



BEcopter

Especificación de Requerimientos de Software

Bastida Eric

mayo, 2019

Fecha	Versión	Descripción	Autor
24/09/18	1.0	Especificación de Requerimientos	Bastida Eric
25/09/18	1.1	Se realizan cambios, comentarios y sugerencias	Marina Murillo
25/09/18	1.2	Se resuelven las sugerencias	Bastida Eric
22/02/18	1.3	Se corrigen y mejora los requerimientos	Lucas Genzeliz
07/05/19	1.3	Sugerencias y correcciones.	Guido Sanchez
15/05/19	1.3	Se resuelven las peticiones.	Eric Bastida

Índice

1. Introducción	2
1.1. Propósito	2
1.2. Alcance	2
1.3. Definiciones, Acrónimos y Abreviaturas	3
1.4. Referencias	5
1.5. Visión general del documento	5
2. Descripción general	6
2.1. Perspectiva del producto	6
2.2. Funciones del producto	6
2.3. Características de los usuarios	7
2.4. Restricciones	7
2.5. Suposiciones y Dependencias	8
2.6. Requisitos Futuros	8
3. Requisitos Específicos	9
3.1. Interfaces Externas	9
3.1.1. Interfaces de usuario:	9
3.1.2. Interfaces de hardware:	11
3.1.3. Interfaces de software:	11
3.1.4. Interfaces de comunicación:	11
3.2. Requisitos funcionales	12
3.3. Requisitos no funcionales	14

1. Introducción

En este documento destinado a la Especificación de Requerimientos de Software (ERS) se describirán los aspectos que están involucrados en el desarrollo del proyecto BEcopter. Dicho proyecto consiste en una plataforma para el guiado y navegación de un vehículo aéreo no tripulado (VANT), es decir, este software permitirá al usuario realizar maniobras aéreas de forma manual con un joystick o bien mediante un conjunto de instrucciones cargadas en modo *batch* dentro del vehículo que le permitirán desplazarse en un determinado ambiente. Además esta plataforma le brindará al usuario toda la información necesaria del estado del vehículo y sensores.

Cabe mencionar que el formato en que se desarrolló este documento fue basado en el estándar 830-1998, proporcionado por la IEEE.

1.1. Propósito

El presente documento tiene como propósito definir los diferentes tipos de requerimientos que han sido obtenidos de los usuarios que utilizarán de manera directa o indirecta el sistema (*stakeholders*) mediante técnicas de ingeniería en requerimientos [1], incluyendo además, los juicios que se han considerado para la planificación y diseño del software. Toda esta información es de utilidad para el desarrollador del presente proyecto para avanzar en el desarrollo del mismo y, sobre todo, informar de manera clara y precisa a los agentes evaluadores del proyecto: directores e integrantes de la cátedra *Proyecto Final de Carrera* (PFC).

1.2. Alcance

El nombre BEcopter es un acrónimo originado por las iniciales del ejecutor del proyecto y el tipo de vehículo que se utiliza para realizar las pruebas. En nuestro caso hacemos uso de un UAV tipo cuadricóptero (o quadcopter en su traducción al inglés), por lo tanto, para el desarrollo del nombre se descarta el prefijo “cuadri” ya que este determina la cantidad de hélices que contiene el vehículo y además esta descripción limita al software al uso de UAVs de cuatro hélices; y como se explicará en la sección 2.6 no únicamente se pretende utilizar cuadricópteros. El sistema proporcionará las acciones necesarias para que el vehículo pueda realizar maniobras en el aire mediante el control de un joystick que será comandado por el usuario o a través de un conjunto de instrucciones que serán cargadas en la plataforma y luego enviadas al vehículo para su ejecución. Estas acciones serán:

- Ascenso,
- Descenso,
- *Hovering*.

Este software proporcionará a la comunidad una plataforma con la cual el usuario final podrá implementar sus propias ideas sin entrar en detalles internos del hardware pertinente al vehículo.

1.3. Definiciones, Acrónimos y Abreviaturas

- **Stackholders:** Este término se utiliza comúnmente para referirse a cualquier persona o grupo que se verá afectado por el sistema de manera directa o indirectamente.
- **Plataforma:** Para este tipo de vehículos, es un sistema que sirve como base para hacer funcionar los módulos de hardware y/o de software en conjunto. En lo que concierne al hardware hacemos referencia a los componentes electrónicos del VANT, como pueden ser:
 - Motores,
 - Sensores como por ejemplo, acelerómetros, giroscopios, magnetómetros, etc.,
 - Periféricos de entrada y/o salida tales como tarjetas SD, puertos USB, entre otros.

Para poder obtener información de cada componente y gestionar sus interacciones es necesario que un software administre en un segundo plano estas tareas, y que de forma sencilla proporcione al usuario funcionalidades para manipular estos componentes electrónicos, con el propósito de realizar acciones de vuelo sobre el vehículo.

- **VANT o UAV:** Un vehículo aéreo no tripulado (VANT), conocido también por su nombre en inglés como *Unmanned Aerial Vehicle* (UAV) o drone, es una aeronave que vuela sin tripulación. El mismo es capaz de mantener un nivel de vuelo controlado, sostenido y propulsado por uno o varios motores de explosión, electrónicos, o de reacción.
- **Cuadricóptero:** Un cuadricóptero, cuadirrotor o quadrotor es un helicóptero con cuatro rotores para su sostén y su propulsión. Los cuatro rotores están generalmente colocados en las extremidades de una cruz. A fin de evitar que el aparato se tumbe respecto a su eje de orientación es necesario que dos hélices giren en un sentido y las otras dos en el otro sentido.
- **Roll, Pitch, Yaw:** [2] Los ángulos de navegación (ángulos de Euler) se utilizan para describir la orientación de un objeto en tres dimensiones como se puede observar en la Figura 1.
- **Firmware:** Es un programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Está fuertemente integrado con la electrónica del dispositivo, es el software que tiene directa interacción con el hardware, siendo así el encargado de controlarlo para ejecutar correctamente las instrucciones externas. Por lo tanto un firmware es un software que maneja físicamente al hardware.
- **Modo batch:** (o procesamiento por lotes) es la ejecución de un programa sin el control o supervisión directa del usuario (que se denomina procesamiento interactivo). Este tipo de programas se caracterizan porque su ejecución no precisa ningún tipo de interacción con el usuario.

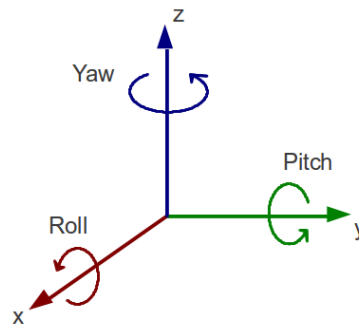


Figura 1: Sistema de referencia.

- **Acelerómetro:** Instrumento que sirve para medir la aceleración de movimiento de un vehículo.
- **Magnetómetro:** Instrumento para medir la fuerza y la dirección de un campo magnético.
- **ESC:** Un control electrónico de velocidad o ESC (Electronic Speed Controller) es un circuito electrónico con el propósito de variar la velocidad de un motor eléctrico, su dirección y posiblemente también actuar como un freno dinámico.
- **GPS:** Sistema estadounidense de navegación y localización mediante satélites.
- **Giroscopio:** Dispositivo electrónico que sirve para medir la orientación en el espacio de algún aparato o vehículo.
- **Brújula:** Instrumento que proporciona una dirección de referencia (respecto al norte) en el plano horizontal y permite la medición de ángulos horizontales con respecto a esta dirección.
- **Barómetro:** Instrumento para medir la presión atmosférica.
- **Horizonte artificial:** El horizonte artificial muestra la orientación longitudinal de la aeronave (la relación del eje longitudinal del vehículo con respecto al plano del suelo), es decir: si está girado, inclinado, con el frente levantado, bajado o todo a la vez. Sirve de gran ayuda en condiciones en que la visibilidad es poca o nula. Su principio mecánico de funcionamiento es giroscópico.

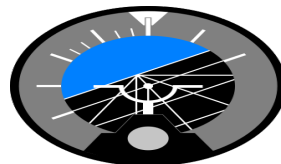


Figura 2: Horizonte artificial indicando un giro a la derecha en descenso.

- **Instrumentos de vuelo:** Conjunto de dispositivos tales como giroscopio, acelerómetro, magnetómetro, ESC, GPS, barómetro, indicador de rumbos, horizonte artificial, brújula que equipan una aeronave y que permiten al piloto llevar a cabo una operación de vuelo en condiciones seguras. Dependiendo de su tamaño o grado de sofisticación, una aeronave puede contar con un número variable de instrumentos [3].
- **Aeromodelismo:** es una afición y un deporte derivado de la técnica de construcción y vuelo de aeroplanos de pequeño, mediano y gran tamaño, denominados aeromodelos, que han sido preparados para volar sin tripulación.

1.4. Referencias

- [1] S. Ian, *Ingeniería del software*. Pearson Educación, 2005.
- [2] J. G. de la Cuesta, *Aviation Terminology = Terminología Aeronautica*. Diaz de los Santos, 2003.
- [3] Wikipedia, “Instrumentos de vuelo - wikipedia, la enciclopedia libre,” 2017. [Internet; descargado 27-agosto-2017].

1.5. Visión general del documento

De manera informativa y para guiar al lector en la lectura del presente documento, en esta sección se presenta un breve resumen sobre la estructura del mismo. En forma general, este documento se compone de cuatro partes importantes:

1. **Introducción:** En esta sección se proporcionará un panorama general sobre el ERS. Este consta de varias subsecciones: 1.1 Propósito, 1.2 Alcance del sistema, 1.3 Definiciones y 1.4 Referencias.
2. **Descripción General:** En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Además, consta de las siguientes subsecciones: 2.1 Perspectiva del producto, 2.2 funciones del producto, 2.3 características de los usuarios, 2.4 restricciones, factores que se asumen y 2.6 futuros requisitos.
3. **Requisitos Específicos** Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los programadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.

2. Descripción general

2.1. Perspectiva del producto

El producto BEcopter tiene como perspectiva ser una plataforma para el guiado y navegación de VANTs, esto incluye, aeroplanos, helicópteros o multirrotores como pueden ser tricópteros, cuadricópteros, hexacópteros, etc. El producto depende de un firmware que es distribuido bajo la licencia GNU por la empresa Ardupilot ¹, este es el encargado de gestionar todo el hardware perteneciente al vehículo mediante un software a bajo nivel, y que a su vez, brinda funcionalidades básicas de control e información sobre el estado del vehículo.

2.2. Funciones del producto

En esta subsección del documento se explicarán, a grandes rasgos, las funcionalidades que proporcionará la plataforma BEcopter:

- **Control del vehículo:** Mediante esta funcionalidad se pretende que el sistema pueda proporcionar de manera clara y sencilla al usuario acciones que puedan ser ejecutadas por el vehículo, como pueden ser: despegar, moverse y aterrizar. En lo que respecta a las funciones se podrán hacer de dos maneras distintas.
 1. **Modo manual:** El usuario podrá controlar el vehículo mediante un joystick que enviará las señales correspondientes al sistema y según una configuración predeterminada serán interpretados por el vehículo para realizar las acciones que han sido fijadas.
 2. **Modo batch:** En este modo el sistema brindará distintos tipos de acciones (denominadas misiones) que el usuario podrá seleccionar de manera secuencial, es decir, siguiendo un orden correspondiente como por ejemplo: primero despegar luego elevarse e ir a una ubicación específica, etc. Además, se le podrán asignar atributos a cada acción como pueden ser: velocidad, posición, altura, etc. Una vez que el usuario selecciona este conjunto o lote de acciones el sistema las enviará al vehículo para que sean ejecutadas.
- **Comunicación inalámbrica:** Para que el sistema pueda comunicarse con el vehículo se debe seleccionar algún tipo de medio por el cual sea posible transmitir los datos; ya que el vehículo se desplazará en grandes espacios despejados, es recomendable que los datos se envíen de manera inalámbrica. Por tanto, el sistema tendrá la responsabilidad de proporcionar las funcionalidades correspondientes a la gestión de comunicación inalámbrica entre la plataforma y el vehículo.
- **Visualización de datos:** Poder representar datos como ubicación, velocidad, altura, estado de motores, entre otros aspectos, es de suma importancia ya que brinda información relevante al usuario para poder realizar, en tiempo real, maniobras de salvaguarda sobre el vehículo o capturar información para realizar algún tipo de análisis para su posterior estudio. Es por eso que el sistema representará de manera numérica todos los datos

¹ArduPilot - www.ardupilot.org

que sean necesarios para la navegación, control y sensores del vehículo. Además, dará la opción de poder visualizar estos datos, que van cambiando continuamente a través del tiempo en forma gráfica. Esto es un aspecto importante ya que permite al usuario ver cómo es el comportamiento temporal de variables críticas del vehículo y/o datos a analizar.

2.3. Características de los usuarios

En lo que respecta al tipo de usuario final que utilizará el sistema, son dos:

1. Usuario normal:

Tipo de usuario	Normal.
Formación	Conocimientos básicos en informática y aeromodelismo.
Habilidades	Navegación en vehículos aéreos.
Actividades	Uso y configuración de la plataforma.

2. Usuario desarrollador:

Tipo de usuario	Desarrollador
Formación	Ciencias de la computación
Habilidades	Programación en Python y navegación en vehículos aéreos.
Actividades	Uso y configuración de la plataforma. Implementación de rutinas.

2.4. Restricciones

Esta subsección corresponde a aquellas limitaciones que se imponen al desarrollador del producto.

- Sistema Operativo: Linux.
- Tipo de software: Desarrollado para ordenadores de tipo escritorio.
- Funciones de vuelo: Momentaneamente se desarrollarán 3 funciones
 1. Ascender,
 2. Descender,
 3. *Hovering*.
- Interfaz gráfica de usuario: Desarrollada mediante la librería PyQt².
- Limitaciones de hardware: Por pertenecer a un software de tipo de control de navegación de vehículos aéreos es necesario tener los siguientes elementos:
 1. Raspberry Pi 3³.
 2. Placa Navio2⁴.

²PyQt - www.riverbankcomputing.com/software/pyqt/intro

³Raspberry - www.raspberrypi.org

⁴Navio2 - www.emlid.com/navio/

3. Joystick.
4. Módulos de comunicación inalámbrica Xbee⁵.
5. Estructura de un VANT.

- Lenguaje de programación: Python 2.7.
- Consideraciones acerca de la seguridad: El producto no implementa un modo de seguridad o *safety mode* por tanto es responsabilidad del usuario cualquier tipo de accidente que se produzca utilizando el producto.
- Desarrollado bajo la Licencia Pública General de GNU.

2.5. Suposiciones y Dependencias

Con lo establecido hasta el momento no se suponen factores que puedan intervenir en modificaciones a un nivel profundo sobre el desarrollo del producto, en caso fortuito sobre cualquier tipo de inconveniente, se incorporarán de manera adicional al documento para solventar el problema sin afectar en su totalidad a los demás requisitos. El producto tiene una gran dependencia tecnológica con el hardware ya que de manera conjunta establecen un canal de comunicación y además, es el paso principal que condiciona el inicio de las funcionalidades del sistema, por esta razón es de suma importancia tener de antemano el vehículo armado y correctamente configurado.

Por distintos factores que intervienen en la gestación del proyecto, el desarrollo de las funcionalidades tendrá foco en la navegación de un UAV tipo cuadricóptero, descartando en primera medida opciones de vuelo orientadas a aeroplanos.

2.6. Requisitos Futuros

Se prevén los siguientes requisitos que pueden ser implementados en el producto.

- **Más acciones de vuelo:** se pretende desarrollar funcionalidades de navegación que equipen al vehículo con menos restricciones al momento de desplazarse en el aire, ya sean:
 - *Desplazamiento en forma circular con respecto a un punto (objetivo) como centro:* esta funcionalidad por lo general es útil al momento de hacer tomas cinematográficas o de algún tipo de video casero, proporcionando así una buena perspectiva del objetivo a analizar.
 - *Follow me:* en esta opción se designara un objetivo (por ejemplo una persona) y una vez ingresado al software el vehículo procederá a seguir este objetivo según parámetros preestablecidos con anterioridad como: altura, distancia al objeto, ubicación, etc.
- **Integración a más tipos de vehículos:** extensión a más tipos de vehículos que puedan ser comandados y guiados por la plataforma, como por ejemplo, aeroplanos, helicópteros, hexacópteros, tricópteros, etc.

⁵XBEE - www.xbee.cl/que-es-xbee/

- **Adquisición de imágenes:** En el caso que vehículo tenga incorporada una cámara la plataforma brindará opciones para poder visualizar las imágenes que se están capturando en tiempo real.

3. Requisitos Específicos

En esta sección se realiza una descripción detallada de cada uno de los requerimientos. Además de describirlos, en esta sección también se clasifican los mismos de acuerdo al módulo al cual pertenecen para facilitar su trazabilidad a través del desarrollo de la aplicación.

3.1. Interfaces Externas

3.1.1. Interfaces de usuario:

Número de requisito	R001
Nombre del requisito	Visualización de datos de vuelo y estado del vehículo
Tipo	Requisito Restricción
Descripción	El sistema debe proporcionar la representación en tiempo real y numérica de los datos de vuelo, esto incluye, información proveniente de todos los sensores del vehículo y niveles de intensidad generados por los periféricos de entrada/salida, tales como joystick y señal de cobertura.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R002
Nombre del requisito	Representación gráfica en tiempo real y evolutiva de la información capturada.
Tipo	Requisito Restricción
Descripción	El sistema debe dar la opción de representar gráficamente y en forma evolutiva la información capturada por algún determinado periférico de entrada. Esto se debe representar mediante una ventana y utilizando la librería Vispy ⁶ .
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R003
Nombre del requisito	Representación gráfica de las variables características para el guiado, navegación y control del vehículo.
Tipo	Requisito Restricción
Descripción	El sistema deberá ilustrar de manera gráfica y representativa los datos provenientes de los instrumentos de vuelo. Como el horizonte de artificial, indicador de rumbos, etc.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R004
Nombre del requisito	Manual de usuario.
Tipo	Requisito Restricción
Descripción	Se deberá realizar un manual que especifique y ejemplifique las principales funcionalidades del sistema. Este manual será un medio por el cual los usuarios podrán aprender a utilizar el sistema, por lo que deberá ser realizado con este propósito.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R005
Nombre del requisito	Inserción de comandos de vuelo
Tipo	Requisito Restricción
Descripción	El sistema ofrecerá una lista de comandos que el usuario podrá elegir de manera secuencial y luego serán enviadas al vehículo para que sean procesadas. Estos comandos serán: ascender, descender y <i>hovering</i> . Además de incluirse sus respectivas características como velocidad, posición y tiempo de ejecución.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R006
Nombre del requisito	Modos de vuelo
Tipo	Requisito Restricción
Descripción	A modo de seguridad el sistema debe permitir el guiado de vehículo en modo manual
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R007
Nombre del requisito	Representación gráfica del vehículo
Tipo	Requisito Restricción
Descripción	Como medida de referencia el sistema ilustrará un UAV representativo mediante un gráfico que simule las acciones que está realizando el vehículo en tiempo real.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R008
Nombre del requisito	Calibración
Tipo	Requisito Restricción
Descripción	El sistema debe informar al usuario si los sensores principales del vehículo se encuentran correctamente calibrados, en caso contrario se dará la opción de calibrarlos.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R009
Nombre del requisito	Posicionamiento geográfico.
Tipo	Requisito Restricción
Descripción	Mediante un mapa el sistema debe ubicar geográficamente utilizando el GNNS del UAV y representarlo con algún tipo de gráfico en dicha ubicación.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R010
Nombre del requisito	Implementación de un Log perteneciente a la plataforma
Tipo	Requisito Restricción
Descripción	Brindar un registro de todo tipo de inconvenientes, errores o precauciones que se presentan sobre el uso de las funcionalidades de la plataforma.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R011
Nombre del requisito	Implementación de un Log perteneciente a los sensores.
Tipo	Requisito Restricción
Descripción	Brindar un log de todo tipo de inconvenientes, errores o precauciones que se presentan sobre el uso de los sensores instalados en el vehículo.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

3.1.2. Interfaces de hardware:

Número de requisito	R012
Nombre del requisito	Identificación y control del comando.
Tipo	Requisito Restricción
Descripción	El sistema deberá identificar el comando de tipo joystick conectado a la computadora y dar opciones de mapeo de las funciones de navegación a elección del usuario, siendo por defecto uno ya estipulado, utilizando la librería Pygame ⁷ .
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R013
Nombre del requisito	Gestión con el hardware de comunicación inalámbrica.
Tipo	Requisito Restricción
Descripción	El sistema deberá identificar el hardware pertinente a la comunicación inalámbrica y brindar las opciones para su correspondiente configuración de red.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

3.1.3. Interfaces de software:

Número de requisito	R014
Nombre del requisito	Integración de Dronekit
Tipo	Requisito Restricción
Descripción	Uso de la librería Drone-kit en Python para poder gestionar todo tipo de interacción con el VANT.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

3.1.4. Interfaces de comunicación:

Número de requisito	R015
Nombre del requisito	Comunicación inalámbrica
Tipo	Requisito Restricción
Descripción	El sistema debe permitir el establecimiento de una comunicación inalámbrica con el vehículo según los módulos de comunicación existentes en ambos extremos, además de poder buscar, establecer o interrumpir la comunicación con el vehículo presente.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

3.2. Requisitos funcionales

Número de requisito	R016
Nombre del requisito	Estado de conexión
Tipo	Requisito Restricción
Descripción	Se deberá proveer una sentencia en el script principal encargada de consultar periódicamente el estado de conexión del vehículo o de acercarse fuera del rango de cobertura. En caso de perder la conexión se deberá invocar una función de salvataje.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R017
Nombre del requisito	Función de salvataje
Tipo	Requisito Restricción
Descripción	Implementar acciones de resguardo del vehículo en caso de perder la conexión o tener un bajo nivel de recepción de señal. Este script deberá consultar los niveles de energía restantes y en caso de obtener niveles moderados, se le enviarán al vehículo las acciones necesarias para que el mismo se estabilice y/o quede suspendido en el aire. En caso de no tener la energía necesaria se procederá a realizar un aterrizaje.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R018
Nombre del requisito	Script base para el chequeo del vehículo.
Tipo	Requisito Restricción
Descripción	El sistema deberá comprobar los estados de cada sensor instalado en el vehículo, con esto el script debe corroborar la existencia de cada sensor y su correcta calibración, en caso contrario se deberá deshabilitar la opción de despegue del vehículo hasta que se solucione el problema. De manera complementaria se deberá informar el origen del mismo.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R019
Nombre del requisito	Estructura de clases para UAV.
Tipo	Requisito Restricción
Descripción	Programar mediante el paradigma orientado a objetos las entidades necesarias para la gestión del vehículo como también conexión y control.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R020
Nombre del requisito	Clase sensor.
Tipo	Requisito Restricción
Descripción	Cada sensor perteneciente al vehículo corresponde a una entidad lógica para la gestión del vehículo, por tanto, este deberá proveer información sobre el estado, dar la opción de calibrar y obtención de información del mismo mediante funcionalidades implementadas en una clase.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R021
Nombre del requisito	Clase conexión.
Tipo	Requisito Restricción
Descripción	Debido a que la conexión entre la plataforma y el vehículo es un aspecto importante se deberá implementar una clase que gestione toda esta interacción.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R022
Nombre del requisito	Clase Comando.
Tipo	Requisito Restricción
Descripción	El joystick encargado de comandar la nave debe ser capaz de contener todas las funcionalidades que serán ejecutadas por el vehículo y además de corresponder dichas funciones sobre un mapeo personalizado. La clase comando deberá gestionar toda esta información.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R023
Nombre del requisito	Clase vehículo.
Tipo	Requisito Restricción
Descripción	Esta clase contendrá todo tipo de información proveniente del vehículo físico, además de sensores que este contiene y se encargará de interpretar las acciones de navegación que han sido enviadas por la plataforma.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R024
Nombre del requisito	Modos de vuelo.
Tipo	Requisito Restricción
Descripción	Proveer una funcionalidad dentro de la clase vehículo que permita guiar el vehículo de manera manual con el comando o mediante una lista de acciones. En caso de utilizar la segunda opción se permitirá además poder manejar el vehículo de manera manual para evitar algún tipo de accidente.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

3.3. Requisitos no funcionales

Número de requisito	R025
Nombre del requisito	Seguimiento del vehículo a través de un mapa.
Tipo	Requisito Restricción
Descripción	Mediante un mapa ubicar la posición geográfica del vehículo con respecto a los datos obtenidos del sistema de posicionamiento global.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R026
Nombre del requisito	Conversión de unidades.
Tipo	Requisito Restricción
Descripción	En aeromodelismo se utilizan una variada cantidad de unidades métricas para magnitudes físicas y la utilización de estos depende de cada usuario, es por esto, que se debe suministrar la opción de un conversor de unidades.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional

Número de requisito	R027
Nombre del requisito	Integración de GitHub
Tipo	Requisito Restricción
Descripción	Con fines de portabilidad el versionado del proyecto o producto final será gestionado mediante la plataforma GitHub.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional