

Informe Proyecto de Grado

Entrega postal usando UAVs

Diego González

Miguel Revetria

Tutora: Raquel Sosa

Cliente: Damián Pintos, Administración Nacional de Correos

Informe de proyecto de grado presentado al tribunal evaluador como requisito de graduación
de la carrera de Ingeniería en Computación

Carrera de Ingeniería en Computación

Facultad de Ingeniería

Universidad de la República

Montevideo - Uruguay

Abril - 2016

Resumen

En el contexto de actividades del Correo es de interés poder llegar a realizar entregas postales a los destinatarios en todo el país. Por diversas razones en algunas localidades y parajes no es posible enviar a un cartero por tierra para hacer la entrega.

Actualmente, para Uruguay, el servicio postal está disponible para una pequeña parte del territorio (3,8%). Varios factores influyen en esto, por ejemplo, los altos costos para la distribución en zonas dispersas, imposibilidad de acceso al destino, etc. El aumento de este porcentaje impacta directamente sobre una parte de la población, que podrá recibir ciertos servicios que actualmente no recibe por no contar con una dirección postal.

Para resolver esta problemática se plantea en el presente proyecto estudiar bajo qué condiciones es viable la entrega postal utilizando aeronaves no tripuladas. En el proyecto se estudiaron las características técnicas y los usos actuales de las aeronaves no tripuladas, se estudió en detalle las necesidades de entregas postales a lugares apartados.

Se propuso un mecanismo para solucionar el problema empleando UAVs conectados a un sistema de monitoreo en tiempo real. Se desarrolló un sistema capaz de tener control en todo momento sobre los drones en vuelo utilizando celulares inteligentes como nexo entre ambos.

Para validar la solución propuesta, se definieron dos grandes objetivos. El

primero, desarrollar un sistema que facilite la comprobación de la factibilidad técnica de la propuesta y las condiciones de uso necesarias. El segundo, utilizar dicho sistema con el fin de obtener métricas que permitan validar la solución propuesta, conocer restricciones que puedan existir, problemas que puedan surgir y mitigar los riesgos inherentes a un proyecto de estas características.

Ambos objetivos se consideran realizados. Por un lado, luego de recopilar y estudiar las métricas se llegó a la conclusión de que los drones actuales son capaces de transportar correspondencia como facturas a distancias de 5 km. Sin embargo, es necesario desarrollar mecanismo de seguridad adicionales que brinden un grado de confianza adicional. Por otro lado, se desarrolló un prototipo flexible que podrá ser utilizado como base para la implantación en producción de la solución.

Adicionalmente, se pudo adquirir dos modelos de drones: AR Drone 2.0 y X8+; para realizar pruebas de campo haciendo uso de la financiación brindada por el Centro de Innovación en Ingeniería. Esto permitió obtener parámetros reales que indican la viabilidad de la propuesta: capacidad de carga, tiempo de vuelo, duración de la batería, velocidad máxima, precisión de los drones, estudio de conectividad en el territorio del país, nivel de dificultad a la hora de planificación y ejecución de rutas.

Palabras clave: Drone, UAV, Entrega Postal

Tabla de contenido

Capítulo 1 - Introducción.....	1
1.1. Contexto	1
1.2. Motivación.....	1
1.3. Objetivos.....	2
1.4. Resultados esperados.....	3
1.5. Resultados alcanzados.....	3
1.6. Organización del documento	4
Capítulo 2 - Fundamentos	5
2.1. Antecedentes.....	5
2.1.1. Amazon Prime Air	5
2.1.2. DHL Parcel.....	5
2.1.3. Google[x] Project Wing.....	6
2.1.4. Topografía en Uruguay con drones	7
2.1.5. Resumen.....	7
2.2. Drones (UAVs)	8
2.2.1. Historia.....	8
2.2.2. Usos civiles	8
2.2.3. Características.....	8
2.3. Herramientas y tecnologías.....	11
2.3.1. Mecanismo de comunicación con drones.....	11
2.3.2. Comunicación bidireccional en tiempo real cliente-servidor.....	11
2.3.3. Comunicación de emergencia por SMS.....	13
Capítulo 3 - Análisis de Requerimientos y Arquitectura del Sistema	15
3.1. Análisis de requerimientos	15
3.1.1. Requerimientos	16
3.2. Consideraciones generales	17
3.2.1. Conexión continua	17
3.2.2. Carga máxima y autonomía	18
3.2.3. Alcance.....	18
3.2.4. Drones	19
3.2.5. Capacidad de programación de los drones.....	19

3.2.6. Regulación nacional de aeronaves	19
3.3. Modelo de Casos de Uso	20
3.3.1. Actores del sistema	20
3.3.2. Casos de uso del sistema	20
3.4. Arquitectura.....	23
3.4.1. Servidor (o sistema) central	24
3.4.2. Estación de control.....	24
3.4.3. Drone con sistema a bordo	24
3.4.4. Descomposición en subsistemas.....	25
3.4.5. Modelo de Despliegue del Sistema	29
3.4.6. Nodos	29
3.4.7. Conexiones	30
Capítulo 4 - Diseño Detallado.....	31
4.1. Casos de uso críticos.....	31
4.1.1. Planificación de rutas.....	31
4.1.2. Ejecución de comando	32
4.1.3. Registro de Drone	33
4.1.4. Envío de datos de estado	34
4.1.5. Aterrizaje de emergencia.....	35
4.2. Modelo de dominio	35
4.2.1. Descripción de entidades	36
4.3. Diseño de subsistemas	37
4.3.1. Sistema central	38
4.3.2. Sistema a bordo	38
4.4. Diseño de algoritmos.....	41
4.4.1. Navegación autónoma.....	41
4.4.2. Detección de estado de emergencia.....	43
4.5. Decisiones de implementación	44
4.5.1. Conexión entre sistema a bordo y sistema central vía WebSockets.....	44
4.5.2. Envío del estado mediante mensajes SMS.....	44
4.6. Capa de servicios	45
4.6.1. Estación de control.....	45
4.6.2. Sistema a bordo	45

5.1. Recopilación de datos y métricas de vuelo	48
5.1.1. Descripción.....	48
5.1.2. Objetivo.....	48
5.1.3. Resultados Esperados	48
5.1.4. Resultados Obtenidos	48
5.2. Pruebas del sistema.....	52
5.2.1. Simulador de vuelo X8+.....	52
5.2.2. Pruebas de casos de uso crítico 1 - Planificación de rutas.....	53
5.2.3. Pruebas de casos de uso crítico 2 - Ejecución de comando	53
5.2.4. Pruebas de casos de uso crítico 3 - Registro de Drone	54
5.2.5. Pruebas de casos de uso crítico 4 - Envío de datos de estado	54
5.2.6. Pruebas de casos de uso crítico 5 - Aterrizaje de emergencia.....	54
5.3. Resultados	55
5.3.1. Verificación del sistema	55
5.3.2. Validación con el cliente	56
Capítulo 6 - Gestión del Proyecto	58
6.1. Planificación.....	58
6.2. Ejecución.....	62
Capítulo 7 - Conclusiones	65
7.1. Restricciones.....	65
7.1.1. Carga Máxima.....	66
7.1.2. Distancia Máxima	66
7.1.3. Condiciones geográficas	66
7.1.4. Destino	66
Capítulo 8 - Trabajo a Futuro	67
8.1. Implantación del sistema.....	67
8.2. Expansión del área de cobertura.....	68
8.3. Implementación del subsistema de Administración	68
8.4. Integración de componentes a bordo	68
8.5. Mecanismo de entrega.....	69
8.5.1. Mecanismo de tubo	69
8.5.2. Mecanismo de pinzas.....	69
8.6. Adquisición de equipamiento.....	69

8.6.1. FIREBIRD I.....	70
8.6.2. Yeair Quadcopter.....	70
Apéndices.....	71
Apéndice I: Glosario.....	71
Referencias.....	74

Tabla de ilustraciones:

Ilustración 1: cobertura actual y aplicando radios de alcance utilizando drones para la entrega.	2
Ilustración 2: drone utilizado por Amazon.....	5
Ilustración 3: drone utilizado por DHL.....	6
Ilustración 4: drone utilizado en Google[x] Project Wing.....	7
Ilustración 5: Lehmann LA-300.....	9
Ilustración 6: 3D Robotics Aero M.....	9
Ilustración 7: 3D Robotics X8+ (multicóptero).....	10
Ilustración 8: T-REX 700 (helicóptero).....	10
Ilustración 9: APM Rover.....	11
Ilustración 10: cobertura móvil de Antel [20].....	12
Ilustración 7: diagrama de comunicación de sistemas comerciales.....	18
Ilustración 12: diagrama de comunicación del prototipo.....	18
Ilustración 13: modelo de casos de uso del sistema.....	21
Ilustración 14: modelo de casos de uso críticos del sistema.....	23
Ilustración 15: arquitectura del sistema.....	23
Ilustración 16: descomposición en subsistemas.....	26
Ilustración 17: diseño del subsistema “Sistema Central”.....	27
Ilustración 18: diseño del subsistema “Sistema A Bordo”.....	28
Ilustración 19: Modelo de distribución.....	29
Ilustración 20: diagrama de secuencia del CU Planificación de rutas.....	32
Ilustración 21: diagrama de secuencia del CU Ejecución de comando.....	32
Ilustración 22: diagrama de secuencia del CU Registrar Drone.....	33
Ilustración 23: diagrama de secuencia del CU Envío de datos de estado.....	34
Ilustración 24: diagrama de secuencia del CU Aterrizaje de emergencia.....	35
Ilustración 25: modelo de dominio.....	36
Ilustración 26: diagrama de comunicación resumido de las principales actividades del sistema a bordo.....	39
Ilustración 27: diseño de clases del driver de drones.....	40
Ilustración 28: relación entre heading y bearing.....	42
Ilustración 29: simulador SITL ejecutando un vuelo.....	52
Ilustración 30: vista trimestral de la planificación inicial.....	59
Ilustración 31: vista trimestral de la planificación actualizada con actividades extras por financiación.....	60
Ilustración 32: vista trimestral de la planificación para la segunda convocatoria.....	62
Ilustración 33: vista trimestral de la ejecución del proyecto.....	64

Índice de tablas:

Tabla 1: autonomía de Parrot AR.Drone 2.0 en relación a la carga transportada.	49
Tabla 2: alcance y velocidades de Parrot AR.Drone 2.0.....	49
Tabla 3: comportamiento de Parrot AR.Drone 2.0 con carga adicional.	49
Tabla 4: autonomía de 3D Robotics X8+ en relación a la carga transportada.	50
Tabla 5: alcance y velocidades de 3D Robotics X8+.....	51
Tabla 6: comportamiento de 3D Robotics X8+ con carga adicional.	51
Tabla 7: condición de los casos críticos en cuanto a su desarrollo y errores detectados. En la columna de errores detectados, puede verse entre paréntesis si el mismo está solucionado (si) o no lo está (no).....	55
Tabla 8: calendario de gastos.	60

Capítulo 1 - Introducción

1.1. Contexto

UAV es la sigla en inglés para Unmanned Aerial Vehicles (Vehículos Aéreos no Tripulados) más conocido como drone. Los drones comenzaron poblando los cielos en zonas de conflictos bélicos con fines militares. Su capacidad para ser pilotados remotamente quitando a los soldados de las zonas de riesgo los hizo populares en los ejércitos de los países desarrollados. Como ocurre normalmente con la tecnología militar, con el paso del tiempo y el abaratamiento de los costos de producción, los drones llegaron al mercado civil, poblando definitivamente los cielos en todos los rincones del planeta.

Estos vehículos cuentan con ciertas características que los destacan para su uso como medio de transporte de paquetes de poco peso. Su bajo precio los hace accesibles para el público en general. Son equipos que funcionan a base de energía eléctrica siendo amigables con el medio ambiente, uno de los compromisos de todo proyecto que aspira a posicionarse en la industria del mañana. Existen proyectos a nivel mundial que buscan estandarizar la construcción de drones, consiguiendo equipos creados a base de módulos interoperables facilitando la customización y expansión de sus capacidades de forma sencilla y económica. Un ejemplo es el proyecto ArduPilot [10]. Esta estandarización va más allá del hardware que lo compone, uniformizando el software también que corre en ellos y controla el equipo. Esto facilita la programación de los sistemas que controlan los drones.

Sin embargo, en la actualidad, solo existen drones comerciales con mandos a corta distancia, es decir que el piloto de la unidad debe estar relativamente cerca del drone para que se pueda controlar manualmente. Esto imposibilita conectarse al drone a través de una conexión continua a larga distancia para conocer su estado o enviar acciones o comandos para modificar los parámetros de su vuelo. Tampoco hay desarrollado al momento de este proyecto ningún sistema que extienda el alcance de control de los drones. Esto es una gran desventaja que presentan estos equipos para el transporte de mercadería a distancias considerables. Es que se necesita contar con control y conocimiento del estado del drone y del paquete en todo momento. El proyecto busca eliminar esta restricción, permitiendo al piloto del vehículo estar conectado continuamente con él sin importar las distancias. Otro inconveniente que presentan los drones comerciales es su baja autonomía y capacidad de carga. Esto presenta problemas al momento de realizar viajes de largas distancias o transportando objetos pesados.

Podemos encontrar en otras partes del mundo proyectos similares llevados a cabo por empresas de transporte de correspondencia. Estos proyectos buscan transportar de forma automática y económica correspondencia usando drones. Podemos destacar los proyectos llevados a cabo por Amazon, DHL y Google. Los mismos serán descritos más adelante en este documento.

1.2. Motivación

Actualmente, para Uruguay, el servicio postal sólo está disponible para el 3,8% del territorio (95% de la población). Existen varios factores que influyen en esto, como por ejemplo los altos costos para la distribución en zonas dispersas, imposibilidad de acceso al destino, etc.

Por otro lado, el Correo cuenta con bases distribuidas en las regiones más pobladas del interior del país. En las zonas con menor densidad de población no existe el sistema de entrega domiciliaria de correspondencia. En las regiones más apartadas, los residentes deben ir hasta una dependencia del Correo para verificar si les ha llegado algún paquete o correspondencia. Para

estos casos es posible que la persona deba recorrer decenas de kilómetros, para concurrir a un local comercial del Correo, solamente para consultar si ha recibido correspondencia. Esto puede implicar varias horas de viaje, la pérdida del día laboral, atrasos en la recepción de su correspondencia, entre otros contratiempos no deseados.

En algunos casos se ha aplicado una mejora en el servicio postal (“Tenés correo”), la cual consta en una notificación mediante mensaje de texto al celular del destinatario indicándole que le ha llegado correspondencia. Aún así, por estas limitaciones, no se pueden enviar facturas con fecha de vencimiento ni correo certificado.

De forma simplificada, la logística del Correo es la siguiente: al recibir correspondencia, para el caso que la recepción sea en el interior, se realiza el transporte a Montevideo. De esta forma el correo centraliza la correspondencia para, entre otras cosas, realizar una clasificación e identificar el destino y camino que seguirá. Posteriormente se realiza el envío de la correspondencia hacia una sucursal del correo. Ésta sucursal realizará el envío hasta el domicilio destinatario en caso de estar dentro del área de cobertura. La solución propuesta involucra cambios en la distribución de correspondencia, solamente en el último tramo (el envío desde la sucursal, hasta el domicilio), y solamente para la correspondencia que actualmente no se entrega por motivos económicos.

En la siguiente imagen puede verse la cobertura actual del Correo, y en colores más claros, se puede apreciar el incremento de la misma suponiendo distintos buffers de cobertura [7].

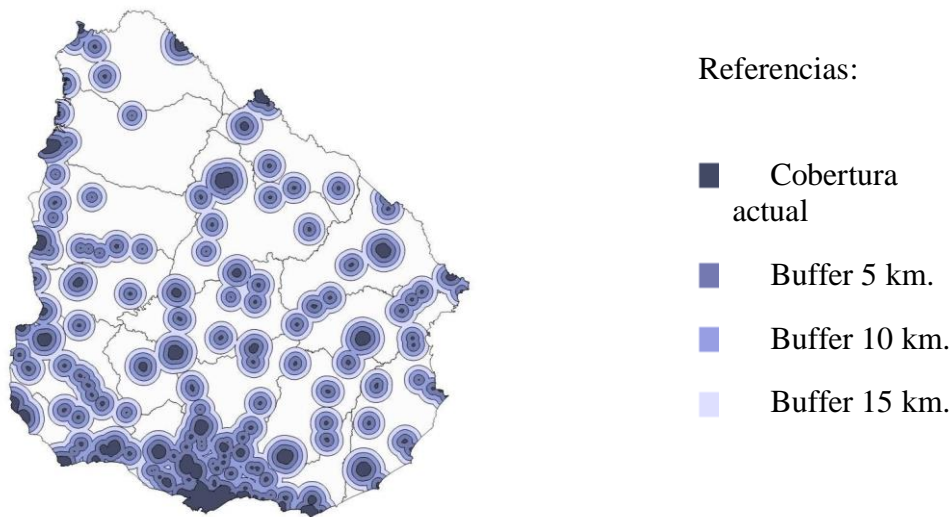


Ilustración 1: cobertura actual y aplicando radios de alcance utilizando drones para la entrega.

1.3. Objetivos

El proyecto plantea dos objetivos principales:

1. Estudiar la viabilidad de extender el área de cobertura del Correo Uruguayo a regiones del interior del territorio nacional utilizando vehículos aéreos no tripulados. En el contexto del correo, hay que estudiar las condiciones en zonas alejadas y poco pobladas del interior del país que afectan la entrega postal utilizando vehículos aéreos. Los datos que son de interés para el proyecto son los siguientes:

- distancias a recorrer entre las bases del correo y los destinos de las entregas.
- volumen y masa de los paquetes a entregar.
- disponibilidad de conexión a internet mediante la red de telefonía móvil

2. Desarrollar un sistema a usar por el correo para la entrega de correspondencia utilizando drones. Con el prototipo desarrollado se resuelven por software las restricciones de los drones inherentes a la comunicación en tiempo real a largas distancia y el manejo automático del vehículo. Con el prototipo desarrollado se podrá dar una visión más precisa de las capacidades (alcance y carga) de estos vehículos para la entrega de correspondencia. El prototipo consiste en el desarrollo de un sistema distribuido, que permitirá la planificación, modificación, almacenamiento y ejecución de rutas.

1.4. Resultados esperados

Con el fin de estudiar la viabilidad de la solución propuesta se desarrollará un prototipo que cumpla con las necesidades básicas del proyecto. Mediante la implementación del prototipo se espera conocer restricciones que puedan existir, problemas que puedan surgir y mitigar riesgos.

En primer lugar, se desea implementar un prototipo flexible, es decir un sistema que permita de forma sencilla el intercambio y la adición de nuevos drones, implementando únicamente un driver que cumpla con las especificaciones. De esta manera el costo de integrar un nuevo dron de mayores capacidades será bajo. Por otro lado, se desea estudiar la performance de los drones adquiridos, para los cuales se desarrollarán los drivers necesarios. Luego de dicha implementación se realizarán múltiples pruebas con el objetivo de obtener métricas que respondan a las inquietudes que motivan el proyecto, como velocidades, distancias máximas, cargas máximas, precisión de los drones, estudio de conectividad en el territorio del país, nivel de dificultad a la hora de planificación y ejecución de rutas.

1.5. Resultados alcanzados

De las pruebas de vuelo realizadas se obtuvieron los datos buscados relacionados con las capacidades de los drones adquiridos para este proyecto. El que mostró mejor desempeño, como era esperado, fue el modelo X8+. El mismo consiguió una autonomía promedio de 00:19:30 y el alcance medio estimado fue de 6,42 Km. Esto posiciona el dron X8+ por encima del primer buffer de cobertura definido. Los resultados se plasmaron en el documento Métricas de vuelo de los drones, ver [5] por información más detallada.

Las pruebas realizadas también probaron que la red de telefonía móvil provista por Antel en el territorio nacional tiene capacidad suficiente como para mantener la comunicación entre el sistema a bordo del dron y servidor central. La calidad de la señal en las distintas áreas del territorio hace que la comunicación en algunos casos no sea tan fluida, pero se considera que la velocidad de transmisión y la tasa de envío de los datos de estado es aceptable. Información en detalle de la velocidad y tasa de transmisión se encuentra disponible en [5].

En cuanto al prototipo desarrollado, se completaron las principales funcionalidades del mismo implementando todos los casos de uso críticos definidos. El sistema es capaz de planificar e iniciar una ruta de entrega, monitorear los drones en vuelo y ejecutar comandos durante los vuelos, vital para responder a escenarios atípicos. Los casos de prueba ejecutados finalmente pasaron de forma satisfactoria. Diversas correcciones fueron realizadas a lo largo de ciclo de test del sistema ajustando principalmente el algoritmo de navegación y la detección de casos de emergencia. El sistema se divide en 3 componentes: el servidor central que se encuentra corriendo en una instancia de OpenShift [12] y otras 2 aplicaciones Android. El servidor central

es una aplicación Java web probada en Apache Tomcat. El código fuente del proyecto está versionado en un repositorio git [25] en Bitbucket [24].

Adicionalmente se generaron documentos que detallan las distintas áreas del proyecto, análisis y estudio previo, diseño, evaluación de riesgos técnicos y consideraciones de implementación y pruebas. Toda esta documentación se encuentra disponible en el mismo repositorio de código fuente.

El equipo se postuló en 2 ocasiones al llamado “Ideas de proyectos de innovación” del Centro de Innovación en Ingenierías (CII), la cual ofrecía un fondo de U\$S 12.500 a proyectos innovadores con objetivos que beneficien a la población uruguaya. La primera postulación se realizó el 30 de abril de 2014 y la segunda el 8 de Setiembre de 2014. En esta última se logró acceder a los fondos y con esta obtener los equipos necesarios para el desarrollo de los prototipos. La ejecución del presupuesto se realizó en 3 desembolsos, con el primero realizado en diciembre del 2014, se obtuvieron equipos de desarrollo y drones Parrot AR. El segundo fue realizado en junio del 2015, con el cual se adquirió el dron Robotics 3DR X8+, junto con los equipos Android. El último desembolso se utilizó para gastos de insumos para la entrega del informe, como fotocopias, CDs, etc.

1.6. Organización del documento

El documento se compone de 8 capítulos. Se inicia estableciendo el contexto en el que fue desarrollado el proyecto, presentando al lector la problemática estudiada, los objetivos planteados al inicio y resumiendo los resultados alcanzados en este proyecto. A continuación, se presenta el marco teórico en torno a la problemática atacada, se profundizan en los temas específicos inherentes al proyecto.

En el capítulo 3 se introducen los requerimientos relevados durante la etapa de análisis junto con una descripción general de la arquitectura del sistema implementado. El capítulo 4 explica detalladamente el diseño del sistema implementado durante el proyecto. El lector puede encontrar en estas secciones la información técnica relevante para la correcta comprensión del sistema implementado. El capítulo siguiente detalla los resultados obtenidos en las etapas de verificación del sistema. También se presentan en este capítulo los resultados de las pruebas realizadas a los UAVs adquiridos. Esto permite conocer las capacidades de cada uno de los modelos de UAV usados.

Se presenta el cronograma del proyecto en el capítulo 6. En esta se detalla el cronograma tentativo del proyecto y el efectivo. Se discuten los motivos que explican el cronograma ejecutado. Los últimos dos capítulos presentan las conclusiones y tratan los puntos a mejorar del proyecto y del sistema implementado. Además, se detallan las tareas necesarias con el objetivo de continuar con el proyecto por parte del correo.

Capítulo 2 - Fundamentos

2.1. Antecedentes

En la actualidad existen antecedentes de proyectos similares en otras partes del mundo, pero ninguno ha sido puesto en producción. A continuación, listamos los más relevantes.

2.1.1. Amazon Prime Air

La empresa Amazon está desarrollando el programa “Amazon Prime Air” [16], el cual consiste en la entrega de productos a domicilio en media hora a partir del momento de compra desde su sitio. La entrega se realiza mediante octocópteros, drones con 8 helices, diseñados por la empresa que pueden transportar cargas de hasta 2,5 kg y tienen una autonomía para lograr alcanzar un radio de 16 km a una velocidad máxima de 22 m/s.

Al llegar al punto de entrega el drone aterriza y libera el paquete de forma automática. Acto seguido, el drone despegue y retorna a la base.



Ilustración 2: drone utilizado por Amazon [16].

2.1.2. DHL Parcel

DHL publicó los vuelos de prueba de sus aeronaves no tripuladas [13]. Las pruebas consistieron en una entrega de medicamentos recorriendo un total de 2,5 km. El drone utilizado para estas pruebas fue el md4-1000, un drone de aproximadamente US\$ 50.000, que puede transportar cargas de hasta 1,2 kg [17] y cuenta con una autonomía de 90 minutos. Muy por encima de otros modelos estudiados, como referencia el modelo X8+ tiene una autonomía de 15 minutos [41].

Los investigadores de DHL modificaron el drone de modo de aumentar la duración del vuelo y el alcance. En un vuelo de prueba se entregaron medicamentos desde el puerto de Norddeich (Alemania) hasta la isla de Juist. La distancia recorrida fue de unos 12 Km a una velocidad

máxima 18 m/s. Al igual que Prime Air, el drone utilizado por DHL debe aterrizar para liberar la carga, la cual es removida manualmente por el destinatario.

El sistema monitorea en todo momento el estado del drone. De la información obtenida no queda claro qué tecnología de comunicación utiliza (radio frecuencia o red telefonía móvil).



Ilustración 3: drone utilizado por DHL [13].

2.1.3. Google[x] Project Wing

En Queensland, Australia, los miembros del proyecto denominado “Project Wing” [14] están haciendo las primeras pruebas con drones de 1.5m de envergadura, y 8.6 Kg de peso. Este drone es elevado por cuatro propulsores y permite transportar hasta 1.4 Kg. El sistema posee una estación de control que permite monitorear el drone en todo momento, a pesar de que apuntan a un sistema totalmente automático.

Este proyecto busca resolver las limitaciones de los drones existentes creando un nuevo prototipo de UAV híbrido: capaz de volar como aeroplano y como helicóptero. Esto aumentaría el alcance del vehículo de forma notoria, permitiendo el despegue y aterrizaje en vertical, además de mantener la posición a la hora de realizar la entrega.

Incluyeron en estos drones un mecanismo de transporte y liberación de los paquetes. Al momento de realizar la entrega el drone pasa a modo helicóptero para mantener la posición fija. Luego el mecanismo de transporte hace descender el paquete sujetado por una cuerda y al momento de tocar el piso, el sistema libera el paquete de la cuerda. De esta manera no es necesario que el drone aterrice, reduciendo la posibilidad de accidentes al mantener el drone lejos de los destinatarios.



Ilustración 4: drone utilizado en Google[x] Project Wing [14].

2.1.4. Topografía en Uruguay con drones

Se está llevando a cabo en Uruguay proyectos de topografía y fotografía aérea utilizando drones. Una de las empresas, con la cual tuvimos una reunión de investigación, es UAV Agrimensura [18]. Ellos utilizan drones tipo multirrotor y aeroplanos para tomar fotografías del terreno y generar más tardes mapas de alta resolución de la topografía uruguaya.

Destacaron algunas características deseables que los drones que utilizan no tienen. Características que aumenten la seguridad del vehículo durante los vuelos. Principalmente la falta de conexión con la estación central durante vuelos de mediano/largo alcance (5+ km). Consideran sumamente útil contar no solo con telemetría a larga distancia sino también la opción de enviar comandos al drone en todo momento durante el vuelo. Otra especificación interesante que se manejó durante la reunión, sería la de incluir un paracaídas de seguridad que evite la caída libre en un accidente.

2.1.5. Resumen

Vistos los antecedentes anteriores, podemos suponer que la entrega de correspondencia utilizando aeronaves no tripuladas será un hecho en un futuro cercano. Obteniendo resultados y métricas favorables durante este proyecto, el correo nacional podría ser el primer caso en Latinoamérica en incorporar esta tecnología y de los primeros en el mundo.

De los resultados de los proyectos anteriores se desprende que las capacidades, en cuanto a carga y autonomía, de los UAVs actuales son suficientes para el transporte de carga de poco peso. Sin embargo, para la entrega de correspondencia, existen falencias en los sistemas de UAV que requiere trabajo adicional para hacer viable un proyecto de estas características. Un ejemplo claro de esto es la falta de conectividad de los drones a largas distancias, o mecanismos de seguridad en caso de que por algún motivo el drone pierda su capacidad de vuelo, etc. Uno de los objetivos de este proyecto es solucionar las carencias actuales de los sistemas y drones existentes en cuanto a software. El otro objetivo es determinar, bajo las condiciones particulares del Uruguay y del Correo, cuánto puede extenderse el área de cobertura de entrega del correo.

2.2. Drones (UAVs)

2.2.1. Historia

Como podemos ver en [29], el primer lanzamiento de una aeronave no tripulada data de 1922 (RAE 1921) lanzado desde una aeronave transportadora. Se emplearon durante la segunda guerra mundial para entrenar a los operarios de los cañones antiaéreos. Sin embargo, no es hasta poco más que a finales del siglo XX cuando operan los UAV controlados mediante radio control. En cuanto a la obtención, manejo y transmisión de la información, gracias a la aplicación de nuevas técnicas de protección de las mismas resulta posible conseguir comunicaciones más seguras, más difíciles de detectar e interferir.

Estas aeronaves no tripuladas también son utilizadas en un pequeño pero creciente número de aplicaciones civiles como en labores de lucha contra incendios [26] o seguridad civil, como la vigilancia de los oleoductos o la entrega postal. Los vehículos aéreos no tripulados suelen ser preferidos en misiones que son demasiado “aburridas, sucias o peligrosas” para los aviones tripulados.

2.2.2. Usos civiles

Alrededor del mundo están surgiendo diversas aplicaciones no militares para estos vehículos. Usos tan variados como el apoyo a la investigación del comportamiento de tormentas tropicales hasta misiones de búsqueda y rescate surgen continuamente a medida que esta tecnología se vuelve cada vez más accesible a personas de todo el mundo. La lista de proyectos que hacen uso de los UAV es cada vez más extensa:

1. **Conservation Drone Project:** han agregado una cámara termal a un UAV para estudiar la distribución de orangutanes, densidad, cambios de territorios, etc.
2. **Agricultura de punta:** la agricultura moderna utiliza drones para monitorear campos incrementando el rendimiento y ahorrando dinero. En Gielow, Alemania hicieron notar que la aplicación de pesticidas, agua o fertilizantes con precisión, en donde los UAV pueden identificar exactamente en dónde es necesario y llevarlo hasta allí, es mejor para el medio ambiente y para el balance de los agricultores.
3. **Búsqueda y rescate:** en mayo del 2013, en Canadá, una víctima de un accidente de tránsito fue la primera persona encontrada por un dron de búsqueda y rescate. Un Dragan Flyer X4-ES con equipamiento de detección de calor de alta sensibilidad halló al accidentado.
4. **Mapeo 3D:** pequeños y livianos drones con forma de aviones pueden sobrevolar un terreno y sacar miles de fotografías de alta resolución
5. **Caza de huracanes:** La NASA junto con la National Oceanic and Atmospheric Administration (NOAA, EEUU) utilizan UAVs de gran autonomía para espiar tormentas y cómo estas evolucionan. Los drones usados en el programa Global Hawk pueden estar 30 horas volando a 17 Km de altura con una envergadura de 35 metros.

Los usos anteriores fueron recabados de las fuentes [32] y [33].

2.2.3. Características

Existe una amplia gama de drones en el mercado. Usaremos una clasificación simplificada según el tipo (clase) de vehículo presentada en [31] a la que agregamos una tercera clase: *aeroplanos de ala fija*, *rotorcrafts* y *rovers*. La clasificación de los vehículos es bastante sencilla. Por ejemplo, si una aeronave obtiene su sustentación aerodinámica de aspas giratorias, la misma

es considerada un rotorcraft. Mientras que, si la elevación es generada por una o varias alas fijas, la aeronave es considerada un aeroplano. Lo drones que se desplazan terrestremente se denominan rover.

Esta clasificación es altamente adoptada y es utilizada por el gobierno de Estados Unidos para la clasificación de UAVs. Agrupa los drones de forma tal que los miembros de una misma clase cuentan con similar performance para las características de alcance, capacidad de carga, forma de despegue/aterrizaje, precisión. Estas son los principales rasgos que nos interesa analizar, por lo que entendemos, esta clasificación se amolda a las necesidades del proyecto.

Aeroplano de ala fija

Los drones de ala fija tienden a ser más seguros en el aire ya que tienen la capacidad natural de planear sin energía. Estos drones también son capaces de llevar mayores cargas a distancias más largas con menos energía. Hay una gran variedad de aeroplanos de ala fija desde pequeños aeroplanos de espuma alimentados por baterías eléctricas hasta las réplicas en madera de gran escala con múltiples motores de combustible líquido y todo lo demás.

Cuando se deben realizar trabajos de precisión, los drones de ala fija se encuentran en desventaja, ya que deben tener aire en movimiento sobre sus alas para generar sustentación. Esto implica que deben permanecer en movimiento hacia adelante, lo que significa que no pueden flotar en un solo lugar al igual que un helicóptero y como consecuencia no pueden ofrecer el mismo nivel de precisión para posicionar la cámara o instrumento que utilice.

Así que para las misiones más largas y con grandes cargas, un drone de ala fija es la mejor opción. Sin embargo, para mantener una cámara en un lugar, se deberá considerar el cambio a un helicóptero en su lugar.

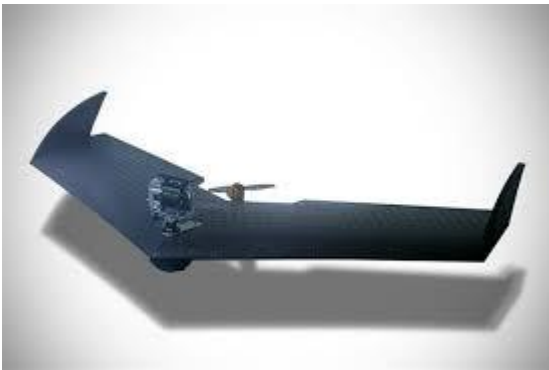


Ilustración 5: Lehmann LA-300 [39].



Ilustración 6: 3D Robotics Aero M [40].

Rotorcrafts

Esta clase de drones puede ser dividida nuevamente en dos subclases

- **Multicópteros:** utilizan un controlador de empuje diferencial con motores de propulsión independientes para proporcionar control de elevación y dirección. Se benefician de una mecánica sencilla y flexibilidad de diseño. Pueden elevar cargas y funcionar correctamente incluso en condiciones de viento fuerte. Contar con varias fuentes de elevación puede darles un margen de seguridad mayor. La variedad de diseños permite contar con convenientes opciones para el montaje de carga.

- **Helicópteros:** típicamente utilizan un único rotor de elevación con dos o más aspas. Mantienen el control direccional variando el ángulo (pitch) de las aspas a través de una conexión mecánica servo-accionado. Son vehículos fuertes, rápidos y eficientes, adecuados para muchas misiones.



Ilustración 7: 3D Robotics X8+ (multicóptero) [37].



Ilustración 8: T-REX 700 (helicóptero) [38].

Rovers

Rover es un proyecto promovido por ArduPilot, el mismo se trata de un avanzado piloto automático de código abierto para guiar vehículos terrestres y embarcaciones. Se puede ejecutar misiones totalmente autónomas que se definen utilizando software de planificación de misión o pregrabado por el conductor durante una ejecución manual.

Estos tipos de vehículos presentan ciertas ventajas que los hacen útiles para el reparto de correspondencia bajo ciertas condiciones. Se pierden los riesgos inherentes de los drones aéreos los cuales pueden llegar a precipitarse bajo condiciones extremadamente anómalas. Un accidente aéreo siempre tendrá consecuencias serias (pérdidas materiales e incluso humanas). Por otro lado, los rovers precisan de caminos, carreteras, etc., en condiciones de ser transitados. Estos caminos además serán transitados por otros vehículos manejados por seres humanos, por lo tanto, se hace imprescindible la implementación de algoritmos de navegación, detección de obstáculos y coordinación con estos vehículos.

Teniendo en cuenta las condiciones en las que se plantea el proyecto, esto es, incrementar el área de cobertura en zonas de difícil o imposible acceso por malas condiciones de los caminos, entre otros inconvenientes. A lo anterior se suma que la implementación de estos algoritmos de navegación y conducción terrestres son los suficientemente complejos como para el desarrollo de un proyecto entero. Debido a lo anterior se dejan de lado para el alcance de este proyecto este tipo de vehículos no tripulados. Sin embargo, pueden ser una buena alternativa para zonas en las que existen caminos en buenas condiciones.



Ilustración 9: APM Rover.

2.3. Herramientas y tecnologías

Para el desarrollo del proyecto y del prototipo en particular se investigó sobre las tecnologías y herramientas existentes que fueran necesarias o simplemente ayudaran a conseguir una mejor solución.

2.3.1. Mecanismo de comunicación con drones

Una métrica importante a la hora de evaluar los drones para considerarlos en el proyecto es el mecanismo de comunicación que ofrecen tanto para el envío de datos de telemetría como para la recepción de comandos. En este sentido consideramos una ventaja que las plataformas (hardware del drone y sistemas internos) fueran de carácter open source. Durante la investigación de los drones disponibles en el mercado encontramos dos plataformas que cumplían con este requisito:

- **AR.Drone 2.0 Parrot:** los modelos AR.Drone 2.0 de Parrot ofrecen un SDK open source en varios lenguajes para Android e iOS. Adicionalmente, los protocolos de comunicación utilizados por los drones están bien documentados. Lo anterior explica la variedad de lenguajes de programación en los que se desarrollaron SDK.
- **ArduPilot:** ArduPilot va un paso más allá que Parrot, definiendo una plataforma completamente open source, desde la especificación de los componentes de hardware que forman los drones hasta el firmware y los SDK de programación en varios lenguajes/ambientes (Android, Python, iOS).

2.3.2. Comunicación bidireccional en tiempo real cliente-servidor

Otro punto fundamental del sistema es que el servidor y el drone se encuentren conectados en todo momento de modo que ambos nodos sean capaces de enviar y recibir información del uno al

otro en simultáneo. Este tipo de conexiones son conocidas como full dúplex. Para esto lo primero es definir los canales físicos de comunicación viables que permitan este tipo de comunicación:

Vía Radiofrecuencia

Uno de los posibles canales de comunicación es mediante radiofrecuencia. Dependiendo del hardware específico que se esté usando, se puede contar con una conexión full dúplex. Se evalúa el alcance que pueden brindar los módulos de radiofrecuencia disponibles en mercado, su precio, su consumo energético y tamaño. Estos puntos son claves dado que cada drone tendrá que contar con su propio módulo encima. Los agrupamos en dos categorías según el alcance que logran:

- **mediano alcance:** con un precio que ronda los USD 40, tienen una cobertura máxima de 2 kilómetros en áreas abiertas. Estos componentes son en general pequeños módulos con antena incluida, pesan unos pocos gramos (15-100). Trabajan con un voltaje que oscila entre los 3-10 V y 2-3 mA. Aproximadamente, en una hora, consumiría el 0.35% de una batería de 1000mAh (capacidad de la batería del drone AR.Drone Parrot 2.0).
- **largo alcance:** llegan a decenas de kilómetros de cobertura (~20-80 km) sin embargo en áreas urbanas el alcance no supera los 2 kilómetros. El precio aumenta hasta unos USD 300, para los más económicos. La serie de sensores de Waspmote están enfocados en módulos para crear nodos autónomos, para esto cuentan con un diseño muy compacto con un peso reducido y batería propia.

Vía IP

Como alternativa a la comunicación vía antenas de radiofrecuencia, se puede utilizar la red de telefonía móvil presente en el territorio nacional y mantener los drones conectados al sistema por internet. Para conocer la viabilidad de usar la red 2-4G de Uruguay como canal de comunicación es necesario primero conocer cuál es la cobertura real de esta red. Según datos brindados por Antel [20] en su sitio online, la cobertura de la red 3G se extiende a casi la totalidad del territorio nacional, como se puede ver en la siguiente ilustración.

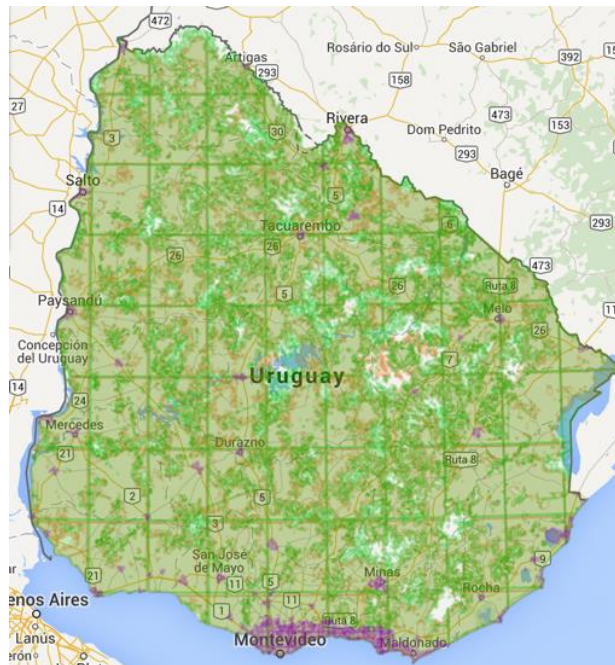


Ilustración 10: cobertura móvil de Antel [20].

Asumiendo que se utiliza la red móvil para conectar los drones con el servidor central por internet, necesitamos conocer qué protocolos existentes sobre IP hay que permitan una conexión full dúplex:

- **WebSockets [27]**: el protocolo de comunicación WebSockets permite una comunicación bidireccional y full dúplex usando una única conexión TCP.
- **Comet [19]**: Comet se refiere a una metodología usada en aplicaciones web en donde se mantiene una conexión HTTP abierta por un largo período de tiempo lo que permite al servidor enviar un mensaje por medio de alguna tecnología push. Existen dos técnicas de implementación Comet:
 - **Streaming**: el cliente Comet abre una conexión persistente con el servidor para todos los eventos Comet. Esos eventos son incrementalmente manejados e interpretados en el cliente cada vez que el servidor envía un nuevo evento, ningún de los dos lados (cliente y servidor) deben cerrar la conexión. Generalmente se utilizan iframes ocultos o el objeto XMLHttpRequest para implementar Comet streaming.
 - **Ajax long polling**: en este caso el cliente envía una petición por Ajax al servidor, la cual se mantiene abierta hasta que el servidor tiene información para enviar al cliente. El cliente luego inicia una nueva petición para obtener subsiguientes mensajes.
- **HTTP/2 [28]**: la siguiente versión de HTTP incorpora grandes mejoras a la versión actual de HTTP como el uso de una única conexión, la compresión de los encabezados usando HPACK y la posibilidad de envío de mensajes server-push.

En comparación con las comunicaciones por radiofrecuencia, la señal 3G tiene claras ventajas:

1. La cobertura y el alcance de la conexión es mayor. Con los módulos de radiofrecuencia de largo alcance es posible conectarse con un drone a unos 50-80 km, esto permitiría tener bases regionales, por ejemplo, la ciudad de Rivera podría contar con su propia base local para el envío de paquetes a las regiones más cercanas. Pero, siguiendo con el mismo ejemplo, la ciudad de Minas de Corrales y Vichadero tendrían que tener sus propias bases locales para expandir el área de entrega del correo a estas zonas. Usando la red de telefonía, una única base nacional sería capaz de conectarse con todos los drones activos.
2. El costo de los equipos necesarios es menor. Un módulo 3G compatible con placas Arduino ronda los USD 70, solo sería necesario comprar un módulo por drone dado que el servidor se conecta a internet por cable.

Por los motivos anteriores se decide usar la red de telefonía móvil para mantener conectados por internet los drones con la base.

2.3.3. Comunicación de emergencia por SMS

Como medida adicional de seguridad se enviarán datos entre los drones y la base mediante mensajes de texto SMS. Esto da un nivel más de confiabilidad a las comunicaciones dado que la red de SMS es más extensa que la de internet móvil.

Usando celulares con sistema operativo Android resulta muy sencillo el envío y la recepción programática de mensajes SMS. Esto apoya la opción de mantener físicamente unido un celular al drone. Del lado del servidor puede usarse algún servicio disponible para el envío/recepción de

SMS o conectar otro(s) celular(es) destinados al envío y recepción de mensajes interactuando con el sistema central directamente. Esta conexión puede ser cableada por USB o por WiFi. Una tercera alternativa sería conectar los celulares por bluetooth al servidor central, pero esto puede plantear alguna dificultad adicional a la hora de programar la comunicación entre ambos.

Capítulo 3 - Análisis de Requerimientos y Arquitectura del Sistema

3.1. Análisis de requerimientos

El relevamiento de requerimientos del sistema se llevó a cabo junto con Sergio Silva, Jefe Departamental del correo en Rivera. El estudio se enfocó en la situación del departamento de Rivera debido a que se tenía información más precisa del mismo. Sin embargo, podemos suponer que existen condiciones similares en otros departamentos del territorio nacional y las deducciones obtenidas en este estudio podrían ser aplicadas en ellos.

Uno de los grandes problemas que se presentan en el departamento de Rivera es la mala condición de las carreteras nacionales. Las mismas se encuentran muy deterioradas debido al tránsito de camiones de carga utilizados en las distintas actividades económicas del departamento: forestación, minería, transportes internacionales, etc. Esto dificulta en gran medida el acceso terrestre a gran parte del departamento elevando los costos del envío de la correspondencia a domicilio. A lo anterior se suma la geografía accidentada del terreno, la presencia de cerros dificulta y enlentece el transporte local.

Se destacan dos problemas actuales en lo que respecta a la entrega de correspondencia, los mismos se corresponden con problemas reales en la actualidad del correo para los cuales se está invirtiendo en mejorar la calidad del servicio. A continuación, se describen estos casos de aplicación del sistema.

1) Entrega de Facturas de servicios

Las facturas de servicios como UTE, OSE, etc. son almacenadas en las sucursales del correo hasta su vencimiento, luego de pasada esta fecha los mismos son descartados. Esto es lo más normal, debido a que es poco frecuente que los destinatarios se movilicen en busca de esta correspondencia. Se podría acordar con los residentes de las zonas sin coberturas la entrega de esta correspondencia en zonas más próximas y distribuidas. Estamos hablando de establecimientos públicos que otorguen ciertas garantías a la privacidad de esta correspondencia, por ejemplo, comisarías de campaña, escuelas rurales, policlínicas, etc. De este modo se reducen las distancias de las entregas a unos 5-20 Km.

En este caso, los paquetes son más livianos. Estamos hablando de paquetes de menos de un Kg, e incluso es viable realizar varios viajes si es necesario dividir la carga total.

2) Entrega de Paquetes de caravanas

En el interior del territorio nacional es una actividad común que los estancieros tengan que ir a las sucursales del correo en busca de caravanas de marcado para el ganado. Esto supone una pérdida de tiempo importante entre los estancieros quienes deben recorrer distancias importantes, decenas de kilómetros, en busca de estos paquetes. En Rivera, esto lleva a que estancieros de poblados como Vichadero y Minas de Corrales tengan que viajar hasta las sucursales de Tranqueras o Rivera en busca de los paquetes.

En cuanto a los paquetes, estos varían en tamaño, llegando a pesar 10 Kg los más grandes y 5 Kg los paquetes más livianos. Estos paquetes deberían ser entregados en el domicilio del destinatario debido a su alto valor monetario, por lo que su entrega debería ser realizada desde la sucursal del correo hasta el domicilio. Por lo que el problema se resume a entregas de 50-100 Km de distancia con cargas de 5-10 Kg.

Se destacó la importancia que tiene la correspondencia de cada ciudadano y que mantener la integridad y privacidad de la misma es una de las prioridades del correo en todo momento. El sistema desarrollado tiene que cubrir condiciones anómalas, siendo éste capaz de:

- alterar el plan de vuelo de forma automática para reducir el riesgo de accidentes
- mantener un registro detallado de la ruta y condiciones del dron y del paquete enviado en todo momento
- permitir que un funcionario del correo vea la posición y el estado del dron en todo momento
- el funcionario debe poder indicar acciones a ser ejecutadas durante el vuelo del mismo
- restringir el acceso a las funcionalidades que afectan el vuelo de un dron solo a personal calificado. Este personal será denominado como controladores de vuelo en el resto del documento.

Más allá de las medidas de seguridad implementadas por el sistema, debido a los riesgos existentes inherentes a un proceso de entrega de última tecnología con pocas experiencias previas, los beneficiarios deberán aceptar condiciones de uso en las cuales se hará explícito que el correo no se responsabiliza por daños o extravío de la correspondencia.

Es deseable poder definir a nivel del sistema, niveles de acceso y permisos sobre los funcionarios del correo que interactúen con los drones. De este modo un funcionario solo podría acceder a la telemetría o manipular los drones para los que tenga el permiso necesario asignado. Se desprende la necesidad de desarrollar una consola de administración que permita realizar la tarea de asignación de permisos de funcionarios a drones.

Cada paquete deberá ser entregado a domicilio o en su defecto en una dependencia del estado habilitada para recibir los paquetes a distribuir entre los ciudadanos de la región. Difícilmente los puntos de entrega puedan ser adaptados para el aterrizaje de los drones al momento de recibir las entregas. Por lo tanto, el vehículo encargado de realizar la entrega deberá ser capaz de aterrizar en terrenos accidentados. En los puntos de entrega tampoco se cuenta con personal capacitado como para iniciar el proceso de retorno del dron, es por esto que el sistema tiene que ser capaz de regresar por sí solo. Sin embargo, en los establecimientos que se utilicen como puntos intermedios en la entrega, se puede suponer que los funcionarios serán capacitados para realizar la tarea de recargar las baterías del dron nuevamente.

3.1.1. Requerimientos

Del análisis anterior obtenemos dos conjuntos de requerimientos: los que definen el sistema-software a implementar y los que restringen los modelos de drones viables para realizar las entregas.

Requerimientos de drones

Los drones utilizados para este sistema tienen que contar con las siguientes cualidades:

- 1) **Telemetría:** notificar el estado del dron en todo momento. La información destacada es la posición, velocidad, estado de la batería.
- 2) **API de control:** el sistema necesita ejecutar acciones de control sobre el dron durante el vuelo. Las operaciones básicas necesarias son las siguientes:
 - a) despegar y aterrizar
 - b) cambiar el vector de velocidad
 - c) indicar la siguiente coordenada a visitar (para drones con capacidades de vuelo autónomo).

- 3) **Conectividad:** tiene que ser posible conectarse al dron e a través de WiFi para recibir los datos de telemetría o indicar que ejecute alguna acción.
- 4) **Alcance:** la distancia mínima que es requerida para este proyecto es de 5 kilómetros. Es deseable poder contar con drones cuyo alcance llegue a los 50 kilómetros.
- 5) **Capacidad de carga:** el dron e debe ser capaz de levantar en el orden de 500-800 gramos como mínimo.

Requerimientos del sistema-software

- 1) **Monitoreo en tiempo real:** de la posición y estado del paquete y del dron e. Los datos más relevantes mencionados son:
 - a) posición
 - b) velocidad
 - c) estado de la batería
- 2) **Mecanismos de control remoto:** tiene que ser posible indicar al dron e que retorne a la sucursal del correo de donde despegó. Opcionalmente el controlador de vuelo podría cambiar la ruta actual de vuelo del dron e o pasar a vuelo manual en el cual él indicaría los comandos de vuelo.
- 3) **Respaldo de la telemetría:** mantener un respaldo de la telemetría de los drones en cada vuelo.
- 4) **Retorno automático:** del dron e a la base de la cual despegó tras completar la ruta planificada. Tener en cuenta que, si el dron e aterriza para dejar el paquete en el destino, este tiene que ser capaz de volver a despegar automáticamente para regresar a la base.
- 5) **Vuelo autónomo:** el dron e debe ser capaz de ejecutar una ruta de entrega de forma autónoma. La ruta más sencilla de entrega es la compuesta solo por un punto de destino y el retorno, el requerimiento mínimo es poder ejecutar esta ruta.
- 6) **Mecanismos de seguridad:** el sistema tiene que contar con mecanismos automatizados para detectar condiciones de emergencia
- 7) **Modo de emergencia:** en caso de que el dron e durante el vuelo entre en un estado de emergencia, el dron e debe aterrizar automáticamente y continuar enviando información de telemetría continuamente.

3.2. Consideraciones generales

3.2.1. Conexión continua

Dado que la ley protege fuertemente la correspondencia, es de vital importancia garantizar la seguridad de la misma, por lo que es necesario tener información y control del dron e en todo momento. Con esto podemos mantener un registro de la ubicación de la correspondencia a lo largo del tiempo. En cuanto a acciones mal intencionadas, no es posible asegurar la integridad de la correspondencia, del mismo modo que sucede hoy en día con los mecanismos utilizados actualmente. Esto es un impedimento para utilizar los drones existentes, ya que la comunicación de éstos con las estaciones de control es mediante WiFi o radiofrecuencia de corto alcance y por lo tanto la cobertura de éstos es inferior a las expectativas del proyecto [34]. Por otro lado, existen drones que permiten la planificación y ejecución de rutas, pero la comunicación de datos y envío de estado del dron e, están acotados al alcance de los canales mencionados anteriormente.

Por otro lado, generalmente, el usuario que planifique las rutas no se encontrará físicamente en la misma localidad, incluso en el mismo departamento que la aeronave. Por lo tanto, en este proyecto se debe definir un canal de comunicación que permita la comunicación entre los drones y la base en todo momento, de este modo el alcance del recorrido del dron e solamente estará

acotado por el área de cobertura de internet móvil, la cual es significativamente mayor a la actual red de distribución del correo.



Ilustración 11: diagrama de comunicación de sistemas comerciales.



Ilustración 12: diagrama de comunicación del prototipo.

3.2.2. Carga máxima y autonomía

Otro aspecto a tener en cuenta es la cantidad de correspondencia a ser entregada en un solo viaje, esto depende directamente de la carga máxima del dron y de su autonomía. En la actualidad, existen drones comerciales que pueden levantar una carga máxima de 2 kg, y volar durante 45 minutos sin tener que descender, pero el costo de estos drones excede el presupuesto del proyecto. No obstante, el sistema desarrollado fue diseñado modularmente, teniendo en cuenta un futuro cambio de modelo. De este modo, brindar soporte a nuevos modelos de drones implica un cambio menor en el sistema, solo sería necesario implementar el módulo de comunicación con el dron e incluirlo en el sistema abordo.

3.2.3. Alcance

En el interior del territorio nacional, las distancias entre las localidades y los centros de distribución del correo pueden llegar a alcanzar varias decenas de kilómetros. Sería ideal contar con drones capaces de alcanzar estas distancias en un solo vuelo. Una buena alternativa a evaluar es el uso de algunos espacios habilitados por el correo como bases intermedias de recarga de energía. Posiblemente estos espacios públicos puedan ser escuelas y comisarías de campaña. Esto permite aumentar el alcance de los vuelos de los drones.

3.2.4. Drones

Según los requerimientos anteriores para los modelos de drones con los cuáles trabajar, consideramos que los que mejor se adaptan son los de tipo rotorcraft. Entre sus cualidades destacan:

- despegue y aterrizaje en vertical, cómo un helicóptero, no requiere de una zona extensa a modo de pista de aterrizaje para descender;
- capacidad de despegue autónomo;
- mantener su posición, esto nos permite la implementación de un mecanismo de receptación de la correspondencia similar al desarrollado por Project Wing;
- tolerancia a vientos fuertes;

Tener en cuenta además que existen modelos de estos tipos de drones con un precio que los hacen accesibles usando los fondos de financiación obtenidos.

Quedan descartados para este proyecto los drones de ala fija, el principal problema es la necesidad de que una persona los lance para iniciar el vuelo y contar con una superficie y en condiciones que les permita aterrizar. Existen modelos de ala fija capaces de despegar de forma autónoma, sin embargo, sus costos los vuelven prohibitivos para este proyecto.

3.2.5. Capacidad de programación de los drones

Minimizar el costo de la entrega de la correspondencia utilizando drones implica contar con un sistema totalmente automatizado, que brinda la opción de poder controlar el dron en todo momento durante el vuelo. Esto requiere que el dron defina una interfaz de comunicación a través de la cual obtener la telemetría del mismo y poder enviar comandos y acciones al vehículo. Otros de los puntos a tener en cuenta al momento de seleccionar los drones con los que trabajar durante el proyecto es el de qué capacidad tienen de ser programados o controlados por un sistema externo.

En la actualidad existe un grupo dedicado al desarrollo de tecnologías de UAVs de modo open source. ArduPilot es una plataforma que incluye el desarrollo del hardware que compone los vehículos además del software necesario para su control programático. Esta plataforma tiene una ventaja importante frente a otros fabricantes de drones, cuentan con una gran variedad de modelos los cuales pueden ser pilotados con sistemas muy similares.

3.2.6. Regulación nacional de aeronaves

La actual regulación de los UAVs en el territorio nacional fue publicada por la DINACIA en setiembre del 2014 [30]. La misma define tres categorías de drones, definidos como *Dispositivos Aéreos operados a distancia*, según su peso sin carga:

- **menores:** hasta 25 kg
- **medianos:** entre 25 y 260 kg
- **mayores:** más de 260 kg

Dentro de las categorías anteriores se diferencian drones dedicados exclusivamente al deporte o la recreación y drones dedicados a actividades remuneradas.

Los modelos de drones manejados en este proyecto entran en la categoría *menores dedicados a actividades remuneradas*. Para estos drones, la actual regulación indica que:

1. La utilización de estos dispositivos se considera incluidas en lo dispuesto en el Art. 122 “Trabajos Aéreos” del Código Aeronáutico Uruguayo, debiéndose dar cumplimiento a los Decretos Nro. 39/977 de 31 de enero de 1977 y 314/994 de 5 de julio de 1994.

2. El interesado deberá contar con seguro de responsabilidad civil o seguro aeronáutico en el caso que se trate de aeronaves.
3. En este tipo de operación, incluso cuando la misma se realice únicamente con Dispositivos Aéreos Operados a Distancia - Menores, los operadores deberán contar con el “Permiso de Operador de Dispositivo Aéreo Operado a Distancia”; sin perjuicio de la correspondiente licencia aeronáutica para el caso de Sistema de Aeronave Pilotada a Distancia (RPAS).
4. Hasta tanto se establezca la reglamentación definitiva y los procesos de certificación correspondientes, la Dirección de Seguridad Operacional determinará para cada caso concreto, las condiciones específicas de operación que garanticen niveles aceptables de Seguridad Operacional, las que serán debidamente notificadas al interesado previo al efectivo inicio de las actividades.

Tener en cuenta que esta es una resolución transitoria expedida por las autoridades reguladoras de Uruguay, mientras la Organización de Aviación Civil Internacional no emite las normas y métodos recomendados (SARPS) para este tipo de actividad. Se estima la emisión de las mismas por el ente internacional a partir del año 2018.

3.3. Modelo de Casos de Uso

3.3.1. Actores del sistema

A continuación, se presentarán los actores involucrados en los casos de uso.

Controlador de vuelo

Los controladores de vuelos serán todos aquellos individuos que sean responsables en la toma de decisiones sobre los drones que estén ejecutando un vuelo planificado. Serán también los encargados de crear las rutas de vuelos y asignarlas a los drones disponibles. Interactuarán con el sistema mediante la interfaz Estación de Control.

Drone

Este actor se corresponde con el drone utilizado para las entregas. Este actor será el encargado de iniciar los casos de uso que se correspondan con notificación del estado del vehículo hacia el sistema. Los drones interactúan con el sistema mediante el envío (notificación de estado) y la recepción (recepción de comandos) de mensajes enviados mediante TCP a puertos predefinidos.

Operario del Drone

El operario del drone será quien interactúe físicamente con el vehículo para dejarlo pronto para ejecutar vuelos planificados, es decir, quien lo enciende y comienza el sistema de abordaje.

Administrador

Los administradores serán los usuarios encargados de mantener la información relevante para la configuración y funcionamiento de los drones.

3.3.2. Casos de uso del sistema

La siguiente imagen muestra el diagrama de casos de uso.

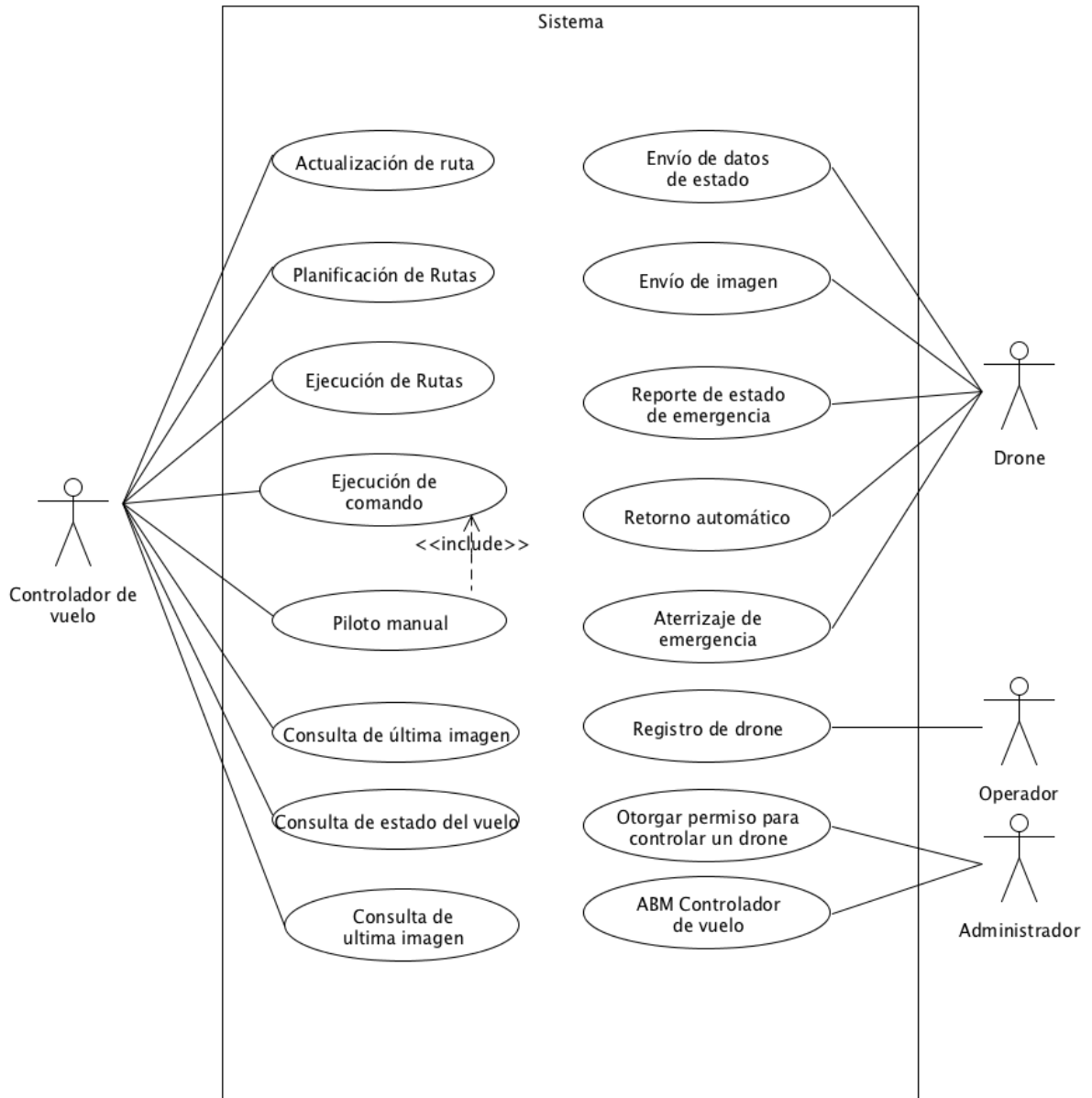


Ilustración 13: modelo de casos de uso del sistema.

A continuación, se detallarán los casos de uso críticos para la arquitectura del sistema.

Planificación de rutas

Las entregas realizadas por los drones deberán estar previamente diseñadas, es para esto que se define el CU “Planificación de rutas”. Una ruta indicará todos los puntos geográficos en donde el drone que la ejecuta deberá entregar una correspondencia. Durante el vuelo, el drone ejecutará la ruta yendo en línea recta entre los puntos sucesivos indicados que forman la ruta. En el momento de crear una planificación el controlador podrá indicar un drone y ejecutarla en ese momento. En cualquier caso, la ruta será almacenada para poder ser utilizada más adelante.

Para simplificar la tarea de indicar la ruta y reducir errores por parte del controlador se solicitará el ingreso de los puntos donde entregar la correspondencia marcándolos en un mapa. El último punto será siempre el lugar de partida y no será seleccionado por parte del controlador.

Ejecución de comando

Durante un vuelo de un drone se ve la necesidad de ejecutar ciertas órdenes para realizar correcciones durante el vuelo. Los comandos soportados son los siguientes:

- **Esperar:** pasa el drone a modo de espera, de este modo el drone se mantiene en su posición hasta que se indique lo contrario.
- **Continuar:** indica al drone que debe continuar con la ruta cargada. Si este ya está en modo de vuelo automático no tiene ningún efecto.
- **Actualizar ruta:** permite al controlador de vuelo modificar la ruta de entrega durante el vuelo. Para este debe indicar la ruta que quiere cargar en el sistema a bordo, seleccionando una de las rutas existentes en el sistema.
- **Mover:** agrega un punto en la ruta que está siendo ejecutada por el drone. Se modifica el plan de vuelo actual de modo que el drone se dirija inmediatamente a este nuevo antes de continuar con la ruta predeterminada. Luego de alcanzar este punto, el drone continúa realizando la trayectoria de la ruta cargada.
- **Aterrizar:** indica al drone que debe aterrizar inmediatamente. Este comando debe ser utilizado con precaución en caso de emergencias.
- **Regresar:** cambia el plan de vuelo haciendo que el drone se dirija al punto de partida.

Los comandos serán transmitidos al drone desde el sistema vía internet, en caso de haber una conexión HTTP entre ambos. En caso contrario se informará al controlador que no hay conexión con el drone en ese momento y no fue posible la ejecución del comando, si desea intentar enviarlo vía SMS.

Registro de Drone

Para que el drone pueda ser utilizado por los controladores de vuelo es necesario hacer el registro del drone. Este registro se realiza desde el sistema a bordo y consiste en el ingreso la información, características y estado del drone. Una vez que se realice este registro el drone podrá comenzar a realizar el envío de datos.

Envío de datos de estado

Para que el controlador de vuelo pueda ver la información de estado del drone, en la estación de control, esta deberá ser comunicada previamente mediante este caso de uso. El drone comunica su estado al sistema. La información enviada contiene, entre otros, la posición en coordenadas geográficas, el nivel de la batería, velocidad, dirección, calidad de la conexión a internet, etc. El sistema notifica esta información a toda estación de control que esté monitoreando el estado de este drone.

Aterrizaje de emergencia

En caso de que la batería alcance cierto mínimo de carga, el sistema indicará automáticamente al drone que aterrice en la posición que se encuentre.

Por otro lado, con el fin que se pueda recuperar el drone, el sistema notificará la posición en la que se produjo el aterrizaje y la carga del mismo. Esta notificación será enviada por todos los medios disponibles (SMS e internet) para asegurar su entrega.

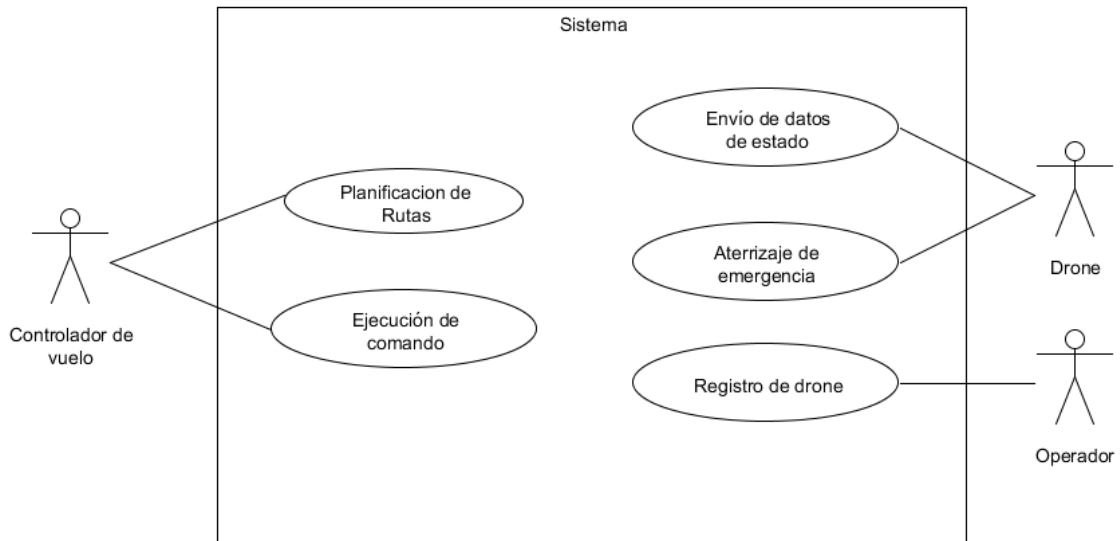


Ilustración 14: modelo de casos de uso críticos del sistema.

Por más información sobre el modelo de Casos de Uso del sistema ver [1]

3.4. Arquitectura

En esta sección se resume la arquitectura del prototipo desarrollado, por más información ver [2]. El siguiente diagrama muestra los distintos componentes físicos que forman parte de la solución propuesta:

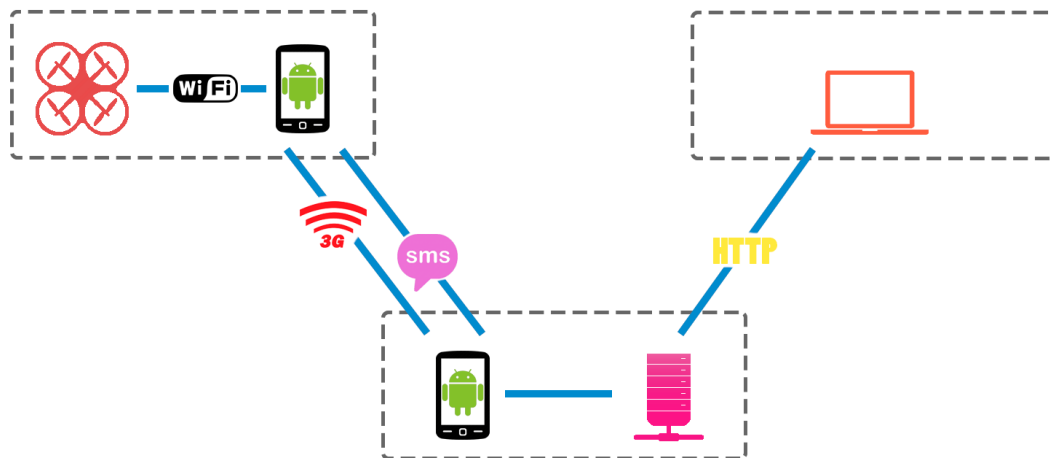


Ilustración 15: arquitectura del sistema.

En la imagen anterior podemos apreciar tres componentes que definen la arquitectura del sistema desde el punto de vista del despliegue de la misma:

3.4.1. Servidor (o sistema) central

El servidor central es el encargado de actuar como intermediario entre las estaciones de control conectadas y los drones existentes. El mismo se comunica bidireccionalmente con los sistemas a bordo de los drones para recabar información de estado y enviar comandos. Para mejorar la fiabilidad de esta comunicación se plantea el uso de internet sobre la red 2G/3G disponible o mediante el envío de mensajes de texto (SMS), por tener este último mayor cobertura. Define la interfaz web usada en las estaciones de control. Además, cuenta con una consola de administración para el ABM de los usuarios del sistema, consultas históricas de los vuelos ejecutados, consulta de los drones registrados en el sistema.

3.4.2. Estación de control

Las estaciones de control son equipos remotos que permiten conocer el estado actual de los distintos drones registrados y además enviar comandos de control a los mismos. Para esto hacen uso del sistema central. Estos equipos ejecutan una aplicación web la cual corre en el servidor central. Otras de las funcionalidades de las estaciones de control son: la planificación de rutas de vuelo, la asignación de un plan de vuelo a un dron y el inicio de una entrega.

3.4.3. Drone con sistema a bordo

El sistema a bordo se encarga de leer el estado del dron y enviarlo al servidor central. Por otra parte, este subsistema recibirá los comandos enviados por las estaciones de control mediante el sistema central, traducirá estos comandos al idioma concreto del dron al que esté conectado y se lo transmitirá por la conexión WiFi. Debido a que no es posible asegurar que se cuenta con conexión en todo momento entre el dron en vuelo y el servidor central, es necesario implementar mecanismos de seguridad en el sistema a bordo del dron. Estos mecanismos de seguridad detectarán cualquier condición que determine un estado crítico del dron durante el vuelo. En caso de entrar en un estado crítico este sistema deberá:

1. Iniciar la rutina de aterrizaje
2. Enviar un mensaje de emergencia al servidor indicando el estado del dron
3. Entrar en modo baliza. En este modo, el sistema de abordo enviará notificaciones periódicas del estado del dron.

De cumplirse alguna de las siguientes condiciones el sistema considera que el dron ha entrado en un estado crítico:

- La batería del dron está por debajo de un mínimo preestablecido.
- La batería del dispositivo móvil a bordo del dron está por debajo de un mínimo preestablecido

Debido a que los drones existentes sólo permiten una comunicación mediante WiFi o radiofrecuencia de corto alcance (hasta 5 Km en los mejores casos), es necesario buscar una alternativa que permita una conexión fiable durante la mayor parte del tiempo de vuelo. Para solucionar esto encontramos las siguientes alternativas:

- Mantener un celular físicamente unido al dron en todo momento que se conecte a internet utilizando la red disponible y se comunique con el dron vía WiFi, radiofrecuencia de corto alcance o bluetooth, o en una conexión cableada (USB OTG) según el caso.
- De forma similar, usar una Raspberry Pi con al menos un módulo GPRS para conectarse a internet.
- En los drones que sea posible, usar su propio CPU para correr nuestro sistema.

La utilización de un celular con sistema operativo Android presenta algunas ventajas por sobre las otras alternativas. Dado que el celular cuenta con su propia batería el sistema a bordo sería capaz de continuar con su ejecución, incluso después de que la batería del dron se haya agotado. Teniendo en cuenta que la autonomía que tienen los celulares es mucho mayor que la de los drones. Un celular normalmente tiene una autonomía de varias horas aún en uso, mientras que los drones en general duran hasta treinta minutos. En caso de emergencia, el sistema a bordo continuará con su ejecución, en una especie de modo de baliza, comunicando el estado del dron (sí puede leerse). Esto daría tiempo a los funcionarios del correo a localizar y recuperar el dron si fuese necesario. Lo anterior representa una ventaja notable por encima de usar una Raspberry pi o el CPU del dron para ejecutar nuestro subsistema. Otra diferencia importante en el uso de un dispositivo Android es la portabilidad de la plataforma. Es decir, el sistema a bordo se implementa en Java, específicamente para Android, y se lo puede correr en casi cualquier dispositivo sin importar el modelo del dron. Esto independiza aún más la implementación con respecto de los drones a utilizar. Adicionalmente, por la experiencia del equipo en desarrollos de aplicaciones móviles Android, resulta más sencillo implementar este subsistema para Android que en C/C++, el lenguaje de programación utilizado generalmente para sistemas embebidos.

Para el desarrollo de este proyecto y del prototipo planteado, utilizar un dispositivo Android parece ser una buena alternativa. Esto nos permitirá realizar un prototipo de forma rápida, independizando los módulos de interacción con los drones de todo el resto del sistema. Sin embargo, hay que tener en cuenta que utilizar el CPU del dron para la ejecución del subsistema a bordo tiene como punto a favor que no se incluye peso adicional durante el vuelo. Esto se torna importante o bien para aumentar el alcance del dron o para poder cargar con mayor carga útil (entiéndase por carga útil, la correspondencia de los usuarios).

3.4.4. Descomposición en subsistemas

En esta sección se presentan los subsistemas identificados durante las etapas de diseño. A continuación, se muestra la descomposición del sistema en los distintos subsistemas identificados.

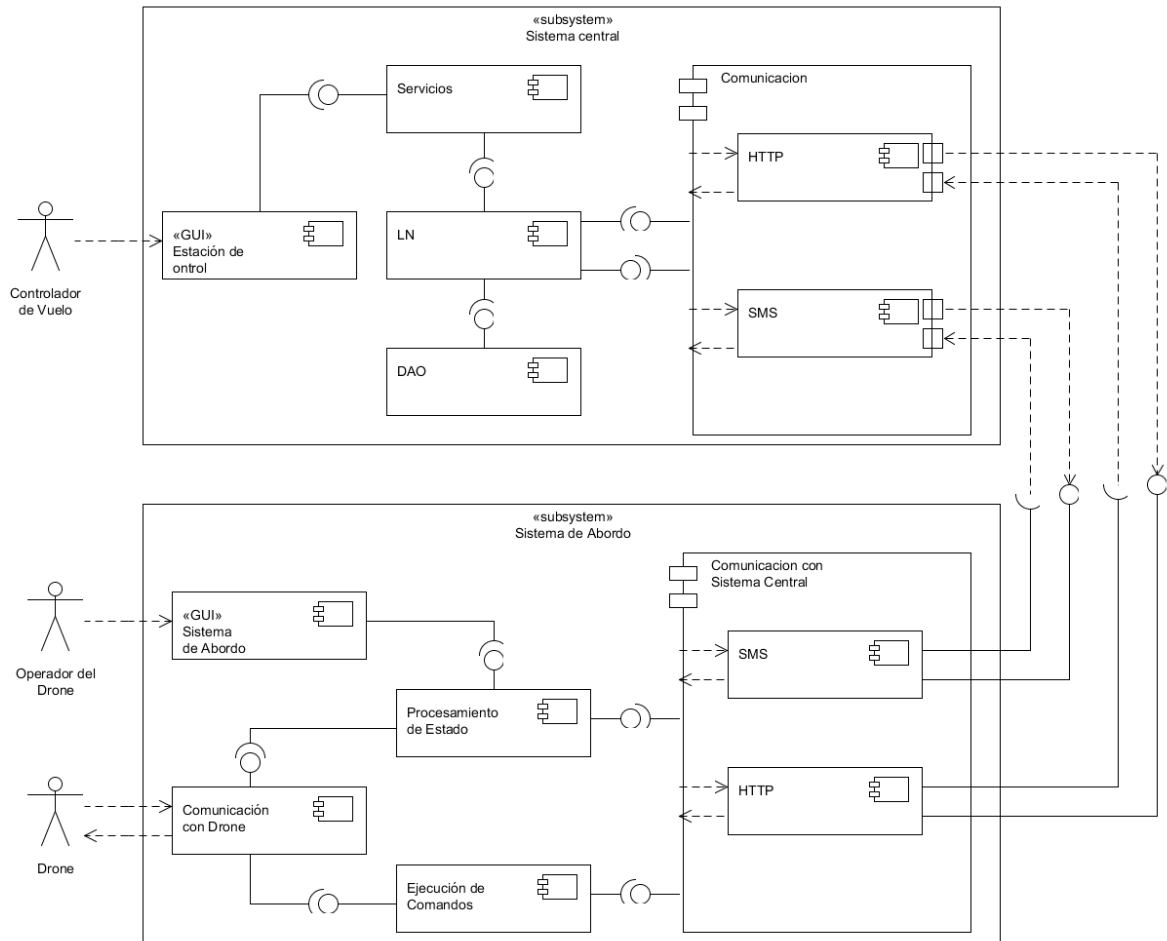


Ilustración 16: descomposición en subsistemas.

Sistema Central

El Sistema Central es el responsable por transmitir la información de estado recibida de cada drone a toda Estación de Control que esté conectada consultando la información de los mismos y de brindar una interfaz para que las estaciones de control puedan comandar los drones que se encuentren en vuelo. La información de estado puede venir de dos conexiones distintas con el drone: HTTP o SMS. La información que el Sistema de Abordo envía por SMS será recibida por un celular¹ conectado físicamente al Servidor del Sistema Central. Este celular enviará la información recibida al servidor al que esté conectado.

¹ Al aumentarse la cantidad de drones registrados, se podría utilizar varios celulares conectados al Servidor para escalar de mejor manera. Se optó por utilizar esta solución para evitar depender de un servicio externo encargado del envío/recepción de SMS por dos motivos:

1. pueden llegar a agregar un tiempo extra de comunicación entre el Sistema Central y dicho servicio lo que entorpece más la comunicación.
2. toda la solución desarrollada se basa en tecnologías open sources, mientras que estos servicios son pagos.

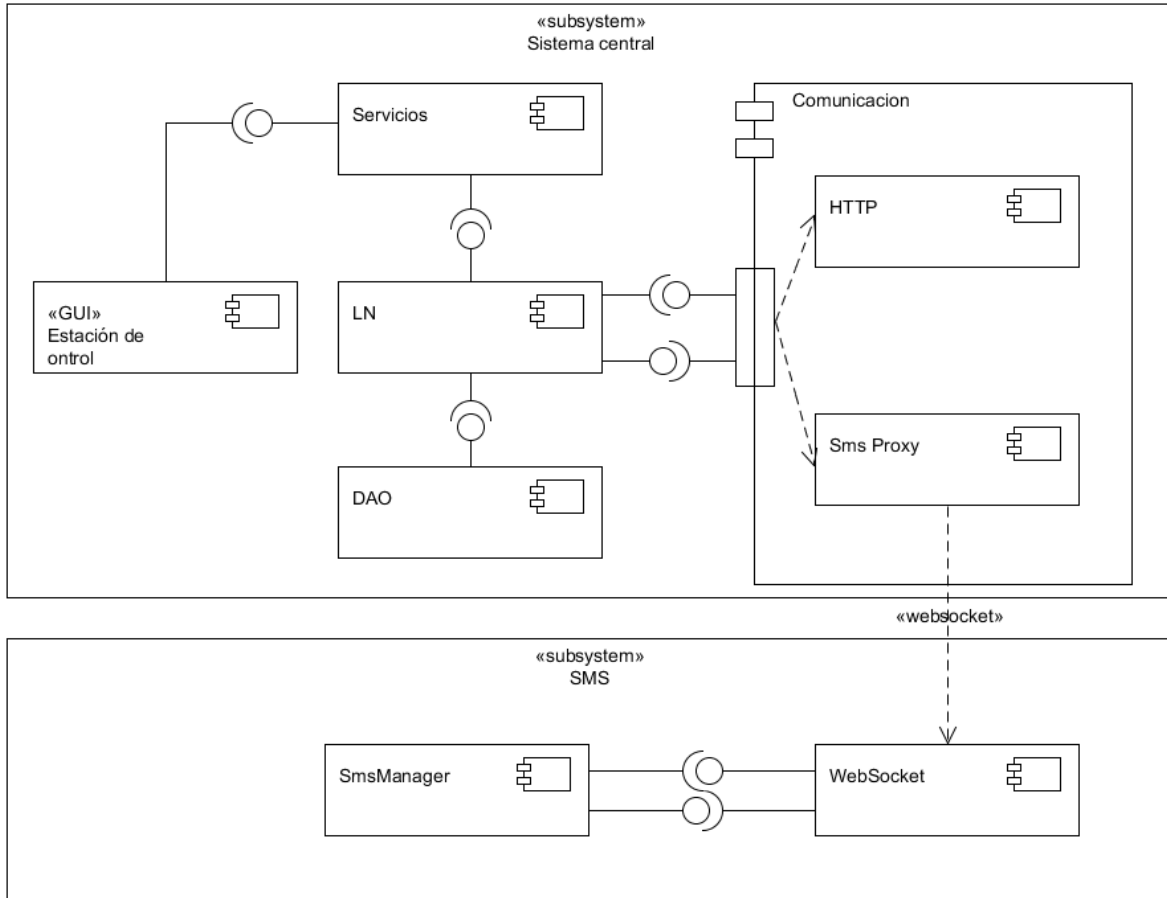


Ilustración 17: diseño del subsistema "Sistema Central"

Los comandos pueden ser enviados hacia el dron por HTTP o SMS. El envío de los comandos por SMS es realizado por el mismo celular anterior.

Toda la información de interés que se transmita desde los drones o desde las estaciones de control son persistidos para poder llevar una traza de los eventos sucedidos en cada ejecución de vuelo.

La comunicación entre la Estación de Control que esté monitoreando un dron en vuelo y dicho dron deberá ser en tiempo real. Este requerimiento plantea varias dificultades técnicas, principalmente las siguientes:

- Conexión continua, asincrónica y bidireccional entre las estaciones de control y el Sistema Central.
- Conexión continua, asincrónica y bidireccional entre el dron y el Sistema Central.

Para resolver el primer punto se utiliza la tecnología de Web Sockets que permite tener un canal abierto entre dos nodos distribuidos conectados por HTTP.

El envío de comandos por SMS y la recepción del estado de los drones enviados por SMS serán procesados por el componente SMS dentro de Comunicación. Este componente se implementa en código nativo Android, para simplificar la tarea de enviar y recibir mensajes de

texto (SMS). La comunicación entre el servidor central y el dispositivo corriendo el módulo SMS se realiza mediante WebSockets, el dispositivo está conectado por cable directamente con el servidor de aplicaciones donde corre el sistema central. Esta conexión permite una comunicación TCP/IP entre ambos nodos.

Sistema de Abordo

El sistema de abordo corre en un dispositivo Android que viaja a bordo del dron. Este subsistema tiene los siguientes objetivos:

- Leer el estado del dron en todo momento, a través de la interfaz que el mismo defina.
- Comunicar periódicamente el estado al sistema central.
- Determinar si el dron se encuentra en un estado crítico y en dicho caso aplicar el protocolo correspondiente.
- Enviar los comandos recibidos desde el sistema central al dron al que se encuentra conectado.

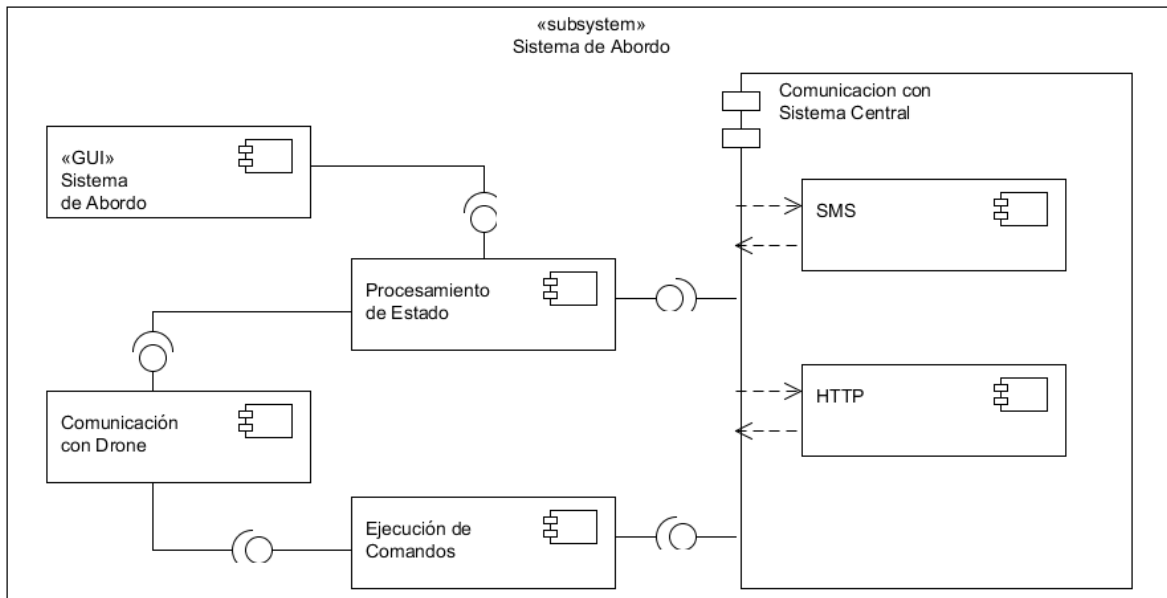


Ilustración 18: diseño del subsistema "Sistema A Bordo".

El sistema a bordo tiene tres interfaces de comunicación con sistemas externos:

- Comunicación TCP vía WiFi con un dron. El protocolo de comunicación depende del dron utilizado. Esto determina la existencia de un módulo especializado para conectarse con cada dron particular.
- Conexión WebSocket vía red de telefonía móvil para envío de estado y recepción de comandos en tiempo real. Esta interfaz se utiliza siempre que haya disponible conexión telefónica en el lugar preciso que se encuentre el dron durante el vuelo.
- Conexión mediante SMS con el sistema central. Brinda un mecanismo de comunicación alternativo al anterior. Se usa este mecanismo dado que cuenta con una mayor cobertura sin embargo es una interfaz no permite asegurar un tiempo de entrega.

El componente “Comunicación con dron” expone una interfaz bien definida que permite:

- Enviar un comando del sistema al dron. Para este caso es responsabilidad de este módulo codificar el comando del sistema en una o varias órdenes nativas del dron con el fin de lograr lo requerido por la estación de control.
- Lectura del estado del dron. En este caso el componente sabe de qué modo el dron presenta su información de telemetría y se encarga de leer esa información, procesarla y enviarla al sistema utilizando el formato predefinido.

Destacar que en caso de contar con un modelo distinto de dron, solo es necesario modificar este módulo para que el sistema sea compatible con el mismo.

3.4.5. Modelo de Despliegue del Sistema

Diagrama de Distribución

A continuación, se muestra el diagrama de distribución física del sistema en términos de cómo se distribuyen sus funcionalidades entre los nodos computacionales.

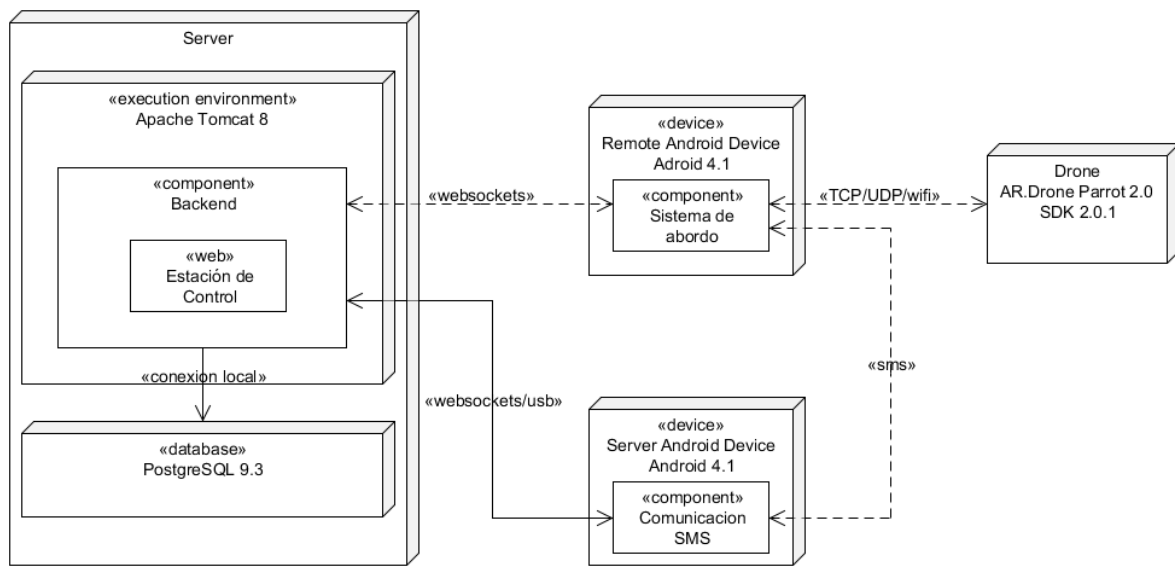


Ilustración 19: Modelo de distribución.

3.4.6. Nodos

Server

El nodo encargado de ejecutar el backend del sistema desarrollado. El mismo brinda la aplicación web Estación de Control la cual es la interfaz por la cual los controladores de vuelo acceden a las funcionalidades del sistema.

Este nodo cuenta con las siguientes características:

- SO compatible con Java EE
- El servidor de aplicaciones es Apache Tomcat 8, compatible con el estándar de WebSockets.
- El servidor de bases de datos es un PostgreSQL versión 9.3

Server Android Device

En este nodo se ejecuta el módulo encargado de enviar comandos y recibir información de estado hacia y desde los sistemas a bordo de los drones. Esta comunicación es realizada mediante mensajes de texto (SMS). Las características de este nodo son las siguientes:

- El sistema operativo es Android con versión entre la 4.0 y la 4.3. Estas versiones simplifican el manejo de SMS dentro de la plataforma ver el riesgo envío de SMS en [9]

Se optó por implementar este módulo en un dispositivo Android por contar con uno para las pruebas y por ser una plataforma open source que permite el desarrollo sobre la misma sin la necesidad de adquirir licencias.

Remote Android Device

Este nodo alberga el sistema de abordaje, cuyo principal objetivo es interactuar con el dron que va adjunto. Las características de este equipo son similares al anterior por los mismos motivos. Hay que agregar que es compatible con el dron utilizado en el desarrollo del proyecto.

Dron

El dron utilizado durante el proyecto. Se empleó un AR.Dron Parrot 2.0 por su bajo costo y rápida adquisición al momento de la compra (disponible para comprar en Uruguay). Cuenta con una interfaz TCP y UDP vía WiFi lo que lo hace compatible con dispositivos con sistema operativo Android.

3.4.7. Conexiones

WebSockets

Las características de esta conexión son las siguientes:

- Implementa el protocolo de WebSockets, ver [RFC-6455]
- La conexión se realiza sobre la red de telefonía celular presente en la zona. La misma puede variar según las capacidades del proveedor (redes 2g, 3g, 4g). En caso de no existir cobertura en el área que se encuentre el dron, esta conexión no será viable.

WebSockets/USB

Esta conexión implementa también el protocolo de WebSockets.

SMS

Esta interfaz utiliza el mecanismo de envío/recepción de SMS de la red de telefonía local. Los mensajes se envían en texto plano utilizando un formato predefinido. Opcionalmente puede definirse un mecanismo de encriptación de la información.

TCP-UDP/WiFi

La conexión entre el dron y el dispositivo móvil que viaja a bordo del mismo se realiza mediante una interfaz TCP o UDP, dicha conexión se realiza sobre la red WiFi definida por el dron y a la cual se conecta el dispositivo móvil.

Capítulo 4 - Diseño Detallado

En este capítulo se presenta el diseño detallado del sistema, enfocado en los casos de uso críticos definidos previamente. Se empieza con el diseño de los casos de uso críticos, para esto se presentan los diagramas de secuencia de los mismos y una explicación para poder entender cómo el sistema interactúa y responde frente a los sucesos externos. Luego se define el modelo de dominio utilizado a lo largo y ancho del sistema. A continuación, se profundiza en el diseño de los subsistemas principales: el servidor central y el sistema a bordo. Para finalizar con esta sección se documentan las decisiones de implementación más significativas del prototipo.

4.1. Casos de uso críticos

En este apartado se detallarán los casos de uso críticos del prototipo desarrollado. Por información más detallada sobre estos casos de uso, así como del resto, ver [3].

A continuación, se describen las instancias de los objetos utilizados en los subsiguientes diagramas de secuencia:

- **Client web:** cliente web que se conecta con el servidor central para obtener información de los drones, iniciar vuelos, controlar drones en vuelo etc. Esta instancia corre en un navegador web del lado del cliente.
- **Servidor central:** instancia del servidor del sistema que corre de forma centralizada.
- **Módulo SMS:** este componente se encarga de comunicar el sistema central y los drones en vuelo haciendo uso de un protocolo de comunicación sobre SMS.
- **Sistema a bordo:** instancia del sistema que corre en el dispositivo móvil, el mismo se encarga de comunicarse con el sistema central para el envío del estado del dron y la recepción de comandos.
- **Drone:** este objeto representa el dron que está ejecutando un vuelo en el momento del caso de uso. El dron puede ser de un modelo u otro. En lo referente a la arquitectura del sistema a bordo, es indiferente el modelo del dron.
- **Drone SDK:** este componente hace referencia a los SDKs respectivos de cada modelo de dron, que es incluido en el sistema de a bordo para poder interactuar con el dron.

4.1.1. Planificación de rutas

El caso de uso inicia en el cliente web. Luego de que el usuario haya definido la ruta geográficamente en el mapa, el cliente envía estos datos al servidor central para crear un nuevo registro en la base de datos con los datos provistos. El servidor retorna un objeto de tipo Result que contiene información sobre si la operación fue ejecutada con éxito. Esto quiere decir que la ruta fue almacenada correctamente a nivel de base de datos. La misma queda entonces disponible para ser ejecutada por un dron más adelante.

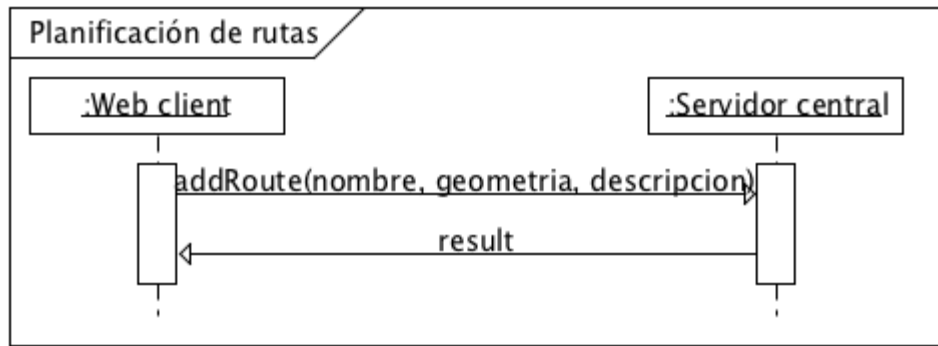


Ilustración 20: diagrama de secuencia del CU Planificación de rutas.

4.1.2. Ejecución de comando

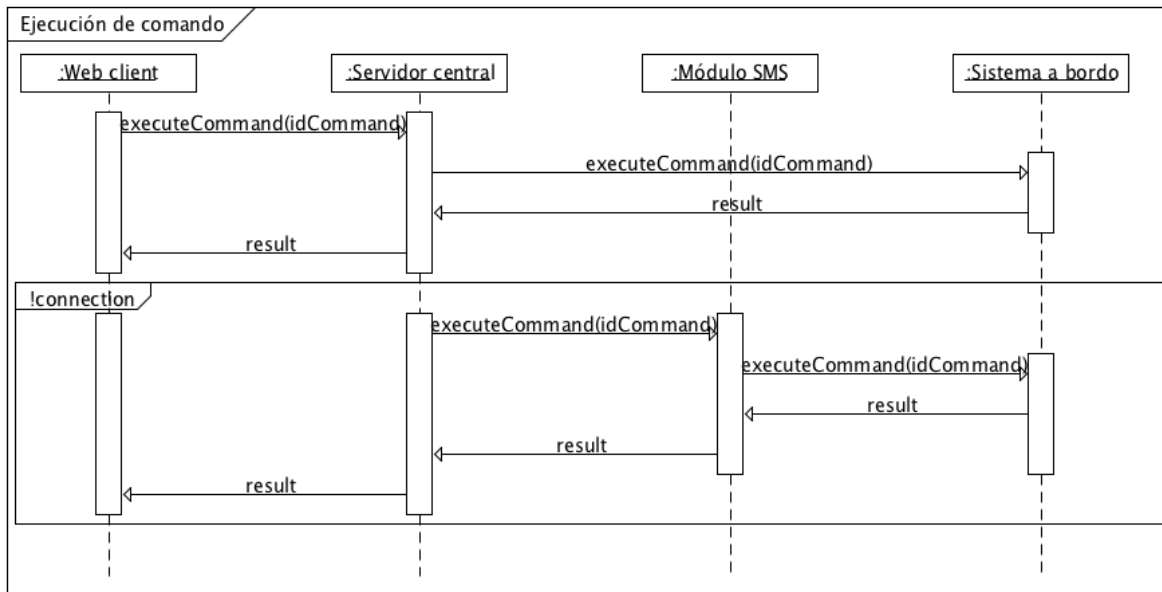


Ilustración 21: diagrama de secuencia del CU Ejecución de comando.

La ejecución de esta funcionalidad involucra todos los módulos del sistema, siendo uno de los más complejos debido a la comunicación entre ellos utilizando las distintas vías soportadas.

El sistema registra el nuevo comando en la base de datos, relacionándolo con el droné indicado y el vuelo actual del mismo. Acto seguido el servidor central comunica el comando al sistema a bordo del droné. Esto se hará de una de las siguientes maneras:

1. **Vía WebSockets:** en primera instancia intenta enviar el comando utilizando el canal de comunicación por WebSockets entre el sistema central y el sistema a bordo. El sistema a bordo del droné recibe el comando en formato JSON a través del cliente WebSocket, el cual notifica del mensaje al componente encargado de decodificar los comandos.

2. **Vía SMS:** en caso de que no haya un canal abierto entre ambos al momento de enviar el comando, el sistema hará uso del módulo SMS para enviar el comando codificado vía mensajes de texto

Habiendo recibido el comando mediante alguna de las vías de comunicación descritas anteriormente, el sistema a bordo instancia el componente encargado de ejecutar el comando recibido. Luego de procesar el comando, el sistema a bordo notifica al servidor central el resultado del mismo.

Finalmente, el servidor central actualiza el estado del comando ejecutado con la información recibida del sistema a bordo del drone. De esta manera el controlador de vuelo es capaz de conocer si el comando que intentó ejecutar fue procesado correctamente en el drone.

4.1.3. Registro de Drone

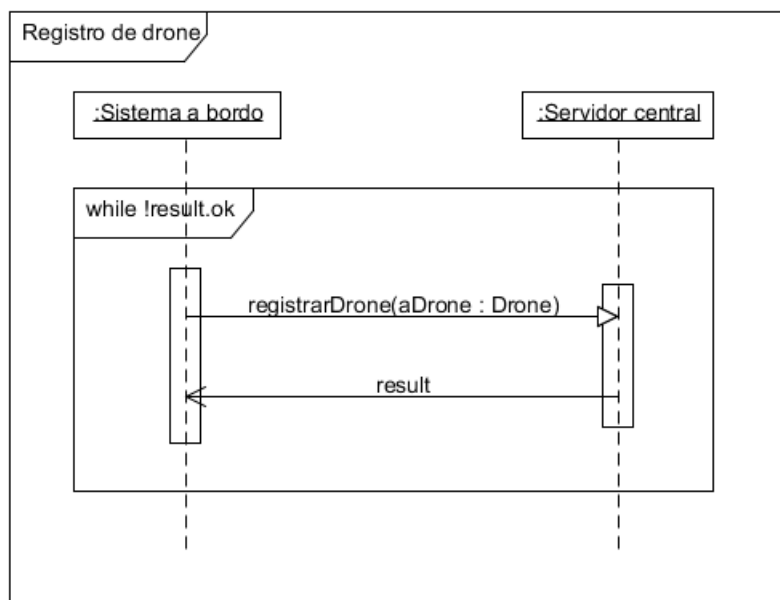


Ilustración 22: diagrama de secuencia del CU Registrar Drone.

Cuando el sistema a bordo inicia, se conecta con el servidor central vía la interfaz WebSocket de forma autónoma. Con el canal abierto, el primero servicio llamado desde el sistema a bordo es el de registro de drone pasando la información necesaria para identificarlo: marca/modelo del drone, versión de firmware instalado, el número de celular conectado, entre otros datos.

Si el drone ya había sido registrado el sistema central solamente actualiza sus datos. En otro caso registra un nuevo drone en la base de datos. Además de esto, el sistema central mantiene una referencia al canal de comunicación WebSocket abierto por este drone de modo de enviar/recibir información y comandos. Con el número de celular actualizado, el servidor central es capaz de enviar comandos mediante mensajes de texto.

4.1.4. Envío de datos de estado

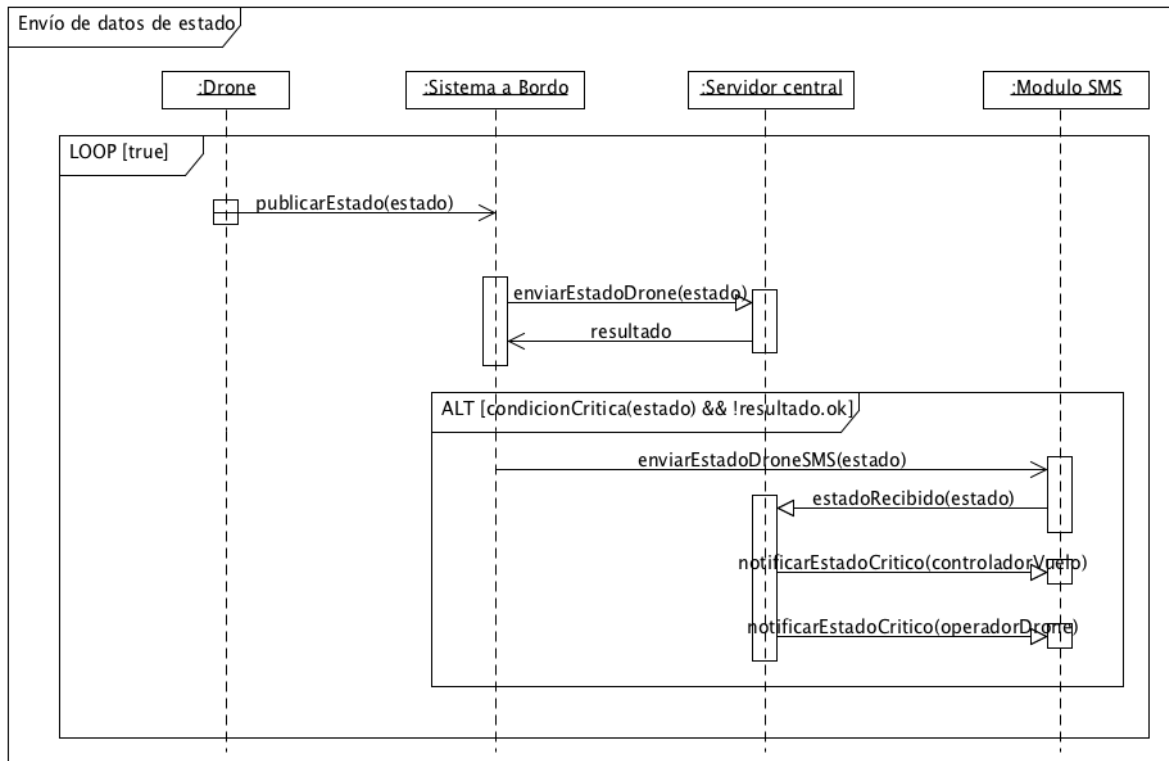


Ilustración 23: diagrama de secuencia del CU Envío de datos de estado.

Los drones en general tienen un subsistema integrado encargado de comunicar información de telemetría del mismo y de recibir parámetros de configuración que alteran el comportamiento del dron. Estos subsistemas se encuentran continuamente enviando y recibiendo información de los sistemas conectados. Los SDK de los distintos modelos de drones utilizan una arquitectura orientada a eventos para la notificación de telemetría. De este modo el código cliente (sistema a bordo) se suscribe para recibir notificaciones cada vez que se reciba telemetría del dron.

4.1.5. Aterrizaje de emergencia

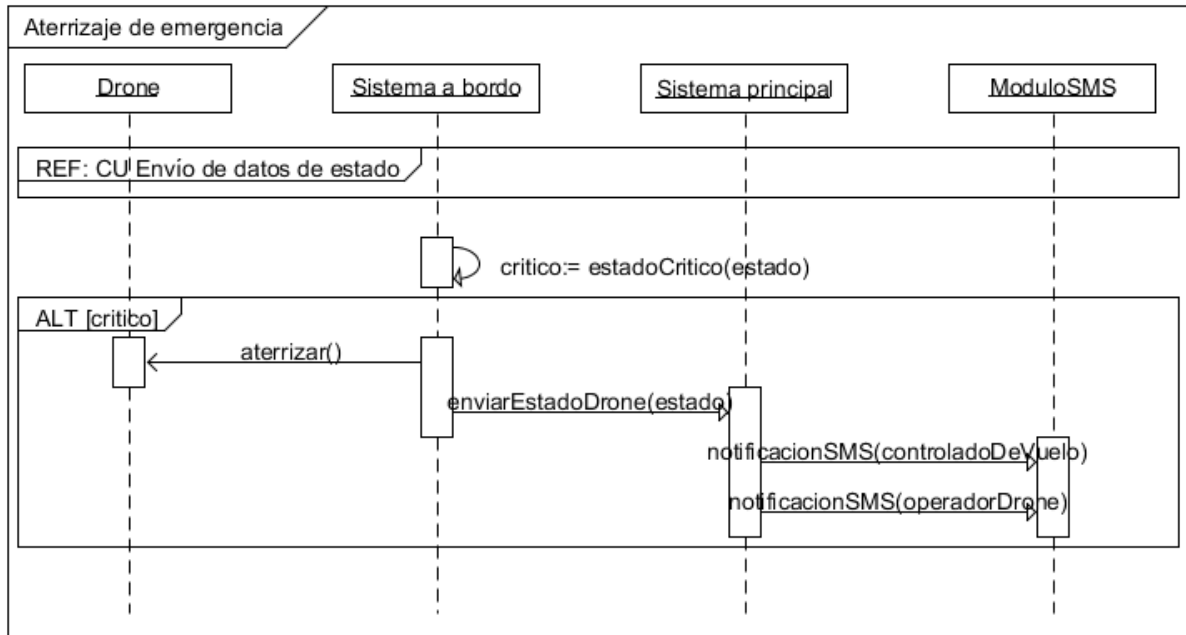


Ilustración 24: diagrama de secuencia del CU Aterrizaje de emergencia.

El sistema a bordo tiene un componente encargado de monitorear el estado del dron de modo de tomar las medidas adecuadas en función de la telemetría recibida sin necesidad de intervención por parte de los controladores de vuelo.

En el diagrama anterior se muestra que, tras recibir nueva telemetría, el sistema a bordo evalúa si el dron se encuentra en estado de emergencia. En caso de ser así, el mismo inicia el aterrizaje y además, notifica el estado actual al servidor central. En caso de no poder enviar el estado al servidor mediante la interfaz WebSocket y tratarse de una condición de emergencia, el sistema a bordo envía el estado vía mensajes de texto.

El servidor central, tras recibir una condición de emergencia, notifica a los controladores de vuelo que estén monitoreando el estado de este dron. En caso de estar registrado el controlador de vuelo, se le notifica enviándole un mensaje de texto a su celular.

4.2. Modelo de dominio

En esta sección se presenta el modelo de dominio de la aplicación, la cual consta de todas las entidades de negocio del sistema. Este modelo es utilizado en el servidor central y es compartido parcialmente y con las modificaciones que sean necesarias con el sistema de a bordo. El siguiente diagrama muestra las entidades de negocio del sistema:

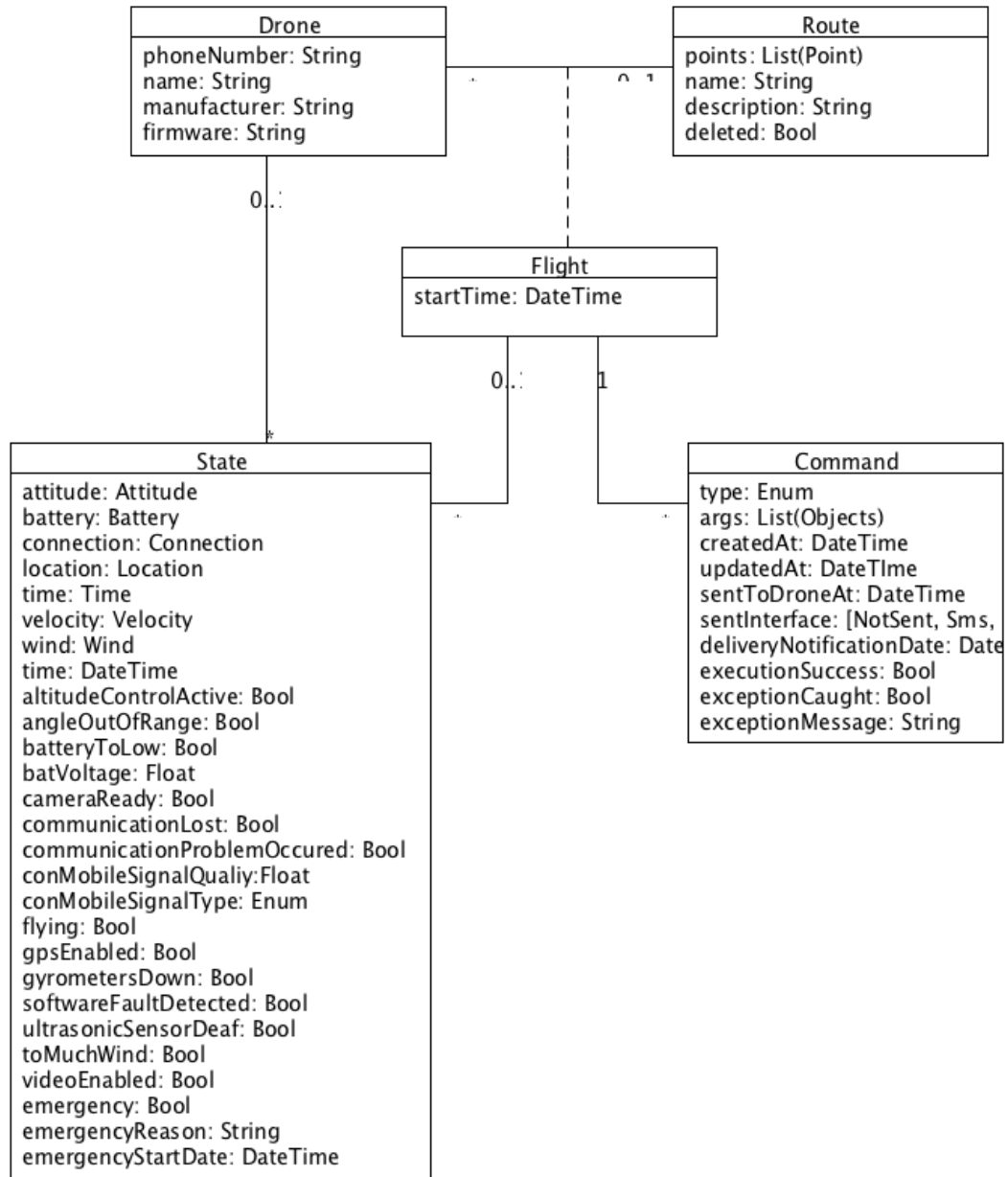


Ilustración 25: modelo de dominio.

4.2.1. Descripción de entidades

uy.pds.model.Command

Representa los comandos que pueden ser enviados a un dron. Estos comandos son ejecutados desde la interfaz web permitiendo al controlador de vuelo manipular el recorrido del dron en tiempo real. Utilizan un objeto JSON como argumento de modo de modelar todos los comandos soportados de forma genérica en una misma clase. Esto simplifica también el envío del comando hacia el Drone haciendo uso de las interfaces de comunicación disponibles (SMS y WebSocket).

Cada comando está asociado con uno y solo un drone, de este modo se sabe en todo momento a cuál drone fue enviado (o se quiere enviar) este comando. Opcionalmente un comando puede estar asociado a un vuelo, con la restricción de que, al momento de crear el comando, el vuelo que se le asocia tiene que ser el vuelo activo del drone relacionado.

uy.pds.model.Drone

Esta clase modela los drones registrados en el sistema. Almacenan la información propia, inmutable a lo largo del tiempo, de cada drone y mantiene relaciones con las distintas entidades que definen el estado momentáneo del drone, esta información varía a lo largo del tiempo.

Los drones son registrados automáticamente cuando se conectan por primera vez con el servidor, y son actualizados en sucesivas conexiones en caso de ser necesario.

uy.pds.model.Flight

Representa los vuelos que están siendo ejecutados o que fueron realizados por los drones. Se mantienen todas las instancias de los vuelos finalizados a modo de histórico, de manera de poder saber lo ocurrido en todo momento en un vuelo particular más adelante.

Esta clase relaciona las distintas entidades que participan y definen un vuelo. Cada vuelo tiene asociado uno y solo un drone que no varía en ningún momento. La ruta que fue ejecutada por ese drone, pudiendo esta ser cambiada durante el vuelo. Además, mantiene una relación con el último estado conocido del drone durante este vuelo.

uy.pds.model.Privilege

Los privilegios son las definiciones de los distintos permisos que tienen sentido en el sistema. Su intención es restringir el acceso a las distintas funcionalidades del sistema de los distintos usuarios del mismo. De este modo, se pretende prevenir que un controlador de vuelo pueda iniciar un vuelo solo usando los drones que están a su alcance (un controlador de vuelo de la sucursal del correo de Rivera no podría iniciar un vuelo con un drone de la sucursal de Colonia)

uy.pds.model.Role

Los roles son una agrupación de privilegios bajo un nombre, esto facilita el proceso de asignación de permisos a los usuarios del sistema.

uy.pds.model.State

Un estado almacena toda la información sobre la condición actual del drone. Los estados son creados por los drones continuamente en el sistema a intervalos de tiempo predefinidos.

Haciendo uso de los últimos estados enviados por el drone, el sistema determina en qué condiciones se encuentra: si la batería restante no es suficiente para completar el vuelo, si hay demasiado viento, etc. De este modo puede notificar a los controladores de vuelo sobre condiciones anómalas, estados de emergencia de los drones en vuelo.

uy.pds.model.User

Almacenan la información de los usuarios que tienen acceso al sistema. Cada usuario tendrá uno (o varios, a definir) roles asociados. Los usuarios solo podrán acceder a las funcionalidades brindadas por el sistema, para las cuales tiene los permisos necesarios.

4.3. Diseño de subsistemas

En esta sección se resume el diseño de los principales subsistemas desarrollados. Por más información ver el documento Modelo de Diseño [3]

4.3.1. Sistema central

El sistema central o PDS por sus siglas en inglés (Postal Delivery System), es bastante sencillo en cuanto a su arquitectura y diseño. El mismo se descompone en las clásicas capas de un sistema de información: presentación (estación de control); capa liviana de servicios; lógica de negocio y acceso a base de datos. Verticalmente hay una capa adicional común a todas las otras compuesta por el modelo de dominio, funciones auxiliares, etc.

Actúa como un canalizador de la comunicación entre las estaciones de control y los drones. Además de almacenar toda la información proveniente de los drones, el servidor almacena las entidades creadas por los usuarios (rutas, vuelos, comandos, etc.) agregándole información adicional de auditoría. Esto permite conocer los responsables de los cambios realizados en los datos del sistema dado que es importante poder responder a quién efectuó tal operación sobre cuál dron para poder asignar las responsabilidades que correspondan.

El sistema central recibe información desde dos fuentes distintas:

- estación de control: esta es la interfaz web definida para que los usuarios puedan interactuar con el sistema. Brinda las funcionalidades para la administración de rutas, drones y vuelos. Accede a los servicios brindados por el sistema central mediante una capa de servicios REST. Se decidió que la interfaz web se comunica mediante servicios en lugar de acceder directamente a la lógica de negocio del backend, para mantenerlos totalmente desacoplados y previendo una posible implementación en otra plataforma, por ejemplo, una aplicación móvil.
- de los drones conectados. Para esto define dos puntos de entrada: una interfaz de servicios sobre el protocolo de WebSockets y otra usando un protocolo de mensajería propio enviados por SMS. Estos dos canales son bidireccionales, en todo momento el tanto el servidor como los drones pueden enviar mensajes a través de ellos. La implementación de la interfaz de comunicación por SMS es hecha en un componente aparte que corre en un dispositivo Android, este sistema es conocido también como PDS-SCI por sus siglas en inglés (Postal Delivery System - SMS Communication Interface).

4.3.2. Sistema a bordo

El sistema a bordo o PDS-DFS por sus siglas en inglés (Postal Delivery System - Drone Flying System), se encarga de realizar tres tareas principales:

1. Controlar el vuelo del dron. Esto es, ejecutar las acciones necesarias como respuesta a los cambios de estado del dron, de modo de que el mismo siga lo más fielmente posible la ruta cargada. Para esto se basa en los datos recogidos de los distintos sensores disponibles. Hace uso del celular y sus propios sensores para suplir alguna carencia del dron, por ejemplo, el uso de la brújula o el GPS si el dron no cuenta con ellos. En todo caso se da preferencia a los datos retornados por los sensores del dron ya que se entiende que estos serán, en general, más precisos que los presentes en el celular. El algoritmo para seguir la ruta de vuelo planificado se detalla en la sección 4.4.1. Navegación autónoma.
2. Notificar activamente el estado del dron al sistema central. El sistema a bordo usa como primer canal de comunicación el establecido mediante WebSockets. Esto se debe a que la comunicación por WebSockets es más veloz y, lo más importante, permite saber si el mensaje fue enviado o no, para tomar las acciones adecuadas en caso de que falle. Si por algún motivo no es posible el envío del estado por este canal, estando en una condición de emergencia, el mismo es enviado por SMS.

- Ejecutar los comandos enviados por los controladores de vuelo desde el sistema central. Se escucha tanto por la interfaz WebSocket como por la de SMS por comandos enviados desde el sistema central. En caso de recibirse algún comando, por alguno de las interfaces anteriores, el sistema a bordo se encarga de decodificar la información recibida y ejecutar las acciones pertinentes.

En el siguiente diagrama se resume las principales interacciones del sistema a bordo para la implementación de las anteriores responsabilidades. Para simplificar el diagrama se omitió el flujo a través de la interfaz SMS.

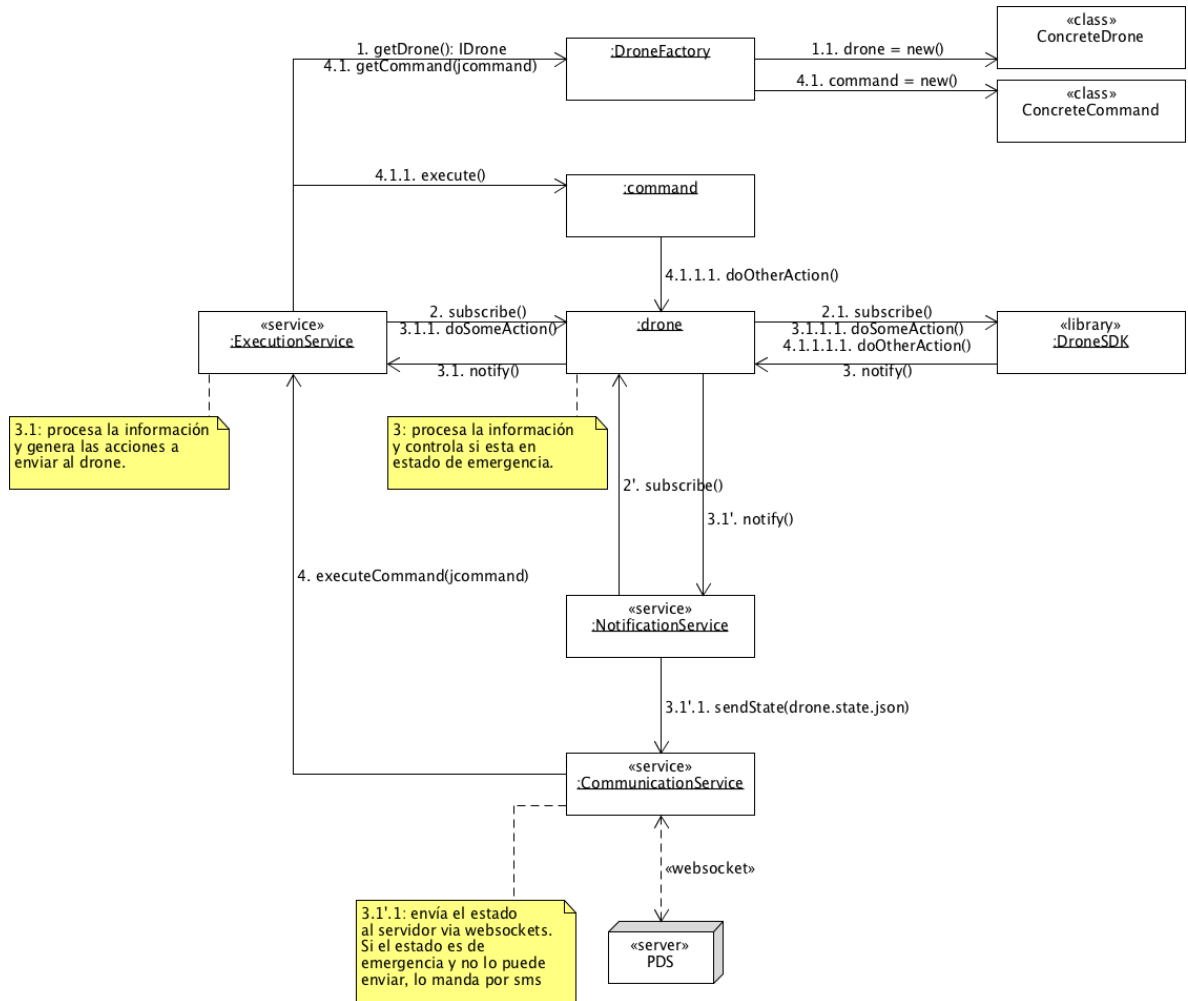


Ilustración 26: diagrama de comunicación resumido de las principales actividades del sistema a bordo.

La clase *ExecutionService* se encarga de monitorear el vuelo del dron y ajustarlo mediante el envío de órdenes al dron a través de la interfaz *IDrone*. La instancia de dron concreta canaliza toda la comunicación entre el resto del sistema a bordo y el SDK del dron. El envío del estado del dron al sistema central es realizado por la clase *NotificationService*, este servicio se encuentra ejecutando en segundo plano continuamente enviando con cierta frecuencia el estado actual del dron. Se ve notificado por el dron cuando cambia su estado, para poder enviar de

forma inmediata la notificación al servidor en caso de estar en emergencia.

Debido a que el monitoreo del dron, el envío del estado al servidor y la recepción de mensajes deben estar siempre disponibles, el sistema a bordo fue diseñado sobre servicios de Android que ejecutan en segundo plano. Estos servicios no finalizan de forma automática, es decir, su tiempo de ejecución es indefinida. Adicionalmente, cuando el celular se enciende, el sistema operativo notifica a nuestra aplicación para iniciar todos los servicios necesarios. Esto se realiza mediante una instancia de *BroadcastReceiver* de Android.

Para entender mejor el diagrama anterior es necesario conocer cómo se diseñó el componente encargado de interactuar con los drones. El sistema a bordo está diseñado de modo de definir de forma genérica una interfaz que permite obtener la telemetría del dron y enviar las operaciones necesarias hacia el dron que permitan la implementación de los distintos comandos definidos en el sistema. Luego, por cada modelo de dron soportado, se debe implementar un módulo o driver encargado de implementar dicha interfaz. De este modo abstraemos la comunicación con los SDK de los distintos drones, manteniendo el resto del sistema a bordo invariante con respecto al dron con el que se esté conectando.

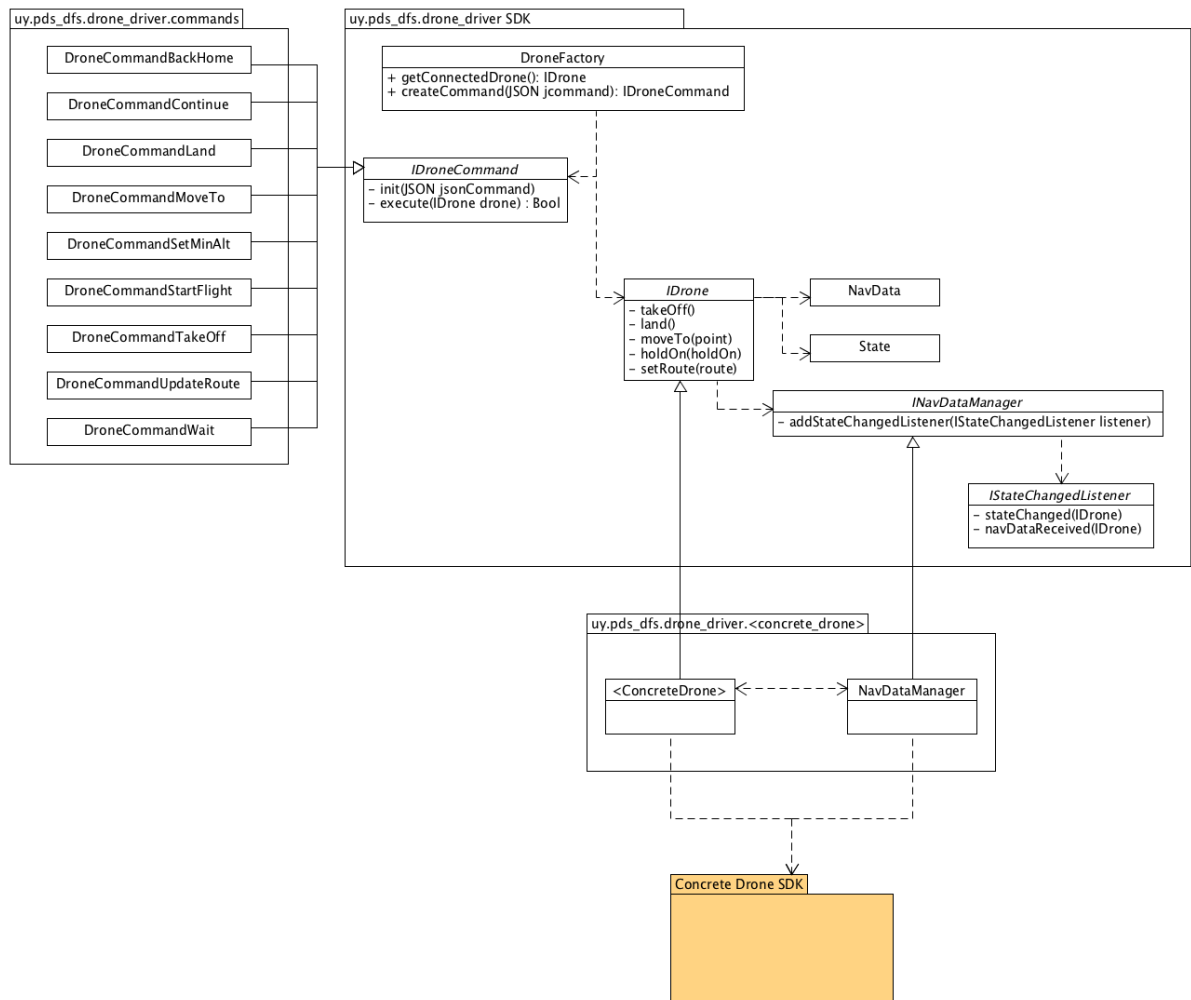


Ilustración 27: diseño de clases del driver de drones.

El diagrama de clases anterior simplifica la arquitectura del drone driver a sus clases más representativas:

- **DroneFactory**: se encarga de retornar la instancia de drone que está actualmente conectado al sistema a bordo. También es el encargado de instanciar cada comando según el JSON recibido del servidor.
- **IDrone**: interfaz de comunicación con los drones. Debe ser implementada por el driver concreto para cada modelo de drone. El mismo define las acciones básicas que pueden ser ejecutadas para una instancia de drone y mantiene una referencia a la información de navegación y estado del drone.
- **INavDataManager**: el módulo drone driver se diseñó con una arquitectura basada en eventos. En el mismo, el código cliente se registra para recibir notificaciones de cuando el estado del drone ha cambiado. Esta interfaz define cómo se registran los observers del estado del drone mientras que la implementación de la misma es la encargada de ejecutar las notificaciones cuando corresponda.

En el paquete de ejemplo `uy.pds_dfs.drone_driver.<concrete_drone>` se implementan las interfaces necesarias `IDrone` e `INavDataManager`. Un paquete similar a este debe ser creado para cada nuevo modelo de drone que sea soportado por el sistema.

4.4. Diseño de algoritmos

4.4.1. Navegación autónoma

El sistema de navegación es el encargado de enviar las órdenes necesarias al drone de modo que este siga lo más fielmente posible la ruta de vuelo cargada. El sistema está diseñado para comandar drones que soporten vuelo automático/semiautomático (no es necesario dar ordenes de pilotaje, solo se envían rutas), como el X8+, y modelos que no cuenten con esta capacidad, como el AR.Drone 2.0. Para los modelos de drones que soportan vuelo (semi) automático, el algoritmo de navegación se ve sensiblemente simplificado.

Dada la ruta de vuelo, como una colección ordenada de coordenadas geográficas, el sistema de navegación ejecuta las acciones necesarias para que el drone siga una trayectoria que conecte cada una de estas coordenadas manteniendo el orden. Al alcanzar la última ubicación el sistema ejecuta la acción de aterrizaje. Para esto el sistema mantiene una referencia a la próxima ubicación de la trayectoria.

Para el caso de drones que no tengan integrado un sistema de navegación (semi) automático, el algoritmo de navegación utiliza como base la información retornada por el GPS y la brújula, incorporados en el drone, para determinar las acciones necesarias que permitan efectuar el vuelo siguiendo la trayectoria debida. En caso de que el drone no cuente con alguno de estos sensores, la información es tomada del celular del sistema. Con esta información el sistema calcula el ángulo entre la orientación actual del drone, o *heading*, con respecto de la ubicación de destino. Este ángulo es conocido como *bearing*.

El valor de *bearing* determina si la nariz del drone apunta en dirección al siguiente punto de la ruta. En caso afirmativo, el drone está en condiciones de avanzar. El sistema calcula el *bearing* como la diferencia entre el *heading* del drone y el ángulo con respecto al norte del destino. La siguiente imagen ilustra la relación entre *heading* y *bearing*.

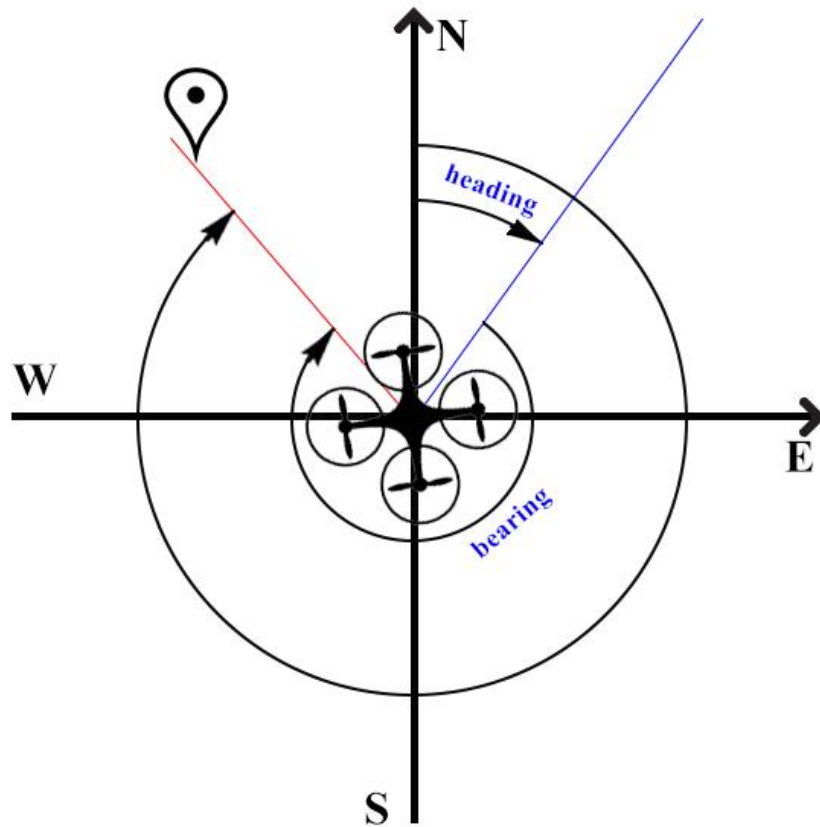


Ilustración 28: relación entre heading y bearing.

Los pasos del algoritmo de navegación se resumen a continuación:

1. Iniciar el vuelo
2. Establecer altura mínima de vuelo
3. Esperar a recibir una nueva lectura de los sensores de telemetría (GPS, brújula, etc.)
4. Calcular la distancia hacia el siguiente punto de la trayectoria
5. Si esta distancia es menor que un mínimo predeterminado
 - a. El dron alcanzó un punto de la ruta
 - b. Si hay más puntos por recorrer
 - i. Seleccionar el siguiente punto de la ruta
 - ii. Continúa en el paso 2
 - c. Si no
 - i. Aterrizar
 - ii. Fin del algoritmo
6. Si el dron soporta vuelo (semi) automático
 - a. Enviar el comando *moveTo* con las coordenadas del siguiente punto de la ruta
 - b. Continúa en el punto 2
7. Si no
 - a. Calcular el bearing al punto de destino

- b. Si bearing es mayor al ángulo de estabilidad predefinido
 - i. Aumentar la velocidad de rotación en el sentido adecuado
 - ii. Continúa en el punto 2
- c. Si no
 - i. Actualiza la velocidad del dron para que avance
 - ii. Continúa en el punto 2

4.4.2. Detección de estado de emergencia

El sistema a bordo recopila todos los cambios en el estado del dron por suscripción a los eventos disparados por el SDK del mismo. Utiliza toda esta información para calcular un factor de riesgo para el momento actual, en caso de este factor supere un umbral preestablecido el sistema a bordo entra en modo de emergencia. Se definen dos sabores de estado emergencia: peligro y crítico. Los datos que se usan para determinar si el dron se encuentra en un estado de emergencia son los siguientes:

- carga actual de la batería, el sistema asegura que la carga de la batería nunca llegue a un mínimo preestablecido con el dron en vuelo. En este caso el sistema pasa a un estado de emergencia crítico.
- calidad de la señal del GPS, si el sistema no puede determinar su posición con una precisión mínima (configurable) no es viable continuar con el vuelo. En este caso el sistema pasa a un estado de emergencia crítico.
- el sistema estima qué carga de la batería es necesaria para completar la ruta de vuelo en función del consumo hasta el momento, si esta estimación indica que se necesita mucha más carga de la disponible, el sistema determina un estado de emergencia.
- velocidad del viento, el sistema controla que la velocidad del viento no supere la resistencia del dron. Para esto, según el modelo de dron, se definen dos máximos para la velocidad del viento. El menor de los dos indica un estado de peligro si la medición del viento es superior a este. El mayor de los máximos se usa para determinar un estado de emergencia crítico.

En este modo el dron realiza los siguientes pasos:

1. si el estado de emergencia es crítico:
 - a. inicia la secuencia de aterrizaje
2. sino (el estado de emergencia es peligro)
 - a. inicia el regreso a la base (RTL)
3. notifica al sistema central vía WebSockets
 - a. si el envío de este mensaje falla, inmediatamente manda un SMS con información resumida sobre el estado del dron y el motivo de la emergencia
4. el sistema central notifica a los operadores de vuelo asignados al dron sobre la emergencia vía SMS. Como no se implementó el control de acceso de los controladores de vuelo a los drones, el sistema notifica a todos los números de celulares registrados
5. el sistema a bordo continúa enviando el estado al servidor central periódicamente
6. el sistema a bordo desactiva la ejecución de comandos dado que en un estado de emergencia el dron no debería volver a volar.

4.5. Decisiones de implementación

4.5.1. Conexión entre sistema a bordo y sistema central vía WebSockets

Para mantener un canal de comunicación full dúplex entre el sistema central y los drones conectados, se decidió utilizar el protocolo de WebSockets. Los principales motivos que nos llevaron a optar por este protocolo fueron, en primer lugar, cumplir con los requerimientos del sistema y en segundo lugar por su gran compatibilidad con las distintas plataformas involucradas.

4.5.2. Envío del estado mediante mensajes SMS

Para el envío del estado utilizando SMS se tuvo que definir un protocolo de comunicación el cual cumple con las siguientes condiciones:

- Permitir al servidor central identificar inequívocamente de que dron proviene el mensaje recibido.
- Reconstruir la información útil del mensaje, es decir la información acerca del estado del dron a partir de uno o varios SMS. Esto se debe a que la longitud en caracteres de cada mensaje SMS está acotada, por lo que el envío de la información de estado del dron probablemente sería dividido en varios mensajes SMS.

Para cumplir con estas condiciones se definió el formato de mensajes enviados sobre el protocolo SMS de la siguiente manera:

- El mensaje enviado es un objeto JSON
- Permite identificar el dron que envió el mensaje
- Cuenta con una clave que identifica cada mensaje enviado por el dron. Cuando se envía el estado del dron al servidor, este estado es dividido en varios mensajes SMS, cada uno de estos mensajes tienen el mismo valor de clave.
- La información útil, es decir, el estado del dron, es enviado codificada en base64.
- Un campo para determinar el orden exacto en el que deben concatenarse los payloads de los mensajes SMS para reconstruir el mensaje original.
- Un campo que determina la cantidad total de mensajes SMS que debe esperar recibir el servidor central antes de intentar reconstruir el mensaje original.
- El largo total del mensaje queda determinado por la capacidad del protocolo SMS que se esté utilizando en el momento. Este puede ser de 140 caracteres [35].

De esta forma, el formato exacto del mensaje es:

```
{
  "id": 9999,
  "n": "01",
  "t": "09",
  "d": "Abc1231274asdav72312..."
}
```

El identificador del dron se corresponde con el número del cual se envió el mensaje de texto.

El total de caracteres adicionales agregados para armar el mensaje, eliminando los blancos innecesarios, es de 34 caracteres, es decir un 24% de la capacidad del mensaje.

4.6. Capa de servicios

En esta sección se presenta un resumen de la capa de servicios expuestos por el servidor central para comunicarse con los demás subsistemas. Por más información ver [4].

4.6.1. Estación de control

La capa de servicios consumida por el frontend del sistema fue diseñada siguiendo los principios REST. A continuación, se listan los servicios expuestos por el servidor

- Listar drones en vuelo: Retorna la lista de drones que se encuentran en vuelo en el momento de la petición
- Estado de dron: Retorna los datos del dron junto con el último estado conocido del mismo.
- Lista de comandos ejecutados para el vuelo actual del dron: Retorna la lista de comandos ejecutados durante el vuelo actual de un dron.
- Ejecutar comando: Permite ejecutar un comando para un dron en vuelo. El comando debe ser alguno de los soportados por el sistema. Retorna un JSON con el comando creado.
- Listar rutas: Retorna todas las rutas habilitadas dadas de alta en el sistema.
- Crear ruta: Crea una nueva ruta en el sistema indicando al menos nombre, ruta geográfica.
- Obtener ruta: Retorna la información de la ruta indicada.
- Modificar ruta: Edita la información de la ruta dada. Retorna el estado de la ruta tras la actualización.
- Eliminar ruta: Elimina la ruta indicada. Se hace una baja lógica de la ruta en lugar de eliminarla físicamente del sistema.
- Listar drones: Retorna la lista de drones en vuelo o listos para iniciar una nueva entrega. De este modo la estación de control puede iniciar un vuelo para una ruta dada o actualizar la ruta de un dron en vuelo.
- Iniciar vuelo: Inicia una entrega por el dron indicado y la ruta dada. Retorna los datos del vuelo creado.
- Actualizar ruta actual de dron en vuelo: Este servicio permite cambiar la ruta de entrega que está ejecutando un dron dado por la ruta indicada.

4.6.2. Sistema a bordo

El sistema a bordo se comunica usando WebSockets sobre el cual se definió el protocolo de comunicación entre el sistema a bordo y el servidor central. A continuación, se listan los mensajes soportados por ambas partes.

Mensajes aceptados por el servidor central

Todos los mensajes recibidos deben ser objetos JSON con el siguiente formato:

```
{  
  "createdAt": "[FECHA_CREADO]",  
  "id": "[ID]",  
  "name": "[NOMBRE]",  
  "payload": { }  
}
```

El identificador es utilizado para evitar procesar mensajes que puedan recibirse de forma duplicada. Por ejemplo, al enviar un comando por WebSocket y SMS a la vez.

El campo *name* identifica el tipo de mensaje, el servidor central usa este valor para determinar qué acción realizar. Los tipos de mensajes soportados son los siguientes:

- **register**: usado por el sistema a bordo para registrar el dron en el sistema, este debe ser el primer mensaje enviado por el dron. El payload de este mensaje contiene toda la información del dron: datos invariantes, estado y telemetría.
- **status_notification**: estos mensajes son enviados por el sistema a bordo para notificar el estado actual del dron
- **command_received**: tras la ejecución de un comando, el sistema a bordo notifica al servidor central el resultado del mismo. El resultado indica si el comando fue ejecutado correctamente o no

Mensajes aceptados por el sistema a bordo

El servidor central envía dos tipos de mensajes al sistema a bordo mediante la interfaz WebSocket. Los mismos tienen el siguiente formato:

```
{
  "name": "[NOMBRE]",
  "payload": { }
}
```

A continuación, se listan los tipos de mensajes enviados desde el servidor central al sistema a bordo:

- **response**: este tipo de mensaje es enviado como respuesta a un mensaje recibido desde el sistema a bordo. Sirve para notificar al sistema a bordo el resultado del procesamiento de su solicitud.
- **execute_command**: comandos enviados por el controlador de vuelo a través del servidor central. El payload de este mensaje es el comando en sí, los mensajes de este tipo se representan de la siguiente forma:

```
{
  "name": "execute_command",
  "payload": {
    "arguments": { },
    "commandId": "[ID]",
    "type": "[NOMBRE_COMANDO]"
  }
}
```

Los comandos soportados son los siguientes:

- **BackHome**: aborta la ruta de vuelo actual e inicia el retorno del dron al punto de partida. No recibe argumentos.
- **Continue**: indica al dron que debe continuar con la ruta de vuelo con normalidad. No recibe argumentos.
- **Land**: aborta el vuelo actual del dron e inicia el aterrizaje del mismo. No recibe argumentos.
- **MoveTo**: agrega un punto intermedio en la ruta del dron. El dron inmediatamente actualiza su plan de vuelo para dirigirse a este punto.

Entrega postal usando UAVs

- **SetAltitude:** modifica la altitud de vuelo del drone.
- **StartFlight:** inicia un nuevo vuelo, recibe la ruta que debe realizar el drone. Luego de cargar la ruta en el drone, inicia el vuelo automáticamente.
- **TakeOff:** inicia el despegue del drone. No recibe argumentos.
- **UpdateRoute:** modifica la ruta actual de vuelo. Puede ser recibido durante el vuelo, en este caso se cambia el plan de vuelo del drone.
- **Wait:** indica al drone que mantenga su posición hasta recibir el comando `Continue`.

Capítulo 5 - Pruebas y Resultados

Este capítulo presenta los resultados obtenidos al realizar la evaluación de las capacidades técnicas de los drones adquiridos, esto es, la medición de ciertas características de interés para el presente proyecto (velocidad, autonomía, alcance, etc.). Por otro lado, se resumen las evaluaciones realizadas al sistema desarrollado.

5.1. Recopilación de datos y métricas de vuelo

5.1.1. Descripción

Realizar pruebas en localidades alejadas para recabar parámetros y restricciones de vuelo y recepción de internet. Identificar posibles problemas técnicos.

5.1.2. Objetivo

Determinar qué buffer de cobertura puede alcanzarse con el drone seleccionado, las condiciones del medio ambiente y capacidad de carga.

Determinar calidad mínima de señal de internet móvil para el correcto funcionamiento del software.

5.1.3. Resultados Esperados

Los resultados esperados para esta actividad son obtener respuestas a las siguientes incógnitas:

- 1) Alcance máximo del drone.
- 2) Velocidad máxima.
- 3) Duración estimada de la batería.
- 4) Capacidad máxima de carga.
- 5) Tipo de conexión de datos móviles óptima para el envío ininterrumpido del estado del drone al servidor.

5.1.4. Resultados Obtenidos

A continuación, se resumen los datos más importantes obtenidos durante estas pruebas, por más información sobre los resultados de esta actividad ver el documento anexo [5].

Parrot AR.Drone 2.0

Se identificaron problemas con la sensibilidad de la brújula del Android que se encuentra a bordo, lo que hace difícil la ejecución de la ruta de modo correcto. Con respecto a las métricas, en resumen, los resultados obtenidos son:

- 1) El alcance promedio logrado fue de 2,07 Km.
- 2) La velocidad máxima conseguida fue de 16.82 Km/H.
- 3) La batería tiene una duración promedio de 9 minutos y 7 segundos, pero esta se ve afectada significativamente según la cantidad de peso que se le agregue.
- 4) La capacidad máxima de carga del drone es de 264 gramos. Con pesos mayores el drone no consigue el despegue, con pesos menores pero cercanos, consigue el despegue, pero se nota dificultad en el vuelo y consumo de batería excesivo.

A continuación, se muestran en detalle datos obtenidos en las pruebas realizadas para el modelo Parrot AR.Drone 2.0.

Entrega postal usando UAVs

Autonomía

En la Tabla 1, vemos la duración del vuelo, con una batería cargada completamente, en función de la carga adicional:

Tabla 1: autonomía de Parrot AR.Drone 2.0 en relación a la carga transportada.

Peso adicional (grs)	Tiempo 1	Tiempo 2	Tiempo 3	Promedio
0	00:09:25	00:09:01	00:08:56	00:09:07
100	00:07:15	00:06:10	00:07:05	00:06:50
120	00:06:08	00:05:31	00:06:01	00:05:53

Alcance y velocidad

Vuelo normal utilizando la carcasa de exteriores. Se realizaron vuelos en segmentos de 50m (ida y vuelta) lo que genera un esfuerzo adicional para cambiar la dirección de vuelo. Los resultados obtenidos se muestran en la Tabla 2.

Tabla 2: alcance y velocidades de Parrot AR.Drone 2.0.

Distancia (Km)	Velocidad promedio (Km/h)	Velocidad máxima (Km/h)	Tiempo
2,1	13,26	16,82	00:09:34
1,9	12,86	15,43	00:08:52
2,2	14,29	16,79	00:09:14

Carga

El comportamiento del drone Parrot AR.Drone 2.0 se ve afectado sensiblemente con cargas mínimos. Es posible realizar vuelos con cargas menores a 200 gramos. En la Tabla 3 se muestra el comportamiento de este drone con distintas cargas.

Tabla 3: comportamiento de Parrot AR.Drone 2.0 con carga adicional.

Peso adicional (grs.)	Comportamiento
80	Despega con normalidad
100	Despega con normalidad
150	Despega con dificultad

200	Despega con dificultad
240	Despega con mucha dificultad (apenas levanta y vuelve a caer)
264	No despega

3D Robotics X8+

Con este modelo se solucionó el problema de la brújula, ya que el 3D Robotics X8+ tiene una brújula integrada con mayor precisión. En resumen, las métricas obtenidas, son las siguientes:

1. El alcance promedio en una trayectoria recta sin carga adicional es de 6,42 Km.
2. La velocidad máxima obtenida 28,09 Km/hr.
3. La batería tiene una duración promedio de 19 minutos y 30 segundos, la batería se ve afectada según la cantidad de peso, pero a diferencia del modelo anterior la carga debe ser más significativa para lograr una reducción de la autonomía.
4. La capacidad máxima de carga del dron es de 800 gramos. Con pesos mayores el dron logra despegar, pero realiza un aterrizaje de emergencia segundos después.

A continuación, se muestran en detalle datos obtenidos en las pruebas realizadas para el modelo 3D Robotics X8+.

Autonomía

En la Tabla 4, vemos la duración del vuelo del dron 3D Robotics X8+, con una batería cargada completamente, en función de la carga adicional.

Tabla 4: autonomía de 3D Robotics X8+ en relación a la carga transportada.

Peso adicional (grs)	Tiempo 1	Tiempo 2	Tiempo 3	Promedio
250	00:20:11	00:19:01	00:019:20	00:19:30
600	00:15:01	00:14:37	00:15:15	00:14:57
900	00:12:40*	00:10:12*	00:10:52*	00:11:14*

* Son valores estimados según la cantidad de batería utilizada, dado que no se pudo descargar completamente la batería, debido al aterrizaje forzado.

Alcance y velocidad

Vuelo normal utilizando la carcasa de exteriores. Se realizaron vuelos en segmentos de 50m (ida y vuelta) lo que genera un esfuerzo adicional para cambiar la dirección de vuelo. Los resultados obtenidos se muestran en la Tabla 5.

Tabla 5: alcance y velocidades de 3D Robotics X8+.

Distancia (Km)	Velocidad promedio (Km/h)	Velocidad máxima (Km/h)	Tiempo
2,47	20,78	26,78	00:07:10
3,22	21,35	27,94	00:08:54
2,70	18,26	24,47	00:09:01
2,53	18,43	24,23	00:08:12
2,79	19,76	27,1	00:08:19
2,48	20,94	26,54	00:07:12
3,15	21,12	28,09	00:08:48

De las tablas de autonomía y alcance anteriores, podemos estimar el alcance máximo del drone. Notar que en las pruebas de alcance no se voló el drone hasta que quedase sin batería, en su lugar se hicieron varias pruebas. Promediando se tiene que el alcance máximo del drone es de 6,42 Km en trayectoria recta.

Carga

El comportamiento del drone 3D Robotics X8+ se mantiene constante con cargas superiores a las del Parrot AR.Drone 2.0. Es posible realizar vuelos con cargas menores a 1 Kilogramo. En la Tabla 6 se muestra el comportamiento de este drone con distintas cargas.

Tabla 6: comportamiento de 3D Robotics X8+ con carga adicional.

Peso (en grs.)	Comportamiento
250	Despega con normalidad
400	Despega con normalidad
600	Despega con normalidad
800	Despega con dificultad
900	Aterriza luego de unos minutos

Red de telefonía móvil nacional

La red de telefonía móvil brindada por Antel abarca la mayor parte del territorio nacional, con protocolos lo suficientemente rápidos como para permitir una comunicación continua con el servidor remoto. La red 3.7 G de Antel está disponible en la mayoría del país [20] y la velocidad de este tipo de red es 14-86 Mb/s de bajada y 2-22 Mb/s de subida [5].

5.2. Pruebas del sistema

5.2.1. Simulador de vuelo X8+

Antes de comenzar con las pruebas del sistema empleando el drone X8+, se empleó un simulador que provee ArduPilot [10]. El ambiente SITL para Linux que puede simular distintos tipos de vehículos. El simulador SITL permite ejecutar Planeadores, Copters o Rovers sin hardware alguno. Este simulador es un ejecutable nativo para Linux que permite probar el comportamiento del código desarrollado sin contar con hardware.

Durante el desarrollo del driver para el drone X8+ de 3D Robotics, parte del sistema de a bordo, se utilizó este simulador para probar el comportamiento de la aplicación. Esto nos permitió detectar varios errores de programación. Adicionalmente, ayudó durante el proceso de desarrollo del driver brindando una interfaz de pruebas ágil para determinar el formato de los mensajes intercambiados entre el drone y la aplicación cliente. Esto compensó la falta de documentación de DroneKit para Android.

El ambiente de pruebas se compone básicamente de los siguientes sistemas:

1. Sistema Mac OS 10.11
2. Simulador de Android (Genymotion): nodo ejecutando el sistema de a bordo.
3. Máquina virtual VirtualBox corriendo Ubuntu 15.04: nodo ejecutando el simulador SITL de ArduPilot configurado con el Copter modelo X8+
4. Servidor central hospedado en [11] OpenShift [12]

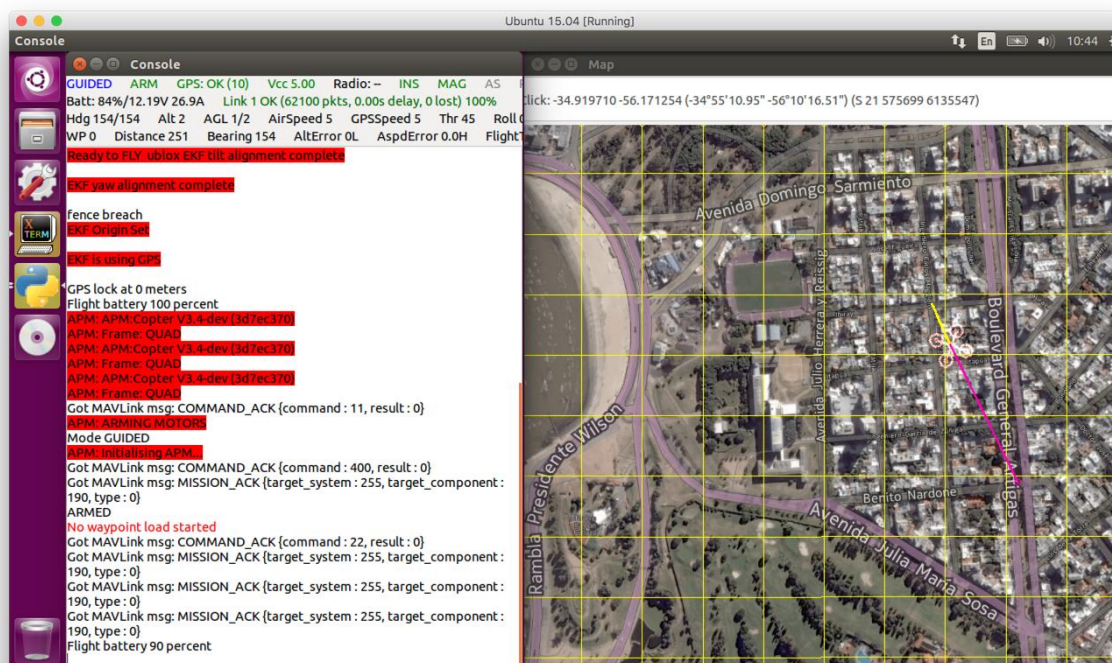


Ilustración 29: simulador SITL ejecutando un vuelo.

El simulador SITL cuenta con las siguientes herramientas:

1. Consola de comandos conectada al drone. Se utiliza el protocolo de comunicación MAVLink para comunicarse con el drone.

2. La consola de control que lista todos los cambios en el estado del dron, incluyendo los comandos recibidos y parte de la telemetría (posición, carga de la batería, etc.).
3. El mapa con la posición del dron. Muestra el dron en movimiento y los puntos de la ruta en ejecución. Además, permite la ejecución de algunos comandos como: RTL, actualización y visualización de la misión, entre otros.

Con este ambiente armado se probaron los siguientes aspectos del sistema:

1. Recopilación de información de telemetría del dron X8+
2. Algoritmo de ejecución de rutas.
3. Envío y ejecución de comandos hacia el dron

De las pruebas realizadas se obtuvieron los siguientes resultados:

- El sistema lee y analiza correctamente los datos de telemetría.
- El algoritmo de ejecución de rutas diseñado inicialmente no era viable ya que no permitía la actualización de la ruta durante el vuelo. Se tuvo que modificar para usar guided points en lugar de misiones nativas del dron.
- Para cada comando brindado por el sistema desde la interfaz web, se logró implementar un algoritmo en el driver específico para este dron.
- No fue necesario realizar modificaciones importantes en la interfaz definida por el sistema a bordo para la comunicación con los drones. Los cambios realizados fueron los siguientes:
 - Se agregó a la interfaz *IDrone* una nueva funcionalidad para determinar si el dron soporta modo guiado (semi-automático) de vuelo;
 - En esta misma interfaz se definió una nueva acción *moveTo(point)* que permite indicarle al dron que debe volar hacia el punto indicado.
 - En el algoritmo de navegación implementado en el sistema a bordo, se controla si el dron soporta modo guiado de vuelo y en tal caso se utiliza la nueva función *moveTo* para establecer el siguiente punto a seguir por el dron.

5.2.2. Pruebas de casos de uso crítico 1 - Planificación de rutas

Descripción

Realizar pruebas de integración sobre el caso de uso crítico de planificación de rutas.

Objetivo

Comprobar el correcto funcionamiento del módulo de código correspondiente al caso de uso.

Resultados Esperados

Se espera ejecutar el caso de uso desde el frontend con el objetivo de agregar rutas y que éstas queden almacenadas correctamente en la base de datos. También se desea probar que el sistema no permite el ingreso de rutas incorrectas.

Resultados Obtenidos

Las rutas correctas especificadas se almacenan correctamente en la base de datos.

5.2.3. Pruebas de casos de uso crítico 2 - Ejecución de comando

Descripción

Realizar pruebas de integración sobre el caso de uso crítico de ejecución de comando.

Objetivo

Comprobar el correcto funcionamiento del módulo de código correspondiente al caso de uso.

Resultados Esperados

Se espera ejecutar correctamente una ruta planificada previamente.

Resultados Obtenidos

Se logró ejecutar una ruta correctamente, la ruta consistía en un conjunto de puntos relativamente cercanos. Se probó también con rutas donde la batería no alcanzaría, rutas de más de 50 km y el sistema detectó que la batería sería insuficiente, mostrando una advertencia al usuario.

5.2.4. Pruebas de casos de uso crítico 3 - Registro de Drone

Descripción

Realizar pruebas de integración sobre el caso de uso crítico de registro de dron.

Objetivo

Comprobar el correcto funcionamiento del módulo de código correspondiente al caso de uso.

Resultados Esperados

Se espera registrar drones al sistema central de forma correcta y visualizarlos desde la estación de control.

Resultados Obtenidos

Se registraron drones de forma correcta para los 2 modelos de drones.

5.2.5. Pruebas de casos de uso crítico 4 - Envío de datos de estado

Descripción

Realizar pruebas de integración sobre el caso de uso crítico envío de datos de estado.

Objetivo

Comprobar el correcto funcionamiento del módulo de código correspondiente al caso de uso.

Resultados Esperados

Se espera obtener información del estado del dron de forma continua.

Resultados Obtenidos

Se realizaron múltiples vuelos para testear esta funcionalidad. Estos test se realizaron con los dos modelos de drones y en ambos casos se enviaron datos desde el sistema a bordo al servidor central.

5.2.6. Pruebas de casos de uso crítico 5 - Aterrizaje de emergencia

Descripción

Realizar pruebas de integración sobre el caso de uso crítico aterrizaje de emergencia.

Objetivo

Comprobar el correcto funcionamiento del módulo de código correspondiente al caso de uso.

Resultados Esperados

Se espera realizar un aterrizaje de emergencia y chequear que el envío de información sea realizado.

Resultados Obtenidos

Se realizaron varios vuelos con un nivel bajo de batería (omitiendo la advertencia de batería baja del sistema). Cuando el sistema detecta un nivel bajo de batería efectivamente aterriza en el lugar que esté enviando la posición al sistema central.

Se detectó un error en el sistema cuando el sistema a bordo no contaba con conexión a internet. El sistema no enviaba la información por medio de mensaje de texto, y continuaba intentando el envío por internet. Este problema fue solucionado posteriormente.

5.3. Resultados

En esta sección se presentan los resultados obtenidos durante las pruebas de vuelo realizadas con los drones junto a los resultados de las pruebas del sistema.

5.3.1. Verificación del sistema

Durante las pruebas realizadas se detectaron varios errores que fueron corregidos. Algunos de ellos quedaron abiertos debido a la complejidad de su solución o la baja prioridad de los mismos. Luego de cada etapa de corrección se volvieron a realizar las mismas pruebas hasta que no se volvieron a detectar nuevos errores, y la condición de los drones lo permitió.

En la siguiente tabla se resumen los resultados de las pruebas. El valor para el estado aprobación de cada caso de uso se calculó como el cociente del tiempo estimado completado sobre el total del tiempo para finalizar. En la Tabla 7 puede verse el estado de completitud de cada caso de uso crítico junto con los errores detectados. Siendo el tiempo total el tiempo completado más el tiempo estimado necesario para corregir los errores pendientes.

Tabla 7: condición de los casos críticos en cuanto a su desarrollo y errores detectados. En la columna de errores detectados, puede verse entre paréntesis si el mismo está solucionado (si) o no lo está (no).

Caso de Uso	Aprobación (%)	Errores detectados (¿corregido?)
Planificación de rutas	100	1. No permite agregar descripción (si) 2. Excepción al tratar de editar una ruta si no se ingresa un nombre. (si) 3. Eliminar ruta falla, trata de eliminar de la base las rutas, aunque tengan referencias (si)
Ejecución de comando	93	4. Vuelo en zigzag, Parrot (si) 5. No aterriza al llegar al último punto de la ruta (si) 6. Brújula del celular no funciona (no) -Sin solución por software, requiere incluir un sensor más sensible. 7. Comando MoveTo no funciona en X8+ (si) 8. Comandos Wait y Continue no implementados en X8+ (si) 9. X8+ inicia vuelo muy rápido, cuando no terminó de despegar (si) 10. En el X8+, no se puede actualizar la ruta en vuelo con el modo de vuelo auto (si) 11. El dron no recibe los comandos por SMS (no)
Registro de Drone	97	12. No se reconecta automáticamente en caso de desconectarse (si)

		<p>13. No se puede configurar url de servicios (si)</p> <p>14. No se puede seleccionar el modelo del dron (si)</p> <p>15. No obtiene el número del celular (si)</p> <p>16. Si el celular tiene dos conexiones sim y la sim conectada está en la segunda entrada, no se obtiene el número del celular (no)</p>
Envío de datos de estado	85	<p>17. Formato inválido en el JSON enviado al server (si)</p> <p>18. Falta parsear respuesta del server en Android (si)</p> <p>19. Se envían demasiados datos por SMS (si)</p> <p>20. Android requiere confirmación tras enviar muchos SMS (no)</p> <p>21. El server no está uniendo los SMS de una misma</p> <p>22. Notificación correctamente (si)</p> <p>23. Se generan muchos datos en los logs en el celular, borrarlos cuando superen cierto límite (no)</p> <p>24. Registrar datos sobre el funcionamiento de la red 3G (si)</p> <p>25. Faltan datos de telemetría X8+ (si)</p> <p>26. Leer el porcentaje de batería X8+ (si)</p>
Aterrizaje de emergencia	100	<p>27. No se tiene en cuenta la velocidad del viento, si está presente (si)</p> <p>28. Calcula mal la cantidad de batería necesaria para terminar el viaje (si)</p> <p>29. Determinar gravedad del estado (emergencia, advertencia) (si)</p> <p>30. No permitir despegar si se detectó una Emergencia (si)</p>

De los mayores errores detectados durante las pruebas del sistema, solo quedaron abiertos un 17%. Se entiende que los errores pendientes más importantes a tener en cuenta para trabajo a futuro son los relaciones con la comunicación vía SMS.

5.3.2. Validación con el cliente

En la última reunión con el cliente se relevó la siguiente validación. Considera que el sistema desarrollado es altamente compatible con los sistemas desarrollados por la Administración Nacional de Correos, ya que las tecnologías utilizadas como: Java, PostgreSQL, PostGIS, OpenLayers también son manejadas por la empresa. También valoró que las tecnologías utilizadas son open source, una tendencia actual en el estado.

El cliente, sugirió continuar con el proyecto con la implementación de un plan piloto, en donde se instale el sistema en un ambiente reducido, en un pueblo del interior. Con el sistema

actual funcionando, se podrán detectar nuevos requerimientos, que surjan de un uso real en lugar de supuestos.

Por otro lado, para evitar el desarrollo del módulo de administración a corto plazo, coincidió con la instalación de diferentes instancias por cada localidad.

Desde el punto de vista social, el correo considera que el impacto del proyecto es positivo, ya que como empresa de comunicación tienen como objetivo brindar el servicio a toda la población en todo momento. Por diversos motivos como inundaciones, incendios, mal estado de las vías de transporte, entre otros inconvenientes, es inviable brindarlo con las capacidades actuales. Con la implantación del sistema se podrán romper estas barreras.

Capítulo 6 - Gestión del Proyecto

En este capítulo se tratan los puntos relacionados a cómo se desarrolló el proyecto en general. Se hace especial énfasis en la divergencia entre la planificación inicial de las actividades que componen el proyecto y la evolución real del mismo a lo largo del tiempo.

6.1. Planificación

La planificación inicial del proyecto se dividió en cinco etapas principales. A grandes rasgos se planificaron las siguientes etapas:

- **Relevo y análisis de requerimientos:** esta tarea se ejecuta en etapas tempranas del proyecto. Se realizan reuniones con el cliente para definir las inquietudes, requerimientos y restricciones.
- **Documentación general:** esta es una tarea que es ejecutada a lo largo de todo el proyecto. Todas las actividades llevadas adelante durante el transcurso del proyecto son documentadas. De esta manera se espera mantener organizada la información generada durante el desarrollo, simplificando las tareas finales, por ejemplo, la redacción del informe.
- **Investigación:** se corresponde con el estudio del estado del arte, investigar proyectos similares, modelos de drones disponibles en el mercado y sus capacidades, tecnologías que puedan ser de ayuda, etc.
- **Arquitectura y diseño:** se define la arquitectura y el diseño del sistema. Se detectan los riesgos técnicos más significativos y se evalúan de qué forma mitigarlos: estudiando una tecnología en particular, desarrollando un prototipo concreto que sirva para validar alguna hipótesis, etc.
- **Implementación:** hay tareas de implementación que son llevadas a cabo durante la fase de arquitectura y diseño para validar las decisiones tomadas. Estas tareas se tratan del desarrollo de prototipos concretos. Con la arquitectura estable y el diseño de los principales componentes del sistema, nos encontramos en condiciones de empezar con el desarrollo propiamente dicho.
- **Verificación:** esta etapa encapsula las tareas de testeo del sistema. Al inicio se piensa que habrá una ida y vuelta importante detectando y corrigiendo defectos del desarrollo. Por lo tanto, habrá actividades de desarrollo que serán ejecutadas luego de comenzada la verificación. Por este motivo, se solapan ambas actividades.

Se puede apreciar la distribución de estas tareas a lo largo de la duración estimada inicial del proyecto en el siguiente diagrama:

Id.	Nombre de tarea	T2 14				T3 14			T4 14			T1 15			
		Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	
1	Investigación	[Barra de planificación original]													
2	Relevo y Análisis de Requerimientos	[Barra de nueva tarea]													
3	Arquitectura y Diseño	[Barra de nueva tarea]													
4	Implementación	[Barra de nueva tarea]													
5	Verificación	[Barra de nueva tarea]													
6	Documentación	[Barra de nueva tarea]													

Leyenda

Planificación original	Nueva tarea
Tarea finalizada	Tarea atrasada

Ilustración 30: vista trimestral de la planificación inicial.

Luego de iniciado el proyecto, se presentó la oportunidad de anotarse a un llamado de financiación de proyectos de innovación tecnológica que tuvieran foco en solucionar un problema en una empresa, una industria o un sector prioritario del país. El llamado fue realizado por el Centro de Innovación en Ingeniería (CII) en conjunto con la Fundación Julio Ricaldoni. Este llamado espera que los proyectos contribuyan a la promoción de la innovación e incorporación de tecnología en el sector productivo.

La posibilidad de conseguir los fondos de la financiación supuso una excelente oportunidad para poder trabajar con equipamiento real, principalmente la incorporación al desarrollo de modelos de drones con capacidades semejantes a las requeridas realmente. Viendo que este proyecto se amoldaba casi a la perfección con las bases y criterios de selección del llamado no dudamos en postularnos. Habiéndonos postulado al llamado en abril del 2014, comenzamos con la definición de la arquitectura del prototipo de sistema teniendo en cuenta los drones que esperábamos conseguir con los fondos. Además, se actualizó el cronograma de actividades de la siguiente manera:

- Se incorporaron nuevas tareas de pruebas en exteriores con los drones a comprar.
- Se ajustó el plan de desarrollo para incluir el desarrollo de prototipos que permitieran probar rápidamente la integración del sistema con el dron.

El diagrama de Gantt anterior se transformó en lo siguiente:

Id.	Nombre de tarea	T2 14				T3 14			T4 14			T1 15			
		Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	
1	Investigación	[Barra de planificación original]													
2	Relevo y Análisis de Requerimientos	[Barra de nueva tarea]													
3	Arquitectura y Diseño	[Barra de nueva tarea]													
4	Implementación	[Barra de nueva tarea]													
5	Verificación	[Barra de nueva tarea]													
6	Documentación	[Barra de nueva tarea]													
7	Pruebas en exteriores	[Barra de nueva tarea]													

Leyenda

Planificación original	Nueva tarea
Tarea finalizada	Tarea atrasada

Ilustración 31: vista trimestral de la planificación actualizada con actividades extras por financiación.

El proyecto no fue seleccionado en la primera convocatoria. Sabiendo de la existencia de un segundo llamado para mediados del año 2014, se volvió a actualizar el cronograma con la suposición de conseguir el financiamiento de este segundo llamado. Para la segunda postulación se planificó usar los fondos disponibles en tres desembolsos. En primera se compraría un drone sencillo de bajo costo para realizar las primeras pruebas del prototipo, además de notebooks para desarrollo. En el segundo desembolso se compraría un modelo de drone más costoso con mayores prestaciones. Por último, el tercer desembolso sería utilizado para los gastos de impresiones, fotocopias, etc., al final del proyecto. En la tabla 8, se puede ver el calendario de gastos planificado.

Tabla 8: calendario de gastos.

Rubros	Mes de cumplimiento Hito	Diciembre	Marzo	Abril
01	Adecuación Edilicia			
02	Equipamiento	5.200	5.200	
03	Actualización de instrumental			

Entrega postal usando UAVs

04	Servicios y seguros		300	1000
05	Personal Incremental			
06	Consultores			
07	Promoción y Difusión			
08	Materiales e Insumos			500
09	Otros costos	300		
Total:		5.500	5.500	1.500
USD				

Como se puede apreciar de la tabla anterior, la financiación obtenida fue destinada casi en su totalidad a la compra de equipamiento para la implementación del sistema. En este rubro, se compraron notebooks para los miembros del equipo, dos drones AR.Drone Parrot 2.0 para la primera parte del prototipo. Con el segundo desembolso se compró otro el drone X8+, con mayores prestaciones que el modelo anterior, repuestos para este drone y celulares para interactuar entre el servidor central y los drones. El resto del dinero fue destinado para el pago de servicios (OpenShift, conexión a internet móvil) e impresiones.

Con el calendario de compras anterior, se adecuaron algunas tareas del proyecto para ser ejecutadas al momento de disponer de los modelos de los drones. Para cada drone se definieron tres tareas:

- Implementación del driver en el sistema a bordo
- Medición de capacidades de vuelo de los drones
- Pruebas del sistema

La planificación resultante fue la siguiente:

Id.	Nombre de tarea	T2 14				T3 14			T4 14			T1 15			T2 15			T3 15			Oct.
		Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	
1	Investigación																				
2	Relevo y Análisis de Requerimientos																				
3	Arquitectura y Diseño																				
4	Implementación de prototipos																				
5	Desarrollo de prototipos																				
6	Documentación																				
7	Pruebas en exteriores																				
8	Implementación de CU críticos																				
9	Implementación de CU																				
10	Implementación driver X8+																				
11	Pruebas primarias																				
12	Pruebas en exteriores																				

Leyenda

Planificación original	Nueva tarea
Tarea finalizada	Tarea atrasada

Ilustración 32: vista trimestral de la planificación para la segunda convocatoria.

6.2. Ejecución

La tarea de investigación fue realizada dentro de los plazos previstos. Sin embargo, el análisis tuvo una extensión en su duración debido a complicaciones para la realización de entrevistas de estudio del estado actual de la distribución de correspondencia del correo y sus necesidades. El principal actor de esta parte es el Jefe Departamental del correo en Rivera, el Sr. Sergio Silva. La distancia y una agenda ocupada dificultaron la coordinación de las reuniones necesarias. Finalmente se tuvieron charlas por Skype entre los miembros del equipo del proyecto y Sergio.

Habernos postulado a los llamados de financiación de proyectos de innovación y haber conseguido financiación en la segunda convocatoria significó una oportunidad única para llevar a cabo el proyecto utilizando equipo real (drones). Incluso se pudo adquirir un dron de excelentes características que de lo contrario no hubiera sido posible incluir dentro del proyecto. Sin embargo, esto significó también algunos retrasos en lo planificado. Existen tres hitos principales relacionados a la convocatoria que rompieron con los planes iniciales:

1. Fallo de primera convocatoria: no se consiguió la financiación. Tras evaluar los motivos por los que el proyecto no fue aprobado y viendo que con ciertos cambios se tendría más

chances de conseguir el beneficio, se volvió a planificar las tareas pendientes y se preparó el formulario de inscripción. Se trabajó mientras en tareas de implementación que no dependían de los drones.

2. Fallo de segunda convocatoria, con resultado positivo en este caso. La noticia de que el proyecto había conseguido la financiación fue recibida el 26 de noviembre del 2014.
3. Acceso a segunda financiación. Tras haber completado las tareas acordadas haciendo uso del primer desembolso de fondos por parte de la FJR y presentado el informe correspondiente, la aprobación del segundo desembolso se realizó el día 23/06 del 2015. Nuestra planificación no había tenido en cuenta este tiempo, por lo que las tareas relacionadas con el segundo modelo de drone (X8+) se vieron retrasadas hasta esta fecha.

Otro factor que incidió en la ejecución del proyecto fue la compra del segundo modelo de drone, el X8+ de la empresa 3D Robotics. Desde los primeros trámites de homologación de equipo radioeléctrico en URSEC [23], pasando por la compra en el exterior y la importación, transcurrieron cerca de dos meses. Esto retrasó aún más el inicio de las pruebas de vuelo con este drone y de las pruebas de integración del sistema. Se pudo acelerar el desarrollo del driver para este drone haciendo uso de un simulador del vehículo.

Adicionalmente, pero en menor medida, la rotura del drone X8+ durante las pruebas del sistema a mediados de diciembre del 2015 terminaron retrasando el proyecto una vez más. En este caso no se pudo continuar con las pruebas del sistema mientras se importaron los repuestos para reparar el drone. A fines de diciembre llegaron los repuestos a Montevideo y se retomaron las pruebas del sistema. A continuación, se muestra un Gantt con la ejecución final del proyecto:

Id.	Nombre de tarea	T2 14			T3 14			T4 14			T1 15			T2 15			T3 15			T4 15			T1 16					
		Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	Abr.	
1	Investigación	■																										
2	Relevo y Análisis de Requerimientos	■																										
3	Arquitectura y Diseño				■																							
4	Implementación de prototipos				■																							
5	Desarrollo de prototipos				■																							
6	Documentación	■																										
7	Pruebas en exteriores										■																	
8	Implementación de CU críticos										■																	
9	Implementación de CU												■															
10	Implementación driver X8+																	■										
11	Pruebas primarias																			■								
12	Pruebas en exteriores													■														
13	Adquisición X8+																	■										
14	Compra de repuestos X8+																										■	
15	2° Pruebas en exteriores																										■	

Legenda

Planificación original	Nueva tarea
Tarea finalizada	Tarea atrasada

Ilustración 33: vista trimestral de la ejecución del proyecto.

Los distintos inconvenientes sufridos durante las postulaciones a los llamados de financiación, trámites de importación, etc., repercutió de manera negativa en el estado de ánimo de los miembros del equipo de desarrollo del proyecto. Esto tuvo consecuencias negativas en la ejecución del proyecto lo que llevó a impedir culminar con tareas independientes de estos factores en las fechas previstas. De todos modos, si bien se retrasó la planificación, gracias a las postulaciones se pudieron obtener datos más fieles y un prototipo más cercano al producto que se debería realizar para poner en producción.

Haber conseguido la aprobación y consiguiente financiación de la convocatoria a proyectos de innovación supuso una opción sumamente positiva. Esto nos dio la oportunidad de adquirir equipo con el que obtener métricas reales y valiosas para determinar la viabilidad del proyecto. Además, nos permitió ver nuestro trabajo llevado a la práctica, una experiencia genial. Por lo que ajustar al cronograma realizado para contar con esta posibilidad fue una decisión altamente positiva.

Capítulo 7 - Conclusiones

Luego de los estudios realizados se considera que los drones de la actualidad cuentan con capacidades suficientes para el transporte de correspondencia de poco peso. Con los modelos utilizados durante el proyecto, se podría aumentar el área de cobertura del correo en un 442% respecto de la actual cobertura. Esto, sin dudas, ofrecerá un mejor servicio para los habitantes de las zonas a las cuales el correo llega con dificultad o que directamente no llega. El total del área del territorio nacional con cobertura de correo pasaría de un 3,8% a 16,79% por extender la cobertura en un área de 5 Km en torno a las oficinas actuales del correo. Sin embargo, quedan abiertas cuestiones a cerca de la seguridad del vuelo de estos vehículos en ambientes urbanos. Se considera la seguridad del público en general como un punto crítico.

Teniendo en cuenta los equipos que se utilizaron para desarrollar el prototipo y en base a las métricas medidas en las distintas pruebas realizadas, se podrán transportar 100 (600 gr) cartas a una distancia superior a los 5 Km, en un tiempo de 19 minutos de vuelo. Se solucionó la limitación del alcance de conexión de los drones comerciales disponibles actualmente mediante la incorporación de un dispositivo Android conectado físicamente al dron en vuelo. Esto nos permite mantener una comunicación con el servidor central por dos canales distintos: SMS e internet móvil. Este enfoque puede ser utilizado en otros proyectos que involucren drones para la ejecución de algún tipo de tareas y su control a distancia, extendiendo las aplicaciones de este proyecto a otros de características similares. Adicionalmente, como resultado de la alta autonomía de los celulares en comparación con los drones, tenemos un nivel de seguridad adicional en caso de un accidente al continuar recibiendo telemetría desde el celular a bordo por varias horas. Esto se vuelve fundamental para el rescate del aparato y de la correspondencia transportada en caso de ser necesario.

Como se mencionó en el capítulo 3, la entrega de caravanas es una actividad muy común e importante para el correo en el interior del país. Los paquetes de caravanas que se suelen enviar pesan entre 5 y 10 Kg y las distancias de las rutas de entregas abarcan entre 50 y 100 Km. Los drones utilizados para este proyecto no tienen la capacidad de transportar paquetes tan pesados ni por distancias tan grandes. Se descarta entonces, el envío de caravanas utilizando modelos de drones similares a los empleados en el proyecto. Sin embargo, existen modelos de drones, como los que se mencionan en el siguiente capítulo, con mayores prestaciones con los cuales parece ser viable el transporte de estos paquetes.

Notamos que el comportamiento del dron X8+ es inseguro para las personas a su alrededor en caso de un accidente. El mismo no cuenta con mecanismos de seguridad para cortar la energía de los motores en caso de estrellarse, por lo que las hélices siguen rotando aún después de impactar contra el suelo, lo que puede poner en riesgo la integridad física de las personas alrededor. Para esto se puede pensar en alternativas para el mecanismo de entrega, de modo que el dron nunca deba aproximarse demasiado al suelo, incorporar chasis de protección alrededor de las hélices, buscar modelos de drones que apaguen los motores en caso de detectar de que alguno se vea obstruido, etc. El dron AR.Drone Parrot 2.0 tiene mecanismos de seguridad que apagan los motores en caso de que alguna de las hélices sea vea obstruida, o si la posición del dron se encuentra invertida. Estas medidas de seguridad deberían ser un requisito a la hora de elegir nuevos modelos de drones.

7.1. Restricciones

Con los datos obtenidos el sistema consigue un comportamiento razonable de alcance y capacidad de carga para un primer prototipo y para el dinero invertido en el mismo. Existen

ciertas limitantes a tener en cuenta a la hora de implantar el sistema en ambiente de producción. Las más importantes están relacionadas con temas seguridad de los vehículos durante el vuelo, la reglamentación nacional sobre el vuelo de aeronaves y las capacidades de los drones en cuanto a autonomía y carga. Para estudiar en mayor profundidad el comportamiento del sistema en un ambiente similar a la realidad del territorio nacional, se sugiere la comenzar con un plan piloto. Para esto, se tendrá que tener en cuenta las restricciones mencionadas a continuación.

7.1.1. Carga Máxima

Teniendo en cuenta que el peso de una correspondencia estándar (por ejemplo, una factura) es de 6 gr, la cantidad de correspondencias que se podrán entregar en un solo viaje no puede ser mayor a 100 cartas, para el drone modelo 3D Robotics X8+. Esto se ajusta al caso de estudio “Entrega de Facturas de servicios”, presentado en la sección 3.1 del presente documento.

7.1.2. Distancia Máxima

Suponiendo condiciones climáticas favorables, es decir que no haya lluvia ni vientos que superen los 45 km/h, la distancia máxima es de:

- 5,34 km si el drone transporta la carga máxima
- 6,42 km/h si el peso de la correspondencia es mínimo.

7.1.3. Condiciones geográficas

Para mantener los vuelos de los drones bajo condiciones de seguridad deseables, se requiere la conectividad con la red de telefonía móvil en todo momento. Hay que tener en cuenta que las características geográficas del terreno pueden afectar esta señal, por ejemplo, las zonas de cerros altos la bloquean. Antes de implantar el sistema en una localidad hay que estudiar las condiciones geográficas de la misma para determinar si el sistema se comportará como es requerido.

7.1.4. Destino

Es necesario asegurarse que la zona de entrega se encuentre en condiciones. Esto es, debe ser un área llana y despejada en un radio de unos 10 metros para evitar accidentes al momento del aterrizaje del drone. No debe haber personas cerca del drone cuando este comience con el aterrizaje. Esto puede mejorarse implementando alguna de los mecanismos de entregas comentados en el capítulo 8.

Capítulo 8 - Trabajo a Futuro

La solución propone que en lugar de dejar la correspondencia a la espera que los destinatarios la retiren, un funcionario de la sucursal coloque la misma en una bandeja en un dron, posteriormente se planifique una ruta, o se cargue una previamente diseñada, e inicie el envío. Debido a que los drones tienen un alcance reducido a unos cuantos kilómetros y que los envíos en el interior del país pueden requerir viajes a largas distancias [6], se podrán usar puntos de abastecimiento intermedios para recargar las baterías del dron antes de continuar. Estos puntos pueden ser centros públicos como escuelas rurales, comisarías, etc. En la actualidad, se cuenta con el apoyo del Jefe Departamental de Policía de Rivera para usar las comisarías de campaña como centros de abastecimiento.

Dado que el proyecto consiste en el estudio de la viabilidad de la entrega postal mediante drones y la construcción del prototipo, existen actividades consideradas necesarias o esperadas para la implantación y puesta en producción del proyecto. En la siguiente sección se detallan las actividades identificadas.

8.1. Implantación del sistema

El prototipo desarrollado cuenta de tres nodos principales. Por un lado, tenemos el servidor central el cual implementa la interfaz web, la lógica de negocio y la capa de servicios utilizados por los restantes subsistemas. El servidor central puede correr en Apache Tomcat 8.0+, las siguientes funcionalidades son requeridas para su correcto funcionamiento:

- Librería de WebSockets. Adicionalmente, en caso de que el servidor donde esté hospedado el sistema modifique el puerto de acceso a los servicios WebSockets, el firewall deberá mantener este puerto accesible desde el exterior.
- Acceso a servidor de base de datos PostgreSQL 9.3 o superior con una versión de Postgis instalada igual o superior a 2.1.4.

Por otro lado, existen dos aplicaciones Android, que deben ser instaladas en los dispositivos móviles utilizados. La aplicación PDS-SCI debe ser instalada en un dispositivo con una versión de Android instalada mayor o igual a 4.2. Este dispositivo además de tener instalado un chip sim de telefonía con capacidad de envío de SMS y acceso a datos móviles. Luego debe configurarse esta aplicación para acceder a los servicios del sistema central. De forma similar la aplicación PDS-DFS debe ser instalado en un dispositivo Android similar al anterior, pero con dos restricciones adicionales: el celular debe ser compatible con OMG y debe contar con brújula y GPS propios.

Por más información sobre los pasos necesarios para desplegar el sistema en los distintos nodos ver [8]. Durante el proyecto se utilizó el servicio OpenShift para desplegar el sistema central. La instalación de backend en los servidores de OpenShift no presentó mayores dificultades, puede verse también en [8] los pasos necesarios para implantar el sistema en esta plataforma.

La capacidad de definir permisos de acceso a los usuarios del sistema, con la finalidad de restringir las acciones a ejecutar por parte de un operador y el acceso a los drones conectados no está disponible en el estado actual del prototipo desarrollado. Esto se debe a que la interfaz web de administración no fue implementada durante la ejecución de este proyecto. Esto implica que cualquier usuario que acceda al sitio web donde esté publicado el sistema central podrá ser capaz de administrar las rutas y los vuelos. Para prevenir que un funcionario del correo de un

departamento pueda controlar los drones de otro, se puede tener una instancia del servidor central implantado en cada uno.

8.2. Expansión del área de cobertura

El área de cobertura está dada en la actualidad por el alcance de los drones utilizados para las entregas. Puede estudiarse la viabilidad de utilizar zonas o establecimientos como bases intermediarias para recargar las baterías de los drones. De este modo, haciendo paradas intermedias de recarga, puede multiplicarse el radio de alcance de los drones. Las zonas de recarga tienen que ser lugares que cuenten con las condiciones y el personal capacitado para esta función. Deben tener un área despejada donde el droné pueda aterrizar. El sistema tiene que ser ajustado para permitir al administrador de vuelo definir puntos de abastecimiento en la ruta. Al llegar a estas ubicaciones el droné deberá aterrizar y esperar a recibir la orden de continuar, la cual sería enviada tras la recarga de la batería. En la versión actual del prototipo esto se puede realizar de manualmente ejecutando comandos en sucesión. Esto es, durante el vuelo, el controlador puede indicar que el droné se dirija a un punto específico, el cual se correspondería con una base de recarga. Al llegar ahí, el controlador deberá enviar el comando *Wait*, seguido de *Land*. De esta manera el droné aterrizará en la ubicación de la base. Luego personal capacitado cargará la batería del droné, pudiendo reemplazarla por otra ya cargada. Por último, el controlador debe enviar el comando *TakeOff* para que el droné continúe con el vuelo. Tener en cuenta que este proceso podría requerir de un tiempo mayor al normal, debido a la demora por la carga de las baterías del droné. Lo que puede tomar entre una y dos horas, dependiendo de la capacidad de la batería y el cargador.

Adicionalmente, a medida que la tecnología de estos vehículos avance, los mismos tendrán mayor autonomía y mayor capacidad de carga. En la actualidad existen drones a la venta con capacidades superiores a los usados en este proyecto, por lo que los resultados obtenidos en este proyecto podrían ser superados con esos modelos. Algunas mejoras ya disponibles se consiguen sustituyendo las baterías clásicas por otras fuentes de energía, como combustibles fósiles o hidrógeno. Yeair [21] es un droné que utiliza gasoil como combustible. Mientras que, Hycopter [36], es un droné con baterías de hidrógeno, según el fabricante tiene una autonomía máxima de 4 horas.

8.3. Implementación del subsistema de Administración

Está pendiente la implementación de la interfaz web de administración para controlar el acceso de los usuarios al sistema. Esta interfaz principalmente debe permitir definir los conjuntos de permisos y roles necesarios para poder determinar a qué drones puede acceder cada usuario.

A nivel del sistema central, el único punto del sistema en su totalidad que debe ser modificado, es necesario realizar varias tareas para la incorporación de este módulo de administración.

- Modificar el esquema de Base de Datos con las nuevas entidades.
- Definir los roles y permisos básicos para el acceso al sistema.
- Autenticación y autorización de los usuarios durante su interacción con el sistema.
- Definir los servicios necesarios para la interfaz web de administración.
- Implementación de la interfaz web de administración

8.4. Integración de componentes a bordo

Una de las posibles mejoras identificadas es que el sistema a bordo se encuentre integrado en el droné. En el prototipo desarrollado es necesario incorporar hardware externo, celular Android,

que se encarga de transmitir los comandos al dron. Para realizar esto es necesario adquirir un dron que permita ejecutar software a medida dentro del chip integrado y que además cuente con conectividad móvil (3G o 4G).

8.5. Mecanismo de entrega

El prototipo realizado solo estudia la viabilidad de la entrega, teniendo en cuenta las variables mencionadas en el capítulo 1 y es capaz de transportar la correspondencia hacia el lugar indicado, pero no tiene en cuenta el mecanismo de entrega de la correspondencia. Por lo tanto, se sugieren 2 mecanismos posibles:

8.5.1. Mecanismo de tubo

El primer mecanismo consiste en acoplar un tubo con una sola tapa al dron, la correspondencia se debe introducir en él como un rollo. La inclinación del tubo deberá asegurar que la correspondencia no caiga, siempre y cuando el dron se mantenga con una inclinación determinada. Cuando el dron llega a destino deberá hacer una maniobra brusca, con el fin de pasar ese ángulo y dejar caer la carta.

Ventajas:

- Es de muy bajo costo, ya que no incluye mecanismos electrónicos.
- Es muy fácil de implementar, teniendo en cuenta el sistema desarrollado.
- Es liviano, por lo que se puede transportar mayor carga útil.

Desventajas:

- Es muy difícil asegurar que no se caiga la correspondencia por error.

8.5.2. Mecanismo de pinzas

El segundo mecanismo consiste en un sistema eléctrico de pinzas, mediante la interfaz USB con el dispositivo móvil a bordo. La correspondencia se deberá introducir en las pinzas al momento de iniciar el viaje, y se deberán abrir cuando llegue. Éste sistema sería similar al utilizado por Amazon. Se puede ver una demostración del dron usado por Amazon en vuelo en [15].

Ventajas:

- Es muy seguro ya que en comparación con el anterior no hay posibilidad de dejar caer la correspondencia por error.

Desventajas:

- Implica un costo mayor que el mencionado anteriormente.
- Se deberá agregar al sistema componente de software que tenga el comportamiento deseado.
- Es más pesado, lo que reduce la carga útil que se puede transportar.

8.6. Adquisición de equipamiento

Si bien el equipamiento con el que se trabajó a lo largo del proyecto se puede utilizar sin problemas para algunas de las actividades mencionadas anteriormente, en ésta sección se menciona el equipamiento que no se incorporó por motivos económicos. Al momento no existen drones comerciales que solucionen el problema de conectividad que soluciona el proyecto, por lo cual estos drones se deben utilizar con el sistema desarrollado.

Tener en cuenta a la hora de adquirir nuevos modelos de drones que estos deben contar con ciertas características que los permita integrar con el sistema desarrollado. Esto es, el dron debe contar con una interface de control y telemetría programática, accesible mediante radiofrecuencia o WiFi. Para el caso de radiofrecuencia, el distribuidor debe proporcionar además un transmisor

de radio en la misma frecuencia que sea compatible con dispositivos Android. El mismo tiene que ser conectado al celular vía USB OTG. De este modo, para soportar un nuevo modelo de dron, solamente se debe desarrollar un driver específico que sirva de nexo entre el sistema a bordo (ejecutando en el celular) y el dron. En principio, esta tarea sería relativamente sencilla.

8.6.1. FIREBIRD I

El dron FIREBIRD I [22] de Helidroid es un equipo altamente recomendable para este tipo de actividades ya que cuenta con una carga máxima de 20 Kg y una autonomía de 45 minutos, alcanzando velocidades entre 30 y 60 km/h. Su costo es de aproximadamente U\$S 3.800,00 en el país origen, por lo que se estima que el costo del producto en Uruguay es de U\$S 7.125,00 incluyendo envío e impuestos de importación.

8.6.2. Yeair Quadcopter

Otra opción que se puede tomar para realizar el proyecto, es utilizar drones con otras fuentes de energía como por ejemplo gasoil. Como es el caso de Yeair que es un cuadricóptero que utiliza gasoil, alcanzando una velocidad máxima de 100 km/h, una carga de 5kgs y un recorrido de 50 km (ver [21]).

Apéndices

Apéndice I: Glosario

3G: Es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles). La velocidad máxima teórica es de 3,6 Mbps.

4G: En telecomunicaciones, 4G son las siglas utilizadas para referirse a la cuarta generación de tecnologías de telefonía móvil. Para que una red móvil se considere una red 4g debe cumplir varios requisitos, entre ellos hay uno muy claro: las velocidades máximas de transmisión de datos deben estar entre 100 Mbit/s para una movilidad alta y 1 Gbit/s para movilidad baja.

Bearing: Es la dirección considerada en el plano del horizonte y, principalmente, cualquiera de las comprendidas en la rosa de los vientos. Precisamente la palabra procede del latín rhombus ('rombo'), que son las formas geométricas que unidas señalan las diferentes direcciones posibles en la rosa de los vientos.

CPU: La unidad central de procesamiento o unidad de procesamiento central (conocida por las siglas CPU, del inglés: central processing unit), es el hardware dentro de una computadora u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

Driver: un driver o un controlador, es un programa que controla un dispositivo. En el contexto de este documento se utiliza la palabra *driver* para hacer referencia al módulo, para del sistema a bordo, encargado de controlar el dron.

Drone: o UAV por su sigla en inglés (Unmanned Aerial Vehicle) refiere a una Aeronave Remotamente Piloteada para la Organización Internacional de Aviación (ICAO por sus siglas en inglés), es una aeronave sin un piloto humano a bordo. Su vuelo puede ser controlado por una computadora a bordo o por control remoto por un piloto en tierra o en otro vehículo.

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 siendo el software que tiene directa interacción con el hardware: es el encargado de controlarlo para ejecutar correctamente las instrucciones externas.

GIT: es un sistema de control de versiones gratuito y de código abierto, distribuido y diseñado para manejar pequeños y grandes proyectos con el velocidad y eficiencia.

GPRS: General Packet Radio Service (GPRS) o servicio general de paquetes vía radio creado en la década de los 80 es una extensión del Sistema Global para Comunicaciones Móviles (Global System for Mobile Communications o GSM) para la transmisión de datos mediante conmutación de paquetes. Existe un servicio similar para los teléfonos móviles, el sistema IS-136. Permite velocidades de transferencia de 56 a 114 kbps.

Heading: Es un término usado en navegación que refiere a la dirección a la que un vehículo está apuntando. Puede que no sea la dirección en la que el vehículo se está moviendo, lo que se conoce como curso o trayectoria.

HTTP: Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

JSON: Acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript, aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

Octocoptero: Se refiere a un helicóptero no tripulado que tiene ocho rotores.

Radiofrecuencia: Frecuencia correspondiente a las ondas de radio, comprendidas entre los 3 kHz y 30 MHz. Aunque se emplea la palabra radio, las transmisiones de televisión, radio, radar y telefonía móvil están incluidas en esta clase de emisiones de radiofrecuencia. Otros usos son audio, vídeo, radionavegación, servicios de emergencia y transmisión de datos por radio digital; tanto en el ámbito civil como militar. También son usadas por los radioaficionados.

Rotorcrafts: categoría de drones en los cuales se utilizan de uno a varios motores con hélices rotativas para generar elevación.

RTL: Return to Lunch, por sus siglas en inglés, se refiere a un comando ejecutado de forma nativa en algunos drones para realizar un retorno automático al punto de partido (desde donde despegó el dron)

SDK: Un kit de desarrollo de software o SDK (siglas en inglés de software development kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de software crear aplicaciones para un sistema concreto, por ejemplo, ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etcétera.

SITL: Software in the Loop, por sus siglas en inglés, refiere a un sistema desarrollado para simular vehículos no tripulados.

Tecnología push: La tecnología push es una forma de comunicación a través de internet en la que la petición de envío tiene origen en el servidor, por oposición a la tecnología pull, en la que la petición tiene origen en el cliente.

Telemetría: La telemetría es una tecnología que permite la medición remota de magnitudes físicas y el posterior envío de la información hacia el operador del sistema.

TCP: Significa Protocolo de Control de Transmisión y es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación, posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él, (es decir, el protocolo IP).

UDP: El protocolo UDP (Protocolo de datagrama de usuario) es un protocolo no orientado a conexión de la capa de transporte del modelo TCP/IP. Este protocolo es muy simple ya que no proporciona detección de errores (no es un protocolo orientado a conexión).

USB OTG: USB On-The-Go, por sus siglas en inglés, es una extensión de la norma USB 2.0 que permite a los dispositivos USB tener mayor flexibilidad en la gestión de la interconexión. Esto permite que ciertos dispositivos, por ejemplo, un celular, actúen como host, por lo que se les puede conectar una memoria USB, una antena de radiofrecuencia, etc.

Referencias

- [1] D. González, M. Revetria, Documento Modelo de Casos de Uso, anexo.
- [2] D. González, M. Revetria, Documento Descripción de la Arquitectura, anexo.
- [3] D. González, M. Revetria, Documento Modelo de Diseño, anexo.
- [4] D. González, M. Revetria, Documento Especificación de Servicios, anexo.
- [5] D. González, M. Revetria, Métricas de vuelo de los Drones, anexo.
- [6] D. González, M. Revetria, Documento Condiciones de Rivera, anexo.
- [7] D. González, M. Revetria, Documento Cobertura del Correo, anexo.
- [8] D. González, M. Revetria, Guía de instalación del prototipo, anexo.
- [9] D. González, M. Revetria, Documento de Evaluación de Riesgos Técnicos, anexo.
- [10] Ardupilot.com, ‘ArduPilot Autopilot Suite’, en línea, disponible en: <http://ardupilot.com> (accedido: 29- Feb- 2016).
- [11] Rhcloud.com, ‘Servidor central’, en línea, disponible en: <http://server20-proygrado.rhcloud.com/PDS/pds.jsp> (accedido: 29 Feb. 2016).
- [12] Openshift.com, ‘OpenShift Online by Red Hat’, en línea, disponible en: <https://www.openshift.com> (accedido: 29 Feb. 2016).
- [13] DHL.com, ‘DHL | Press Research: Parcel’, en línea, disponible en: http://www.dhl.com/en/press/releases/releases_2014/group/dhl_parcelcopter_launches_initial_operations_for_research_purposes.html (accedido: 29 Feb. 2016).
- [14] Warwick, Graham, “Google Details ‘Project Wing’ Unmanned Package-Delivery R&D”, Aviation Week & Space Technology, 3- Nov- 2014.
- [15] Amazon.com, ‘Determining Safe Access with a BestEquipped, Best-Served Model for Small Unmanned Aircraft Systems’, en línea, disponible en https://images-na.ssl-images-amazon.com/images/G/01/112715/download/Amazon_Determining_Safe_Access_with_a_Best-Equipped_Best-Served_Model_for_sUAS.pdf (accedido: 20 Abr. 2016).
- [16] Amazon.com, ‘Amazon Prime Air’, en línea, disponible en: <http://www.amazon.com/b?node=8037720011> (accedido: 29 Feb. 2016).
- [17] Microdrones.com, ‘MD4-1000 drone usad por DHL’, en línea, disponible en: <https://www.microdrones.com/en/products/md4-1000/at-a-glance> (accedido 29 Feb. 2016).
- [18] UAV Agrimensura, en línea, disponible en: <http://www.uavagrimensura.com.uy> (accedido 29 Feb. 2016).
- [19] WebReference.com, ‘Comet Programming: Using Ajax to Simulate Server Push’, en línea, disponible en: <http://www.webreference.com/programming/javascript/rg28/index.html> (accedido: 20 Abr. 2016).
- [20] Antel.com, ‘Antel - cobertura móvil’, en línea, disponible en: <http://www.antel.com.uy/antel/personas-y-hogares/movil/cobertura/cobertura-movil> (accedido: 29/02/16).

- [21] Yearir.de, ‘Drone Yearir’, en línea, disponible en: <https://yearir.de/> (accedido: 29 Feb. 2016).
- [22] Helioidroid.com, ‘Helioidroid | Soluciones’, en línea, disponible en: <http://www.heliodroid.com/soluciones.php> (accedido: 02 Mar. 2016).
- [23] URSEC, ‘Unidad Reguladora de Servicios de Comunicación’, en línea, disponible en: <https://www.ursec.gub.uy/> (accedido: 29 Feb. 2016).
- [24] Bitbucket.org, ‘Bitbucket - The Git solution for professional team’, en línea, disponible en: <https://bitbucket.org/> (accedido: 29 Feb. 2016).
- [25] D. González, M. Revetria, ‘Repositorio de código fuente’, en línea, disponible en: <https://bitbucket.org/remer/proygrado> (accedido: 29 Feb. 2016).
- [26] Adtsgroup.com, ‘Control de incendios utilizando drones’, en línea, disponible en: <http://adtsgroup.com/control-de-incendios> (accedido: 20 Abr. 2016).
- [27] I. Fette, A. Melnikov, ‘The WebSocket Protocol’, RFC 6455, Dic. 2011.
- [28] M. Belshe, R. Peon, ‘Hypertext Transfer Protocol Version 2 (HTTP/2), RFC 7540, May. 2015.
- [29] M. Arjomandi, S. Agostino, M. Mammone, M. Nelson, T. Zhou, ‘Classification of Unmanned Aerial Vehicles’, en School of Mechanical Engineering, The University of Adelaide, Australia, 01 Ene. 2007.
- [30] Dirección Nacional de Aeronáutica, ‘PODER EJECUTIVO MINISTERIO DE DEFENSA NACIONAL DIRECCIÓN NACIONAL DE AVIACIÓN CIVIL E INFRAESTRUCTURA AERONÁUTICA - D.I.N.A.C.I.A. 1 Resolución 291/014’, en Diario Oficial, República Oriental del Uruguay, 04 Set. 2014, disponible en: http://www.dinacia.gub.uy/ciudadania/diario-oficial-imp/publicaciones-realizadas/item/download/892_f524b6949068a5b049783f3fa067f54f.html (accedido: 29 Feb. 2016).
- [31] J. Maddalon, K. Hayhurst, D. Koppen, J. Upchurch, ‘Perspectives on Unmanned Aircraft Classification for Civil Airworthiness Standards’, en Langley Research Center, Hampton, Virginia, Feb. 2013.
- [32] T. Cox, C. Nagy, M. Skoog, I. Somers, ‘Civil UAV Capability Assessment’, en línea, disponible en: http://www.nasa.gov/centers/dryden/pdf/111761main_UAV_Capabilities_Assessment.pdf, Dic 2004, (accedido: 29 Feb. 2016).
- [33] National Geographic, ‘5 Surprising Drones Uses (Beside Amazon Delivery)’, en línea, disponible en: <http://news.nationalgeographic.com/news/2013/12/131202-drone-uav-uas-amazon-octocopter-bezos-science-aircraft-unmanned-robot/> (accedido: 29 Feb. 2016).
- [34] D. González, M. Revetria, Documento Investigación sobre drones, anexo.
- [35] E. Wilde, A. Vaha-Sipila, ‘URI Scheme for Global System for Mobile Communications (GSM) Short Message Service (SMS)’, RFC 5724, Ene. 2010.
- [36] HUS Unmanned Systems, ‘HYCOPTER hydrogen fuel cell UAV’, en línea, disponible en: <http://www.hus.sg/#!/hydrogen-multi-rotor/ccm4> (accedido: 20 Abr. 2016).

[37] 3DR, 'X8 | 3DR | Drone & UAV Technology', en línea, disponible en: <https://3dr.com/x8/> (accedido: 20 Abr. 2016).

[38] Align-Trek.co.uk, 'T-REX 700 (2015 Model) Nitro DFC Super Combo RH70N01AT', en línea, disponible en <https://www.align-trek.co.uk/t-rex-700-nitro-dfc-super-combo-rh70n01aw.html> (accedido: 20 Abr. 2016).

[39] Lehmannaviation.com, 'L-A Series', en línea, disponible en: <http://www.lehmannaviation.com/l-a-series.html> (accedido: 20 Abr. 2016).

[40] 3DR, 'Aero-M Support | 3DR | Drone & UAV Technology', en línea, disponible en: <https://3dr.com/kb/aero-m/> (accedido: 20 Abr 2016).

[41] D. González, M. Revetria, Documento Investigación sobre drones, anexo.