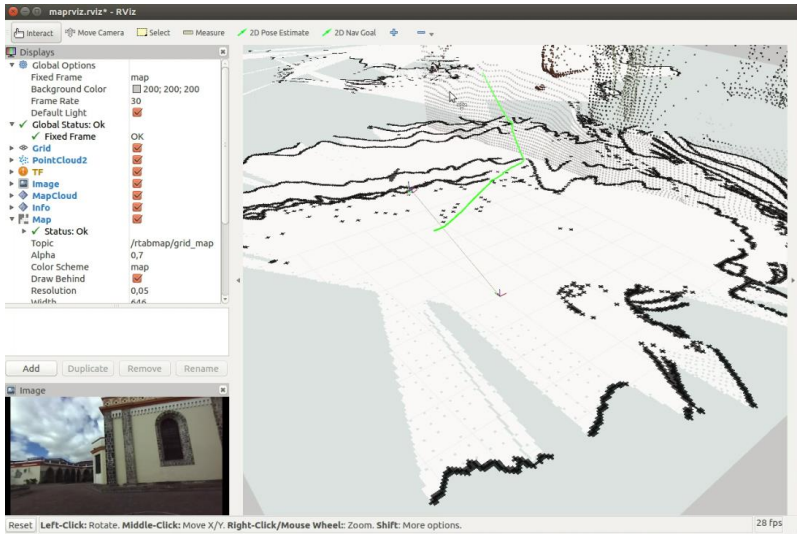
PRUEBAS Y RESULTADOS

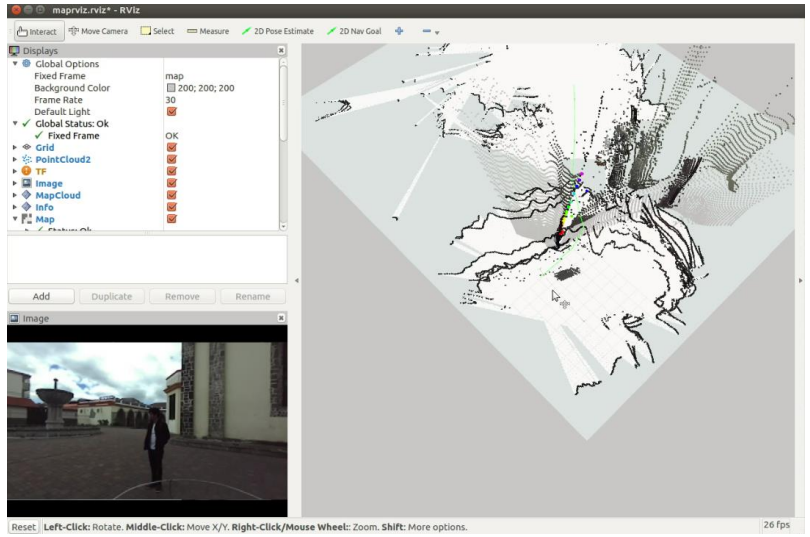
- **Evaluación del algoritmo de detección de obstáculos**



PRUEBAS Y RESULTADOS

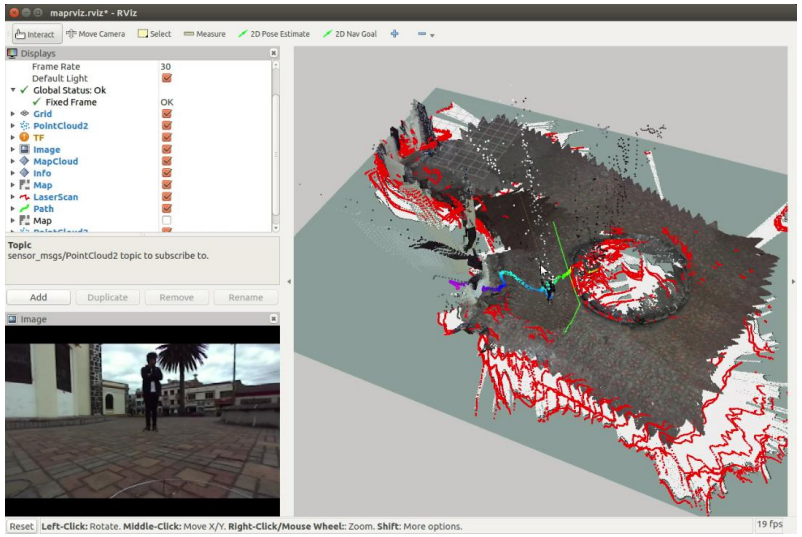
- **Evaluación de el algoritmo de evasión de obstáculos y seguimiento de ruta**

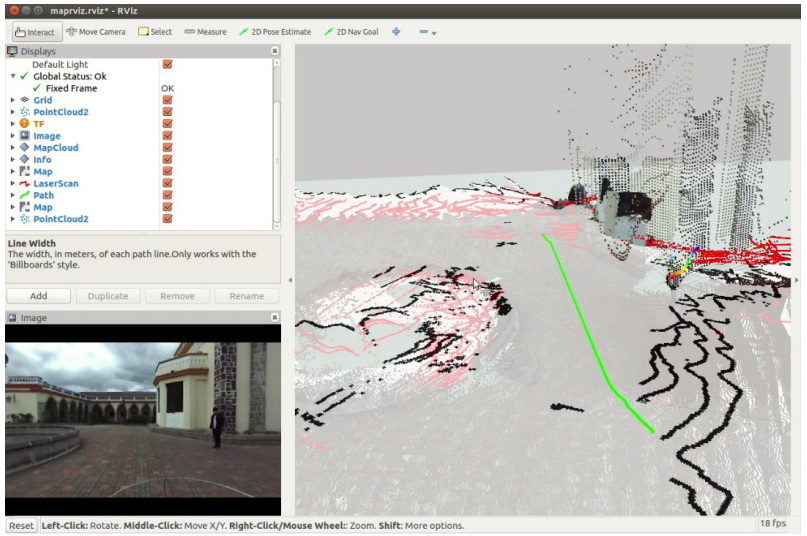




PRUEBAS Y RESULTADOS

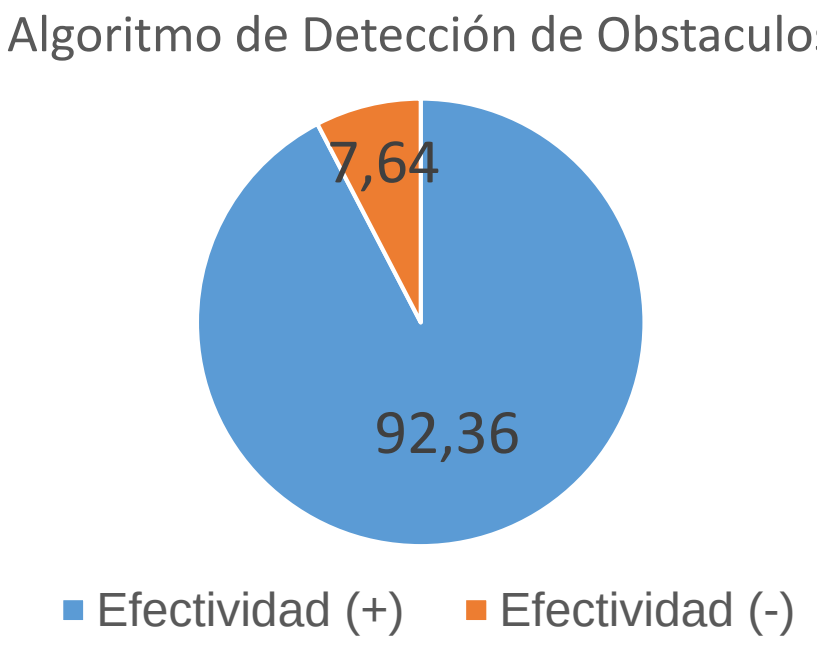
- **Evaluación de el algoritmo de evasión de obstáculos y seguimiento de ruta**





PRUEBAS Y RESULTADOS

- **Evaluación de el algoritmo de evasión de obstáculos y seguimiento de ruta**



PRUEBAS Y RESULTADOS

- **Ortomosaico**

CONCLUSIONES

- El uso del controlador PIXFALCON en el quadrotor fue de gran ventaja ya que, no solo se puede transmitir los parámetros de vuelo mediante protocolos MAVLink, sino que también, permite la inclusión de algoritmos complejos controlados por un sistema operativo que englobe todas las funciones.
- El software Mission Planner permite configurar y calibrar los sensores del quadrotor como: el giroscopio, GPS y su unidad de medición inercial. Además, facilita cargar la última versión de firmware "*ArduCopter*" disponible para el módulo PIXFALCON.

CONCLUSIONES

- Las herramientas SDK de la cámara estereoscópica ZED permiten variar la resolución de trabajo de esta, disminuyendo la cantidad de información que será procesada por la GPU del PC, ayudando a un muestreo más apropiado.

- El procesamiento de imágenes obtenidos de la cámara ZED, consume una gran cantidad de recursos computacionales. Esto se ve reflejado en la velocidad de comunicación entre los diferentes nodos de ros, también limita la velocidad de muestreo para el algoritmo de evasión de obstáculos lo que representa una dificultad al tratarse de obstáculos dinámicos.

CONCLUSIONES

- El uso de ROS facilita el desarrollo de algoritmos que permitan realizar las diferentes tareas que comprenden un sistema de piloto inteligente, como el: control de vuelo del quadrotor, la detección y evasión de obstáculos, usado al momento de generar un vuelo autónomo.

- El sistema operativo robótico ROS, presenta la ventaja de contar con ciertas herramientas que facilitan la localización y mapeado usando algoritmos para SLAM en 2D y 3D; procesos necesarios para la navegación autónoma y evasión de obstáculos utilizando cualquier plataforma robótica.

CONCLUSIONES

- ROS al ser un sistema operativo de código abierto, posee un repositorio oficial con varios trabajos desarrollados y probados en sistemas robóticos complejos por la gran comunidad de desarrolladores que aportan a la investigación de nuevos avances en la robótica.

- En este Proyecto se ha logrado comprender algunas de las funcionalidades básicas de ROS, además se ha empleado la librería rospy y roscpp que permiten crear nodos en lenguaje Python y C++, los cuales pueden interactuar entre sí con facilidad.

CONCLUSIONES

- El quadrotor presenta problemas de “armado” cuando se encuentra rodeado de edificios ya que no recibe una correcta señal de GPS. Esto causa un fallo en el chequeo previo del estado del UAV antes de que este puede encender sus motores.
- El desempeño del algoritmo de evasión de obstáculos, el cual se encarga de generar la ruta que debe seguir el quadrotor, depende principalmente de la calidad de imagen que proporciona la cámara ZED; por lo que se puede concluir que si se trabaja con la resolución más baja que ofrece la cámara y a una frecuencia de trabajo no superior a los 15 Hz se puede generar una ruta correcta con tiempo para que el quadrotor pueda responder.