



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN
División de Estudios Profesionales

Trabajo de Titulación

Opción TI: Tesis

Proyecto: "Evaluación De Api's Para Vant Mediante Un Framework En El Monitoreo De Contaminantes Criterio"

ALUMNO(S):	Miguel Angel Nava Martínez
No. CONTROL:	14481037
CARRERA:	Ingeniería en Sistemas Computacionales
ASESOR DE RESIDENCIA:	M.C José Isidro Hernández Vega
REVISORES:	

Guadalupe, N.L.

Marzo, 2020

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Nuevo León

División de Estudios Profesionales
Ingeniería en Sistemas Computacionales

TESIS DEFENDIDA POR

MIGUEL ANGEL NAVA MARTINEZ

Y APROBADA POR EL SIGUIENTE COMITÉ

M.C José Isidro Hernández Vega

Director del Comité

Miembro del Comité

Miembro del Comité

*Jefa del Departamento de Sistemas y
Computación*



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Sector NACE: RSGC 602 37
Terminación: 2012.09.22
Alcance: Proceso Educativo desde la
Inscripción hasta la entrega de Título Profesional



**EVALUACION DE API'S PARA VANT MEDIANTE UN FRAMEWORK EN EL
MONITOREO DE CONTAMINANTES CRITERIO**

TESIS

PARA OBTENER EL GRADO DE

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTA

MIGUEL ANGEL NAVA MARTINEZ

NO. DE CONTROL 14481037

DIRECTOR DE TESIS

M.C. JOSÉ ISIDRO HERNÁNDEZ VEGA

CO-DIRECTORA DE TESIS

M.C. DOLORES GABRIELA PALOMARES GORHAM

Cd. Guadalupe, Nuevo León, México, Enero de 2019

RESUMEN del proyecto de tesis de **MIGUEL ANGEL NAVA MARTINEZ**, presentada como requisito parcial para la obtención del grado de INGENIERO EN SISTEMAS COMPUTACIONALES. Guadalupe, Nuevo León. Enero de 2019.

TITULO DE TESIS

EVALUACION DE API'S PARA VANT MEDIANTE UN FRAMEWORK EN EL MONITOREO DE CONTAMINANTES CRITERIO

.

Resumen aprobado por:

M.C. JOSE ISIDRO HERNANDEZ VEGA

RESUMEN

En este proyecto de tesis se presenta la evaluación de API'S usadas para el control de Vehículos Aéreos No Tripulados (VANT) a través de un framework, su aplicación fue en el VANT utilizado en el proyecto de Monitoreo de Contaminantes Criterio.

Para la solución del problema se programó el framework llamado DRONEKIT para el envío de información de manera transparente entre el Dron, tarjeta de adquisición de datos y estación de tierra usados en el proyecto.

Con la configuración y programación de la API en el DRONEKIT se realizan pruebas experimentales transmisión de datos, tanto de la tarjeta de adquisición, como de los sensores del VANT utilizado.

Se comparó la eficiencia del framework DRONEKIT con respecto al de DJI y 3DR, en sus características, lenguajes de programación utilizados, manipulación y obtención de información del VANT.

Se realizó una evaluación de los frameworks mencionados, mostrada en el capítulo de resultados de la presente tesis.

Se dan especificaciones técnicas de los frameworks evaluados que sirvieron para desarrollar el prototipo de proyecto Monitoreo de contaminantes en el aire utilizando un sistema mecatronico inteligente para plumas industriales en el área metropolitana de Monterrey.

Palabras Clave:

API, VANT, Frameworks para VANTS, interfaces, Programación API, Plataformas de drones.

Abstract of the thesis presented by **MIGUEL ANGEL NAVA MARTINEZ** as a partial requirement to obtain the Engineering in Computer System. Guadalupe, Nuevo Leon, Mexico, January 2019.

TITLE OF THESIS

**EVALUATION OF API'S THROUGH A FRAMEWORK FOR VANT IN THE
MONITORING OF CONTAMINANTS CRITERIA**

Abstract approved by:

M.C. JOSE ISIDRO HERNANDEZ VEGA

ABSTRACT

This thesis project is presented the evaluation of API'S used for the control of Unmanned Aerial Vehicles (UAV) through a framework, its application was in the UAV used in the Polluter Monitoring Project Criterion.

For the solution of the problem, the framework called DRONEKIT was programmed to send information transparently between the Dron, data acquisition card and ground station used in the project.

With the configuration and programming of the API in the DRONEKIT, experimental data transmission tests are carried out, both on the acquisition card and on the UAV sensors used.

The efficiency of the DRONEKIT framework was compared with that of DJI and 3DR, in its characteristics, programming languages used, handling and obtaining information from the VANT.

An evaluation of the aforementioned frameworks was carried out, shown in the results chapter of this thesis.

Technical specifications of the evaluated frameworks that were used to develop the project prototype Monitoring of pollutants in the air are given using an intelligent mechatronic system for industrial pens in the metropolitan area of Monterrey.

Key Words: API, VANT, Frameworks for VANTS, interfaces, API Programming, Drone platforms.

DEDICATORIAS

Esta tesis se la dedico con mucho cariño a mi padre Inés Nava Castillo su gran sacrificio que realizo a lo largo de mi formación educativa el cual he obtenido el apoyo económico y laboral, también les quiero dedicar la tesis a mi asesor M.C. José Isidro Hernández Vega por su gran apoyo y asesoramiento en este proyecto ya que sin su ayuda y conocimientos no hubiera sido posible esta tesis, a mi profesora Martha Alicia Casillas Careaga que con su ayuda y sus conocimientos que nos brindó y con ayuda de sus materias que me impartió obtuve muy buenas herramientas para realizar este proyecto, a mis compañeros Mario Humberto Uriegas de León, Andrés González Rodríguez, Rodrigo Aldahyr Guadalupe Marroquín Rodríguez los cuales fueron un gran apoyo emocional y gracias a sus conocimientos logre obtener mis propios conocimientos en base a los de ellos .

AGRADECIMIENTOS

Al Instituto Tecnológico de Nuevo León que me ha permitido realizar mis estudios de licenciatura, así como utilizar sus instalaciones para el desarrollo de mi proyecto.

Al Programa de Desarrollo Profesional Docente (PRODEP) por la beca otorgada al incorpórame al proyecto de investigación: Monitoreo de contaminantes en el aire utilizando un sistema mecatrónico inteligente para plumas industriales en el área metropolitana de Monterrey, aprobado en la Convocatoria 2017 para el Fortalecimiento del Cuerpos Académico: Tecnologías Mecatrónicas Sustentables e Inteligentes, clave ITNLE-CA-5

También un agradecimiento a mis asesores Ing. José Isidro Hernández Vega y al profesor Natividad Hernández Romero que fueron los que me ayudaron a aplicar los conocimientos que he obtenido en la carrera y lo cual obtuve más conocimientos que el profesor me brindo y le agradezco con mayor felicidad a mi señor padre Inés Nava Castillo que es el que me ha sacado adelante y el cual me da motivaciones para salir adelante con mis estudios.

EVALUACION DE API'S MEDIANTE UN FRAMEWORK PARA VANT EN EL MONITOREO DE CONTAMINANTES CRITERIO

CONTENIDO

Contenido

LISTA DE FIGURAS.....	VIII
Página.....	VIII
Capítulo I Introduccion	1
1.1 Motivación del problema	1
1.2 Planteamiento del problema.....	1
1.3. Definición del problema	3
1.4. Justificación	4
1.5 Objetivos	4
1.5.1 Objetivo general.....	4
1.5.2 Objetivos particulares	5
1.6 Hipótesis	5
1.7 Alcances y Limitaciones	5
Capítulo II MARCO TEORICO.....	5
2.1 Que es una API	5
2.2 Tipos de API para VANTS	6
2.3 IMPORTANCIA DE LAS APIS EN LOS VANTS	7
2.4 Frameworks para trabajar con VANTS	7
2.5 APIS PARA VANTS	10
2.6 conceptos de comunicación API-VANT	14
Capítulo III.....	17
METODOS Y PROCEDIMIENTOS	¡Error! Marcador no definido.
3.1 Definición de la metodología empleada para el proyecto.....	17
3.2 Material y equipo utilizado (SW)	26
3.3 Consideraciones para el diseño	32
Capítulo IV	33
Resultados	¡Error! Marcador no definido.

4.1 Experimentación del prototipo en toma de muestras.....	33
4.2 Análisis de resultados	34
4.3 Comparación entre las diferentes APIS para VANTS.....	34
Capítulo IV	36
Conclusiones.....	¡Error! Marcador no definido.
REFERENCIAS	38

LISTA DE FIGURAS

<i>Figura</i>		Página
1	Figura. 1 Características del DJI	12
2	Figura. 2 Manipulación de vuelos autónomos	12
3	Figura. 3 Mapeo de figuras en 2D Y 3D	13
4	Figura. 4 Topografías y Fotografías de terrenos	13
5	Figura. 5 Interfaz de Raspberry con sus librerías y los puertos hacia el drone	21
6	Figura. 6 Programa de Python con Dronekit Funcionando	26
7	Figura. 7 Resultados adquiridos del dron y la API	26
8	Figura. 8 Dron de cuatro hélices. Cuadricoptero	27
9	Figura. 9 Computadora de vuelo Pixhawk	27
10	Figura. 10 GPS	27
11	Figura. 11 Antenas de Telemetría	28
12	Figura. 12 Batería de Litio	28
13	Figura. 13 Motores para los drones	28
14	Figura. 14 Radiocontrol para operar el dron.	29
15	Figura. 15 Una laptop que hace función de estación de tierra	29
16	Figura. 16 Placa raspberri pi 3 versión B	30
17	Figura. 17 Símbolo de distribución Rabian Jessie de Linux	30
18	Figura. 18 Manejador de Base de Datos	31
19	Figura. 19 Mission Planner	32
20	Figura. 20 Resultados Obtenidos	32

LISTA DE TABLAS

Tabla		Página
I	Tabla 1. Diferencia en los criterios de transferencia de datos entre el Dronekit y el DJI	34
II	Tabla 2. Diferencia en los criterios de transferencia de datos entre el Dronekit y el 3DR	35

CAPÍTULO I INTRODUCCIÓN

1.1 Motivación del problema

Un VANT es un vehículo aéreo tripulado que no lleva abordo a un operador humano, utiliza las fuerzas aerodinámicas para generar la sustentación, puede volar autónomamente o ser tripulado de forma remota, que puede ser recuperable y que puede transportar una carga. Existen plataformas de trabajo cerradas y de arquitectura abierta.

Para su manipulación y programación en las arquitecturas abiertas hay una variedad de Interfaz de programación de aplicaciones en inglés Application Programming Interface.

Es importante saber cuáles son las más adecuadas para el VANT que se está trabajando. La variedad de protocolos ya desarrollados contribuye a desarrollar aplicaciones para el VANT de manera flexible y rápida.

Sin embargo es importante conocer cuáles son las API'S más adecuadas para el control del VANT.

La tesis esta contextualizada en un proyecto de investigación para el monitoreo de contaminantes. La contaminación es un problema que ha ido en crecimiento y el monitoreo de estos contaminantes se ha estado realizando desde 1992 en el estado de Nuevo León por estaciones fijas de monitoreo que tienen un área de escaneo más limitado. El uso de nueva tecnología abre las posibilidades para una expansión en las áreas de monitoreo y obtener resultados más acertados

1.2 Planteamiento del problema

La contaminación ambiental ha estado en monitoreo gracias a las estaciones fijas proporcionadas por el SIMA [Sistema Integral de Monitoreo Ambiental del estado de Nuevo León] proporciona los datos de calidad del aire por medio de su página web en la que la información es presentada por medio de gráficas e imágenes que son fáciles de interpretar por el usuario.

Un punto en contra en cuanto a las estaciones de monitoreo es que se encuentran fijas en distintos puntos [de la ciudad/estado] y la información proporcionada puede ser solo relevante según las zonas donde se encuentran implantadas, por ello la información que se obtiene no es de mucha ayuda para otras áreas de la ciudad porque no cubren un mayor rango posible.

El monitoreo de contaminantes se realiza en puntos específicos del Area Metropolitana de Monterrey, su monitoreo está en función del alcance de las estaciones fijas. Realizar un monitoreo en un aérea especifica donde las estaciones no registren un monitoreo se convierte en una problemática para el seguimiento de los contaminantes, tal como en chimeneas industriales o fuentes más generadores de contaminantes ambientales.

El hecho que el monitoreo solo se haga a partir de estaciones fijas puede ocasionar que la información que se muestra por medio de una página web no sea tan relevante y que no abarque todas las zonas dentro de un área determinada.

Con este proyecto se puede resolver ese problema porque el monitoreo será constante y en tiempo real y en lugares donde las estaciones fijas no llegan.

Tener un sistema de monitoreo de contaminantes móviles podría en primer lugar abarcar mayores áreas en las cuales monitorear las condiciones de la zonas, en segundo ver que contaminantes predominan y por último ver qué soluciones se pueden ofrecer para reducir la contaminación según los resultados obtenidos.

Desarrollar aplicaciones para drones requiere de herramientas diversas, como interfaces, que permitan una comunicación entre un dron y otras aplicaciones que se relacionan a la recopilación de información de los sensores que se interconectan al dron. Las herramientas para integrar una aplicación mediante algún dron, requiere de plataformas de trabajo adecuadas, estas herramientas se convierten en un marco de trabajo, tal como la usada en este proyecto.

Ver la efectividad de esta herramienta es de importancia para la asertividad de los resultados, compararla con otras que existan en el mercado contribuirá a una mejor selección de frameworks para drones

1.2.1 Antecedentes Del Problema

Actualmente, el país cuenta con el Reglamento de Estándares Nacionales de Calidad Ambiental del Aire, aprobado mediante Decreto Supremo N° 074-2001-PCM que es un documento de gestión de la calidad del aire en el país, el cual contribuye a determinar los criterios para la protección de la calidad ambiental, así como los lineamientos estratégicos para alcanzar progresivamente la protección de la salud de las personas. El Decreto Supremo N° 074-2001-PCM en su artículo 12 señala que el monitoreo de la calidad del aire y la evaluación de los resultados en el ámbito nacional es una actividad de carácter permanente, a cargo del Ministerio de Salud a través de la Dirección General de Salud Ambiental (DIGESA), quien podrá encargarse a instituciones públicas o privadas dichas labores. Por otra parte, se cuenta con el Reglamento de los Niveles de Estados de Alerta Nacionales para Contaminantes del Aire, aprobado mediante Decreto Supremo N° 009-2003-SA, documento de gestión que permite la implementación de un conjunto de medidas predeterminadas para la prevención de riesgos a la salud y la exposición aguda de la población a los contaminantes del aire. Es por ello, que los datos generados de los programas de monitoreo de la calidad del aire debe contar con un nivel establecido de confiabilidad y comparabilidad, pues serán una herramienta fundamental para la toma de decisiones.

Es posible el empleo de los sistemas de vehículos aéreos no tripulados (drones) en investigaciones del medio ambiente, ciencias en general, monitoreo, etc.

Es por eso que llegamos a la conclusión de realizar este proyecto para el mejor monitoreo del aire y saber en qué nivel están los contaminantes. Y la función que es importante de este proyecto grande es el de la comunicación e interconexión entre un dron y una estación de tierra y eso se logra mediante una aplicación o software. [3]

1.3. Definición del problema

Los VANT en la actualidad tienen una amplia variedad de modelos y tamaños en el mercado, pero no toda la disponibilidad de adaptabilidad de componentes externos,

carga extra (carga útil) y compatibilidad de comunicación con dispositivos externos, que son las principales características que debe de cumplir

Es importante conocer las características de la API para el VANT del proyecto de investigación y realizar un comparativo con las existentes en el mercado, para contar con una interfaz de programación adecuada para el VANT. Tomando como base El Dronekit Framework

Las características que deben de cumplir son:

- Transferencia de datos en tiempo real
- Compatibilidad con los manejadores de Bases de Datos
- Datos que se puedan interpretar en la web

1.4. Justificación

Construir apps remotas vía web que controlen drones, requiere de una integración de tecnologías. Hoy en día se utilizan una variedad de frameworks que ayudan a facilitar el desarrollo de este tipo de aplicaciones, pero su uso requiere de conocimientos especializados, pruebas experimentales, desarrollo de código. El Dronekit Framework cuenta con estas necesidades, además de ser open source, por lo que es necesario probar y aplicar este entorno de trabajo en el proyecto monitoreo de contaminantes.

El desarrollo de sistemas de control para este tipo de vehículos no es trivial, debido principalmente a la dinámica tan compleja inherente a los sistemas aerodinámicos. El Dronekit Framework es una plataforma que nos ayuda a realizar configuraciones a los códigos abiertos para poder recopilar información de los diferentes sensores con los que cuenta cada drone y poder transferir esa información mediante una conexión a una base de datos.

1.5 Objetivos

1.5.1 Objetivo general

Evaluar 2 Interfaces para la programación de aplicaciones (API) en el control de VANTS, tomando como referencia el Dronekit Framework

1.5.2 Objetivos particulares

- Realizar un código para generar la transmisión de datos
- Utilizar e implementar Frameworks o APPI's para obtener información de los sensores que se encuentran en un drone y la información de los sensores que se interconectan con una tarjeta embebida en tiempo real.
- Obtener mediante el código la información recopilada de los contaminantes del aire
- Evaluar el trabajo realizado con el Dronekit Framework con respecto a las 2 API estudiadas en la tesis

1.6 Hipótesis

Existe una diferencia en las plataformas para el control de drones mediante una API que ayuden a desarrollar aplicaciones en sistemas de tiempo real

1.7 Alcances y Limitaciones

El alcance para este proyecto es la evolución de la extracción de información de un VANT mediante una aplicación, la cual sirvió para enviar información a una estación de tierra y así mismo transferirla a una base de datos. La información que se recopiló del drone fue de utilidad para la toma de decisiones y precauciones acerca del estado en el que se encuentra el medio ambiente y para informar o realizar acciones para que la gente se dé cuenta de lo que está pasando y realizar alguna solución. El límite establecido en el proyecto fue una API en la que se realizara la conexión y envió de datos de los contaminantes del aire que se llegaron a captar mediante el drone y a su vez enviar esos datos a una estación de tierra. Ya que esto no es visible para el usuario es muy importante que en el receptor pueda verificar si los datos están llegando de manera adecuada y gracias a esa interfaz los datos pueden ser mandados a una base de datos para su mejor manipulación.

CAPÍTULO II MARCO TEORICO

2.1 Que es una API

Una API (siglas de 'Application Programming Interface') es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse

entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Las API pueden servir para comunicarse con el sistema operativo (WinAPI), con bases de datos (DBMS) o con protocolos de comunicaciones (Jabber/XMPP). En los últimos años, por supuesto, se han sumado múltiples redes sociales (Twitter, Facebook, Youtube, Flickr, LinkedIn, etc) y otras plataformas online (Google Maps, WordPress...), lo que ha convertido el social media marketing es algo más sencillo, más rastreable y, por tanto, más rentable.

Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software (o de la infraestructura ya existente en otras plataformas) para no estar reinventando la rueda constantemente, reutilizando así código que se sabe que está probado y que funciona correctamente. En el caso de herramientas propietarias (es decir, que no sean de código abierto), son un modo de hacer saber a los programadores de otras aplicaciones cómo incorporar una funcionalidad concreta sin por ello tener que proporcionar información acerca de cómo se realiza internamente el proceso. [9]

2.2 Tipos de API para VANTS

Existen dos tipos de APIS para los VANT unos son de código abierto y el otro de código cerrado.

Código abierto:

Código abierto (en idioma inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado Software libre. [10]

Código cerrado:

Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones, o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

Para la Fundación para el Software Libre (FSF) este concepto se aplica a cualquier software que no es libre o que sólo lo es parcialmente (semilibre), sea porque su uso, redistribución o modificación está prohibida, o requiere permiso expreso del titular del software. En el software no libre una persona física o jurídica (compañía, corporación, fundación, etc.) posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades (donde el acceso al código fuente es una condición previa); de distribuir copias; o de mejorar el programa y hacer públicas las mejoras (para esto el acceso al código fuente es un requisito previo). De esta manera, un software sigue siendo no libre aún si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución. [7]

2.3 Importancia de las apis en los VANT´S

Una API, en su nivel más básico, permite que los productos y servicios puedan hablar unos con otros; es un software que expone las funciones con las que cuenta su sistema para que otras aplicaciones puedan integrarse a éste y aprovechen sus capacidades.

En la actualidad las compañías también utilizan sus propias APIs para construir sistemas internos, sitios web y aplicaciones móviles hechas a la medida. Cabe señalar que gracias a las APIs es posible generar nuevos tipos de relaciones, con lo cual la web puede volverse más social.

Dependiendo de sus características, las APIs pueden utilizarse para satisfacer las necesidades del comercio electrónico, pagos en línea, redes sociales, cómputo en la nube, entre muchas otras que también pueden ser enfocadas a los drones y la función de diferentes aplicaciones en drones son para monitoreo, para propósitos del medio ambiente, también se puede manejar para el mundo de la fotografía entre otros. [1]

La importancia que veo en las APIS en los drones es que gracias a estas APIS pueden lograr muchos proyectos futuros, en los cuales los drones serán más

utilizados porque pueden llegar a lugares donde otros vehículos e incluso las personas no pueden alcanzar, son muy importante las APIS porque con ellas podemos realizar instrucciones en los cuales el dron puede entender y realizar esas acciones que programamos y lo que importa es que la API que se realice para cada proyecto sea compatible y viable para el proyecto que se le indique.

2.4 Frameworks para trabajar con VANT´S

La mayoría de los proyectos tienen partes comunes necesarias para el funcionamiento como, por ejemplo, acceso a base de datos, validación de formularios o seguridad. Un framework nos evita tener que programar estas partes, de esta manera nos resulta más fácil centrarnos en programar la aplicación.

Los frameworks están basados en patrones de desarrollo, normalmente MVC (Modelo-Vista-Controlador) que ayudan a separar los datos y la lógica de negocio de la interfaz con el usuario. Vamos, que gracias a ellos, lo tenemos todo más ordenado.

Está claro que un framework siempre te va permitir hacer cosas de una manera fácil y segura, que para ti serían imposibles o al menos te costaría mucho tiempo hacerlas.

Si tenemos en cuenta los puntos anteriores, sabremos que desarrollar una aplicación con un framework nos permite hacerlo más rápido, más limpio y más seguro.

Una vez que tenemos claro por qué usar un framework, tenemos que formularnos la siguiente pregunta:

2.4.1 ¿Qué framework usar?

Debemos tener claro cuáles son las características y necesidades de nuestro proyecto para elegir que tecnología utilizar. Si con un framework sencillo tenemos bastante, no hay necesidad de usar uno más complicado, ya que lo único que vamos a conseguir es complicarnos la vida. Por otro lado, utilizar el mismo framework que te fue bien en otro proyecto, tal vez no sea la decisión correcta. Dicho esto te

exponemos una serie de factores que debes considerar a la hora de elegir uno u otro:

2.4.2. Factores a considerar a la hora de elegir un framework u otro

1.-Desarrollo web con PHP 5

2.-Soporte de comunidad

Es muy importante que el framework elegido tenga una comunidad activa y participativa. De esta manera te va a ser más fácil conseguir ayuda en los momentos difíciles. Visita la web de la comunidad y asegúrate que obtendrás consejos y soluciones cuando los necesites.

3.-Documentación

La documentación de un framework debe estar actualizada y ser lo bastante amplia para que te ayude con los problemas que te puedan surgir. Una herramienta bien documentada es mucho más fácil de utilizar.

4.-Simplicidad y potencia

Algunos framework son muy potentes pero difíciles de entender. El código generado por el framework debe ser claro y limpio, que no tengas que preocuparte de las peculiaridades internas del framework.

5.-Arquitectura MVC

Como hemos comentado anteriormente, este tipo de arquitectura nos ayuda a tener separado la lógica de negocios, los datos y la presentación. De esta manera nuestro código esta mucho más ordenado, nos hace más fácil cualquier tipo de modificación y actualización y facilita mucho el trabajo en equipo.

6.-Reutilización

Es imprescindible que los elementos desarrollados puedan ser reutilizados en varios proyectos.

7.-Patrón Active Record

Es interesante tener una clase especifica que se encargue de realizar las operaciones de modificación y consulta de una determinada tabla de una base de datos. De esta forma nuestra aplicación quedará aislada del trabajo con SQL.

8.-Posicionamiento

Dependiendo del proyecto puede ser que no nos interese estar bien posicionado, pero si no es así, es interesante que el framework ofrezca funcionalidades como url's amigables, etc., para no tener que rehacer el proyecto más adelante.

9.-Seguridad

Puede ser un tema que en un principio pasamos por alto y luego ya es demasiado tarde. Asegúrate que el framework este desarrollado tomando la seguridad como punto de partida. Si un framework es muy potente pero no es seguro no nos sirve de nada.

Después de todos estos consejos lo último que te queda es probar, ya que es la única manera de saber si un framework se adapta a las necesidades de tu proyecto.

[6]

2.5 APIS PARA VANTS

DJI SDK

El DJI se especializo más en el mundo de la fotografía ya que cuenta con sus propios productos de cámaras que tienen una alta resolución y gracias a eso pueden realizar trabajos como los que son fotografías de paisajes o mapeo de ciertos lugares. El DJI no es muy compatible con todos los drones es decir que es compatible por lo regular con drones de su misma empresa. [5]

El SDK de 3DR

3DR Site Scan es la plataforma completa de software para drones diseñada para profesionales de la construcción y la ingeniería.

La construcción es dura. Los drones lo hacen más fácil.

Construimos Site Scan porque creemos que hay un mejor futuro para la construcción donde los equipos trabajan sincronizados, los problemas se solucionan pronto y los proyectos se terminan a tiempo.

Estamos trabajando para lograr este futuro transformando los drones en potentes herramientas de captura de la realidad que apoyan a su equipo desde la primera encuesta hasta la inspección final.



Figura. 1 Características del DJI

Captura datos aéreos en tu sitio de trabajo

Elige tu modo de vuelo y desliza para despegar. Vuela de forma autónoma con drones 3DR con nuestra aplicación móvil para capturar tu proyecto de forma rápida y fácil en detalle.



Figura. 2 Manipulación de vuelos autónomos

Convierta las fotos de drones en mapas 2D precisos y modelos 3D con nuestra plataforma basada en la nube. Nuestras herramientas de análisis facilitan el seguimiento de los cambios, realizar mediciones, superponer archivos de diseño, compartir actualizaciones de progreso y más.

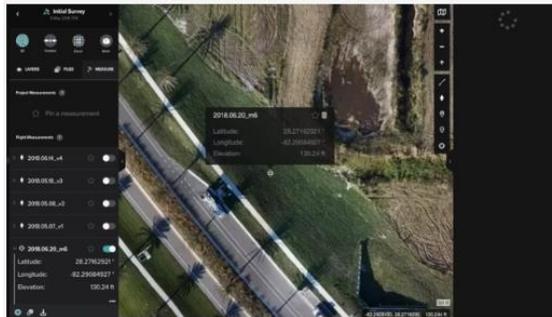


Figura. 3 Mapeo de figuras en 2D Y 3D

Los datos de Site Scan funcionan a la perfección con las herramientas que usa todos los días. Exporte a GIS y a los formatos de archivo nativos de Autodesk con solo un clic y comparta datos a la perfección con nuestra integración BIM 360.

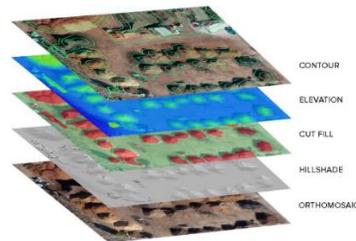


Figura. 4 Topografías y Fotografías de terrenos

Como el SDK de DJI el de 3DR es muy similar lo utilizan mas para mapeo y conseguir imágenes, fotografías y videos de diferentes lugares en donde no se pueden llegar fácilmente. En este SDK no es muy compatible con muchos tipos de drones solo con muy pocos. [4]

Dronekit

DroneKit-Python es un SDK, permite a los desarrolladores crear aplicaciones que se ejecutan en una computadora de a bordo y comunicarse con el controlador de vuelo ArduPilot utilizando un enlace de baja latencia lo cual gracias a las telemetrías se puede realizar la transferencia de datos por medio de la radio frecuencia.

Las aplicaciones incorporadas pueden mejorar significativamente el piloto automático, agregando mayor inteligencia al comportamiento del vehículo y realizando tareas que son computacionalmente intensivas o sensibles al tiempo (por

ejemplo, visión artificial, planificación de ruta o modelado 3D). DroneKit-Python también se puede utilizar para aplicaciones de estaciones terrestres, comunicándose con vehículos a través de un enlace RF de latencia más alta.

La API se comunica con los vehículos a través de MAVLink. Proporciona acceso programático a la información de telemetría, estado y parámetros de un vehículo conectado, y permite la administración de la misión y el control directo sobre el movimiento y las operaciones del vehículo.

Compatibilidad

DroneKit-Python es compatible con vehículos que se comunican mediante el protocolo MAVLink (incluida la mayoría de los vehículos fabricados por 3DR y otros miembros de la base DroneCode). Se ejecuta en Linux, Mac OS X o Windows.

DroneKit-Python está validado contra, y por lo tanto es más compatible con, la plataforma ArduPilot UAV. Los vehículos con otros pilotos automáticos pueden ser menos compatibles debido a las diferencias en la adherencia / interpretación de la especificación MAVLink

Características de API

La API proporciona clases y métodos para:

- Conectarse a un vehículo (o varios vehículos) desde un script
- Obtenga y configure el estado del vehículo / telemetría e información de parámetros.
- Reciba notificaciones asincrónicas de cambios de estado.
- Guíe un UAV a la posición especificada (modo GUIADO).
- Envíe mensajes personalizados arbitrarios para controlar el movimiento UAV y otro hardware (modo GUIADO).
- Crear y administrar misiones de waypoint (modo AUTO).
- Anular la configuración del canal RC. [11]

2.6 Conceptos de comunicación API-VANT

Radio Frecuencia

Radiofrecuencia (abreviado RF), también denominado espectro de radiofrecuencia, es un término que se aplica a la porción menos energética del espectro electromagnético, situada entre los 3 hercios (Hz) y 300 gigahercios (GHz).

El hercio es la unidad de medida de la frecuencia de las ondas, y corresponde a un ciclo por segundo. Las ondas electromagnéticas de esta región del espectro, se pueden transmitir aplicando la corriente alterna originada en un generador a una antena.

El primer sistema práctico de comunicación mediante ondas de radio fue el diseñado por Guillermo Marconi, quien en el año 1901 realizó la primera emisión trasatlántica radioeléctrica. Actualmente, la radio toma muchas otras formas, incluyendo redes inalámbricas, comunicaciones móviles de todo tipo, así como la radiodifusión.

En este proyecto la transmisión de los datos que se realizaron por radio frecuencia fueron obtenidos gracias a las antenas de telemetría lo cual contienen un emisor y un receptor.[12]

Telemetría

Telemetría es una técnica automatizada de las comunicaciones con la ayuda de que las mediciones y recopilación de datos se realizan en lugares remotos y de transmisión para la vigilancia. Esta técnica utiliza comúnmente transmisión inalámbrica, aunque original de los sistemas de transmisión utilizados por cable. Los usos más importantes de telemetría incluir el clima de recopilación de datos, supervisión de plantas de generación de energía y hacer el seguimiento de vuelos tripulados y no tripulados.

Un sistema de telemetría normalmente consiste de un transductor como un dispositivo de entrada, un medio de transmisión en forma de líneas de cable o las ondas de radio, dispositivos de procesamiento de señales, y dispositivos de

grabación o visualización de datos. El transductor convierte una magnitud física como la temperatura, presión o vibraciones en una señal eléctrica correspondiente, que es transmitida a una distancia a efectos de medición y registro.

Gracias a la radiofrecuencia y a las herramientas que utilizamos para realizar la comunicación y transferencia de datos que son las telemetrías ahora falta el cómo interpretar los datos que se quieren transferir y eso se menciona en el siguiente punto.

Raspberry Pi 3 Modelo B

Construido con el último procesador Broadcom 2837 ARMv8 de 64 bits, el Raspberry Pi 3 Model B es más rápido y más potente que sus predecesores.

Ha mejorado la administración de energía para admitir dispositivos USB externos más potentes y ahora viene con conectividad inalámbrica y Bluetooth incorporada. Para aprovechar al máximo la administración de energía mejorada en la Raspberry Pi 3 y brindar compatibilidad incluso para dispositivos aún más potentes en los puertos USB, se recomienda un adaptador de 2.5 A.

Especificaciones técnicas:

- Broadcom BCM2837 de 64 bits del procesador de 64 bits ARMv8 Quad Core potencia de placa única

ordenador funcionando a 1,2 GHz

- 1 GB de RAM

- Wi-Fi a bordo BCM43143

- Bluetooth Low Energy (BLE) a bordo

- 40 pines GPIO extendida

- 4 x puertos USB2

- 4 polo de salida estéreo y un puerto de vídeo compuesto

- Tamaño completo HDMI

- puerto de la cámara de CSI para conectar la cámara Frambuesa Pi
- puerto de pantalla DSI para la conexión de la pantalla táctil Frambuesa Pi
- puerto MicroSD para cargar el sistema operativo y el almacenamiento de datos
- Mejorado cambiaron Micro Fuente de alimentación USB (ahora admite hasta 2.5 Amps)

Esta tarjeta inteligente fue la que se utilizó para lograr este proyecto de transferencia de los datos ya que en esta tarjeta se puede instalar cualquier sistema operativo y el cual se pueden efectuar diversas tareas en ella como realizar programas en distintos lenguajes de programación.

Lenguaje de programación Python

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos.

La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, <http://www.python.org/>, y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables

Este lenguaje hizo que el proyecto funcionara correctamente ya que las librerías del dronekit están representadas en este lenguaje de programación.

CAPÍTULO III MÉTODOS Y PROCEDIMIENTOS

3.1 Definición de la metodología empleada para el proyecto

Se basó en la metodología de prototipos consiste en la obtención de requisitos en el cual se plantea lo que se realizara,

- El diseño global en el que comprende la simulación del sistema, los diseños de circuitos en que esta integrado nuestro drone, así como la estructura del prototipo y componentes a utilizar,
- En la construcción del prototipo consiste en tomar las mediciones del material a utilizar

- Desarrollo del prototipo aquí se empieza a armar el prototipo con todas los circuitos o componentes necesarias como por ejemplo las tarjetas inteligentes, los sensores MQ, la configuración de la API que se realizó y cada una de las partes de nuestro dron para construirlo,
- Refinamiento del prototipo en este paso solo se verifica que el prototipo está en funcionamiento y algún desperfecto que pueda tener
- Por último el sistema terminado, el prototipo funcionando.

Procedimiento

Primeramente se tuvo que instalar el raspbian en la raspberry y posteriormente se tuvo que realizar todos estos comandos aplicados en una terminal de la raspberry. para habilitar i2c y SPI

```
sudo raspi-config
```

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install i2c-tools
```

```
sudo adduser pi i2c
```

-Pin 3 (SDA) -> Arduino A4 (SDA) Pin 5 (SCL) -> Arduino A5 (SCL)

```
i2c python support
```

```
sudo apt-get install python-smbus
```

-Soporte UART python

```
sudo apt-get install python-serial
```

-Python libs y Mavlink Proxy

-Instalación del Python

```
sudo apt-get install screen python-wxgtk2.8 python-matplotlib python-opencv  
python-pip python-numpy python-dev libxml2-dev libxslt-dev
```

```
sudo pip install pymavlink
```

```
sudo pip install mavproxy
```

```
sudo -s
```

-Instalacion del Mavlink Proxy

```
mavproxy.py --master=/dev/ttyUSB0 --baudrate 57600 --sitl 127.0.0.1:5501 --out  
127.0.0.1:14550 --aircraft DroneName
```

-A través de un cable USB

```
mavproxy.py --master=/dev/ttyACM0 --baudrate 115200 --sitl 127.0.0.1:5501 --out  
127.0.0.1:14550 --aircraft DroneName
```

-Marco DroneKit

```
sudo apt-get install python-pip python-dev python-numpy python-opencv python-  
serial python-pyparsing python-wxgtk2.8 libxml2-dev libxslt-dev
```

```
sudo pip install droneapi echo "module load droneapi.module.api" >> ~/.mavinit.scr
```

```
pip install dronekit
```

```
pip install dronekit-sitl
```

-Instalar bibliotecas de sensores DHT22

```
sudo apt-get update
```

```
sudo apt-get install build-essential python-dev
```

-Ingresar a esta pagina

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git cd  
Adafruit_Python_DHT
```

```
sudo python setup.py install
```

-El MPL3115A2 requiere un comando de arranque repetido apropiado en su comunicación I2C, esta línea es necesaria en cada arranque.

```
sudo su -
```

```
echo -n 1 > /sys/module/i2c_bcm2708/parameters/combined
```

```
exit
```

- Pruebas

-Lista de direcciones disponibles

```
i2cdetect -y 1
```

-Encuentra tu dispositivo USB

```
ls /dev/tty*
```

```
Init Apps en rapi startup
```

```
sudo crontab -e
```

-al final del archivo

```
@reiniciar
```

Todos estos comandos fueron necesarios para realizar la instalación de las librerías e incluso para obtener comandos en los cuales se requerirán para el proyecto.

También en esta metodología investigamos acerca de la forma en que se realiza la transmisión de los datos a través de la radio frecuencia y de cómo representar eso en el programa que se desarrolló.

Uno de los factores que se ocupó fue la de representar las variables que se obtendrán del dron a través de la transmisión de estos datos es decir, saber qué tipo de datos estamos manejando para que al momento de recibirlos saber cómo manejar esos datos en la base tierra.

En el siguiente diagrama se muestra la interfaz de la Raspberry Pi 3 Modelo B con sus librerías, los puertos hacia el dron

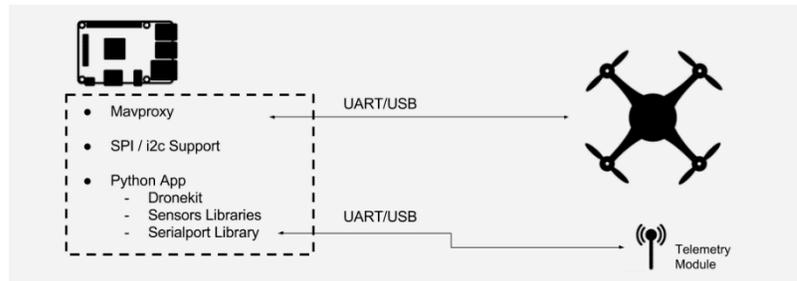


Figura. 5 Interfaz de Raspberry con sus librerías y los puertos hacia el drone

Este es el código que se utilizó para la transmisión de los datos utilizando las librerías del dronekit.

Se muestra las librerías que utilizamos, también se muestra la conexión a la telemetría

```

from dronekit import connect, VehicleMode, LocationGlobal,
LocationGlobalRelative
# En esta parte es donde se importan las librerías del dronekit
y es donde se manda a llamar a esas librerías
from pymavlink import mavutil

import spidev
import sys
import time
import argparse
import json

import smbus
import serial

# I2C Constants
# ADDR = 0x60
baro_addr = 0x60
CTRL_REG1 = 0x26
PT_DATA_CFG = 0x13

# Connecting telemetry module
try:
    ser = serial.Serial('/dev/ttyUSB0', 57600, timeout=5)
# este comando es para realizar la conexión al vehículo a
través de USB
except e:
    print "Serial is not available."

```

En este punto del código se extrae los datos que queremos obtener específicamente del drone.

```

# Connecting Mavproxy
try:
    global vehicle
    parser = argparse.ArgumentParser(description='Print out
vehicle state information. Connects to SITL on local PC by
default.')
    parser.add_argument('--connect',
default='127.0.0.1:14550',help="vehicle connection target.
Default '127.0.0.1:14550'")
    args = parser.parse_args()
    print "\nConnecting to vehicle on: %s" % args.connect
    vehicle = connect(args.connect, wait_ready=True)
    vehicle = connect('127.0.0.1:14550', wait_ready=True)
# el primer parametro especifica la direccion de destino
# el wait_ready se usa para determinar si connect()regresa
inmediatamente en la conexión o si espera hasta que se
completan algunos parámetros y atributos del vehículo
#En la mayoría de los casos, debe usar wait_ready=Truepara
esperar en el conjunto predeterminado de parámetros
except Exception, e:
    print "Error to Connect"

```

En esta parte del código se expresa el método en donde nos manda información si el dron está armado o no que quiere decir con esto, es cuando los motores están encendidos o listos para el vuelo.

También se muestra en qué tipo de modo de vuelo se encuentra el dron.

```

# BGN: GENERAL FUNCTIONS

def writeNumber(value):
    bus.write_byte(address, int(value))
    return -1

def readNumber():
    number = bus.read_byte(address)
    return number

def set_arm(arm):
    if arm:
        print "Arming motors"
        vehicle.armed = True
    else:
        print "Disarming motors"
        vehicle.armed = False

```

```

def set_mode(mode):
    if mode == 1:
        vehicle.mode = VehicleMode("STABILIZE")

    if mode == 2:
        vehicle.mode = VehicleMode("ALT_HOLD")

    if mode == 3:
        vehicle.mode = VehicleMode("LOITER")

    if mode == 4:
        vehicle.mode = VehicleMode("RTL")

    if mode == 5:
        vehicle.mode = VehicleMode("AUTO")

    print "Change mode: " + str(mode)

```

En esta parte se guardan los diferentes métodos anteriores en variables los cuales posteriormente se van a mandar a llamar para imprimir la información que arroje el dron, estos datos son los del estado del dron.

```

# MAIN LOOP
def main():
    try:
        while True:
            ## Vehicle data ##
            #estado del VANT#
            mode = str(vehicle.mode.name)
            batt = str(vehicle.battery.level)
            arm = str(vehicle.armed)

```

Estos datos son los de la posición en la que se encuentra el dron.

```

#posicion global#
    lat = vehicle.location.global_frame.lat
    lon = vehicle.location.global_frame.lon
    alt = vehicle.location.global_frame.alt

```

En este apartado del código se muestran más variables referentes al dron

```

#ESTADO DEL ENTORNO DEL vant#
    vel = str(vehicle.velocity)#velocidad del VANT con
respecto a sus tres ejes
    ground_speed = str(vehicle.groundspeed)#velocidad a la
que se mueve el Vant con respecto a tierra

```

```

        air_speed = str(vehicle.airspeed)#(velocidad del
aire*estimacion)

```

Esta parte del código es donde se manda a imprimir las variables que necesitamos.

```

print "Mode: %s" % mode
    print "Latitud: " + str(lat)
    print "Longitud: " + str(lon)
    print "Altitude: " + str(alt)
    print "Battery: " + str(batt)
    print "velocidad del vehiculo: " + str(vel)
    print "velocidad del aire: " + str(air_speed)

```

```

data = "[]"

```

Por ultimo esta parte es la principal ya que es el orden en cómo se mandan a llamar e imprimir las variables y en qué tipo de variable se están representando para posteriormente en el recibimiento de los datos coloquen el mismo orden.

```

## Sending data ##

```

```

    data = '[' + str(mode) + ',' + str(batt) + ',' +
str(arm) + ',' + str(lat) + ',' + str(lon) + ',' + str(alt) +
',' + str(vel) + ',' + str(ground_speed) + ',' +
str(air_speed) + ']'

```

```

ser.write(str(data) + '\n')
time.sleep(0.1)

```

```

callback = ''

```

```

if ser.inWaiting() > 1:
    print "Reading data..."
    callback = ser.readline()
    command = callback.split(':')
    print "%s:%s" % (str(command[0]),
str(command[1]))
    if str(command[0]) == 'mode':
        set_mode(int(command[1]))

    if str(command[0]) == 'arm':
        set_arm(int(command[1]))

    if str(command[0]) == 'takeoff':

```

```

        takeoff(int(command[1]))

    ser.flush()

except KeyboardInterrupt:
    GPIO.cleanup()
    ser.close()

if __name__ == '__main__':
    print "=="Inicio=="
    main()

```

Y en la siguiente imagen se muestra el programa funcionando correctamente.

```

Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Connecting to vehicle on: 127.0.0.1:14550
>>> APH:Copter V3.5.2 (4322fda)
>>> PX4: 1d6bf64c NuttX: 1a99ba58
>>> Frame: QWAD
>>> PX4v3 00220030 33355107 33373538
>>> APH:Copter V3.5.2 (4322fda)
>>> PX4: 1d6bf64c NuttX: 1a99ba58
>>> Frame: QWAD
>>> PX4v3 00220030 33355107 33373538
>>> Link timeout, no heartbeat in last 5 seconds
==Inicio==
Mode: ALT_HOLD
Latitude: 0.0
Longitude: 0.0
Altitude: 11.95
Battery: None
velocidad del vehiculo: [0.0, 0.0, 0.01]
velocidad del aire: 0.0
Reading data...

```

Figura. 6 Programa de Python con Dronekit Funcionando

Posteriormente en la siguiente imagen pueden observar el recibimiento de los datos que se mandaron de la raspberry a través de las antenas de telemetría en donde se realiza él envío de los datos mediante radio frecuencia y la utilización de las librerías del dronekit.

```

C:\WINDOWS\system32\cmd.exe - index.js
data: [ALT_HOLD, None, False, 25.664972, -100.2444624, 0.79, [0.03, 0.12, -0.04], , 0.129808500409, 0.552000045776]
12
Data inserted!
data: [ALT_HOLD, None, False, 25.6649746, -100.2444983, 490.14, [-0.02, 0.35, 0.07], , 0.34900790453, 0.355000019073]
12
Data inserted!
data: [ALT_HOLD, None, False, 25.6649743, -100.2444977, 490.13, [-0.03, 0.36, 0.06], , 0.364088833332, 0.151000007987]
12
Data inserted!
data: [ALT_HOLD, None, False, 25.6649744, -100.2444975, 490.12, [-0.02, 0.35, 0.06], , 0.353564918041, 0.122000008821]
12
Data inserted!
data: [ALT_HOLD, None, False, 25.6649747, -100.2444975, 490.08, [0.0, 0.33, 0.05], , 0.348777486563, 0.109000004828]
12
Data inserted!
data: [ALT_HOLD, None, False, 25.6649747, -100.2444975, 490.07, [0.0, 0.33, 0.05], , 0.33465886116, 0.295000016689]
12

```

Figura. 7 Resultados adquiridos del dron y la API

3.2 Material, equipo y software utilizado

- Dron

Un dron es esencialmente un robot que vuela. Puede ser controlado vía remota o puede volar de manera autónoma a través de unos planes de vuelo controlados por el software que está integrado al sistema y el cual trabaja en conjunto con el GPS.



Figura. 8 Dron de cuatro hélices. Cuadricoptero

- Computadora de vuelo

Es el sistema o computadora que dirige los movimientos del dron a través de instrumentos de medición como el giroscopio y el acelerómetro, y en otros casos, también utilizando GPS o barómetro.

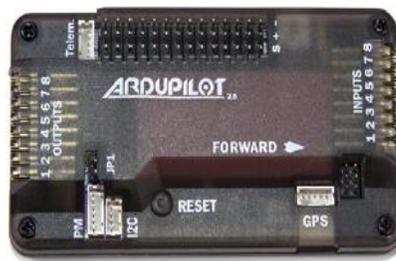


Figura. 9 Computadora de vuelo Pixhawk

- GPS

El GPS es el sistema de posicionamiento global, el cual nos puede dar la ubicación del dron ya que está basado por satélites y cuenta con 24 satélites puestos en órbita.



Figura. 10 GPS

- Antena

Permite la comunicación entre el dron y la estación de tierra y/o la radio.



Figura. 11 Antenas de Telemetría

- Batería

Las baterías utilizadas para dar poder al dron. Es un tipo de batería recargable de tecnología de ion de litio en un empaque suave, haciéndolas más ligeras pero también quitándole rigidez a la cobertura.



Figura. 12 Batería de Litio

- Motores

Los motores son parte del sistema de propulsión. Los motores que se utilicen tienen un impacto enorme en la capacidad de carga máxima que el dron pueda soportar, así como el tiempo de vuelo.



Figura. 13 Motores para los drones

- Radio

Utilizado para controlar el dron en modo manual o de respaldo cuando se vuela en modo automático.



Figura. 14 Radiocontrol para operar el dron.

- Estación de tierra

La estación de tierra es el equipo que un segundo operador debe mantener en constante vigilancia para poder vigilar el estado del dron, el estado de los aparatos que el dron este cargando (cámaras, sensores) y la ruta de vuelo que se haya trazado.



Figura. 15 Una laptop que hace función de estación de tierra

- Raspberri pi 3

Placa de computadora simple integra conectividad BlueTooth y Wi-Fi. Se le pueden instalar diversos sistemas operativos y aplicaciones hechos especialmente para este tipo de placas.



Figura. 16 Placa raspberri pi 3 versión B

Los cables se utilizan para interconectar a la computadora de vuelo lo necesario para volar (gps, motores, buzzer), así como también conectar el dron a la estación de tierra para verificar que este en un estado óptimo para el vuelo.

- Raspbian Jessi OS

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Stretch (Debian 9.4) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática.

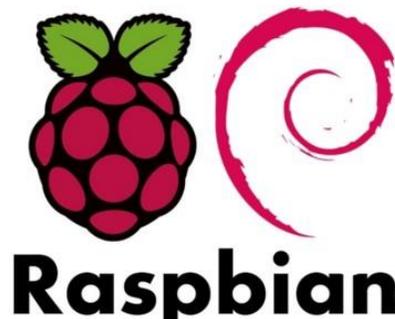


Figura. 17 Símbolo de distribución Rabian Jessie de Linux

- Dronekit

Utilerías que ofrecen un SDK y API web para el fácil desarrollo de aplicaciones para drones.

- Nodejs

Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node está diseñado para construir aplicaciones en red escalables.

- Postgresql

Es un manejador de base de datos libre, basado en el modelo de SQL y permite la recopilación y almacenamiento de los volúmenes de datos entrantes.



Figura. 18 Manejador de Base de Datos

- SublimeText

Es un editor de texto que permite trabajar con diversos lenguajes de programación, haciendo que la codificación se haga de una manera más ágil y dinámica.

- Mission Planner

Mission Planner tiene como objetivo ofrecer a sistemas abiertos de controladores de vuelo la posibilidad de trazar rutas para el vuelo de sus vehículos, así como también dar la posibilidad de calibrar los aparatos para comprobar su buen funcionamiento antes, durante y al finalizar la operación.



Figura. 19 Mission Planner

3.3 Consideraciones para el diseño

El diseño de la API fue en base a los datos que se quieren recopilar, y enviar o transmitirse a la estación de tierra. Ya que la API es en base al lenguaje de programación de Python entonces se tuvo que realizar la estructura con la sintaxis de este lenguaje y estructurar muy bien las cadenas de código para que las variables que se obtengan lleguen de manera adecuada para posteriormente poder manipular esos datos en una base de datos.

De esta manera cuando el drone este en vuelo la API funciona internamente en el drone así que lo único que se debe de hacer durante el vuelo es conectarlo hacia la estación de tierra y así se pueden visualizar la llegada de los datos que se obtienen de los sensores del drone.

```

C:\WINDOWS\system32\cmd.exe - index.js
data: [ALT_HOLD,None,False,25.664972,-100.2444624,0.79,[0.03, 0.12, -0.04],,0.129808500409,0.552000045776]
12
Data inserted!
data: [ALT_HOLD,None,False,25.6649746,-100.2444983,490.14,[-0.02, 0.35, 0.07],,0.34900790453,0.355000019073]
12
Data inserted!
data: [ALT_HOLD,None,False,25.6649743,-100.2444977,490.13,[-0.03, 0.36, 0.06],,0.364088833332,0.151000007987]
12
Data inserted!
data: [ALT_HOLD,None,False,25.6649744,-100.2444975,490.12,[-0.02, 0.35, 0.06],,0.353564918041,0.122000008821]
12
Data inserted!
data: [ALT_HOLD,None,False,25.6649747,-100.2444975,490.08,[0.0, 0.33, 0.05],,0.340777486563,0.109000004828]
12
Data inserted!
data: [ALT_HOLD,None,False,25.6649747,-100.2444975,490.07,[0.0, 0.33, 0.05],,0.33465886116,0.295000016689]
12

```

Figura. 20 Resultados Obtenidos

CAPÍTULO IV RESULTADOS

4.1 Experimentación del prototipo en toma de muestras

Dentro del laboratorio se realizó la experimentación correspondiente. Mediante las pruebas realizadas se comprobó que la conexión del dron estuviera bien hecha, es decir que se comprobó que los sensores, antenas y controles del dron respondieran bien y estuvieran calibrados. Se comprobó también la conexión de la computadora de vuelo del dron hacia la placa raspberri pi para que de esta obtuviera los datos de los sensores y los enviara por medio de la aplicación que se realizó con el DRONE KIT a la estación de tierra donde la información fue almacenada y por último comprobar que la conexión al estar corriendo el programa estuviera funcionando correctamente.

Cada prueba realizada fue en un estado controlado, lo que quiere decir que las pruebas de vuelo fueron mínimas y los resultados obtenidos fueron obtenidos más por medio de la conexión de los dispositivos dentro del laboratorio, siempre

comprobando que todo funcionara correctamente para que los resultados no variaran en cada prueba.

4.2 Resultado de la experimentación

El resultado final que se llegó a lograr en este proyecto fue el de realizar una aplicación que logró interconectar varias tarjetas mediante el DRONE KIT para recibir y enviar datos de un dron y en el que se representara la información de contaminantes en el aire enviada por los sensores en el dron y almacenada en una base de datos.

4.3 Comparación entre las diferentes APIS para VANTS

Como se vio anteriormente existen diferentes APIS para los VANTS así que en este proyecto se realizó la comparación del Dronekit con otras APIS y se demostrara que la mejor opción para proyectos de este tipo es mejor utilizar la herramienta Dronekit.

Tabla 1. Diferencia en los criterios de transferencia de datos entre el Dronekit y el DJI

Criterios	Dronekit	DJI
VELOCIDAD	El dronekit tiene una velocidad de tiempo real al transferir los datos que se quieren obtener de los sensores de un dron	El DJI también maneja su velocidad en tiempo real pero no de los sensores que se conectan al dron.
ALMACENAMIENTO	Tiene un amplio manejo de espacio de memoria la cual facilita la transferencia y manipulación de los datos obtenidos	El almacenamiento del DJI debe de tener una computadora la cual la capacidad de su memoria sea un poco amplia o de mayor capacidad.
COMPATIBILIDAD	El Dronekit tiene una mayor cantidad de compatibilidad con la mayoría de los drones que se fabrican hoy en día ya que se basa en el lenguaje de	El DJI es solo compatible con drones que son fabricados en su propia empresa y algunos de la empresa 3DR, también utiliza el lenguaje de programación Python

	programación Python y la mayoría de los drones también están programados en este lenguaje.	pero no manda datos de sensores.
CONECTIVIDAD	La conectividad del Dronekit es una de las mejores ya que se puede conectar a distintos drones a la vez y realiza una conexión excelente con las tarjetas inteligentes.	La conectividad del DJI también es buena con los diferentes drones en los cuales puede manejar, sin embargo no es posible que se realicen las conexiones con otras tarjetas inteligentes.
TIPOS DE DATOS	Los datos que puede mandar y recibir el Dronekit son datos enteros, booleanos, flotantes, cadenas de caracteres (string), entre otros.	El DJI solo puede realizar el envío y recibimiento de imágenes ya sea en 2D O 3D

Tabla 2. Diferencia en los criterios de transferencia de datos entre el Dronekit y el 3DR

Criterios	Dronekit	3DR
VELOCIDAD	El dronekit tiene una velocidad de tiempo real al transferir los datos que se quieren obtener de los sensores de un drone	El 3DR también maneja su velocidad en tiempo real al transferir los datos pero son datos que solo el 3DR maneja.
ALMACENAMIENTO	Tiene un amplio manejo de espacio de memoria la cual facilita la transferencia y manipulación de los datos obtenidos	El almacenamiento del 3DR es buena como la del Dronekit también facilita la transferencia de los datos.
COMPATIBILIDAD	El Dronekit tiene una mayor cantidad de compatibilidad con la mayoría de los drones que se fabrican hoy en	El 3DR también es compatible con una cantidad favorable de drones y en los cuales su mayor compatibilidad es

	día ya que se basa en el lenguaje de programación Python y la mayoría de los drones también están programados en este lenguaje.	con los drones con cámaras.
CONECTIVIDAD	La conectividad del Dronekit es una de las mejores ya que se puede conectar a distintos drones a la vez y realiza una conexión excelente con las tarjetas inteligentes.	La conectividad del 3DR es muy buena ya que se puede conectar con varios drones a su vez pero no puede enviar datos simultáneamente.
TIPOS DE DATOS	Los datos que puede mandar y recibir el Dronekit son datos enteros, booleanos, flotantes, cadenas de caracteres (string), entre otros.	El 3DR fue diseñado solo para el envío y recibimiento de imágenes ya sea en 2D O 3D o un mapeado completo de algún terreno.

CAPÍTULO IV CONCLUSIONES

Evaluar 2 Interfaces para la programación de aplicaciones (API) en el control de VANTS, tomando como referencia el Dronekit Framework

Se logró el objetivo el cual en las pruebas que se hicieron obtuvimos los resultados esperados en cuanto a envío y recibimiento de datos en una velocidad esperada y en la que los datos de los contaminantes fueron a un rango determinado por el sensor.

La hipótesis se cumplió debido a que se hizo la comparación entre el Dronekit y 2 diferentes APIS, las cuales fueron el DJI y el 3DR y se obtuvieron los resultados que se esperaban en cuanto a la velocidad en la cual se enviaron los datos que se obtenían de los sensores MQ para posteriormente reflejarlos y realizar graficas de los contaminantes y sus niveles que hay en ese lugar y también uno de los resultados fue el de un mejor rendimiento y una mejor transmisión de datos de un dron a una estación de tierra.

Este proyecto en general fue una prueba que se logró cumplir y se logró adquirir conocimientos nuevos acerca de los contaminantes e incluso acerca de diferentes tipos de monitoreo que se pueden realizar y con ayuda de eso se pueden generar muchos más proyectos para el futuro de este tipo o incluso en otras áreas que se requiera investigar.

5.1 Recomendaciones

Si se quisiera utilizar otro tipo de API como por ejemplo el 3DR es favorable que se piense bien en el proyecto que se quiere realizar ya que diferentes Frameworks no son especialmente para proyectos o investigaciones científicos o ambientales. Recomiendo que tomen en cuenta la velocidad de transmisión, la capacidad de interactuar con otros lenguajes de programación y también que sea compatible con diferentes tarjetas inteligentes y el más viable es el DRONEKIT.

5.2 Trabajos futuros

En el futuro se pueden realizar APIS para celulares en los cuales se puedan configurar e incluso manejar más rápida la información que se obtenga de los drones que están en vuelo y así poder representar esos datos en una serie de graficas desde su celular.

Se recomienda adquirir un framework de licencia junto con el VANT, debido a que las librerías de configuración ya están desarrolladas y solo se integran a la aplicación de manera flexible, sin necesidad de meterse a la parte del hardware del VANT, debido a que se puede llegar a dañar con las modificaciones en programación, una variación de corriente dada por un comando puede quemar el VANT.

REFERENCIAS

- [1] Luigi Forestier. (2014). La importancia de las APIs en el nuevo mundo. 31/06/2014, de Mundo contact Sitio web: <https://mundocontact.com/la-importancia-de-las-apis-en-el-nuevo-mundo-hiperconectado/>
- [2] Gustavo Arencibia Carballo . (2016). ¿Es posible el uso de drones en la investigación científica y el monitoreo ambiental? . 2018, de REDVET® Revista Electrónica de Veterinaria Sitio web: <http://www.veterinaria.org/revistas/redvet/n090916/091601.pdf>
- [3] DIGESA. (2005). PROTOCOLO DE MONITOREO DE LA CALIDAD DEL AIRE Y GESTIÓN DE LOS DATOS. 2018, de DIRECCION GENERAL DE SALUD AMBIENTAL Sitio web: http://www.digesa.minsa.gob.pe/norma_consulta/Protocolo-de-Calidad-del-Aire.pdf
- [4] Liam Kirk. (2014). Sitio Oficial 3DR. 2018, de 3DR Sitio web: <https://3dr.com/>
- [5] Empresa DJI. (2015). DJI. 2018, de DJI Corporation Sitio web: <https://www.dji.com/es>
- [6] Orix. (2016). ¿Qué es un framework y para qué se utiliza?. 2018, de Orix Systems Sitio web: <https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza>
- [7] EcuRed. (2015). Software propietario. 2018, de EcuRed Sitio web: https://www.ecured.cu/Software_propietario
- [8] Wikipedia. (2015). Vehículo aéreo no tripulado. 7/11/2018, de Fundación Wikimedia, Inc. Sitio web: https://es.wikipedia.org/wiki/Veh%C3%ADculo_a%C3%A9reo_no_tripulado
- [9] Robert Lowe. (2014). ¿Qué es una API y para qué sirve?. 2018, de Ticbeat Sitio web: <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>
- [10] EcuRed. (2015). Código Cerrado. 2018, de EcuRed Sitio web: https://www.ecured.cu/Código_abierto
- [11] Dronekit 3DR. (2015). Acerca de DroneKit. 2016, de Robótica 3D Sitio web: <http://python.dronekit.io/about/overview.html>

[12] Radio Comunicaciones . (2016). TELEMETRÍA – APLICACIONES DE MEDIDA A DISTANCIA. 2017, de Radio Y Engineering Company SL Sitio web: <http://www.radiocomunicaciones.net/radio/telemetria/>