

**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**SkyGuard: Gestión remota y automatizada de  
ganado**

Entregado como requisito para la obtención del título de Ingeniero en Sistemas

**Dan Blanco – 194544**

**Gastón Donadío – 192203**

**Victoria Rocha – 161604**

**Agustín Rodríguez – 194412**

**Emiliano Rodríguez – 195358**

**Tutora: Mariel Feder**

**2020**

## Declaración de autoría

Nosotros, Agustín Rodríguez, Dan Blanco, Emiliano Rodríguez, Gastón Donadío y Victoria Rocha, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano.

Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto de grado de la carrera de Ingeniería en Sistemas;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Victoria Rocha  
[5 de marzo de 2020]



Agustín Rodríguez  
[5 de marzo de 2020]



Gastón Donadío  
[5 de marzo de 2020]



Emiliano Rodríguez  
[5 de marzo de 2020]



Dan Blanco  
[5 de marzo de 2020]

## **Agradecimientos**

Queremos agradecer a la Universidad ORT, por la formación académica que nos brindó y especialmente a nuestra tutora Mariel Feder por guiarnos y apoyarnos durante el transcurso de este proyecto tan importante para nosotros.

También a los revisores de las diferentes instancias de evaluación, Álvaro Ortas, Gastón Mousques y Darío Macchi, por los comentarios y las críticas constructivas que nos ayudaron a mejorar en el desarrollo del proyecto.

A la empresa Todosoft, por confiar en nosotros para llevar adelante esta idea, por la compra de los materiales necesarios y por ofrecernos su conocimiento.

A los expertos de negocio, quienes estuvieron dispuestos a colaborar y dedicaron su tiempo permitiéndonos conocer el negocio y obtener retroalimentación para construir el producto.

Por último, un agradecimiento especial a nuestras familias, pilares fundamentales de nuestra formación como personas y como profesionales, por su apoyo incondicional a lo largo de toda la carrera.

A todos ellos, muchas gracias.

## Abstract

Dos de los principales problemas de los campos dedicados a la ganadería extensiva son el abigeato [1] y la muerte de animales por el ingreso de los mismos a zonas definidas como peligrosas.

Para esto hoy en día los peones de los campos dedican muchas horas al conteo y verificación de la integridad del ganado de manera de detectar, de forma temprana, entre otras cosas, la falta de ganado por extravío o robos de los mismos.

Los capataces definen las zonas a monitorear en los campos, los peones realizan las recorridas, pero si las condiciones climáticas no son apropiadas, se dificulta la realización de los controles provocando que los mismos, en ocasiones, no puedan ser realizados.

En campos de más de 500 hectáreas hay personal destinado únicamente a realizar estos controles pero aun así no se logra evitar la pérdida de los animales.

El proyecto SkyGuard consiste en un sistema completo para el monitoreo de campos utilizando drones y tecnología RFID, ayudando a disminuir la pérdida de los animales a través de monitoreos automáticos.

Una vez que se ingresa la ubicación y los límites del campo, el sistema SkyGuard genera la mejor forma de recorrerlo utilizando drones. Los drones realizan esta recorrida automáticamente a la hora definida y ante cualquier fallo están programados para volver a la base e intentar hacer la recorrida más adelante.

El sistema envía notificaciones en tiempo real a los usuarios ante cualquier problema detectado, por ejemplo, cuando un animal ingresa en una zona peligrosa o se encuentra fuera del perímetro del campo.

El sistema permite también generar reportes de las recorridas del dron, mapa de calor con las zonas visitadas por el ganado, los animales leídos y los faltantes.

SkyGuard consiste en una aplicación web y una aplicación móvil, disponible para Android e iOS, que le permite a los usuarios gestionar el sistema, visualizar y cargar información, generar reportes, recibir alarmas, entre otros.

Ambas aplicaciones consumen una API REST desarrollada en .NET Core utilizando una base de datos MySQL.

A su vez, la solución incluye un componente de *software* dentro del dron que es el encargado de dotar al vehículo de la inteligencia necesaria para realizar las recorridas.

Se utilizó una adaptación de la metodología ágil Scrum tanto para la gestión del proyecto como para el desarrollo del producto.

## Palabras clave

.NET; Abigeato; Aeronaves; Android; APM; ArduPilot; Campos; *Cross-platform*; DJI; Drones; Ganado; Inventario; iOS; Mission Planner; *Mobile*; Móvil; RFID; React; Scrum; Web.

## 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) [2].

**API (*Application Programming Interface*):** Conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción [3].

**Arduino:** Es una compañía de desarrollo de *software* y *hardware* libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de *hardware* para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real [4].

**ARM:** ARM, anteriormente Advanced RISC Machine, originalmente Acorn RISC Machine, es una arquitectura RISC (*Reduced Instruction Set Computer*=Ordenador con Conjunto Reducido de Instrucciones) de 32 bits y, con la llegada de su versión V8-A, también de 64 Bits, desarrollada por ARM Holdings [5].

**Bash:** GNU Bash o simplemente Bash (Bourne-again shell) es un lenguaje de comandos y shell de Unix escrito por Brian Fox para el Proyecto GNU como un reemplazo de *software* libre para el shell Bourne [6].

**Benchmarking:** Consiste en tomar "comparadores" o *benchmarks* a aquellos productos, servicios y procesos de trabajo que pertenezcan a organizaciones que evidencien las mejores prácticas sobre el área de interés [7].

**Bluetooth:** Es una especificación industrial para redes inalámbricas de área personal (WPAN) creado por Bluetooth Special Interest Group, Inc. que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz [8].

**CI<sup>2</sup>:** El Centro de Innovación en Ingeniería es una iniciativa conjunta de las Facultades de Ingeniería de la Universidad de la República, Universidad Católica del Uruguay, Universidad ORT Uruguay y Universidad de Montevideo [\[9\]](#).

**CRUD:** En informática, CRUD es el acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: *Create, Read, Update and Delete*), que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un *software* [\[10\]](#).

**DevOps:** Acrónimo inglés de *development* -desarrollo- y *operations* -operaciones-, es una práctica de ingeniería de *software* que tiene como objetivo unificar el desarrollo de *software* (Dev) y la operación del *software* (Ops) [\[11\]](#).

**DynDNS:** DynDNS (Dynamic Network Services, Inc.) es una compañía de Internet de los Estados Unidos de América dedicada a soluciones de DNS en direcciones IP dinámicas. Solía ofrecer servicios gratuitos de redirección a IP de subdominios de una gran lista de nombres disponibles, ofreciendo a particulares la oportunidad de crear un servidor en Internet gratuitamente con una dirección del tipo *misubdominio.dyn dns.com* [\[12\]](#).

**Ethernet:** Es un estándar de redes de área local para computadores y aviones web, por sus siglas en español Acceso Múltiple con Escucha de Portadora y Detección de Colisiones (CSMA/CD) [\[13\]](#).

**Expo:** Es una plataforma de código abierto para crear aplicaciones nativas universales para Android, iOS y web con JavaScript y React [\[14\]](#).

**Fail Safe:** En ingeniería, un sistema a prueba de fallas es una característica o práctica de diseño que, en el caso de un tipo específico de falla, responde de manera inherente de una manera que no causará daños mínimos o mínimos a otros equipos, al medio ambiente o a las personas [\[15\]](#).

**Firmware:** El *firmware* o soporte lógico inalterable es un programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Está fuertemente integrado con la electrónica del dispositivo, es el *software* que tiene directa interacción con el *hardware*, siendo así el encargado de controlarlo para

ejecutar correctamente las instrucciones externas. De hecho el *firmware* es uno de los tres principales pilares del diseño electrónico [16].

**FPV (First-person view):** El pilotaje con visión remota o pilotaje en inmersión es un procedimiento que consiste en dirigir un modelo teledirigido (avión, helicóptero, aerotrén, coche, etc.) por medio de una cámara de vídeo a distancia y de una pantalla o unas gafas de vídeo [17].

**Framework:** Entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar [18].

**GPIO:** GPIO (*General Purpose Input/Output*, Entrada/Salida de Propósito General) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución [19].

**GPS:** El Sistema de Posicionamiento Global (GPS; en inglés, *Global Positioning System*), y originalmente NAVSTAR GPS, es un sistema que permite determinar en toda la Tierra la posición de cualquier objeto (una persona, un vehículo) con una precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión [20].

**GRASP:** En diseño orientado a objetos, GRASP son patrones generales de *software* para asignación de responsabilidades, es el acrónimo de "GRASP (*object-oriented design General Responsibility Assignment Software Patterns*)" [21].

**HD:** La alta definición (AD), más conocida como HD o HQ (siglas del inglés *High Definition* o *High Quality*, respectivamente), es un sistema de imagen, vídeo o sonido con mayor resolución que la definición estándar, alcanzando resoluciones de 1280×720 píxeles y 1920×1080 píxeles [22].

**IDE:** Es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica [23].

**IP43:** El estándar IP (*International Protection*, 'Protección Internacional' en castellano) de un LED es aquel que define su grado de protección anti-polvo y humedad. Sirve para orientar sobre su colocación, o no, en diversas estancias, tanto exteriores como interiores. El IP43, concretamente, señala que protege de objetos mayores de 1 mm y del goteo de agua desde un ángulo de 60 grados [\[24\]](#).

**IPA:** El formato de archivos .ipa es el formato utilizado para las aplicaciones de Apple en los dispositivos iPhone, iPod Touch y iPad [\[25\]](#).

**JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico [\[26\]](#).

**LTE:** Acrónimo de *Long Term Evolution*. Es un estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad para teléfonos móviles y terminales de datos. El 3GPP está definida por unos como una evolución de la norma 3GPP UMTS (3G) y por otros como un nuevo concepto de arquitectura evolutiva (4G) [\[27\]](#).

**Material Design:** Es una normativa de diseño enfocado en la visualización del sistema operativo Android, además en la web y en cualquier plataforma [\[28\]](#).

**Mockup:** Un *mockup*, mock-up, o maqueta es un modelo a escala o tamaño real de un diseño o un dispositivo, utilizado para la demostración, evaluación del diseño, promoción, y para otros fines [\[29\]](#).

**ngrok:** ngrok es un proxy inverso que crea un túnel seguro desde un *endpoint* público hacia un servicio web corriendo localmente. ngrok captura y analiza todo el tráfico que pasa por el túnel para luego inspeccionarlo y reproducirlo [\[30\]](#).

**No-IP:** No-IP es un proveedor de servicios pagos y gratuitos de DNS dinámico [\[31\]](#).

**Open Source:** El código abierto es un modelo de desarrollo de *software* basado en la colaboración abierta. Se enfoca más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el *software* libre. Para muchos el término libre hace referencia al hecho de adquirir un *software* de manera gratuita [\[32\]](#).

**PaaS (*Platform as a Service*):** Es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en un entorno conteniendo una pila básica de sistemas, componentes o APIs preconfiguradas y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un sistema Linux, un servidor web, y un ambiente de programación como Perl o Ruby) [\[33\]](#).

***Product Breakdown Structure*:** En la gestión de proyectos bajo la metodología PRINCE2, es una herramienta para analizar, documentar y comunicar los resultados de un proyecto, y forma parte de la técnica de planificación basada en el producto [\[34\]](#).

**RFID:** RFID o identificación por radiofrecuencia (del inglés *Radio Frequency Identification*) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio [\[35\]](#).

**SD:** Es un dispositivo en formato de tarjeta de memoria para dispositivos portátiles, por ejemplo: cámaras digitales (fotográficas o videograbadoras), teléfonos móviles, teléfonos inteligentes, computadoras portátiles y videoconsolas (de sobremesa y portátiles), tabletas y entre muchos otros [\[36\]](#).

***Script*:** En informática, un *script*, archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo general se almacena en un archivo de texto plano. Los *scripts* son casi siempre interpretados, pero no todo programa interpretado es considerado un *script*. El uso habitual de los *scripts* es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los intérpretes de órdenes sean a la vez intérpretes de este tipo de programas [\[37\]](#).

**SOLID:** En ingeniería de *software*, SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion) es un acrónimo mnemónico

introducido por Robert C. Martin a comienzos de la década del 2000 que representa cinco principios básicos de la programación orientada a objetos y el diseño [\[38\]](#).

**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 [\[39\]](#).

**TypeScript:** Es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases [\[40\]](#).

**Unix:** Registrado oficialmente como UNIX®, es un sistema operativo portable, multitarea y multiusuario; desarrollado en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Arturo Roman, Cristoph Edmond [\[41\]](#).

**UI (*User Interface*):** La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo [\[42\]](#).

**Waypoint:** Es una coordenada para ubicar puntos de referencia tridimensionales utilizados en la navegación basada en GPS. La palabra viene compuesta del inglés *way* y *point*. Los *waypoints* se emplean para trazar rutas mediante agregación secuencial de puntos [\[43\]](#).

# Índice

1. Introducción	23
1.1. Cliente	23
1.2. Proyecto seleccionado	24
1.3. Objetivos del proyecto	24
1.4. Estructura del documento	26
2. Problema y solución	28
2.1. El problema y su contexto	28
2.2. ¿Qué es el abigeato?	29
2.3. ¿Qué es una zona de riesgo para un animal?	29
2.4. Principales interesados	30
2.5. Descripción de la solución	31
2.6. Beneficios de utilizar drones en el campo	33
2.7. Componentes de la solución	33
2.8. Plan de vuelo	34
3. Marco metodológico	35
3.1. Características del proyecto	35
3.2. Ciclo de vida del proyecto	35
3.3. Artefactos	38
3.3.1. Project backlog	38
3.3.2. Iteration backlog	38
3.4. Ceremonias	39
3.4.1. Reunión de planificación	39
3.4.2. Demo con el cliente	39
3.4.3. Reunión de retrospectiva	40
3.4.4. Reunión de alineamiento	41
	13

3.5. Expectativas de un miembro del equipo	42
3.6. Roles	42
3.7. Relación con el cliente	43
4. Ingeniería de requerimientos	44
4.1. Relevamiento	44
4.1.1. Observación empírica	45
4.1.2. Entrevistas con expertos de negocio	45
4.1.2.1. Entrevistas con José Luis Callero	46
4.1.2.2. Entrevistas con Rodrigo Pereira	51
4.1.3. Reuniones con el cliente	53
4.1.4. Benchmarking	55
4.1.4.1. SmartXFarm	56
4.1.4.2. G&D Argentina	56
4.1.4.3. ACGDrone	56
4.1.5. Prototipos	57
4.2. Especificación	57
4.2.1. Requerimientos funcionales	57
4.2.2. Priorización de los requerimientos funcionales	59
4.2.3. Requerimientos no funcionales	61
4.2.4. Restricciones	61
4.3. Validación	62
4.3.1. Prototipos aplicación móvil	62
4.3.2. Prototipos aplicación web	66
4.4. Entregables	69
5. Investigación	71
5.1. Investigación de drones	71

5.1.1. DJI Inspire 2	72
5.1.1.1. Descripción	73
5.1.1.2. Problemas	73
5.1.2. DJI S1000 Plus	74
5.1.2.1. Descripción	74
5.1.2.2. Problemas	74
5.1.3. DJI Mavic 2 Enterprise	75
5.1.3.1. Características	75
5.1.4. DJI Matrice 100	76
5.1.4.1. Descripción	76
5.1.4.2. Características	76
5.1.4.3. Especificaciones técnicas	77
5.1.5. Empresas Parrot y Yuneec	77
5.1.6. DJI Matrice 200	78
5.1.6.1. Características	78
5.1.7. Placas ArduPilot	79
5.1.7.1. Características	80
5.1.8. Accsys - Sky Track	81
5.1.8.1. Características del dron	82
5.2. Tipo de caravana RFID a utilizar	83
5.2.1. Tipos de tags	84
5.3. Procesamiento de imágenes	85
5.3.1. Google Cloud Vision API	85
5.3.1.1. Precio de la API	86
5.3.1.2. Instalación	86
5.3.1.3. Prototipo	86

5.4. Digitalización de planos	88
6. Arquitectura	90
6.1. Descripción general	90
6.2. Diseño	93
6.3. Atributos de calidad	101
6.3.1. Modificabilidad	102
6.3.2. Confiabilidad	105
6.3.3. Seguridad	107
6.3.4. Performance	108
6.3.5. Testeabilidad	108
6.3.6. Interoperabilidad	109
6.3.7. Portabilidad	109
6.4. Tecnologías	110
6.4.1. Web API REST	111
6.4.2. Aplicaciones de frontend	112
6.4.2.1 Aplicaciones móviles	112
6.4.2.2 Aplicación web	112
6.4.3. Aplicación del dron (SkyGuard-Copter)	113
6.4.4. Aplicación de instrucciones al dron (SkyGuard-Pathfinder)	114
6.4.5. Base de datos	114
7. Desarrollo de software	115
7.1. Herramientas de desarrollo	115
7.1.1. Frontend	115
7.1.2. Backend	116
7.1.3. Integración con el dron	116
7.2. Flujo de trabajo	117

7.2.1. Modelo de ramas: GitFlow	117
7.2.2. Pull Requests	118
7.3. Despliegue	119
7.4. Distribución de responsabilidades	119
8. Usabilidad	121
8.1. Heurísticas de Nielsen	121
8.2. Material Design	121
8.2.1 Principios	122
8.3. Resultados	122
8.3.1. Análisis de las heurísticas de Nielsen	123
8.3.2. Análisis de Material Design	125
9. Plan de vuelo del dron	131
9.1. Algoritmo	131
9.1.1. Problema	131
9.1.2. Métricas de éxito	131
9.1.3. Entradas	131
9.1.4. Salida	132
9.1.5. Consideraciones	132
9.1.6. Proceso	134
9.1.7. Implementación	134
9.1.8. Futuros pasos	136
10. Gestión del proyecto	137
10.1. Gestión del alcance	137
10.1.1. Definición del alcance	137
10.1.2. Plan de release	140
10.1.2.1. Número de release: 1	140

10.1.2.2. Número de release: 2	141
10.1.2.3. Número de release: 3	142
10.2. Gestión del esfuerzo	142
10.2.1. Métricas de esfuerzo	143
10.2.1.1. Esfuerzo por actividad en el proyecto	144
10.2.1.2. Total de horas trabajadas	147
10.2.1.3. Dedicación de horas por persona	148
10.2.1.4. Horas de desarrollo por iteración	149
10.2.1.5. Horas de desarrollo por persona	150
10.2.1.6. Horas de documentación por persona	151
10.2.1.7. Cumplimiento de los objetivos por iteración	152
10.2.1.8. Desviación entre esfuerzo estimado y esfuerzo real en las tareas de ingeniería	152
10.3. Gestión de riesgos	153
10.3.1. Identificación	154
10.3.2. Actividades preventivas	155
10.3.3. Seguimiento y evolución	158
10.4. Ejecución de iteraciones	161
10.4.1. Velocidad	161
10.4.2. Hitos del proyecto	163
10.5. Gestión de la comunicación	164
10.5.1. Espacio de trabajo en Slack	164
10.5.2. Calendario de Google	165
10.6. Conclusiones	165
11. Calidad	167
11.1. Objetivos de calidad	167
11.1.1. Objetivos de calidad del proceso	167

11.1.2. Objetivos de calidad del producto	167
11.1.3. Objetivos de calidad académicos	168
11.2. Plan de calidad	168
11.3. Aseguramiento de la calidad	171
11.3.1. Aplicación de estándares	171
11.3.1.1. Estándares de codificación	171
11.3.1.2. Estándares de documentación	171
11.3.2. Pruebas de software	172
11.3.2.1. Pruebas cruzadas	172
11.3.2.2. Pruebas de integración entre el hardware y el software	173
11.3.2.3. Pruebas con usuarios	175
11.3.2.4. Pruebas de conectividad	176
11.3.2.5. Pruebas de satisfacción con el cliente	176
11.4. Gestión de incidentes	177
11.5. Revisiones	178
11.6. Retrospectivas	178
11.7. Validación	179
11.8. Gestión de la configuración	179
11.8.1. Identificación de los elementos de configuración	179
11.8.2. Elección de las herramientas	180
11.8.2.1. Software	181
11.8.2.2. Documentación	183
11.9. Métricas de calidad	185
11.9.1. Métricas del proceso	185
11.9.2. Métricas del producto	186
12. Dificultades del proyecto	189

12.1. Rechazo de la solicitud de financiación	189
12.2. Necesidad de cambio de proveedor de dron	190
12.3. Demora en la compra y construcción del dron final	190
12.4. Instalación y configuración de placa al dron de prueba	192
12.5. Cortocircuito de placa durante la instalación	193
12.6. Necesidad de repuestos debido a la rotura de componentes	193
12.7. Necesidad de buen clima para realizar pruebas	194
13. Conclusiones	196
13.1. Estado actual	196
13.2. Conclusiones generales	196
13.3. Lecciones aprendidas	197
13.4. Próximos pasos	198
14. Referencias bibliográficas	200
Anexo 1: Investigación de RFID	210
Anexo 2: Registro de dedicación de horas por iteración	213
Anexo 3: Tabla de análisis de riesgos	214
Anexo 4: Documentos de financiación entregados a CI <sup>2</sup>	218
Anexo 5: Análisis de herramientas para seguimiento de horas	242
ClickUp	242
Toggl	242
Bitrix24	243
Clockify	244
Conclusión	245
Anexo 6: Historial de retrospectivas	246
Retrospectiva 28 de mayo de 2019	246
Retrospectiva 15 de junio de 2019	249

Retrospectiva 25 de junio de 2019	251
Retrospectiva 09 de julio de 2019	254
Retrospectiva 25 de julio de 2019	256
Retrospectiva 06 de agosto de 2019	258
Retrospectiva 12 de setiembre de 2019	260
Retrospectiva 26 de setiembre de 2019	262
Retrospectiva 10 de octubre de 2019	264
Retrospectiva 24 de octubre de 2019	267
Retrospectiva 07 de noviembre de 2019	268
Retrospectiva 24 de noviembre de 2019	270
Retrospectiva 09 de diciembre de 2019	272
Retrospectiva 23 de diciembre de 2019	273
Anexo 7: Informe de revisiones	275
Informe revisión 1	275
Fortalezas del Grupo	275
Oportunidades de Mejora	275
Acciones a Realizar	276
Informe revisión 2	278
Fortalezas del Grupo	278
Oportunidades de Mejora	278
Acciones a Realizar	278
Informe revisión 3	280
Fortalezas del Grupo	280
Oportunidades de Mejora	280
Acciones a Realizar	281
Anexo 8: Mejoras propuestas por el cliente	283

Anexo 9: Revisión de código	285
Anexo 10: Metodología de trabajo	286
Anexo 11: Empresas de drones investigadas	293

# 1. Introducción

El presente documento tiene como objetivo describir el desarrollo del proyecto SkyGuard realizado como requisito para la obtención del título de Ingeniero en Sistemas de la Universidad ORT Uruguay.

El mismo fue desarrollado por Dan Blanco, Gastón Donadío, Victoria Rocha, Agustín Rodríguez y Emiliano Rodríguez durante el período marzo 2019 - marzo 2020.

## 1.1. Cliente



Ilustración 1: Logo de la empresa TodoSoft

El proyecto SkyGuard consiste en un sistema para monitoreo de ganado utilizando drones y tecnología RFID. El cliente es la empresa Todosoft.

Todosoft es una empresa uruguaya con más de 15 años de trayectoria dedicada a desarrollar e implantar soluciones y productos tecnológicos. Desde 2001 es parte de la CUTI [\[44\]](#) (Cámara Uruguaya de Tecnologías de la Información) y ha implementado el Sistema de Gestión de la Calidad bajo las normas ISO 9001:2000 lo cual le permite brindar servicios de excelencia con capacidad de adaptación a las cambiantes exigencias del mercado.

El equipo acordó con el cliente, una vez finalizado el proyecto final de carrera, entregar el código, la documentación y los derechos de comercialización del producto así como también el *hardware* que se utilizó para las pruebas.

## 1.2. Proyecto seleccionado

El proyecto SkyGuard fue presentado a Universidad ORT por parte del equipo ya que se conocía el interés de la empresa por desarrollar un sistema de estas características.

El alcance del proyecto y las funcionalidades fueron definidas por el equipo luego de una etapa inicial de investigación ya que el cliente solo tenía la idea general de desarrollar un sistema de control de ganado utilizando drones y tecnología RFID. Los conceptos resultantes de la investigación fueron luego validados con el cliente y expertos de negocio antes de transformarse en el producto final implementado.

El equipo seleccionó este proyecto por tratarse de un producto innovador y un proyecto en el que se contaba con la libertad de elegir las tecnologías, relevar los requerimientos con expertos, armar el diseño de la solución, la interacción de *software* con *hardware*, además de programar y volar drones. El nivel de dificultad fue un motivador adicional pues le imprimía al proyecto un elemento de desafío que el equipo quería afrontar.

El proyecto resultaba desafiante para el equipo ya que involucra el uso de una tecnología que era desconocida para la mayoría de los integrantes (RFID), un contexto de negocio diferente, incorporación e integración de elementos de *hardware* y mucho trabajo de investigación de tecnologías de *software* y *hardware* de última generación.

## 1.3. Objetivos del proyecto

En conjunto se definieron los objetivos del proyecto de forma tal que cumplieran con las expectativas que cada integrante tenía. Los mismos fueron una guía a lo largo del desarrollo del proyecto y se utilizaron como forma de evaluar el éxito del mismo.

Los objetivos definidos se dividen en las siguientes categorías:

### **Objetivos académicos:**

- **Aprobar el proyecto final de carrera con excelencia:** El equipo se propuso aprobar el proyecto de fin de carrera de manera exitosa, realizando un trabajo de calidad que pueda ser reconocido por personas externas al equipo.

- **Aplicar los conocimientos obtenidos en la carrera:** Para cerrar de la mejor manera posible la carrera cursada en la Universidad ORT, se considera fundamental poder aplicar los conocimientos adquiridos en la misma. Es muy importante poder identificar capacidades adquiridas en los años de estudio y sentir la satisfacción de poder aplicarlas en un proyecto propio.

#### **Objetivos de proceso:**

- **Aprender a gestionar un proyecto real:** Siendo un equipo cuyos integrantes no habían trabajado con anterioridad en conjunto, fue un desafío poder gestionar el proyecto de buena manera. Definimos como buena gestión llegar con al menos un 90 % del alcance planteado, en tiempos que no se desfasen en más de un 20 % de las tareas estimadas, logrando definir y cumplir con una metodología que se encuentra documentada, pudiendo evidenciar el cumplimiento de la misma y permitiendo satisfacer los objetivos planteados.
- **Lograr una organización del equipo óptima:** Se propuso lograr una buena organización del equipo, trabajar en conjunto, aprender del resto de los integrantes del equipo brindando cada uno la mayor predisposición con el fin de enriquecerse mutuamente. Su cumplimiento se medirá dos semanas antes de finalizar el proyecto a través de una encuesta que realizarán los miembros del equipo.

#### **Objetivos del producto:**

- **Conocer el negocio y el contexto:** El equipo se propuso conocer el negocio y el contexto del problema de forma tal que, al plantear una solución en base a los lineamientos del cliente, la misma satisfaga tanto sus necesidades como las de los dueños de los campos, sea validado por los expertos de negocio como una solución real y permita identificar el *hardware* adecuado para implementar la solución.
- **Desarrollar la totalidad de los requerimientos de prioridad alta relevados en una versión Beta del producto:** Brindando verdadero valor a los usuarios finales del mismo.

- **Obtener la aprobación del cliente:** Es fundamental para nosotros poder trabajar en conjunto al cliente, cumpliendo con sus expectativas, dejando así una buena impresión de la Universidad, sus métodos y el trabajo con sus estudiantes.

## 1.4. Estructura del documento

A continuación se presentan los capítulos contenidos en este documento junto con una breve descripción del contenido de cada uno.

**Capítulo 2:** Problema y solución: Contiene una descripción del contexto general en donde se produce el problema que el proyecto busca atacar, cuáles son aquellos puntos del problema que SkyGuard resuelve y cuáles no resuelve, así como de qué manera y con qué tecnologías lo hace.

**Capítulo 3:** Marco metodológico: Se describe el ciclo de vida utilizado durante el proyecto, la metodología de trabajo utilizada a lo largo del mismo y los diferentes roles que se tomaron dentro del equipo.

**Capítulo 4:** Ingeniería de requerimientos: Se detallan las actividades y tareas llevadas a cabo para el relevamiento, especificación y validación de los requerimientos del sistema, tanto funcionales como no funcionales.

**Capítulo 5:** Investigación: Se detallan las investigaciones realizadas a lo largo de la tesis. Las mismas requirieron un esfuerzo importante en el proyecto, y a partir de ellas se tomaron decisiones importantes.

**Capítulo 6:** Arquitectura: Define la arquitectura del sistema SkyGuard, detallando las decisiones de diseño tomadas en función de la priorización de determinados atributos de calidad. Se incluye el análisis de las tecnologías utilizadas.

**Capítulo 7:** Desarrollo de *software*: Se describe cómo se llevó a cabo el desarrollo del *software*, siendo el mismo la etapa del proyecto que más tiempo requirió.

**Capítulo 8:** Usabilidad: Se detalla cómo se estudió la usabilidad del sistema en torno a la experiencia de usuario, los análisis de usabilidad y los impactos de los mismos en las pantallas de las aplicaciones de *frontend*.

**Capítulo 9:** Plan de vuelo del dron: Se describe el plan de vuelo desarrollado para identificar el camino que el dron debe seguir para recorrer un campo de manera segura y confiable.

**Capítulo 10:** Gestión del proyecto: Se especifican los principales hitos del proyecto así como la metodología de gestión que el equipo decidió llevar adelante junto con las métricas obtenidas y los riesgos analizados.

**Capítulo 11:** Calidad: Se describen los objetivos de calidad del proyecto, cómo se asegura la calidad, los estándares seguidos y las pruebas realizadas al sistema.

**Capítulo 12:** Dificultades del proyecto: Se documentan todas las dificultades no planeadas que el equipo tuvo a lo largo del proyecto. Impedimentos, roturas de hardware, demoras externas, entre otras.

**Capítulo 13:** Conclusiones: Se presentan las conclusiones obtenidas, los objetivos cumplidos y los que no (expresando las razones correspondientes), las lecciones aprendidas por el equipo a lo largo del desarrollo del proyecto así como también el trabajo a futuro.

**Capítulo 14:** Referencias bibliográficas: Se presentan las referencias bibliográficas que el equipo consumió durante el desarrollo del proyecto.

## 2. Problema y solución

A continuación se presenta el problema que existe actualmente junto con su contexto y la solución desarrollada por el equipo.

### 2.1. El problema y su contexto

Actualmente uno de los principales problemas de los campos dedicados a la ganadería extensiva, es el abigeato y la muerte de animales por el ingreso de los mismos a zonas definidas como peligrosas. Esto se traduce en pérdidas económicas importantes.

Para evitar estas situaciones, hoy en día los peones de los campos dedican muchas horas al conteo y verificación de la integridad del ganado de manera de detectar, de forma temprana, entre otras cosas, la falta de ganado por extravío o robos.

Los capataces definen las zonas a monitorear en los campos, los peones realizan las recorridas, pero si las condiciones climáticas no son apropiadas, se dificulta la realización de los controles provocando que los mismos en ocasiones no puedan ser realizados.

En campos de más de 500 hectáreas hay personal destinado únicamente a realizar estos controles pero aun así no se logra evitar la pérdida de los animales.

Si un animal desaparece es difícil detectarlo y más aún saber cómo y dónde buscarlo debido a que los dueños de campos no tienen una herramienta de monitoreo de su ganado ni ubicación de los mismos (dificultad que se ve acentuada al aumentar la extensión del campo en cuestión). Además, en ocasiones, los terrenos son difíciles de transitar para una persona y bajo determinadas situaciones climatológicas es hasta peligroso.

En Uruguay, en el año 2018 fueron reportados 1362 casos de abigeato, lo cual representa un 13 % más que en el año 2017 según estudio realizado en noviembre de 2018 [\[45\]](#). Para tener más claridad sobre la magnitud del problema, dicho estudio estima que se pierden cerca de 10 millones de dólares al año por hurto de ganado.

A su vez en un estudio realizado en 2015 [46] también mostraba una tendencia creciente respecto al abigeato destacando que no más de 6 % de los mismos, para los años 2015, 2014 y 2013 fue aclarado.

## 2.2. ¿Qué es el abigeato?

El abigeato es un delito punible y estipulado en muchos códigos legales de la mayoría de los países ganaderos y que consiste en el robo o hurto de ganado o animales domésticos, principalmente caballos y vacas, aunque también se da en ovinos.

El ladrón del ganado recibe el nombre de cuatrero o abigeo. Este delito aún se da en las áreas ganaderas de todo el mundo, en especial si aumenta el precio de la carne.

Actualmente en Uruguay el abigeato ha llevado a que se limite la cría de ovejas en las cercanías de los pueblos rurales. Se realizan hasta 25.000 denuncias al año.

Los ladrones suelen matar los animales para quitarles la carne, mayoritariamente. El abigeato más frecuente en Uruguay consiste en que el o los ladrones, carnean al animal enteramente en el lugar del robo o zona cercana, en menos de 30 minutos, y entierran su cuero en las cercanías del lugar del hecho.

## 2.3. ¿Qué es una zona de riesgo para un animal?

Una zona de riesgo para un animal en un campo es una zona definida por el capataz de campo como peligrosa y por la cual los animales no deberían circular debido a que corre en riesgo su vida.

Las zonas peligrosas pueden ser:

- Zonas de **inundaciones o cañadas** ya que los animales podrían quedar atrapados allí y terminar muriendo si no son encontrados a tiempo.
- Zonas **cercanas a una ruta** ya que un animal podría escaparse y un auto chocar contra él, lo cual es muy peligroso tanto para los pasajeros de los vehículos, como para el animal. Incluso la responsabilidad del accidente recae sobre el dueño de campo.

- Zonas con **jaurías de perros** ya que los mismos podrían atacar a los animales y terminar muriendo ya que esto es muy difícil de detectar a tiempo por los peones en las recorridas diarias.

## 2.4. Principales interesados

Los principales interesados de la solución son:

- Empresa Todosoft
- Estancieros
- Capataces de campo
- Peones de campo

La empresa **Todosoft** es el cliente, es decir, es para quien el proyecto fue desarrollado; la misma será quien continuará con el proyecto hasta llegar a una versión final del producto para poder ser vendido a los estancieros.

Los **estancieros** son los propietarios de una estancia, es decir, un establecimiento rural dedicado a la cría de ganado. Es el principal interesado en resolver el problema ya que es quien pierde dinero con el abigeato y la muerte de los animales. Será quien pagará por el sistema y se verá beneficiado económicamente de su implementación. Entendemos que será el principal usuario en lo relativo a reportes y consultas de gestión.

El **capataz** de campo es la persona encargada del control y administración del campo así como de las personas que trabajan en el mismo. Será el principal usuario del sistema, principalmente en lo relativo a las funciones operativas del mismo.

El **peón** de campo es un trabajador rural que se dedica a tareas de control de campo y de animales. El sistema a implementar afectará directamente sus tareas diarias.

## 2.5. Descripción de la solución



Ilustración 2: Esquema de la solución

La solución propone ayudar a disminuir la muerte de los animales por estar en zonas definidas como peligrosas así como también uno de los tipos de abigeato que consiste en el robo del ganado de forma lenta.

El robo de ganado de forma lenta se basa en el hurto de pocas cabezas de ganado y cada determinada cantidad de días, de forma de evitar ser detectados. Para esto se suele romper una parte del alambrado del campo y dejar que los animales se vayan por ahí para posteriormente capturarlos fuera del campo.

La solución planeada para resolver el problema utiliza un sistema de drones que sobrevuelan los campos equipados con lectores RFID que permite escanear los animales y a partir de esto generar reportes y alertas a los estancieros y/o capataces al finalizar la recorrida, notificando los animales detectados y en caso de detectarse algún problema como un animal fuera de la

zona permitida, un animal no encontrado o un animal en una zona peligrosa, se envía una alerta.

Para la realización del escaneo de los animales, los mismos deben tener colocado en su oreja un tag RFID específico para ser leído por la lectora montada en el dron.

En cada campo se genera un plan de vuelo inteligente que le permite al dron recorrerlo de forma óptima teniendo en cuenta la distribución del mismo, la velocidad del dron y el alcance de la lectora RFID.

El plan de vuelo se define para determinados días y horarios, y el dron se encarga de ejecutarlo según lo estipulado enviando los datos a un sistema que los procesa y genera los reportes y alertas.

Los datos se envían por WiFi o 3G mientras el dron está sobrevolando el campo, en caso de tener la señal suficiente para hacerlo. De lo contrario, la sincronización de los datos se realiza cuando el dron llega a la base.

Los reportes generados por el sistema contienen las recorridas del dron, los animales leídos, los animales no leídos y las alarmas disparadas.

El sistema web también permite la visualización de un mapa de calor que le permite al usuario saber por qué zonas se mueven los animales dentro del campo.

La solución implica no solo un producto de *software* sino que también una solución de *hardware* que debe interactuar con el mismo cumpliendo con las restricciones que se mencionan en el [capítulo 5.1](#) para el caso del *hardware* del dron pero también existen otros dispositivos de *hardware* que deben interactuar entre sí y que son necesarios para que la solución cumpla con todas las funcionalidades. Estos otros dispositivos son, la lectora y *tags* RFID que se especifican en el [capítulo 5.2](#) y la placa programable que utiliza el dron que se especifica en el [capítulo 5.1.7](#).

## 2.6. Beneficios de utilizar drones en el campo

El uso de drones para la vigilancia y monitoreo del ganado en campos permite la optimización de recursos, permitiendo el control de grandes extensiones de terreno desde el aire, llegando a lugares de difícil acceso para una persona.

Estudios de mercado demuestran que el uso de drones viene creciendo y lo seguirá haciendo en los próximos años, siendo más del 60 % de los drones vendidos para su uso en campos [\[47\]](#).

Existen varios beneficios en el uso de drones para el campo, los más destacables para los fines del presente trabajo son:

- Mirada del campo desde otro punto de vista: el dron puede dar una mirada general desde el aire al mismo tiempo que se pueden obtener datos de cada animal.
- Posibilidad de recorrer lugares de difícil acceso: zonas complicadas de acceder para una persona ya sea debido a inundaciones, cañadas, entre otros.
- Alertas sobre animales que pueden encontrarse en peligro: ya sea porque salieron de su zona y se encuentran atrapados en una zona peligrosa o porque fueron robados.
- Búsqueda de situaciones puntuales: un ejemplo la búsqueda de un animal. El dron permite recorrer un campo de 500 hectáreas en 1 hora y 15 minutos aproximadamente.
- Conteo e identificación del ganado utilizando la tecnología RFID.
- Análisis de información a través del procesamiento de imágenes: el dron puede tomar imágenes con una buena calidad dependiendo del tipo de cámara usada.
- Seguridad: ya que se puede vigilar el campo utilizando drones.
- Análisis del suelo por el que se mueven los animales: creando un mapa de calor con el movimiento de los animales se pueden elaborar estadísticas para conocer el estado de las pasturas.

## 2.7. Componentes de la solución

La solución SkyGuard está formada por los siguientes componentes:

- SkyGuard-Core: *Backend* encargado de la lógica de negocio del sistema así como la persistencia de los datos.
- SkyGuard-UI: *Frontend* desarrollado para permitir la interfaz con el usuario de la aplicación móvil. Interactuando con el *backend* para la persistencia y manejo de la lógica. Funciona bajo las plataformas Android e iOS.
- SkyGuard-Web: Aplicación web para la gestión del sistema por parte de los usuarios administradores. Interactúa con el *backend* para el procesamiento de la información.
- SkyGuard-Copter: Aplicación que envía las instrucciones al dron para ejecutar un plan de vuelo en determinado momento.
- SkyGuard-Pathfinder: Aplicación que dado un campo genera el plan de vuelo para que el dron lo ejecute a través de SkyGuard-Copter.

## 2.8. Plan de vuelo

El plan de vuelo consiste en generar un listado de puntos geográficos por los cuales el dron debe pasar durante su recorrida, de manera de cubrir con precisión un determinado campo.

La buena definición del plan de vuelo hace a SkyGuard un sistema eficiente y seguro permitiendo:

- Terminar la recorrida cumpliendo con el objetivo de monitorear todo el campo.
- Realizar el trabajo en el menor tiempo posible.
- No tener colisiones durante el vuelo.
- Volver a la base y quedar preparado para la siguiente recorrida.

El perímetro del campo será el que definirá la forma de realizar el trabajo por parte del dron.

Ante cualquier imprevisto durante el vuelo tal como una colisión, poca batería u otra situación que le impida al dron seguir volando, el mismo volverá a la base.

Más adelante, en el [capítulo 9](#) se describe en detalle el plan de vuelo.

### 3. Marco metodológico

Al comenzar el desarrollo de este proyecto una de las primeras cosas que se hizo como equipo fue definir una metodología de trabajo en la que todos los integrantes estuvieran de acuerdo. Dicha metodología se estableció con el objetivo de que fuese una guía y un conjunto de reglas a seguir durante el desarrollo del proyecto.

En el presente capítulo se detalla la metodología que el equipo seleccionó así como la evolución que la misma tuvo.

Se menciona también la forma de comunicación del equipo, el ciclo de vida utilizado y los roles de los integrantes dentro del equipo de proyecto.

Más detalles se encuentran en el [anexo 10](#).

#### 3.1. Características del proyecto

El equipo se basó en las siguientes características del proyecto para la elección del ciclo de vida:

- Requerimientos no definidos por parte del cliente.
- Desconocimiento del negocio por parte del equipo.
- Desconocimiento de algunas de las tecnologías a utilizar.
- El equipo nunca trabajó junto por lo tanto no conocemos la velocidad.
- Duración del proyecto.
- Relación con el cliente.

#### 3.2. Ciclo de vida del proyecto

En base a las características mencionadas, el equipo analizó las opciones de los modelos Evolutivos e Incremental iterativos, considerando que ambos permiten:

- Feedback temprano del cliente y de los usuarios
- Entrega continua y temprana de un producto funcional

La diferencia entre ambos, es que un enfoque incremental implica definir un alcance de alto nivel de todo el proyecto, para así poder diseñar una arquitectura relativamente estable sobre la que construir los sucesivos incrementos.

Considerando que uno de los requerimientos del proyecto incluye la recomendación del *hardware* a utilizar, en particular en cuanto al dron y sus componentes, y que además es necesario realizar una inversión importante para adquirir dicho componente como elemento necesario para el desarrollo y pruebas del propio proyecto, es que optamos por el enfoque incremental frente al evolutivo.

Si bien este enfoque implica un tiempo más largo de relevamiento al comienzo del proyecto, permite identificar los requerimientos de alto nivel que deberán tenerse en cuenta a la hora de definir la arquitectura y el *hardware* relacionado, disminuyendo el nivel de riesgo en la inversión realizada a la hora de comprar el dron y dispositivos adicionales. Con un enfoque evolutivo, el riesgo de descubrir en forma tardía un requerimiento que pudiera implicar que el equipo comprado no fuera apropiado y hubiera que comprar otro, implicaría una pérdida económica para el cliente, además de un impacto muy fuerte en los cortos plazos del proyecto, considerando el tiempo requerido para el proceso de selección, importación, configuración y puesta en marcha del equipamiento.

En base a estas consideraciones, se selecciona utilizar un ciclo de vida **iterativo e incremental** para el desarrollo del proyecto ya que el mismo tiene un nivel de incertidumbre alto por lo que es difícil tener un plan definido desde el inicio.

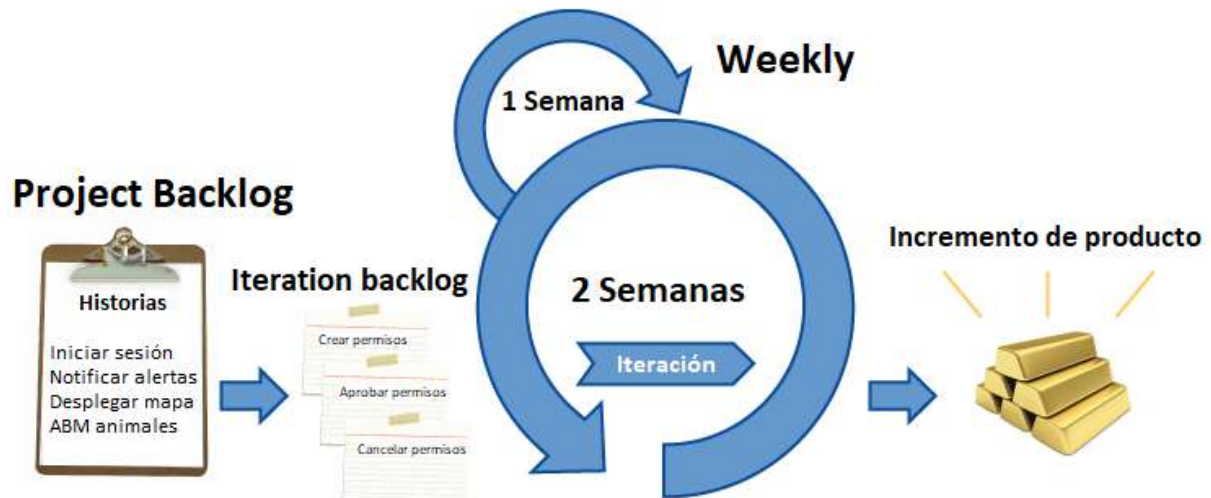


Ilustración 3: Eventos de la metodología

El mismo consiste en liberar al cliente periódicamente partes del producto de software, agregando cada vez más funcionalidad al sistema. A su vez, en cada iteración se revisa y mejora el producto, consiguiendo así una versión más estable del producto en cada iteración [48].

La iteración se definió de **2 semanas** de manera de tener suficiente tiempo para trabajar en agregar valor al producto. Fue un período que resultó adecuado, ya que el equipo pudo adaptarse fácilmente, logrando incrementos del producto en el tiempo justo.

Este ciclo de vida permite

- La **intervención del cliente a lo largo de todo el proyecto**, a través de las revisiones periódicas del mismo.
- Una **rápida adaptación a los cambios** que surjan como resultado del punto anterior. Sobre todo porque los requerimientos no están claros, el negocio es desconocido para el equipo así como parte de las tecnologías a utilizar.
- **Entregas progresivas y funcionales** del producto final en cada iteración, lo cual permite un mejor manejo de las expectativas del cliente.

Para implementar esta estrategia, que está fuertemente inspirada en Scrum [49], tomamos un conjunto de elementos, estrategias y herramientas de las metodologías ágiles [50] como las que se detallan a continuación.

### 3.3. Artefactos

#### 3.3.1. *Project backlog*

Está compuesto por el *product backlog* más las tareas de gestión del proyecto, calidad, soporte y entregables adicionales al producto.

Se denomina *product backlog* a todos los requerimientos del producto, especificados en formato de historias de usuario, es decir, una breve descripción de la funcionalidad, contada desde la perspectiva del usuario e indicando el objetivo.

El mismo está visible para todos los integrantes del equipo y permite tener una visión general de todo lo que se espera realizar.

Se asignan prioridades sobre el *project backlog*, con base en las necesidades del cliente y la complejidad de cada historia de usuario o tarea.

#### 3.3.2. *Iteration backlog*

Una vez definido el *project backlog* inicial y priorizado, en la ceremonia de planificación el equipo determina qué puede completar durante la iteración que comienza. Esos ítems se mueven del *project backlog* al *iteration backlog*.

Para ello, cada historia de usuario se divide en una o más tareas, así el equipo puede tener una visión más clara del trabajo a realizar.

## 3.4. Ceremonias

### 3.4.1. Reunión de planificación

La reunión de planificación es una ceremonia en donde participa todo el equipo, se realiza al finalizar la iteración, con el objetivo de revisar el trabajo terminado y pendiente de la iteración actual, analizar los resultados y planificar las tareas a realizar en la iteración que comienza.

El tiempo de esta reunión está definido en 1 hora pero puede extenderse en caso de ser necesario.

El objetivo es mover historias del *project backlog* al *iteration backlog* de acuerdo a lo que puede abarcar el equipo en base a su velocidad.

### 3.4.2. Demo con el cliente

Es una reunión informal realizada con el cliente periódicamente de manera de mostrarle los avances del producto y recibir retroalimentación sobre el desarrollo del mismo para ajustar el producto en caso de ser necesario.

Participa todo el equipo más el cliente. En la misma se explican los ítems del *project backlog* que fueron finalizados y los que no.

El equipo responde a preguntas técnicas que el cliente pueda tener relacionadas con el desarrollo del producto.

Se revisa todo el *project backlog* junto con la estimación de cada historia y se modifica lo que sea necesario.

Como resultado de la demo se obtiene retroalimentación del cliente junto con un *project backlog* revisado que refleja los objetivos del negocio.

### 3.4.3. Reunión de retrospectiva

Es una reunión que se realiza al finalizar la iteración con el objetivo de mirar hacia atrás a eventos o situaciones pasadas para entender lo que el equipo hizo bien y lo que quiere mejorar.

La reunión de retrospectiva es una oportunidad para que el equipo se inspeccione a sí mismo y cree un plan para que se realicen mejoras durante la siguiente iteración y así enfocarse en la mejora continua.

Se realiza el día siguiente a la finalización de la iteración.

La modalidad es la siguiente:

1. Cada miembro del equipo le asigna un puntaje a la iteración pasada, entre 1 y 10.

**Duración:** 5 minutos

2. Cada miembro del equipo plantea:
  - a. Cosas buenas de la iteración
  - b. Cosas malas de la iteración
  - c. Ideas

**Duración:** 5 minutos

3. Votación a las cosas malas de la iteración que interesa enfocarse en mejorar en la siguiente iteración. Cada uno tiene 5 puntos para asignar en total.

**Duración:** 5 minutos

4. Discutir un plan de acción y asignar un responsable a las 3 más votadas.

**Duración:** 10 minutos

En el [anexo 6](#) podemos ver el historial de las retrospectivas realizadas por el equipo.

### 3.4.4. Reunión de alineamiento

La reunión de alineamiento es una ceremonia en donde participa todo el equipo, se realiza de forma semanal con el objetivo de alinearse en lo que cada uno está haciendo.

Cada miembro del equipo inspecciona el trabajo que el resto está realizando, analizando así las dependencias entre tareas, progreso hacia el objetivo de la iteración y realizar las adaptaciones necesarias para cumplir con lo definido.

Cada miembro del equipo debe responder a las siguientes preguntas:

- ¿Qué metas conseguiste en la semana pasada?
- ¿Cuáles metas no pudiste conseguir y por qué?
- ¿Cuáles son tus metas para esta semana?
- ¿Hay algún tema que te gustaría discutir en la reunión?

De esta manera se facilita la transferencia de información y la colaboración entre los miembros del equipo, mejorando así la productividad y el compromiso del equipo dado que cada uno manifiesta delante del resto las siguientes problemáticas:

- Tareas que pueden afectar a otros miembros del equipo: ya sea porque tienen un impacto o una dependencia con las tareas realizadas por los demás integrantes.
- Impedimentos para realizar una tarea: permite identificar problemas a tiempo. El resto del equipo puede ofrecer ayuda.
- Tareas no planificadas que surgieron y que el equipo no conoce.
- Necesidades de otros miembros del equipo: favorece a que cada uno pueda colaborar y adaptar su trabajo para proporcionar un beneficio al resto del equipo.
- Ritmo de trabajo: se hace visible de manera continua si un miembro del equipo está trabajando por debajo del rendimiento esperado.

### 3.5. Expectativas de un miembro del equipo

Con el fin de estar alineados en lo que se espera de un miembro del equipo es que se definen las siguientes expectativas. Un miembro del equipo debe:

- Contribuir, discutir y opinar con libertad en la toma de decisiones. En caso de no haber acuerdo entre todos los miembros del equipo aceptar lo que la mayoría decida y una vez tomada la decisión comprometerse a cumplirla.
- Dar retroalimentación constructiva y con respeto al resto de los integrantes del equipo ya que se considera que la retroalimentación es una herramienta fundamental para el crecimiento personal y profesional de los miembros del equipo y por ende del éxito del proyecto.
- Estar disponible en los canales de comunicación definidos y responder a los mensajes en menos de 24 horas a excepción de que haya avisado de su ausencia.
- Mantener un promedio de cumplimiento de los objetivos del *sprint* en al menos 70 % y mostrar empatía con los compañeros que por algún motivo no hayan podido cumplir con los objetivos planteados.
- Notificar al resto del equipo los períodos de ausencia o vacaciones agregando un evento compartido en Google Calendar (Tesis - ORT). La notificación debe darse con al menos una semana de anticipación.
- Ser proactivo a la hora de ayudar a otros miembros del equipo.
- Ser puntual al asistir y moderar las reuniones.
- Tener iniciativa a la hora de proponer mejoras a los procesos de trabajo y/o producto.

### 3.6. Roles

Se definieron los siguientes roles dentro del equipo. Si bien todos los integrantes participaron de todas las tareas, hubo roles responsables de cada parte del producto final.

- Arquitecto de *software*: Dan Blanco
- Responsable de calidad (SQA): Gastón Donadío
- Responsable de la gestión de la configuración (SCM): Victoria Rocha
- Gerente del producto: Agustín Rodríguez

- Gerente del proyecto: Emiliano Rodríguez

Los roles definidos le permitieron al equipo tener un responsable del cumplimiento de las tareas asociadas a cada rol por lo tanto cada integrante estaba al tanto del estado y cumplimiento de las tareas en el plazo establecido.

### 3.7. Relación con el cliente

Se definieron reuniones periódicas con el cliente de manera de tener retroalimentación continua y estar alineados en el avance del proyecto de manera que el producto final estuviera alineado con sus expectativas.

Además, el cliente tiene mucho conocimiento sobre la tecnología RFID ya que ha realizado proyectos utilizándola, por lo tanto, las reuniones también son útiles para aprender de aquello que el equipo desconocía al inicio.

Las reuniones iniciales con el cliente permitieron analizar el negocio, especificar los requerimientos del producto y validar la solución propuesta por el equipo.

Durante la etapa de desarrollo del producto, las reuniones se utilizaron para validar el trabajo hecho con el cliente, obtener retroalimentación y ajustar el desarrollo en las siguientes iteraciones.

## 4. Ingeniería de requerimientos

En este capítulo detallaremos el proceso que el equipo llevó adelante para la identificación y validación de los requerimientos del sistema a desarrollar.

Se explican los métodos utilizados para relevar, especificar y validar los requerimientos así como también los expertos de negocio que se consultaron.

Una vez culminada la validación, el equipo realizó un listado de los requerimientos funcionales y no funcionales del sistema.

### 4.1. Relevamiento

Esta actividad tiene como objetivo realizar el relevamiento de todos los requerimientos del sistema a construir. Los mismos son tanto funcionales como no funcionales.

Como se mencionó anteriormente, el cliente no tenía definidos los requerimientos del sistema sino que solo tenía la idea de que quería desarrollar un sistema de monitoreo de ganado utilizando drones y tecnología RFID.

El cliente planteó como sugerencia utilizar como lenguaje de desarrollo de *backend* .NET debido a que es la tecnología con la que desarrolla todos sus proyectos y este lo seguirá desarrollando la empresa en un futuro ya que parte del entregable al cliente incluye el código fuente.

El equipo desconocía el negocio y por lo tanto fue necesario contactar a expertos que pudieran dar una idea de los problemas que actualmente existen en el campo, las tareas de monitoreo que se realizan, quién las hace, cómo las hace, con qué frecuencia y los impedimentos que actualmente tienen.

Debido también a la falta de conocimiento del negocio fue fundamental investigar soluciones similares que actualmente existieran en el mercado, por lo tanto fue parte de la estrategia a seguir.

En base a lo anteriormente mencionado las estrategias que se siguieron fueron las siguientes:

- Observación empírica
- Entrevistas con expertos de negocio
- Reuniones con el cliente
- *Benchmarking*
- *Prototipos*

#### 4.1.1. Observación empírica

A través de la observación empírica el equipo pudo estudiar y entender el entorno de trabajo de los usuarios, clientes e interesados del proyecto.

Para ello el equipo observó cómo se trabaja en un campo con el fin de documentar la situación y detectar oportunidades de mejora en los procesos actuales.

El tipo de observación que se llevó a cabo fue del tipo pasivo, el equipo se limitó únicamente a observar sin hacer preguntas de manera de no interferir en el desempeño normal de las tareas realizadas.

Las observaciones que se obtuvieron fueron las siguientes:

- Los peones no utilizan tecnología para sus tareas diarias.
- Los peones tienen muy incorporada su forma de trabajo.
- Los peones se mueven en el campo por su conocimiento del mismo en base a la experiencia, pero también utilizan elementos del campo como referencias.
- Por momentos se perdía la conexión LTE y la señal de Wi-Fi no llegaba a todas partes del campo.
- Era muy tedioso acceder a algunas zonas específicas del campo.
- Los animales no se asustaron por la presencia de personas ajenas al campo, ni siquiera cuando se elevaba el tono de voz.
- Las vacas se movían poco y lento por el campo.

#### 4.1.2. Entrevistas con expertos de negocio

El objetivo de estas entrevistas fue descubrir las necesidades existentes e identificar los problemas que se quieren resolver en el campo.

También sirvió para validar conceptos del producto y comprender el negocio con el que se estaba trabajando.

El objetivo no era vender un producto sino que identificar problemas, validar soluciones y consultar por las formas que usan actualmente para resolver esos problemas.

Las preguntas que se realizaron fueron de tipo abierta, algunas de ellas fueron las siguientes:

- ¿Qué problemas tienen en el campo y cómo los resuelven?
- ¿Cómo son los monitoreos que se realizan al ganado?
- ¿Qué información del ganado les interesa controlar?
- ¿Hay buena señal Wi-Fi en el campo?
- ¿Qué costo tiene un animal aproximadamente?
- ¿Utilizan algo de tecnología para facilitar sus tareas?

#### 4.1.2.1. Entrevistas con José Luis Callero

José Luis Callero es médico veterinario, integra el cuerpo de asistentes académicos dentro del decanato de la Facultad de Veterinaria.

José está muy metido en el rubro de la ganadería y la lechería y posee varios campos en el departamento de Durazno, Uruguay.

El equipo se reunió personalmente en diferentes oportunidades en Facultad de Veterinaria y la información que se obtuvo fue la siguiente:

- Los grandes campos tienen entre 10 y 12 personas trabajando todos los días en el campo. Él tiene entre 4 y 5 actualmente recorriendo el campo todos los días.
- Cada 4 meses juntan todo el ganado y lo hacen pasar por mangas para contarlos, luego los pesan y los vacunan. Las vacunas son lo único obligatorio a realizar cada 4 meses, el resto es opcional.
- El problema para realizar las tareas de conteo y control de ganado es que los animales tienen que caminar varios kilómetros, lo que hace que el animal se estrese y pierda peso. Se calcula un costo de USD 1.5 por kilogramo, pudiendo tener pérdidas de USD 50 por animal solo por el traslado hacia las mangas.

- Actualmente los controles los realizan cada 4 meses por el trabajo que lleva, pero le atrae la idea de poder controlarlos con mayor frecuencia (al menos una vez por semana).
- Recalcó que no perder ganado es muy importante.
- Cada animal cuesta aproximadamente USD 1.000 en Uruguay y USD 700 en Paraguay.
- Encontrar el ganado perdido es super difícil y costoso. Hay que contratar personas específicas para esto.
- Actualmente hay una tendencia a no estresar al animal para que no pierda calidad el producto.
- Personas expertas podrían calcular el peso de un animal con una foto, con un margen de error de  $\pm 5$  kg.
- El proceso de control de ganado no es exacto ya que muchas veces hay animales que no se encuentran pero luego aparecen, no fue robado ni está muerto.
- El avance de un animal se mide por la altura, si creció determinada medida quiere decir que todo va bien. Esto se podría realizar con imágenes.
- Actualmente se detecta a “ojo” si un animal está enfermo pero midiendo la temperatura corporal se podría detectar si el animal tiene fiebre, lo que indicaría que está enfermo.
- También hay otros indicadores como la diarrea, por lo cual si el animal tiene la cola manchada es porque tiene diarrea y tiene algún problema. Esto se podría ver también en fotografías.
- En la búsqueda de ganado, a veces algunos se van para adentro del monte y es imposible verlos.
- Luis está muy interesado en detectar depredadores ya que los mismos pueden matar al ganado (mayoritariamente crías). En este problema hay mucho dinero (se habló de billones de dólares).
- El estanciero calcula la cantidad y la cantidad de comida que debe obtener un animal para engordar determinada cantidad y muchas veces el ganado no come todo lo que está previsto y es necesario hacer un pastoreo forzado.

- Si el ganado come siempre en el mismo lugar se generan malezas que luego son difíciles de sacar y tienen un costo extra. Para obtener esto, actualmente obtienen una foto satelital, que le llaman de índices verdes, donde se ven los detalles de las pasturas.
- Los incendios son muy costosos y sería importante detectarlos de forma temprana.
- Los animales no beben agua de pozo por lo que tienen bebederos esparcidos a lo largo del campo que se comunican por tuberías. Es común que esos caños se rompan por mordeduras de animales o pisadas. El ganado puede vivir hasta 3 o 4 días máximo sin agua, por lo que es necesario que una persona recorra casi todos los días estos bebederos para detectar esos problemas.
- Los animales permanecen en promedio 18 meses en el campo, un mínimo de 12 meses si el engorde fue muy bueno y las situaciones climatológicas ayudaron y un máximo de 24 en caso contrario.
- Las estancias tienen buena conectividad a internet, capaz no en todo el campo pero sí en muchas zonas.
- Realizan fumigaciones al campo utilizando drones.
- Sufren 3 tipos de robo:
  - Matan al animal y lo carnean en el momento.
  - Va un camión, los cargan y se los llevan en manada.
  - Cortan el alambrado y los dejan irse para atraparlos a un par de kilómetros.

De las entrevistas con Luis pudimos concluir que dedican muchas horas hombre a recorrer los campos para controlar el ganado, el alambrado, los bebedores y todas las instalaciones en general.

También que la pérdida de los animales es costosa y por lo tanto para evitarlo hacen controles a los animales pero los mismos no pueden hacerse tan seguido como quisieran debido al trabajo que les lleva y a que de la forma que lo hacen los animales se estresan, lo cual se quiere evitar.

En las recorridas no saben con exactitud si están todos los animales porque muchos están escondidos y no se ven, pero luego aparecen.

Ya utilizan algo de tecnología, en particular usan drones para las fumigaciones.

Luego del análisis de toda la información por parte del equipo, se profundizó en los siguientes temas en las entrevistas siguientes:

### 1. Además de la ubicación del ganado, ¿Qué más sería útil controlar?

- a. La **ubicación** de los animales: Con esto sabrían si el animal está en una zona peligrosa, prohibida o fuera del campo. También controlarían el movimiento del animal de manera de saber si está comiendo en los lugares previstos o si hay que forzarlos a comer en otros lugares.
- b. La **cantidad** que hay en el campo: Esto serviría para saber si desapareció ganado ya sea por robo, muerte o porque salió de las zonas estipuladas.
- c. El **tamaño** de cada animal: Ya que esto muestra la evolución del mismo. Si está creciendo bien es porque no tiene ningún problema. Hay personas expertas que viendo al animal pueden saber el peso que tiene con un margen chico de error. Esto se podría hacer a través de fotos.
- d. El **estado de salud** de cada animal: A través de sensores de temperatura se podría saber si el animal tiene fiebre, lo que indicaría una enfermedad. Expertos también pueden detectar otros síntomas como diarrea a través de fotos del animal.
- e. El **estado del campo**: Controlar inundaciones, incendios y humedad en la tierra.
- f. El **estado de los bebederos**: Controlar que esté llegando agua a los bebederos ya que sin eso los animales podrían morir. Controlar también las tuberías.
- g. **Depredadores**: Detectar si hay depredadores merodeando la zona (Ejemplo: perros sueltos)

### 2. Con la ubicación, ¿Qué se podría controlar?

- a. Se puede controlar si el animal está en zonas peligrosas como ser una cañada. Esto es muy útil ya que muchas veces los animales se encuentran en zonas que son de difícil acceso para los peones de campo.
- b. Permite también detectar si un animal está fuera de la zona permitida como por ejemplo si salió de la zona cercada del campo.
- c. También permite detectar las zonas por las que se movió el animal con lo cual saber si está comiendo en los lugares previstos pudiendo llevar un control de si

la gordura del animal está siendo óptima o es necesario forzarlo a comer en otros lugares.

**3. ¿Qué alertas le gustaría tener que se puedan calcular con la ubicación?**

- a. Alertas de si un animal se encuentra en una zona peligrosa, ya que corre riesgo su vida, o si ha desaparecido alguno.

**4. ¿Lo ideal es que los controles se realicen todos los días?**

- a. No es necesario que sea todos los días.
- b. Actualmente el conteo de todos los animales lo hacen cada 4 meses, por lo tanto hacerlo cada 1 semana estaría bien.
- c. Sí sería importante hacer el control de los bebederos todos los días.

**5. ¿Cómo hacen hoy en día el control de ganado?**

- a. Los peones de campo hacen recorridas diarias para obtener diferente información, por ejemplo, para el chequeo del estado de los bebederos hay una persona dedicada solo a eso.
- b. Sin embargo para el recuento de los animales, la vacunación y control de peso, contratan personas adicionales, tiene una duración de 1 semana y lo hacen cada 4 meses ya que cada 4 meses es obligatorio vacunarlos.

**6. ¿Cuánto tiempo dedican a esto?**

- a. Tienen 4 o 5 personas que trabajan todos los días, y contratan 5 o 6 personas extra cada 4 meses para hacer el control general por un tiempo de una semana.

**7. ¿Cuentan con buena señal en el campo?**

- a. El Internet llega y cada vez más está mejorando esto.

**8. ¿Cuáles serían las zonas peligrosas?**

- a. Hay zonas que está estipulado que el animal debería encontrarse, ya sea porque se realizaron pasturas o porque son las zonas más adecuadas y preparadas.
- b. Las zonas peligrosas podrían ser zonas que están fuera de las mencionadas. Esto sería porque no hay alimento de calidad, hay peligro de que el animal quede atrapado o hay inundaciones.

**9. ¿Interesa identificar al ganado o simplemente ubicar donde está?**

- a. Es importante identificar al ganado para tener un control del avance del animal.

- b. Incluso hay empresas grandes que tienen un historial de los animales con su árbol genealógico, lo que les permite saber con anterioridad la calidad del animal lo que se termina reflejando en el precio.

**10. ¿Tienen idea cuantos dólares pierden al año por muerte o robo de ganado? y en caso afirmativo ¿Qué porcentaje representa en sus ingresos?**

- a. No sabe exactamente pero en un campo con 3.500 cabezas de ganado, podrían tranquilamente faltar entre 10, 20 y hasta 100 animales lo cual implica un costo de hasta USD 100.000 al año.

#### 4.1.2.2. Entrevistas con Rodrigo Pereira

Rodrigo es el dueño de la empresa Zafrales, empresa que se dedica a la contratación de personal temporal para zafras, trabajan con distintos rubros, entre ellos la ganadería y tienen experiencia trabajando en campos.

El equipo se reunió con Rodrigo vía Hangouts y le contó la idea de utilizar drones en el campo para el monitoreo; se obtuvo la siguiente información y retroalimentación:

- Los principales problemas que tiene el campo son:
  - Abigeato, sobre todo en los ovinos.
  - Muerte de animales por inundaciones.
  - Muerte de animales por mordeduras o ataques de perros sueltos.
- Los peones de campo cuentan los animales todos los días para saber si les falta alguno. También controlan los candados en las porteras y el alambrado del perímetro.
- Se controla que los alambrados estén en orden, que los palos estén y por lo tanto que los animales no se vayan.
- Se enteran que les robaron animales pero no saben exactamente cuándo porque muchas veces los controles no se pueden realizar todos los días por situaciones climatológicas o muchas veces no se ven a animales que pueden estar escondidos en montes.
- Vecinos compraron un dron y se lo dieron a la policía para que monitoree los campos.

En base a la información que el equipo ya tenía en ese momento, decidió consultarle a Rodrigo sobre los siguientes temas:

### **Riesgos que encuentra al uso de drones para monitoreo de campos:**

Si el campo tiene montes se puede complicar el buscar a un animal ya que es muy común que los animales se oculten allí en determinados momentos. Esto podría generar que en una recorrida el dron no detecte a un animal.

Los peones de campo ya conocen los horarios en los que los animales se meten en los montes, entonces el dron tendría que hacer la recorrida evitando esos horarios.

### **Adaptación de los peones a la tecnología:**

Hay que trabajar en la adopción de la tecnología, hoy en día los peones no están acostumbrados a usar tecnología.

El peón en general no sabe leer entonces todo lo que se pueda hacer con figuras y opciones es mejor. Botones grandes porque tienen dedos gordos. Buscar una buena combinación de colores porque en general en los campos hay mucha o poca luz. Utilizar un vocabulario sencillo.

En caso que la solución involucre el uso de una aplicación, Rodrigo sugiere que la utilice el capataz en lugar del peón ya que el primero tiene en general conocimientos universitarios debido a que hoy en día es un requisito tener el título de técnico agropecuario mínimo.

### **Uso de referencias para moverse en el campo:**

La gente que trabaja en el campo se mueve sobre el mismo utilizando referencias tales como potreros, arroyos, entre otros.

En general utilizan una moto para moverse hasta el lugar. Una solución que utilice drones para el monitoreo de campos lo considera mejor para campos grandes con productores de alta adopción a la tecnología.

### **Ganado de alto valor agregado:**

Un nicho de mercado interesante podría ser el de los productores agropecuarios con ganado de alto valor agregado.

Un animal de alto valor agregado tiene un valor 10 veces mayor que el resto del ganado y por lo tanto un robo o carneo a dichos animales le generaría al estanciero una pérdida de mucho dinero.

### **Zonas de peligro:**

Una zona de peligro puede ser por ejemplo una zona contra una ruta, porque si un animal se escapa y un auto choca contra él y muere es culpa del productor y el mismo puede tener graves sanciones legales.

También zonas de cañadas o zonas sensibles a las inundaciones o incendios. En esas zonas estaría bueno para el estanciero o capataz recibir notificaciones y poder prevenir el problema.

#### 4.1.3. Reuniones con el cliente

El equipo buscó en todo momento mantener una relación fluida con el cliente para asegurar que el producto estuviera alineado a sus expectativas.

Es por esto que se realizaron reuniones a lo largo del desarrollo del proyecto, con el objetivo de:

- Entender el problema a resolver
- Definir requerimientos y restricciones
- Validar requerimientos e implementaciones
- Priorizar requerimientos e implementaciones
- Comunicar tiempos, bloqueos y pasos a seguir
- Solicitar *hardware* necesario

Al principio las reuniones fueron específicas de relevamiento de requerimientos y luego se transformaron en reuniones de validación y seguimiento, donde surgieron requerimientos de cambios menores a medida que se iba avanzando en la construcción del producto.

En las primeras reuniones con el cliente el foco estuvo en conocer la empresa Todosoft y entender la idea que tenían de lo que quería resolver, para llegar finalmente a un listado de requerimientos priorizados para el sistema a construir.



Ilustración 4: Reunión en TodoSoft

Se impusieron restricciones para el proyecto, tales como utilizar determinadas tecnologías, las cuales se detallan en el [capítulo 4.2.4](#).

El equipo se reunió con expertos de negocio (ver [capítulo 4.1.2](#)) y en base a la información recolectada y junto con el cliente se definió una lista inicial de requerimientos funcionales que el sistema debía cumplir.

Esta lista fue priorizada entre el cliente y el equipo con el fin de que al momento de comenzar con la fase de desarrollo estuvieran claras las prioridades.

El equipo le planteó a Todosoft la posibilidad de postularse a una financiación de CI<sup>2</sup> para obtener fondos para la compra del *hardware* necesario, el cliente accedió y se comenzó a definir la información necesaria para la postulación, así como también el comienzo de una investigación de un *hardware* que cumpliera con las necesidades del producto, la cual se detalla en el [capítulo 5](#).

El equipo definió un plan de *release* junto con el cliente, es decir, un conjunto de historias agrupadas por *releases* o versiones del producto. Este plan sirvió para darle visibilidad al producto a lo largo de la etapa de desarrollo. El plan de release se encuentra especificado en el [capítulo 10.1.2](#).

Una vez que el equipo comenzó con la fase de desarrollo, las reuniones con el cliente se volvieron reuniones de validación y seguimiento del trabajo realizado. De dichas reuniones surgieron cambios menores a realizar de manera de ajustarse a las expectativas del cliente. En el [anexo 8](#) se detallan cambios propuestos que surgieron de dichas reuniones.

Se aprovechó el conocimiento que Todosoft tiene en cuanto a la tecnología RFID, la cual era desconocida, al inicio, para la mayoría de los integrantes del equipo. Dicho conocimiento se utilizó sobre todo para validar el plan de vuelo desarrollado (ver [capítulo 5.2](#)), en función del tipo de RFID con el que se trabaje (activo o pasivo).

Una vez que se conoció el resultado negativo del CI<sup>2</sup> para la financiación, se acordó con el cliente la compra de un dron que cumplía con todo lo necesario para el proyecto y que tenía un costo menor que el original. Una vez llegó el dron a la empresa Todosoft se realizaron reuniones de forma de realizar el conjunto de pruebas necesarias para garantizar que el dron estuviera funcionando correctamente. Como se menciona en el [capítulo 12.6](#) varias partes del dron llegaron rotas por lo que hubo que solicitarlas nuevamente y volver a realizar las pruebas correspondientes.

#### 4.1.4. *Benchmarking*

Para conocer las soluciones actuales que aplican drones para monitoreo de campos se hizo un *benchmarking* y se encontraron las siguientes empresas que si bien no tienen proyectos iguales a SkyGuard, abarcan muchas partes de la problemática que se intenta solucionar.

#### 4.1.4.1. SmartXFarm

Proponen una aplicación web, una móvil y *scanners* de código de barras o RFID para el monitoreo de animales en campos [\[51\]](#).

Junto con estas tecnologías plantean varios tipos de soluciones para granjas:

- Seguimiento de activos (tractores, maquinaria, entre otros)
- Seguimiento de semillas (cuando se plantaron, cuando se tienen que cultivar y demás)
- Salud y ubicación del ganado
  - Aquí plantean agregan a la solución ya conocida, la apertura de puertas para la clasificación del ganado y habilitación de alimentación individual.
- Seguridad del personal (ver que no estén en zonas peligrosas o restringidas)

Además, ofrecen otros tipos de soluciones como la localización de herramientas y materiales en una obra de construcción. Su solución plantea una localización en tiempo real.

#### 4.1.4.2. G&D Argentina

Esta empresa además de ofrecer la solución de ubicación de ganado, ofrece lo siguiente [\[52\]](#):

- Desplazar o no el boyero eléctrico (cercas eléctricas).
- Evaluar pastoreo rotativo (El orden en el que el ganado se va alimentando).
- Determinar si los animales se desplazan a lugares inundados.
- Evaluación del estado corporal de las vacas.
- Reforzar la seguridad para evitar robos.

#### 4.1.4.3. ACGDrone

Esta empresa también monitorea el ganado en campos pero no utiliza RFID sino que utiliza cámaras térmicas [\[53\]](#).

Esta tecnología agrega información como ver la temperatura del ganado lo que puede indicar una enfermedad u otros datos.

ACGDrone no lo hace pero esta tecnología permite también detectar incendios e inundaciones.

#### 4.1.5. Prototipos

Como se menciona más adelante en el [capítulo 4.3](#), los prototipos se utilizaron como una técnica de validación de los requerimientos funcionales pero también como técnica de relevamiento ya que los mismos le permitieron al equipo tomar decisiones de implementación en base a la retroalimentación recibida por parte de los expertos de negocio y del usuario.

A su vez fueron útiles para comprender, probar y ajustar el producto final antes de comenzar a desarrollarlo.

Los prototipos fueron realizados para la aplicación web y la aplicación móvil. Los mismos permitieron simular lo que sería el producto final, con el fin de que el usuario pueda interactuar con el mismo y solicitar las mejoras que correspondan.

## 4.2. Especificación

Una vez culminadas las etapas de relevamiento, el equipo realizó una lista de requerimientos funcionales y no funcionales.

### 4.2.1. Requerimientos funcionales

Los requerimientos funcionales se especifican a través de historias de usuario.

Por lo tanto las mismas siguen el siguiente formato:

**Como [Rol], quiero [Funcionalidad] para [Objetivo]**

A continuación se detallan las historias de usuario identificadas para el proyecto:

- **RF1.** Como dueño de campo quiero poder agregar y eliminar las zonas a escanear por el ejército de drones para poder armar las rutas que recorrerán los drones.
- **RF2.** Como dueño de campo quiero poder definir zonas peligrosas para los animales, para poder controlar que los mismos no se encuentren en dichas zonas.

- **RF3.** Como dueño del campo quiero poder asignar zonas a uno o más drones para que el dron recorra dicha zona mientras escanea a los animales.
- **RF4.** Como dueño de campo quiero poder agregar, modificar y eliminar ganado para poder llevar el control de cada animal asociándolo al *tag*.
- **RF5.** Como dueño de campo quiero poder ver la información cacheada de las zonas, animales y lecturas para tener acceso a información sin tener conexión a Internet.
- **RF6.** Como dueño de campo quiero poder definir alertas en caso de que un animal ingrese a una zona peligrosa para evitar que los mismos mueran sin ser detectados.
- **RF7.** Como dueño de campo quiero recibir una notificación si un animal sale de la zona definida para evitar robos.
- **RF8.** Como dueño de campo quiero poder ver en un mapa la posición de cada animal en el momento en que fue leído su chip para saber dónde está ubicado.
- **RF9.** Como dueño de campo quiero poder ver un reporte en forma de mapa de calor con la trazabilidad del ganado.
- **RF10.** Como dueño de campo quiero poder ver un reporte de cada una de las recorridas que hizo cada dron sobre las zonas definidas para tener un control de los mismos.
- **RF11.** Como dueño de campo quiero poder iniciar sesión al sistema para tener acceso a las funcionalidades de la aplicación.
- **RF12.** Como dueño de campo quiero poder marcar como leída una alerta para tener a la vista solo las alertas sin leer.
- **RF13.** Como dueño de campo quiero poder buscar por un animal y que me muestre la última posición.
- **RF14.** Como usuario administrador de la plataforma quiero poder agregar, modificar y eliminar dueños de campos para poder añadir nuevos clientes al sistema.
- **RF15.** Como dueño de campo quiero poder definir referencias en el mapa de mi campo para poder utilizarlas para ubicar animales.
- **RF16.** Como dueño de campo quiero poder ver dónde están mis drones en un mapa para poder saber cuánto les falta para terminar la ruta, saber si el sistema está funcionando y saber dónde fue su última ubicación ante un imprevisto.

- **RF17.** Como dueño del campo quiero poder seleccionar “zonas a escanear” y que el sistema seleccione automáticamente qué drones las recorrerán, para poder configurar el trabajo de los drones sin tener que hacerlo manualmente.
- **RF18.** Como dueño de campo quiero poder ver el estado de cada uno de los drones (Volando/No volando, nivel de batería, capacidad de almacenamiento, etc) para poder asignar el dron con mejor disponibilidad a un recorrido.
- **RF19.** Como dueño de campo quiero poder agregar, modificar y eliminar drones para poder asignarlos a las rutas.
- **RF20.** Como dueño de campo quiero poder ver fotos del ganado en un reporte para analizar el estado de salud de los mismos.

#### 4.2.2. Priorización de los requerimientos funcionales

Los requerimientos funcionales se priorizaron de la siguiente manera de acuerdo a los intereses de los expertos del negocio y del cliente Todosoft:

##### **Prioridad alta:**

- **RF1.** Como dueño de campo quiero poder agregar y eliminar las zonas a escanear por el ejército de drones para poder armar las rutas que recorrerán los drones.
- **RF2.** Como dueño de campo quiero poder definir zonas peligrosas para los animales, para poder controlar que los mismos no se encuentren en dichas zonas.
- **RF4.** Como dueño de campo quiero poder agregar, modificar y eliminar ganado para poder llevar el control de cada animal asociándolo al *tag*.
- **RF8.** Como dueño de campo quiero poder ver en un mapa la posición de cada animal en el momento en que fue leído su chip para saber dónde está ubicado.
- **RF13.** Como dueño de campo quiero poder buscar por un animal y que me muestre la última posición.
- **RF3.** Como dueño del campo quiero poder asignar zonas a uno o más drones para que el dron recorra dicha zona mientras escanea a los animales.
- **RF6.** Como dueño de campo quiero poder definir alertas en caso de que un animal ingrese a una zona peligrosa para evitar que los mismos mueran sin ser detectados.

- **RF7.** Como dueño de campo quiero recibir una notificación si un animal sale de la zona definida para evitar robos.

#### **Prioridad media:**

- **RF11.** Como dueño de campo quiero poder iniciar sesión al sistema para tener acceso a las funcionalidades de la aplicación.
- **RF14.** Como usuario administrador de la plataforma quiero poder agregar, modificar y eliminar dueños de campos para poder añadir nuevos clientes a nuestro sistema.
- **RF15.** Como dueño de campo quiero poder definir referencias en el mapa de mi campo para poder utilizarlas para ubicar animales.
- **RF5.** Como dueño de campo quiero poder ver la información cacheada de las zonas, animales y lecturas para tener acceso a información sin tener conexión a Internet.
- **RF9.** Como dueño de campo quiero poder ver un reporte en forma de mapa de calor con la trazabilidad del ganado.
- **RF10.** Como dueño de campo quiero poder ver un reporte de cada una de las recorridas que hizo cada dron sobre las zonas definidas para tener un control de los mismos.
- **RF12.** Como dueño de campo quiero poder marcar como leída una alerta para tener a la vista solo las alertas sin leer.

#### **Prioridad baja:**

- **RF16.** Como dueño de campo quiero poder ver dónde están mis drones en un mapa para poder saber cuánto les falta para terminar la ruta, saber si el sistema está funcionando y saber dónde fue su última ubicación ante un imprevisto.
- **RF17.** Como dueño del campo quiero poder seleccionar “zonas a escanear” y que el sistema seleccione automáticamente qué drones las recorrerán, para poder configurar el trabajo de los drones sin tener que hacerlo manualmente.
- **RF18.** Como dueño de campo quiero poder ver el estado de cada uno de los drones (Volando/No volando, nivel de batería, capacidad de almacenamiento, etc) para poder asignar el dron con mejor disponibilidad a un recorrido.

- **RF19.** Como dueño de campo quiero poder agregar, modificar y eliminar drones para poder asignarlos a las rutas.
- **RF20.** Como dueño de campo quiero poder ver fotos del ganado en un reporte para analizar el estado de salud de los mismos.

Un factor importante al momento de seleccionar los requerimientos funcionales a implementar fue la priorización de los mismos ya que se comenzaron a implementar los requerimientos funcionales de mayor prioridad primero.

#### 4.2.3. Requerimientos no funcionales

- **RNF1.** El componente web del sistema debe ser compatible con Google Chrome, Mozilla Firefox, Internet Explorer y Microsoft Edge.
- **RNF2.** La aplicación móvil del sistema debe ser compatible con Android e iOS.
- **RNF3.** Las lecturas RFID deben poder capturarse y no perderse aún sin conexión Wi-Fi durante el vuelo del dron (se deben almacenar localmente en el dron).
- **RNF4.** El dron debe funcionar en el exterior con condiciones climáticas moderadas, es decir, calor, lluvia leve y poco viento.
- **RNF5.** El sistema debe ser diseñado y construido pensando en que a futuro se agregarán módulos y se continuará con el desarrollo por parte de otros desarrolladores.
- **RNF6.** El sistema debe ser fácil de usar, intuitivo y con las funcionalidades bien definidas entre la aplicación web y las aplicaciones móviles para aprovechar las ventajas de cada plataforma.

#### 4.2.4. Restricciones

El sistema a desarrollar, tiene las siguientes restricciones las cuales fueron impuestas por la empresa Todosoft al inicio del proyecto:

- Uso de tecnología RFID como forma de identificar a los animales en el campo.
- Uso de drones como forma de recorrer el campo de forma automática con la lectora RFID montada sobre el mismo.

## 4.3. Validación

Previo a comenzar el desarrollo, el equipo validó la especificación de requerimientos con el cliente, la empresa Todosoft, y con expertos de negocio, a través de prototipos.

Los prototipos fueron realizados para la aplicación web y la aplicación móvil, con el objetivo de estar alineados en las decisiones tomadas para la implementación del sistema SkyGuard.

Permitieron simular lo que sería el producto final, con el fin de que el usuario pueda interactuar con el mismo y solicitar las mejoras que correspondan.

A su vez fueron útiles para comprender, probar y ajustar el producto final antes de comenzar a desarrollarlo.

### 4.3.1. Prototipos aplicación móvil

A continuación se presentan los prototipos de la aplicación móvil, los cuales permiten:

- *Login* al sistema
- Acceso al menú principal
- Registrar un animal
- Registrar un dron
- Visualizar el mapa con las referencias, los animales y las zonas peligrosas
- Visualizar la información de un animal
- Recibir alertas



Ilustración 5: Prototipo pantallas de la aplicación móvil para *login* y menú principal.

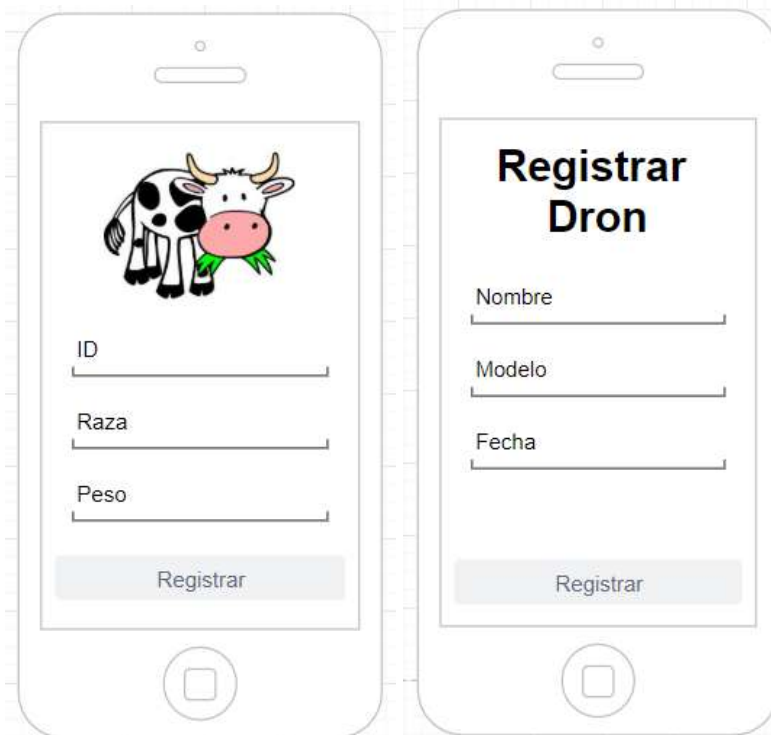


Ilustración 6: Prototipo pantallas de la aplicación móvil para alta de animal y dron.

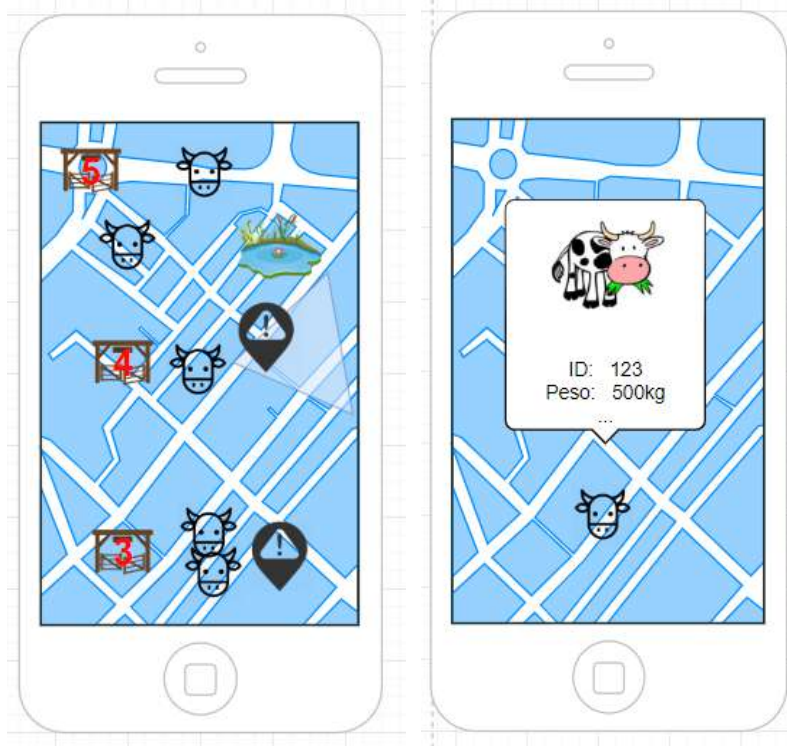


Ilustración 7: Prototipo pantallas de la aplicación móvil para mapas con ubicación de animales, referencias y zonas peligrosas.

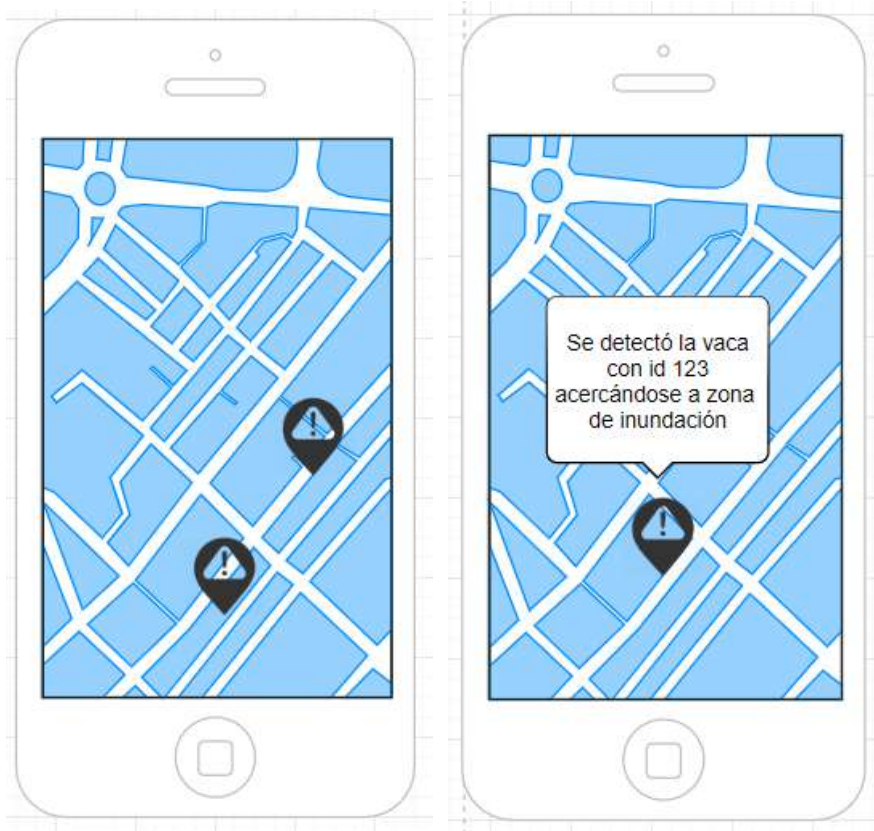


Ilustración 8: Prototipo pantallas de la aplicación móvil para notificación de alertas.



Ilustración 9: Prototipo pantallas de la aplicación móvil para notificación de alertas.



Ilustración 10: Prototipo pantallas de la aplicación móvil para notificación de alertas.

### 4.3.2. Prototipos aplicación web

A continuación se presentan los prototipos de la aplicación web, los cuales permiten:

- *Login* al sistema
- Gestión de drones
- Gestión de animales
- Gestión de zonas peligrosas y referencias
- Gestión de alertas

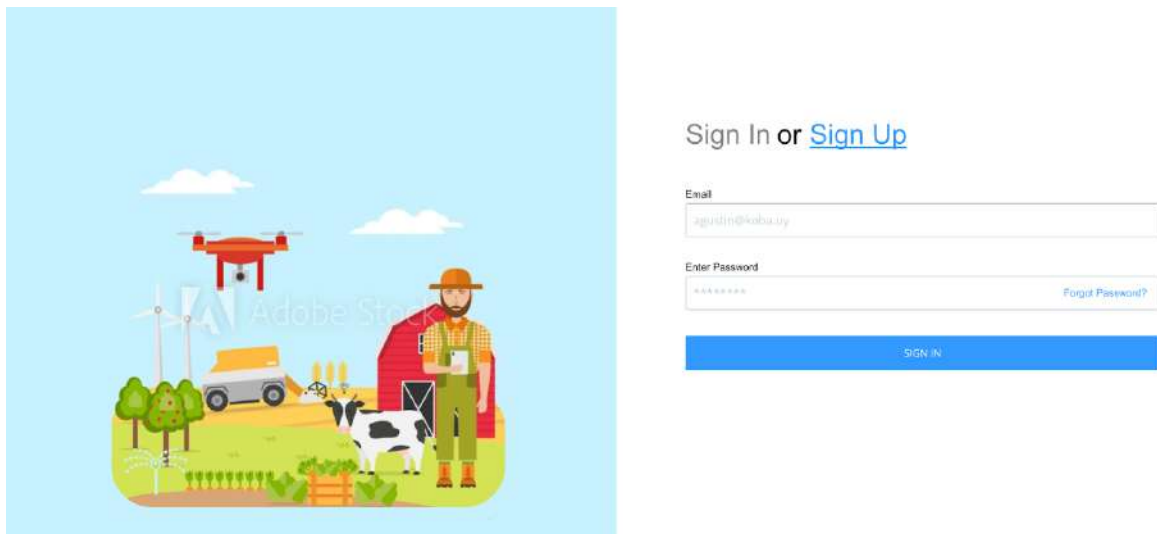


Ilustración 11: Prototipo pantallas de la aplicación web para *login*.

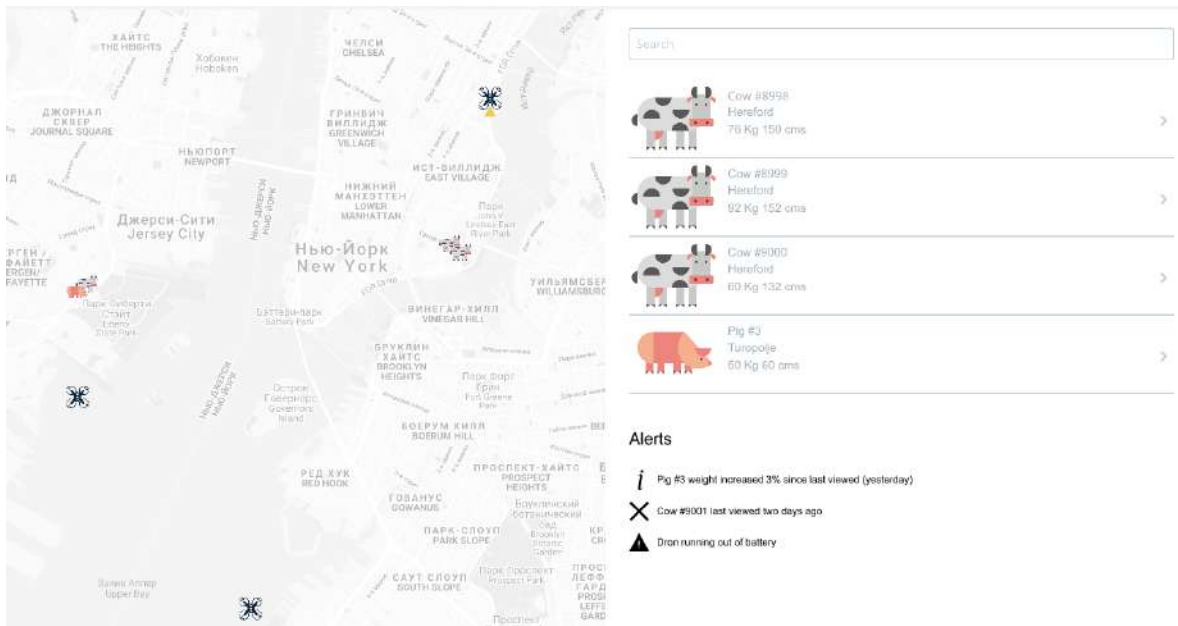


Ilustración 12: Prototipo pantallas de la aplicación web para visualización de animales y alertas.

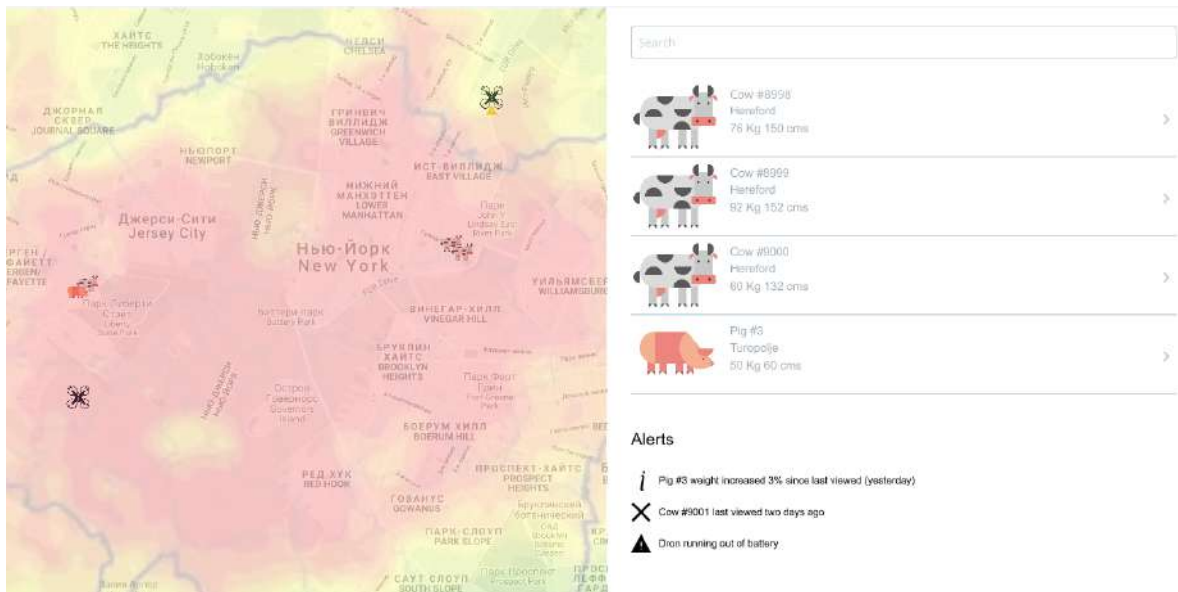


Ilustración 13: Prototipo pantallas de la aplicación web para visualización de mapa de calor.

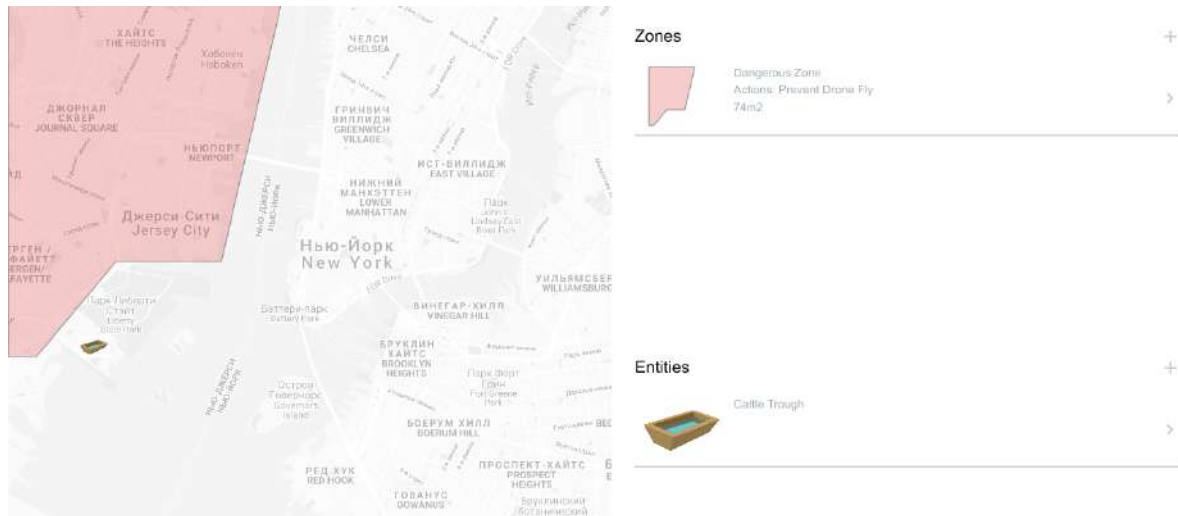


Ilustración 14: Prototipo pantallas de la aplicación web para visualización de zonas peligrosas y referencias.

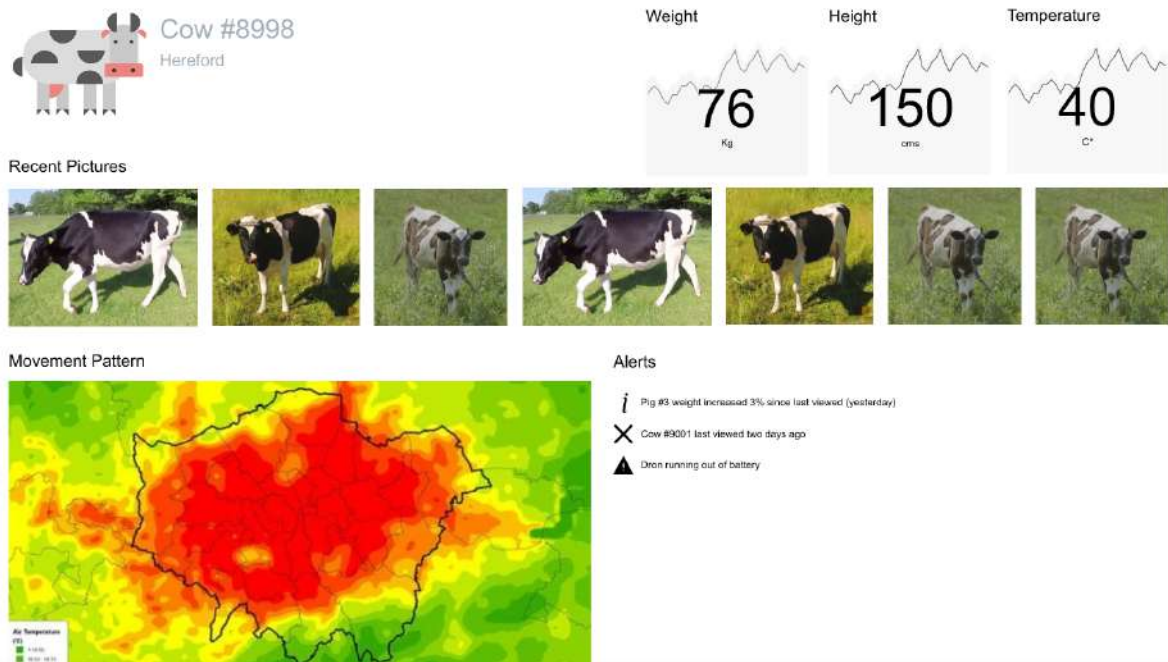


Ilustración 15: Prototipo pantallas de la aplicación web para visualización de la información de un animal



Drone #123456

DJI MegaDrone III

Battery Level



Map



Alerts




-  Fig #3 weight increased 3% since last viewed (yesterday)
-  Cow #0001 last viewed two days ago
-  Dron running out of battery

Ilustración 16: Prototipo pantallas de la aplicación web para visualización de la información de un dron.

## 4.4. Entregables

A continuación se detallan los entregables del proyecto por parte del equipo a la empresa Todosoft una vez que el mismo finalice:

- Código fuente de las aplicaciones *frontend* (SkyGuard-UI) y acceso al repositorio en Github.
- Código fuente de la solución de backend (SkyGuard-Core) y acceso al repositorio en Github.
- Código fuente de la solución que genera el plan de vuelo del dron (SkyGuard-Pathfinder) y acceso al repositorio en Github.
- Código fuente de la solución que ejecuta el plan de vuelo en el dron (SkyGuard-Copter) y acceso al repositorio en Github.
- Documentación del plan de vuelo desarrollado.
- Documentación de la arquitectura del sistema.
- Documentación de las pruebas que garantizan la calidad del producto.
- Transferencia de conocimiento por parte del equipo hacia la empresa Todosoft.

- Relevamiento de las necesidades de negocio por parte de los expertos.
- Investigaciones realizadas, las cuales incluyen:
  - Investigación de drones.
  - Investigación de la placa Ardupilot.
  - Investigación de RFID.
  - Investigación del procesamiento de imágenes.
  - Investigación de la digitalización de planos.

El código fuente a entregar de la solución SkyGuard cumple con todos los requerimientos de prioridad alta y media, no así los de prioridad baja, quedando los mismos para ser desarrollados por la empresa Todosoft en un futuro. Más detalles se encuentran en el plan de releases definido en el [capítulo 10.1.2](#).

## 5. Investigación

La investigación fue un componente muy importante dentro del proyecto debido a que en esta etapa fue necesario definir los componentes bases que serían utilizados a lo largo de todo el proyecto. Esta etapa fue muy exhaustiva y detallista debido a que el margen de error permitido era muy pequeño, ya que el costo de los componentes necesarios es muy alto y las características debían adecuarse exactamente a las necesidades del proyecto. Además de eso, los componentes debían ser compatibles entre sí en un 100 % y permitir una gran facilidad a la hora de escalar y/o cambiar los mismos. El error en la adquisición de uno de estos componentes podría haber impactado fuertemente en los costos del proyecto.

### 5.1. Investigación de drones

Para definir con qué dron/drones se iba a trabajar (tanto para la realización de pruebas como para cuando esté finalizada la solución) se realizó una investigación recabando información de los diferentes vehículos no tripulados disponibles en el mercado.

La valoración de cada uno de los drones investigados se hizo teniendo en cuenta los requerimientos que se tenían, los cuales eran:

- **Que el dron sea programable**, debido a que parte importante del alcance de la tesis era realizar la configuración del plan de vuelo del dron, para lograr así una autonomía y un nivel de modificabilidad importante para que el sistema pueda ser personalizable a futuro.
- **Que el dron tenga la posibilidad de soportar el peso de un lector de RFID** colgando de él, ya que de esto depende poder realizar las lecturas mientras el dron sobrevuela los campos.
- **Que el dron tenga un costo total que no lo haga inviable para el proyecto**. Esto fue impuesto por la empresa Todosoft pensando en la futura comercialización del sistema.
- **Que el dron pueda almacenar las lecturas de RFID localmente**, para asegurarse de que las lecturas no se pierdan por problemas de batería, falta de conectividad u otros errores.

- **Que el dron tenga un tiempo de vuelo aceptable** (duración de la batería) para que la recorrida de campos de medianas y grandes extensiones sea posible.
- **Que el dron tenga un tamaño adecuado** para cumplir con los requerimientos: que sea robusto, resistente pero trasladable con facilidad.

Se investigaron más de 30 empresas, con varios modelos cada una. La lista completa se encuentra en el [anexo 11](#).

Algunos de los drones destacados que se analizaron pero que por diferentes motivos se terminaron descartando se presentan a continuación.

### 5.1.1. DJI Inspire 2



Ilustración 17: Dron DJI Inspire 2

Es un dron de la categoría profesional que cuenta con las siguientes características:

- **Tiempo de vuelo:** 25-27 minutos
- **Velocidad:** 94 km/h
- **Rango de detección:** 30 metros
- **Rango de control:** 7 km
- **Resolución de video:** 5.2 K/4K

- **Video HD en directo:** 1080P

#### 5.1.1.1. Descripción

Características principales [\[54\]](#):

- Va de 0 a 80 km/h en 5 segundos.
- Sistema de batería dual que prolonga el tiempo de vuelo a un máximo de 27 minutos.
- Sistema de anticolidión en 2 direcciones. Le permite evitar obstáculos de hasta 30 metros de distancia.
- Cámara FPV a bordo que permite transmisión de video en directo.
- Grabación de video que puede ser guardado tanto en DJI como en la tarjeta SD.
- Control remoto que soporta una distancia máxima de 7 km.
- Muy buena calidad de video (es utilizado por cineastas para producción de cine).
- La batería dual posibilita que en el caso de que falle una de las baterías la otra puede continuar el vuelo para permitir un aterrizaje seguro.
- Regreso al punto de origen: Los sistemas de visión delantero y trasero le permiten al dron crear mientras vuela un mapa en tiempo real de su ruta de vuelo. Aunque se pierda la conexión es capaz de regresar al punto de origen.

#### 5.1.1.2. Problemas

Principales problemas identificados [\[55\]](#):

- No cuenta con lectores incluidos (ni para lectura de chips RFID, ni cámara de calor ni ninguna otra tecnología que permita identificar al ganado).
- No cuenta con integración Onboard SDK [\[56\]](#).
- Si vuela lejos del dispositivo que accionó el vuelo, se desconecta y vuelve a la base sin recabar los datos solicitados.

## 5.1.2. DJI S1000 Plus



Ilustración 18: Dron DJI S1000 Plus

### 5.1.2.1. Descripción

- *Hardware* robusto, dándole estabilidad y fuerza.
- Soporta un peso máximo de despliegue de 11 kg.
- La batería le permite volar hasta 15 minutos (depende del peso que cargue). Los 15 minutos son teniendo en cuenta que no hay viento, con una carga de 9.5 kg y a una altura aproximada de 2 metros.

### 5.1.2.2. Problemas

- No tiene sistema anti-colisión.

### 5.1.3. DJI Mavic 2 Enterprise



Ilustración 19: Dron DJI Mavic 2 Enterprise

Dron diseñado para empresas que añada características y accesorios para convertirlo en toda una máquina para diversas tareas como búsqueda, rescate, monitorización de incendios y otras actividades [\[57\]](#).

#### 5.1.3.1. Características

- Zoom óptico x2
- 8 KM de distancia al control
- Hasta 31 minutos de batería
- Fotos en 12 MP – filmación en 4K
- Velocidad hasta 72 km/h
- Menos ruidos
- Sistema anticolidión 260 grados
- Almacenamiento interno de 24 GB con protección por contraseña

Tiene muy buena velocidad, un muy buen sistema anticolidión pero no puede cargar el peso de un lector de RFID.

#### 5.1.4. DJI Matrice 100



Ilustración 20: Dron DJI Matrice 100

##### 5.1.4.1. Descripción

Es completamente programable, ideal para que los desarrolladores puedan adecuarse a las necesidades de cada usuario, tanto de *hardware* como de *software*. De ahí la multiplicidad de usos que se le puede dar [\[58\]](#).

##### 5.1.4.2. Características

- Radio de acción de más de 3 kilómetros.
- Autonomía de vuelo de 20 minutos. Con batería suplementaria puede volar hasta 40 minutos con carga de un kilogramo.
- Multiplicidad de puertos y modularidad. Se pueden conectar todo tipo de sistemas o periféricos, desde diferentes cámaras y sensores a sistemas comunicación.
- Está equipado con el primer sistema anti-colisión vertical y horizontal, asequible, denominado Guidance, capaz de detectar obstáculos a menos de 20 metros.
- Puede llevar a bordo múltiples sensores, entre ellos sensores térmicos.

- Puede transmitir video en streaming de cualquier tipo de imagen, incluyendo imágenes infrarrojas o térmicas.

#### 5.1.4.3. Especificaciones técnicas

- Capacidad máxima de peso: 3,6 kg.
- Alcance operativo: 3,1 millas (5 km).
- Distancia de funcionamiento máximo: 5000 m.
- GPS mejorada con una montura plegable.
- Ancho máximo dispositivo: 17 cm.
- Capacidad de batería: 4500 mAh.
- Funciona con Zenmuse X3 Cámara/*Gimbal* DJI SDK para el desarrollo personalizado

#### 5.1.5. Empresas Parrot y Yuneec



Ilustración 21: Dron de Parrot

Otras empresas que se tuvieron en cuenta para la investigación fueron Parrot o Yuneec, sin embargo sus SDK están disponibles solo para desarrollo *mobile* [59]; son soluciones pensadas para tener los dispositivos que accionan a los drones cerca ya que el dron no puede alejarse mucho del mismo [60]. Por ese motivo estas empresas fueron descartadas.

### 5.1.6. DJI Matrice 200



Ilustración 22: Dron DJI Matrice 200

#### 5.1.6.1. Características

Principales características [\[61\]](#):

- Cámara FPV integrada. Sistema de doble batería con largos tiempos de vuelo.
- 38 minutos de tiempo máximo de vuelo.
- Certificado IP43.
- Protección contra el polvo y el agua.
- 2 kg carga máxima.

Dentro de los drones de la empresa DJI es el que mejor se adapta a las características del proyecto por su capacidad de integración de desarrollo y su capacidad de cargar diversos sensores. También resiste climas inhóspitos y polvo [\[62\]](#).

Fue el dron que se había seleccionado como el mejor y el que se iba a comprar, pero por los inconvenientes detallados en el [capítulo 12.2](#) debió ser descartado.

### 5.1.7. Placas ArduPilot



Ilustración 23: Placa ArduPilot APM 2.5

Una vez descartada la marca DJI por sus costos, se continuó con la investigación para la compra del dron.

Teniendo en cuenta que la mayoría de los proveedores que se investigaron no venden drones que se adapten a los requerimientos, hubo que darle un giro a la investigación, y así fue que el equipo conoció las placas madre creadas por ArduPilot.

#### ¿Qué es ArduPilot?

Ardupilot es un proyecto Open Source que provee funcionalidades para el manejo de drones. Cuenta con una documentación [\[63\]](#) muy detallada y un foro de consulta [\[64\]](#) manejado por la comunidad.

Esta placa, y el conjunto de sensores que la complementan está diseñada para usarse como piloto automático en aviones, helicópteros, multirrotores, barcos, coches, entre otros [\[65\]](#).

ArduPilot, para realizar las funciones de procesamiento de información cuenta con dos procesadores desde los que es capaz de gobernar todos los sistemas integrados, tal y como estabilización autónoma, navegación por GPS, vuelo por *waypoints*, sistemas *Fail Safe*, control de cámara, etcétera.

#### 5.1.7.1. Características

- *Firmware* gratuito soportado para diferentes vehículos no tripulados.
- Configuración sencilla mediante la descarga del *firmware* a la placa vía USB, si no se quieren mayores complicaciones no es necesaria la programación, aunque si se quiere mejorar o añadir nuevas características usa el mismo código que Arduino [\[66\]](#).
- Programación de misiones haciendo clicks sobre el mapa de la estación de tierra.
- Soporta cientos de *waypoints* 3D.
- Telemetría bidireccional en tiempo real usando el protocolo MAVlink [\[67\]](#).
- Elección de múltiples estaciones de tierra para gobernar el control del vehículo en tiempo real, incluyendo múltiples características como la recepción de vídeo o la síntesis de instrucciones de voz.
- Aterrizaje y despegue autónomo.

Estas placas, al ser genéricas, son utilizadas por muchas empresas fabricantes de drones.

Se decidió que era la mejor opción ya que permite que en un futuro Todosoft pueda cambiar de proveedor de dron y tener una gran gama de alternativas, debido a que todos los drones que utilicen la placa de ArduPilot son compatibles con el *software*.

A su vez, las empresas que fabrican drones y utilizan estas placas suelen brindar un servicio mucho más personalizado, por lo que el cliente puede comprar drones a futuro y adaptar diferentes lectores de RFID, sumarle cámaras fotográficas, nuevos sensores, protectores, etcétera.

Existen diferentes aplicaciones que trabajan con la SDK de ArduPilot, algunos de ellos son:

- Mission Planner - escrito en C# [\[68\]](#)
- QGroundControl - escrito en C++ [\[69\]](#)
- APM Planner 2.0 - versión vieja de Mission Planner [\[70\]](#)

Estos programas fueron de mucha utilidad para aprender a utilizar la SDK, hacer pruebas y validar el *software* desarrollado.

### 5.1.8. Accsys - Sky Track



Ilustración 24: Accsys Innovation

Finalmente, luego de analizar todas las opciones de drones, se decidió adquirir el dron modelo Sky Track, fabricado por la empresa mexicana Accsys.

Se compró un dron que viene con el lector RFID integrado (algo fundamental para reducir al mínimo el trabajo eléctrico con *hardware*), que viene con la placa de ArduPilot, cuenta con soporte inmediato y a la empresa Todosoft le sirve ya que se le pueden cambiar partes de *hardware*, adaptar lectores, hacer modificaciones, debido al nivel de personalización que ofrece este proveedor.

Accsys es una empresa mexicana, reconocida a nivel mundial y que ha obtenido premios por la calidad de sus productos (primer premio por innovación en “Industria 4.0” 2018).

#### 5.1.8.1. Características del dron

##### **Gran rendimiento Plataforma**

- Aplicaciones de trabajo pesado como la recolección masiva de datos, la visualización en tiempo real, se pueden operar de manera fluida.
- Procesador integrado ARM Cortex-A53, 1.2Ghz.
- *Network* 10/100 Ethernet y 2.4 Ghz 802.11n Wi-Fi.
- Bluetooth 4.1.
- GPIO 40 pines.
- 1Gb en RAM.
- Pantalla *touch* capacitiva de 3.5” instalada en dron para poder visualizar información.

##### **Android**

- Toda la información de vuelo se puede recibir automáticamente desde un dispositivo Android o una PC.

##### **Materiales y características**

- *Frame*, bases, hélices y trenes de aterrizaje están hechos 100 % de fibra de carbono, lo cual lo hace ligero y resistente, con un peso total de 3.0 kgs.
- Soporta una carga total de hasta 4.8 kgs (Incluyendo peso del dron).
- Medidas: ancho de ala 23 cms, largo 64 cms.
- Velocidad máxima 50 km/h.

##### **Componentes de óptima calidad**

- La cámara integrada, así como los procesadores y componentes RFID en el dron son los mejores del mercado y aseguran la estabilidad en campo como tarjetas de control de procesadores ARM.

##### **Ultra largo tiempo de operación**

- Sistema de baterías intercambiables, hace que el dron utilice la batería de reserva más conveniente, por lo que el tiempo de funcionamiento es ilimitado.

### **Módulo de gama alta UHF RFID**

- Con el *software* Sky Track y el módulo de lectura RFID Impinj Indy R2000 [\[71\]](#) (uno de los mejores del mercado para tags pasivos), ofrece el mejor rendimiento y una excelente capacidad de identificación de múltiples etiquetas en la industria con alcance de hasta 8 metros (depende del tag), donde las lecturas de RFID y las coordenadas GPS se pueden enviar por Wi-Fi o descargar a una PC vía ethernet o memoria micro SD, el envío de datos 3G es opcional.

### **Motores de alto poder**

- Equipado con 4 motores *Brushless* de 592 W, que permite cargar hasta 3 kgs de peso y poder operar en condiciones de viento moderadas.

### **Hélices de alto desempeño**

- Equipado con 4 hélices tipo APC 11 x 55 intercambiables con resistencia a impactos leves.

## **5.2. Tipo de caravana RFID a utilizar**



Ilustración 25: Tags para vacas utilizados en el proyecto

La idea es que el dispositivo de lectura que se ubica en el dron (antena o *handheld*) emite una sonda de radiofrecuencia y los chips (*tags*) le responden, logrando así la identificación del mismo.

El propósito fundamental de la tecnología RFID es identificar mediante un lector, sin contacto y a distancia, una tarjeta o etiqueta (*tag*) portada por una persona, un animal, un vehículo en movimiento o cualquier producto que se encuentra en un almacén o en una cadena de producción automatizada.

Existen diferentes tipos de caravanas o *tags* dependiendo del:

- Alcance
- Tamaño
- Material

El código de identificación que contienen estas caravanas es único y puede ser personalizado durante la fabricación de la misma.

Los *tags* incluyen en su interior un pequeño chip y una antena impresa o bobinada para comunicarse con el lector. El chip tiene grabado un número (ID) de serie único que lo identifica entre los demás, y puede disponer de una pequeña memoria para guardar datos, que los lectores son capaces de leer y escribir.

### 5.2.1. Tipos de *tags*

#### **Pasivos**

No necesitan alimentación interna, toman la energía de la propia emisión de las antenas y solo se activan cuando se encuentran en el campo de cobertura de la misma.

#### **Activos**

Utilizan alimentación propia de una pequeña batería y por ende pueden comunicarse con el lector a una distancia mucho mayor y procesar una cantidad de datos superior.

Por lo tanto para el desarrollo del sistema SkyGuard se utilizan *tags* RFID de tipo activo debido a que tienen un alcance mucho mayor de lectura, lo que permite que el dron sea más rápido en la recorrida, lo cual es importante para optimizar el uso de la batería del mismo.

El [anexo 1](#) de este documento contiene información más detallada sobre la investigación de RFID realizada.

### 5.3. Procesamiento de imágenes

Teniendo en cuenta que la gran mayoría de los drones cuenta con cámara de fotos y videos, se realizó una investigación para evaluar la viabilidad de utilizar las fotografías obtenidas por el dron para reconocer jauría de perros en los campos, debido a que fue uno de los problemas surgidos en las entrevistas con expertos, y aunque finalmente no formó parte del alcance del presente proyecto, fue tenido en cuenta a la hora de recomendar una solución extensible que en un futuro pudiera incorporar esta prestación.

Se encontraron varias librerías (la mayoría pagas) para reconocer objetos a partir de fotografías. A continuación se muestra cómo sería el proceso utilizando la API de Google.

#### 5.3.1. Google Cloud Vision API

Es una API de Google que permite obtener información de las imágenes utilizando el aprendizaje automático pre entrenado [\[72\]](#).

En SkyGuard este reconocimiento de imágenes se podría utilizar para el reconocimiento de jaurías de perros en campos. Por lo tanto alcanza con utilizar la funcionalidad de la API que permite detectar objetos y retornarlos como una lista de etiquetas.

La clasificación de los objetos reconocidos se puede hacer con etiquetas predefinidas o personalizadas.

### 5.3.1.1. Precio de la API

PRICE PER 1,000 UNITS, BY MONTHLY USAGE			
FEATURE	1-1,000 UNITS/MONTH	1001-5,000,000 UNITS/MONTH	5,000,001-20,000,000 UNITS/MONTH
Label detection	Free	\$1.50	\$1.00

Ilustración 26: Precios de Google Cloud Vision API

### 5.3.1.2. Instalación

Soporta diferentes lenguajes [\[73\]](#) como: C#, GO, Java, NodeJS, PHP, Python, Ruby.

### 5.3.1.3. Prototipo

El prototipo construido para probar la API arrojó los siguientes resultados. Dada esta imagen:



Ilustración 27: Ovinos guiados por canino

La llamada a la API desde NodeJS retorna los siguientes elementos reconocidos:

```
C:\Users\PC\Desktop\Test Reconocimiento>node reconocimiento_etiquetas.js
Labels:
Mammal
Vertebrate
Dog
Canidae
Sheep
Sheep
Herd
Dog breed
Herding
Border collie

C:\Users\PC\Desktop\Test Reconocimiento>_
```

Ilustración 28: *Tags* reconocidos por la librería de Google

En la imagen anterior se puede observar que la API identificó un “*Dog*” (un perro) entre varias ovejas, lo que coincide con las expectativas que el equipo manejaba.

El código necesario, escrito en JavaScript (para correrlo sobre NodeJS), es el siguiente:

```
quickstart();

async function quickstart() {
  // Imports the Google Cloud client library
  const vision = require('@google-cloud/vision');

  // Creates a client
  const client = new vision.ImageAnnotatorClient();

  // Performs label detection on the image file
  const [result] = await client.labelDetection('./resources/wakeupcat.jpg');
  const labels = result.labelAnnotations;
  console.log('Labels:');
  labels.forEach(label => console.log(label.description));
}
```

## 5.4. Digitalización de planos

Para facilitar la creación del perímetro de los campos, se realizó una investigación para ver cómo se podía obtener esta información de manera digital.

Luego de asesorarse con un estudio de ingenieros agrimensores, se llegó a la conclusión de que la mejor alternativa es utilizar la información disponibilizada en el sitio oficial de Catastro de cada país [74].

La Dirección Nacional de Catastro (Uruguay) cuenta con una plataforma en donde se encuentran digitalizados todos los padrones registrados en el país, y su consumo es gratuito. Los datos se encuentran categorizados por departamentos.



Dirección Nacional de Catastro

### Shapes del parcelario rural y urbano

1 usuarios siguen este Conjunto de Datos Compartir

Archivos shape para descargar del parcelario del Uruguay. Para todo el país e individualmente por departamento.

Campo	Valor
Fuente	www.catastro.gub.uy
Autor	Augusto Azeni
Mantenedor	DNC
Última actualización	7 de febrero de 2020, 12:28 (UTC-03:00)
Creado	26 de julio de 2016, 16:30 (UTC-03:00)
Licencia	Licencia de DAC de Uruguay

Ilustración 29: Sitio web de Catastro con el parcelario

Desde la plataforma en línea de Catastro se pueden descargar el parcelario uruguayo, para así obtener las coordenadas geográficas de los diferentes padrones del Uruguay.

Los datos se descargan en archivos .SHP, los cuales internamente almacenan los padrones y el polígono que delimita cada uno de ellos, puede observarse en la siguiente imagen realizada desde el programa ArcGis [75] para el padrón 105 del departamento de Florida.



Ilustración 30: Polígono del campo seleccionado

Una vez obtenido el polígono para un padrón en específico, se podrían obtener las coordenadas e importarlas en el sistema SkyGuard, para de esta manera automatizar la creación del perímetro de un campo.

COORD					
	FID	Shape *	PADRON	X	Y
▶	0	Point	105	-56,3958	-34,3174
	1	Point	105	-56,3964	-34,3214
	2	Point	105	-56,3999	-34,3207
	3	Point	105	-56,3982	-34,3157
	4	Point	105	-56,3982	-34,3154
	5	Point	105	-56,3954	-34,316
	6	Point	105	-56,3955	-34,3162
	7	Point	105	-56,3956	-34,3164
	8	Point	105	-56,3956	-34,3167
	9	Point	105	-56,3957	-34,3167
	10	Point	105	-56,3957	-34,3169
	11	Point	105	-56,3957	-34,317
	12	Point	105	-56,3958	-34,3172
	13	Point	105	-56,3958	-34,3173
	14	Point	105	-56,3958	-34,3174

Ilustración 31: Lista de coordenadas del padrón seleccionado

## 6. Arquitectura

A continuación se expone en detalle la arquitectura llevada a cabo, con el fin de cumplir los requerimientos tanto funcionales como no funcionales del sistema, que se detallan en el [capítulo 4](#).

En este capítulo se encuentra una descripción detallada de la arquitectura, seguido por las principales decisiones de diseño tomadas, el análisis de las mismas, y los atributos de calidad del sistema priorizados con su justificación correspondiente.

Varios aspectos de la arquitectura están fuertemente atados a las tecnologías que se eligieron para la solución del sistema y son de tal importancia que se dedica una sección entera ([capítulo 6.4](#)) para hablar sobre su elección y por qué se decidió por las mismas.

### 6.1. Descripción general

La arquitectura planteada está enfocada en solucionar los requerimientos del sistema acordados por hacer, tanto funcionales como no funcionales, respetando las restricciones (ver [capítulo 4.2.4](#)) del mismo, y pensando en su futura evolución en etapas posteriores que no se encuentran en el alcance de este proyecto.

La solución tiene cinco componentes principales:

1. Una aplicación móvil que compila a lenguaje nativo Android y Swift (iOS).
2. Una aplicación web, pensada para gestionar el sistema desde un ordenador.
3. Una API REST, los servicios web de la misma serán consumidos por las aplicaciones anteriormente mencionadas, tanto la móvil como la web.
4. Un *backend* de servicios web que permite ejecutar la aplicación que envía instrucciones al dron.
5. Un *backend* que se ejecuta, y hace de nexo entre los servicios web del *backend* y las instrucciones que debe ejecutar el dron por telemetría.

Para las aplicaciones de *frontend* (siendo estas las aplicaciones móviles y la aplicación web) se decidió utilizar una tecnología que favorece al reuso de código (se detalla con mayor precisión en el [capítulo 6.4](#)) por lo cual entre ellas se comparte toda la lógica de llamadas a los

servicios web, además ambas se comunican con la API de Google Maps para disponibilizar la información del campo y el ganado en un mapa.

Los servicios web con los que se comunican estas aplicaciones se encuentran casi en su totalidad alocados en una única aplicación monolítica, ya que lo que se priorizó en la solución fue maximizar los requerimientos funcionales a pedido del cliente.

Fue necesario hacer una segunda aplicación de servicios web para enviarle instrucciones al dron, esto es debido a que el mismo recibe instrucciones a través de telemetría, y por ende necesita estar a no más de 200 metros de distancia del sistema que desea ejecutar las instrucciones. Debido a esta restricción es necesario ubicar un servidor físico cerca de la base del dron; el servidor expone servicios web para poder recibir solicitudes de la API REST que se ubica en la nube y poder comunicarse con el dron a través de estos servicios.

Debido a que el dron necesita exponer servicios REST desde el campo es necesario utilizar alguna herramienta que permita exponer una conexión para que la API REST pueda encontrarla. Para eso puede utilizarse cualquiera de los servicios de resolución de nombres con IP dinámica, como por ejemplo DynDNS, No-IP o ngrok. En el caso del prototipo realizado para la tesis elegimos DynDNS.

Dentro de los servicios web, hay servicios específicos que se comunican con el *backend* de servicios web del dron; la finalidad del mismo es poder ejecutar la aplicación que envía instrucciones al dron, encapsulando las complicaciones estas y proveyendo una interfaz clara para que el *backend* de servicios web se comunique con la misma.

En el siguiente diagrama [\[76\]](#), se puede ver cómo se comunican las aplicaciones entre sí.

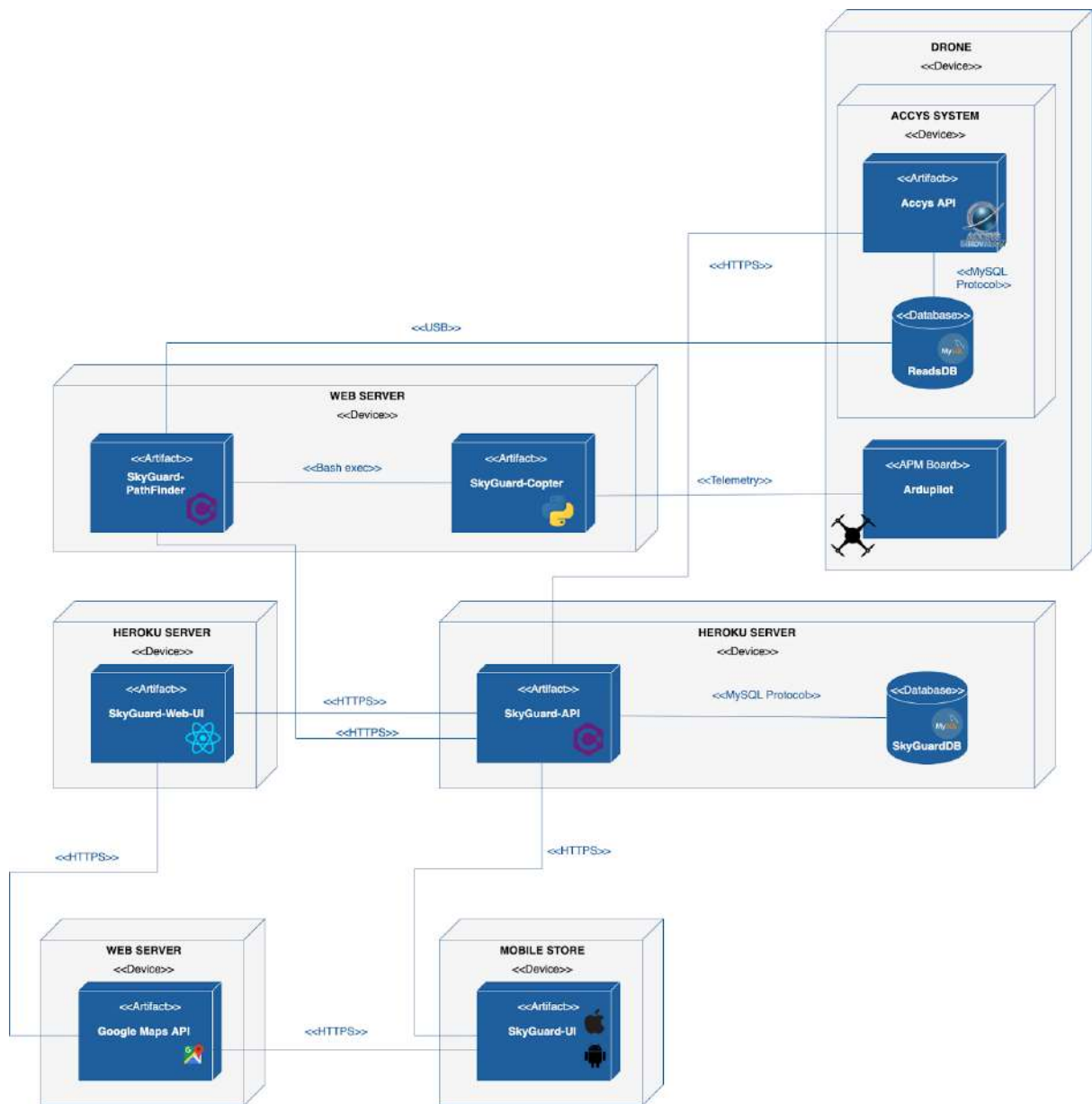


Ilustración 32: Diagrama de despliegue

### Catálogo de elementos

Elemento	Descripción
Heroku server	Plataforma en la cual se encuentran desplegadas las aplicaciones web y la base de datos.
Drone	Vehículo aéreo que recibe instrucciones del sistema.
Web server	Servidor de aplicaciones web.

Accys System	Sistema de terceros Accys, contiene la lectora RFID, guarda sus lecturas locales, envía lecturas en tiempo real por un servicio web, y deja abierto un puerto para conectarse a la base de datos interna.
Mobile store	Servicio en donde se exponen aplicaciones móviles para poder ser descargadas por los usuarios desde sus celulares.
SkyGuard-UI	Aplicación <i>frontend</i> móvil que usan los usuarios para interactuar con el sistema como punto de entrada.
SkyGuard-Web-UI	Aplicación <i>frontend</i> web que usan los usuarios para interactuar con el sistema como punto de entrada.
SkyGuard-API	Servicios web que son consumidos desde las aplicaciones de <i>frontend</i> , guardan datos en la base de datos y se comunican con servicios que permiten manipular al dron.
SkyGuard-Pathfinder	<i>Backend</i> que crea un plan de vuelo y se comunica con el <i>backend</i> del dron.
SkyGuard-Copter	<i>Backend</i> que envía instrucciones al dron.
SkyGuardDB	Base de datos donde se almacena toda la información del sistema.
Accys API	Envía lecturas del dron a la API en tiempo real.
ReadsDB	Guarda las lecturas obtenidas por el lector RFID que se encuentra en el dron.
Google Maps API	API de terceros para geolocalización.

Tabla 1: Catálogo de elementos del diagrama de despliegue

## 6.2. Diseño

En esta sección se mencionan las principales decisiones de diseño tomadas y el por qué de las mismas, además se detallan los flujos más característicos y las diferentes interacciones que tienen entre sí las aplicaciones de la solución propuesta.

Los diagramas presentados en esta sección son representaciones primarias de los mismos y pueden no tener todos los detalles de la solución implementada en código.

Se muestra para comenzar una vista de los componentes más característicos con el fin de entender cómo se relacionan los mismos en tiempo de ejecución.

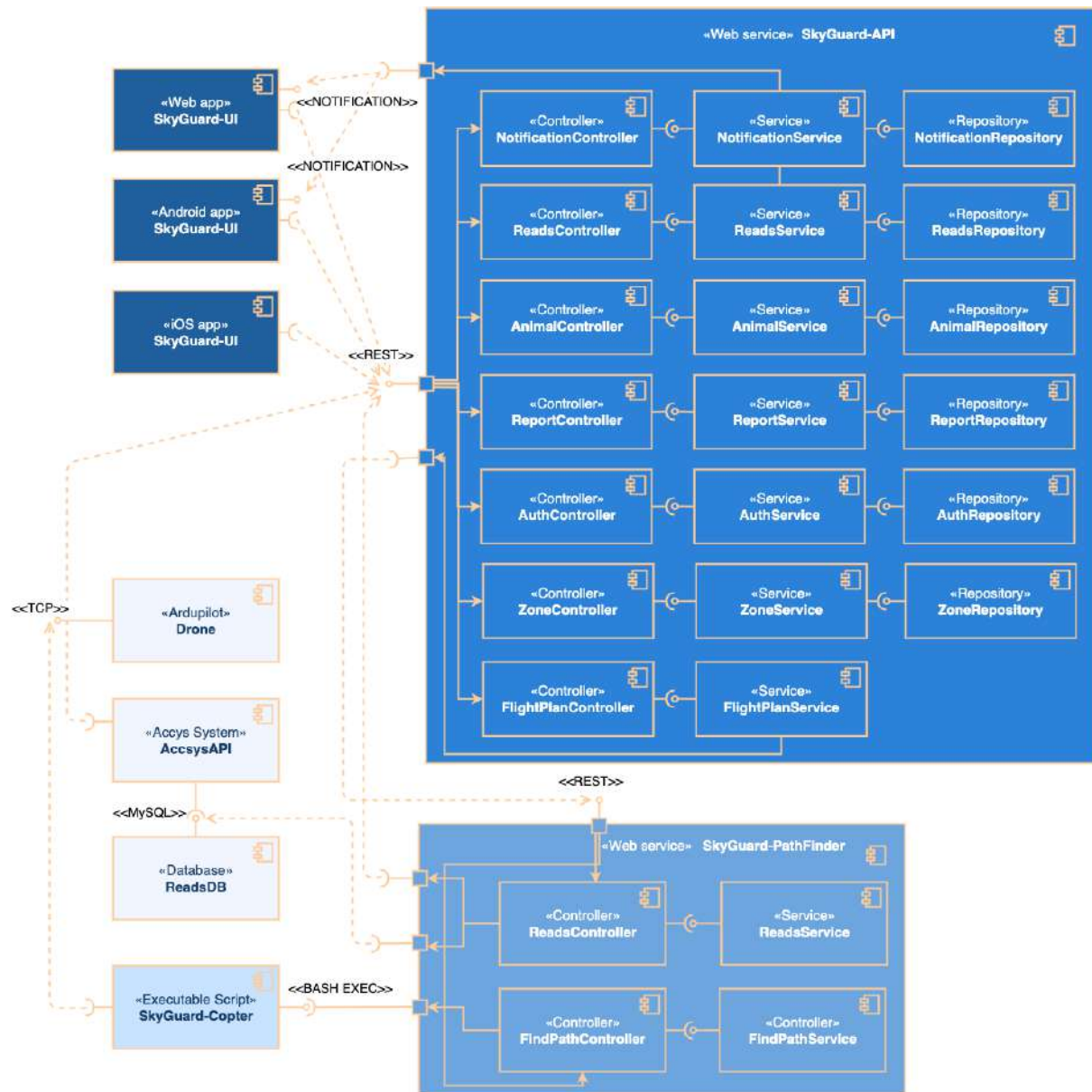


Ilustración 33: Diagrama de componentes

### Catálogo de elementos

Elemento	Descripción
SkyGuard-UI	Componente de <i>frontend</i> , ya sea web o móvil.
Drone	Vehículo aéreo que viaja por el campo.

AccsysAPI	Componente que se comunica con la API para enviar lecturas en tiempo real.
ReadsDB	Base de datos donde se almacenan las lecturas recolectadas por el sistema de Accsys, además permite que se hagan consultas a ella en caso de que fallen las transmisiones <i>online</i> en tiempo real.
SkyGuard-Copter	<i>Script</i> que genera un set de instrucciones al dron, además de armar al mismo (dejarlo listo para el comienzo de su vuelo).
SkyGuard-API	Componente que expone una API REST, único punto de entrada desde las aplicaciones de <i>frontend</i> .
SkyGuard-Pathfinder	Componente que ejecuta el <i>script</i> de SkyGuard-Copter y además expone una API REST para hacer un pedido manual de las lecturas del dron.
NotificationController	Punto de acceso para <i>endpoint</i> de notificaciones.
ReadsController (SkyGuard-API)	Punto de acceso para <i>endpoint</i> de lecturas.
AnimalController	Punto de acceso para <i>endpoint</i> de animales.
ReportController	Punto de acceso para <i>endpoint</i> de reportes.
AuthController	Punto de acceso para <i>endpoint</i> de login y accesos.
ZoneController	Punto de acceso para <i>endpoint</i> de zonas.
FlightPlanController	Punto de acceso para <i>endpoint</i> de plan de vuelo.
NotificationService	Servicio que contiene la lógica de las notificaciones.
ReadsService (SkyGuard-API)	Servicio que contiene la lógica de las lecturas.
AnimalService	Servicio que contiene la lógica de los animales.
ReportService	Servicio que contiene la lógica de los reportes.
AuthService	Servicio que contiene la lógica del <i>login</i> y los accesos.
ZoneService	Servicio que contiene la lógica de las zonas.
FlightPlanService	Servicio que contiene la lógica de los planes de vuelo.
NotificationRepository	Componente que controla el acceso a datos de las notificaciones.
ReadsRepository	Componente que controla el acceso a datos de las lecturas.

AnimalRepository	Componente que controla el acceso a datos de los animales.
ReportRepository	Componente que controla el acceso a datos de los reportes.
AuthRepository	Componente que controla el acceso a datos de la autenticación.
ZoneRepository	Componente que controla el acceso a datos de las zonas.
ReadsController (SkyGuard-Pathfinder)	Punto de acceso para <i>endpoint</i> de lecturas.
ReadsService (SkyGuard-Pathfinder)	Servicio que contiene la lógica de las lecturas.
FindPathsController	Punto de acceso para <i>endpoint</i> de plan de vuelo.
FindPathsService	Servicio que contiene la lógica de los planes de vuelo.

Tabla 2: Catálogo de elementos del diagrama de componentes

Para comprender cómo se relacionan e identifican las distintas entidades del dominio del sistema se presenta el siguiente diagrama:

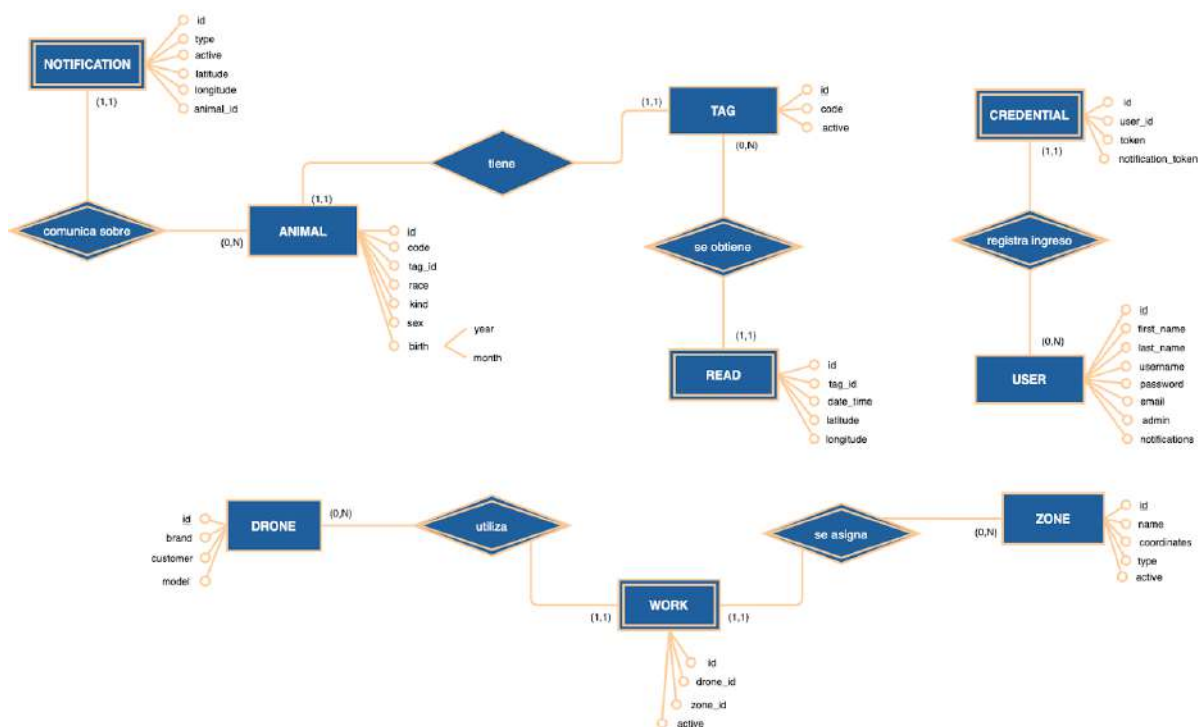


Ilustración 34: Diagrama Modelo Entidad-Relación

Debido a que realizar un recorrido con el dron resulta uno de los flujos más interesantes y complejos del sistema se decide explicarlo en detalle.

Para comenzar un recorrido con el mismo, es necesario indicar en qué zona se desea que se realicen las lecturas, e indicarle al dron que comience su recorrido.

Se ejemplifica con un diagrama de secuencia de alto nivel:

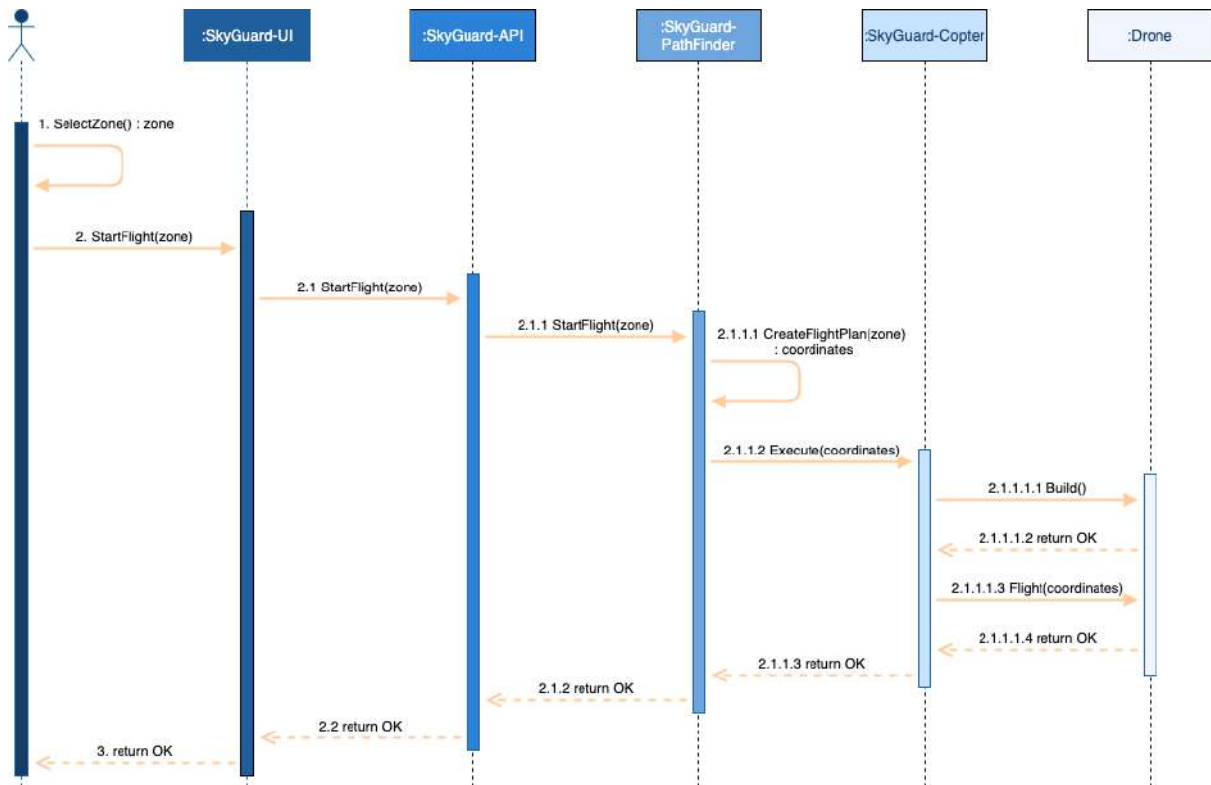


Ilustración 35: Diagrama de secuencia - iniciar recorrido

1. El usuario elige desde la aplicación web que desea volar el dron en una zona específica.
2. Se notifica a la API que se desea iniciar un vuelo.
3. La API consume un servicio web de SkyGuard-Pathfinder y le indica la zona que desea recorrer.
4. SkyGuard-Pathfinder arma un plan de vuelo con la estrategia de cómo recorrer esa zona.

- Una vez que SkyGuard-Pathfinder sabe por dónde debería volar el dron ejecuta SkyGuard-Copter (el *script* escrito en Python), indicando el recorrido, que es el encargado de armar al dron y pedirle que comience con su misión.

En ese viaje el dron empezará a recolectar las lecturas RFID con una lectora que lleva montada, la forma de obtener las lecturas y generar alertas en base a ellas se puede dar tanto en tiempo real, como luego de que el dron vuelve a su base. Esto es debido a que en el campo es esperable quedarnos sin conexión a internet, y por esto mismo, es necesario que esas lecturas se hagan una vez que el dron cuente con internet estable (es decir, se encuentre en la base), en caso de no poder hacerlas en tiempo real, para un mejor entendimiento.

El siguiente diagrama muestra el “camino feliz” de la recolección de lecturas en el dron, enviándolas a la API a medida que las va obteniendo; en caso de observar que los animales no se encuentran en el predio, o los mismos se encuentran dentro de una zona peligrosa, se disparan alertas que llegan como notificación a los celulares de los usuarios.

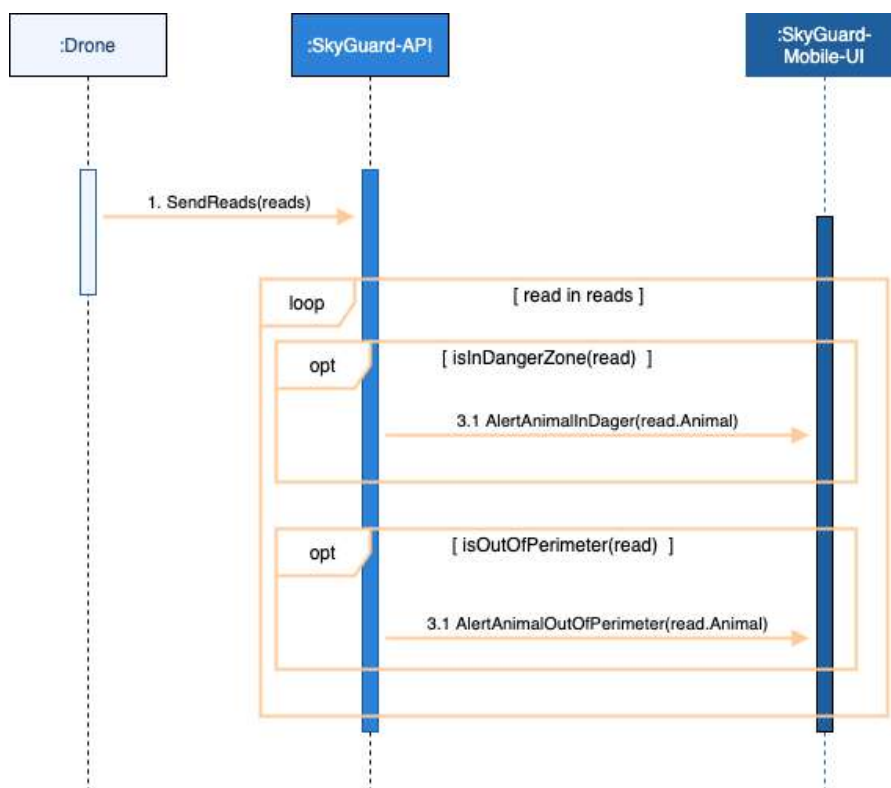


Ilustración 36: Diagrama de secuencia - lecturas en tiempo real

Como se mencionó anteriormente, existe la posibilidad de no tener una conexión de internet estable para enviar las lecturas a medida que se van recolectando y es por esto mismo que es necesario una opción manual, luego de que el dron regresa a la base, la conexión es siempre estable y es necesario que una persona lo conecte al servidor web para poder pedir las lecturas como se puede observar a continuación.

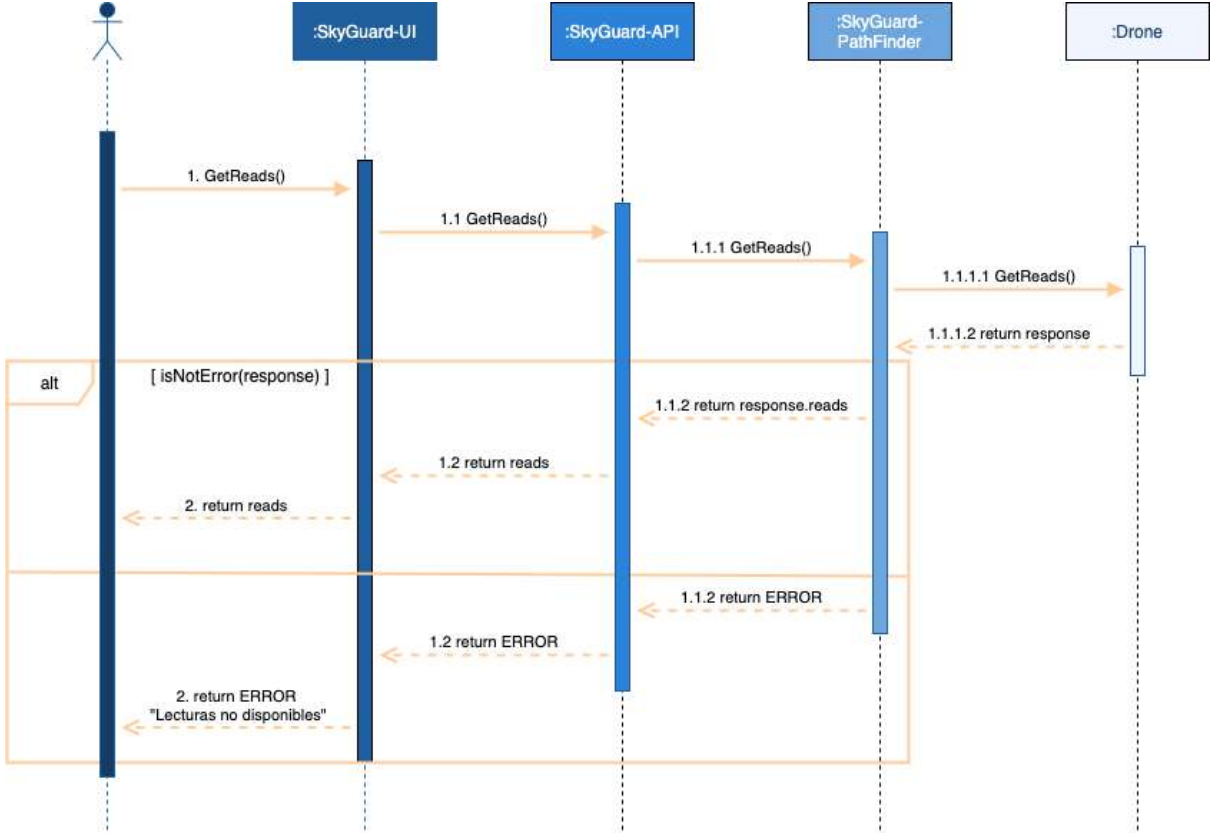


Ilustración 37: Diagrama de secuencia - lecturas a demanda

El pedido manual de lecturas requiere que el dron esté conectado al servidor para poder obtenerlas de la base de datos interna de lecturas del mismo, en caso de no poder conectarse a la base del dron se le retorna un error al usuario.

Se decide incluir el flujo del alta de un animal para mostrar cómo se modelaron los CRUD de la aplicación ya que en general su solución es análoga al resto de los componentes.

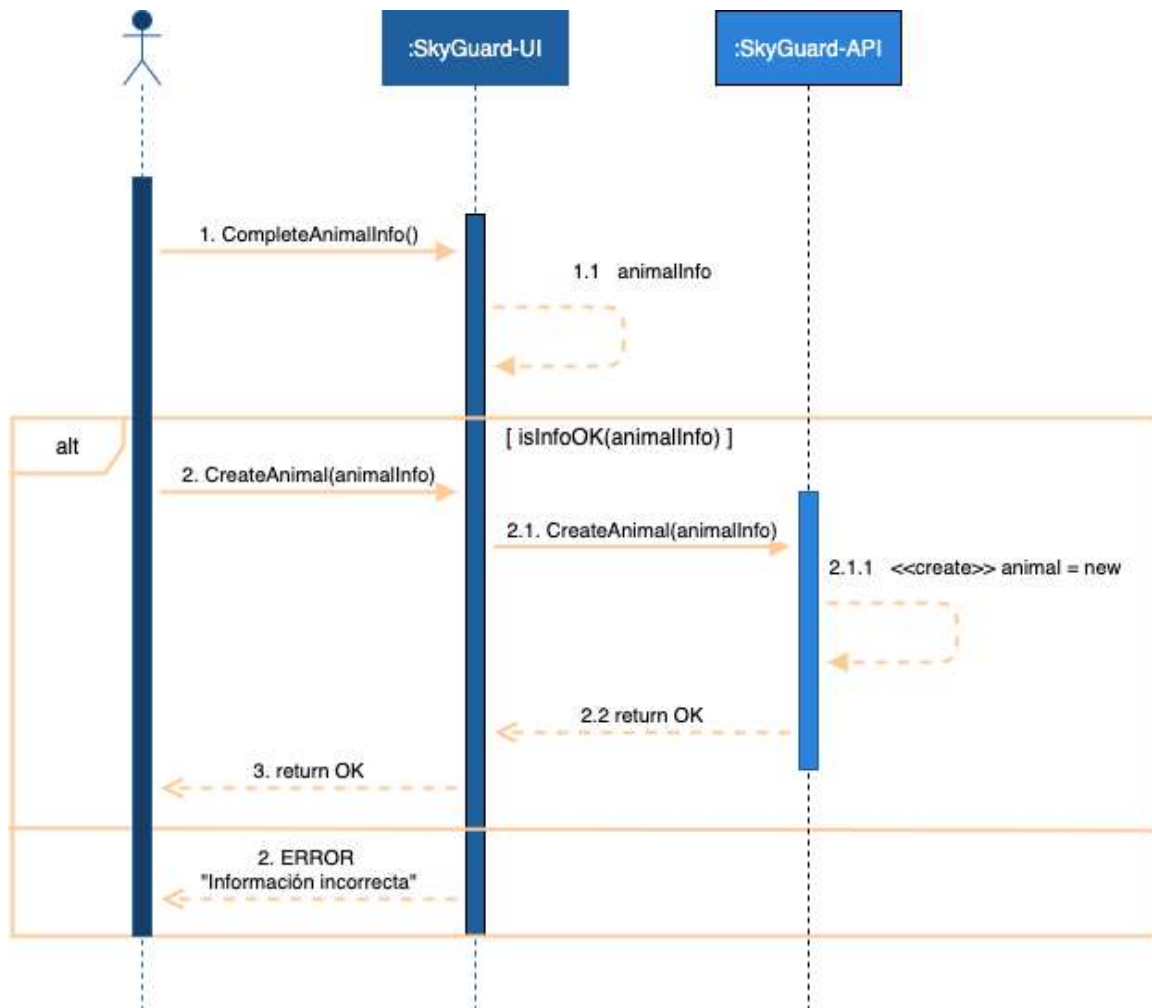


Ilustración 38: Diagrama de secuencia - alta de animal

Como se puede observar en el diagrama anterior, es un flujo muy sencillo, donde las aplicaciones que participan son solo dos, un *frontend* (ya sea web o móvil) y la API del sistema. Al ser este un flujo muy pequeño del sistema, se decide mostrar las principales clases que intervienen en él y en el resto de las operaciones de animales.

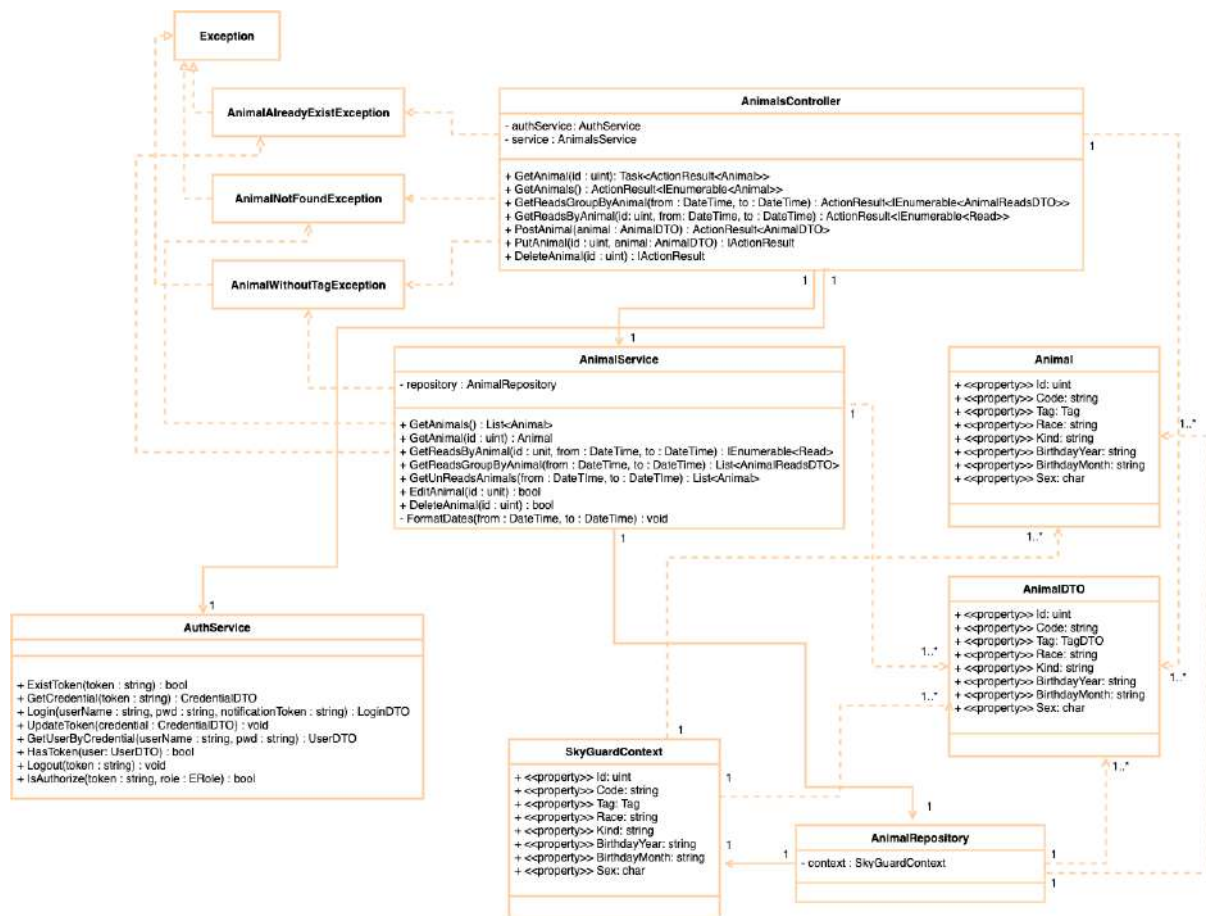


Ilustración 39: Diagrama de clases - CRUD Animales

En este diagrama se puede ver claramente la responsabilidad de cada clase en la gestión de animales.

### 6.3. Atributos de calidad

Se detalla a continuación cuáles fueron los atributos de calidad considerados a la hora de contemplar los requerimientos no funcionales del sistema. Se priorizaron con el cliente y se eligió atacar en profundidad dos de ellos; aunque también se trabajó con el resto de los atributos de calidad que aparecen en esta sección.

Cabe destacar que algunos de los requerimientos no funcionales se resuelven en su totalidad a la hora de elegir la tecnología, como el requerimiento no funcional de la compatibilidad entre distintos navegadores (que se detalla con mayor profundidad en el [capítulo 6.4](#)) y la

resistencia a las inclemencias climáticas moderadas por parte del dron, eligiendo un dron que pueda volar con dichas inclemencias.

### 6.3.1. Modificabilidad

Uno de los dos atributos de calidad más importantes del sistema es la modificabilidad [77], debido a que el alcance del proyecto es una versión inicial del mismo, y se desean incorporar muchas funcionalidades nuevas luego de culminado el mismo, dichos desarrollos futuros los planea llevar a cabo el cliente.

Además, el cliente cuenta con otra idea de negocio para realizar inventario de mercaderías en depósitos con drones y era fundamental para él poder reutilizar toda la lógica que fuera posible.

#### **Objetivos**

El sistema debe ser capaz de incorporar nuevas funcionalidades con un impacto bajo en las ya existentes; a su vez debe soportar cambios previsibles de implementación.

Los componentes del sistema deben ser mantenibles y tener un bajo nivel de acoplamiento para que los cambios en el sistema impacten de la menor forma posible y sirva para que los miembros del equipo puedan trabajar lo más independientemente posible entre ellos, reduciendo así la cantidad de conflictos que se pueden producir entre tarea y tarea.

Existen varias funcionalidades nuevas conocidas que se desean agregar a futuro, como el reconocimiento de animales a través de imágenes, sensores de calor que junto con la lectura RFID reconozcan al animal que está siendo leído, entre otras.

#### **Implementación**

En principio se decidió utilizar el patrón de arquitectura en capas [78], las mismas dividen al *software* en unidades, cada capa es una agrupación de módulos que ofrece un conjunto cohesivo de servicios, además particionan completamente la lógica del *backend* estableciendo así un estricto orden de relaciones.



Ilustración 40: Diagrama de arquitectura en capas

Se puede visualizar los módulos establecidos en las capas mencionadas en el siguiente diagrama:

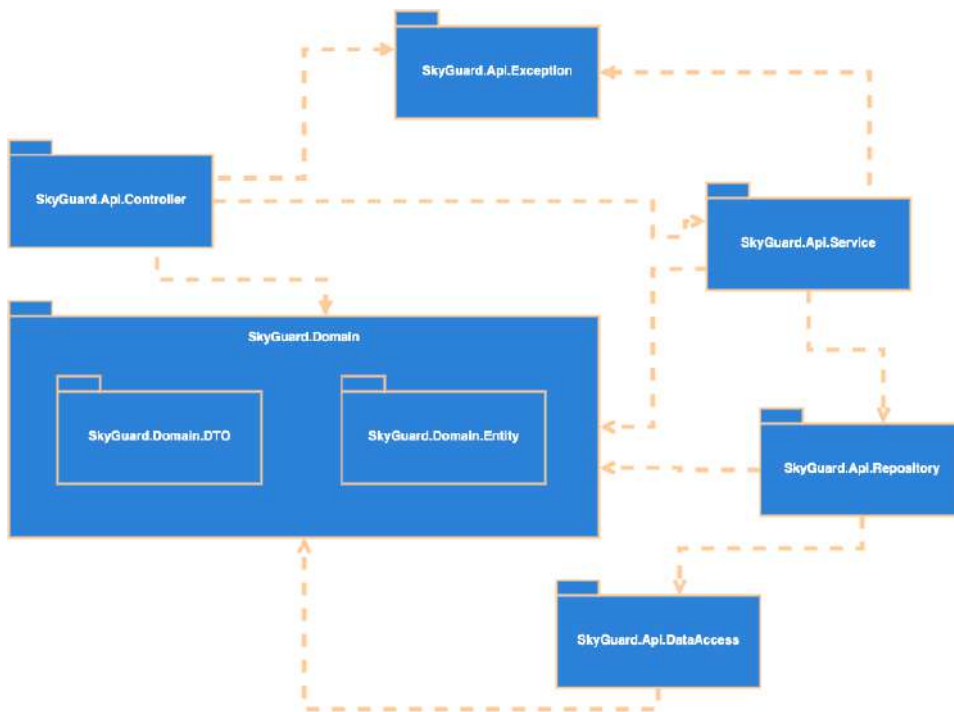


Ilustración 41: Diagrama de módulos - SkyGuard-API

## Catálogo de elementos

Elemento	Descripción
SkyGuard.Api.Controller	Módulo de la capa de presentación que contiene todos los controladores de la API.
SkyGuard.Domain	Módulo con los todos los datos del dominio, tanto entidades como las clases de transferencia de datos.
SkyGuard.Domain.DTO	Módulo con las clases de transferencia de datos.
SkyGuard.Domain.Entity	Módulo con las entidades del dominio.
SkyGuard.Api.Service	Módulo que contiene toda la lógica de negocio de la API.
SkyGuard.Api.Repository	Módulo que contiene la lógica de acceso a datos.
SkyGuard.Api.DataAccess	Módulo que contiene el contexto que accede a datos de la base de datos.
SkyGuard.Api.Exception	Módulo que contiene las excepciones derivadas de los flujos excepcionales de la lógica de la API.

Tabla 3: Catálogo de elementos del diagrama de módulos

Este patrón además permite tener un bajo acoplamiento entre los componentes del sistema *backend*.

Se eligió utilizar las tecnologías React y React Native para las aplicaciones de *frontend* con el fin de reutilizar lógica y beneficiar a la mantenibilidad del sistema y reuso de código (se detalla en mayor profundidad en la [sección 6.4](#)).

Para lograr este cometido se decidió usar el patrón de diseño High Order Component [\[79\]](#), el cual recibe un componente genérico de la vista y le introduce los datos que el mismo debe mostrar. Esto ayudó a que todas las llamadas a la API estén contenidas en un único lugar y se puedan reutilizar en distintos componentes de la UI, lo único que cambia es la forma en la que presentan los mismos datos.

Se utilizaron los principios de diseño de clase SOLID y los principios de asignación de responsabilidades GRASP, para que sean una guía en la codificación del *backend* de la solución, con el fin de generar un código entendible y que fuera sencillo de mantener. Se

establecieron interfaces claras para la comunicación entre las distintas capas del código, además de seguirse los estándares de codificación de los lenguajes utilizados que pueden ser leídos en el [capítulo 11](#), y documentación sobre el algoritmo de plan de vuelo, que puede ser leída en el [capítulo 9](#), con el fin de que sea más sencillo mantenerse en un futuro por el cliente.

### 6.3.2. Confiabilidad

El otro atributo de calidad más importante para cumplir con los requerimientos no funcionales del sistema es la confiabilidad, que la tomamos como un subatributo de la disponibilidad que usa el libro de referencia [\[80\]](#). Es clave para el éxito del sistema no perder ni estropear ninguna de las piezas del *hardware* ya sea el dron o la lectora RFID, debido al costo que tienen estos dispositivos y a la importancia de los mismos dentro de la solución general.

También es muy importante que el usuario pueda confiar en el sistema, percibiendo la menor cantidad de fallas posibles, ya que es gente que si bien tienen conocimientos suficientes de tecnología como para operar el sistema, son personas que vienen realizando algunas de las tareas del sistema de forma manual, y es necesario que vean el beneficio al uso del sistema y cómo mejora sus labores diarias.

#### **Objetivos**

El objetivo más importante es prevenir los fallos que impliquen la pérdida o rotura del *hardware* ya que de ocurrir tendría un costo lo suficientemente alto como para que el sistema no fuera ni rentable ni confiable; además, mientras no se solucione la falla de *hardware* el sistema quedaría fuera de funcionamiento en la mayoría de sus componentes.

Otros de los objetivos que se plantearon fueron detectar fallos y recuperarse de los mismos. Esto es fundamental debido a que el sistema no contará con internet en varios momentos durante la recorrida del dron por el campo, sin embargo se mandan lecturas en tiempo real. A su vez se desea generar confianza en los usuarios a la hora de utilizar el sistema.

## Implementación

En caso de que el dron se encuentre sin batería, o encuentre algún fallo inesperado, el mismo regresará a la base más cercana para poder resguardar su integridad física, además de no salir a hacer su recorrida si las inclemencias climáticas son muy desfavorables, ya que el dron resiste inclemencias climáticas moderadas. Esto último se debe realizar de forma manual, sin embargo se piensa consultar a algún servicio de meteorología en futuras versiones de la solución.

Por otro lado, para contemplar el caso en el cual el dron se queda sin internet y no puede mandar las lecturas a la API, se ignora el comportamiento fallido, es decir, es esperable que se quede sin conexión durante sus recorridos por lo cual, son fallos que si ocurren no tienen mayor implicación, una vez llegado el dron a la base se podrá sincronizar las lecturas que no fueron enviadas durante el recorrido del dron en forma manual.

Se decidió desplegar las aplicaciones en plataformas que permitieran contar con redundancia activa poder contar con varias instancias tanto de en los servicios web, como de la aplicación web, pudiendo así contar con varios nodos activos por aplicación. De esta forma en caso que alguna instancia se encuentre fallando, se pueda utilizar una instancia saludable reduciendo así el tiempo en el cual se podría percibir al sistema fallando.

Para la solución presentada se decidió utilizar Heroku [\[81\]](#) por un tema de costos y practicidad, en cambio para su versión comercial se prevé el uso de una nube como Azure ya que el cliente cuenta con otros sistemas de *software* funcionando en la misma.

Es una restricción del sistema actual tener que conectarse en forma manual para acceder a la base de datos interna del dron y obtener las lecturas, ya que ese componente de *software* está dado por terceros, es posible que ciertos cambios se negocien a futuro con Accsys, en torno a este punto en particular con el fin de automatizar más la solución.

Por último se tienen en cuenta las posibles excepciones que pueden ocurrir en el sistema, ya sea en tiempo de ejecución utilizando programación defensiva [\[82\]](#) e intentando no provocar estos fallos, también teniendo en cuenta los flujos excepcionales conocidos, capturando las excepciones no solo en el *backend* sino en la llamadas a la API desde el *frontend*, y

manejándolas de forma tal que el sistema se pueda recuperar del fallo, pudiendo reintentar cierta operación y registrar el fallo para que pueda ser solucionado, todas estas casuísticas fueron probadas para asegurar el correcto funcionamiento del sistema, información más detallada sobre las pruebas se encuentra en el [capítulo 11](#).

### 6.3.3. Seguridad

#### **Objetivos**

Se busca identificar a los usuarios del sistema y limitar sus accesos según un rol del usuario, éste rol deberá tener relación con las interacciones que el mismo debe realizar con el sistema. Además se necesita que los datos de las lecturas almacenadas por el dron no puedan ser leídos por terceros en caso de hurto [\[83\]](#).

#### **Implementación**

Se creó un *login* en el cual se identifica el usuario que quiere acceder al sistema, se lo autentica y autoriza al uso del mismo siempre y cuando sea un usuario legítimo del sistema, para ello el usuario brinda su nombre de usuario y su contraseña, el *backend* es el encargado de verificar esta información y crea un *token* de sesión que expira luego de un tiempo configurable (por defecto 20 minutos). El *token* se guarda en el almacenamiento local del navegador del usuario, y con ese mismo *token* se puede acceder al resto de los servicios de la API, debido a que para el consumo de los mismos es necesario pasarle un *header* de autenticación con el *token*.

Todos los servicios web para ser consumidos deben brindar el *token* ya que el mismo limita accesos según el rol que tenga el usuario. Al día de hoy existen dos roles en el sistema, el de administrador y el de gestor, y tienen acceso a distintas funcionalidades de acuerdo al rol de los mismos.

En caso de que el usuario ingrese mal su contraseña tres veces, se restringe su acceso por veinticuatro horas.

Para solucionar el problema de que las lecturas no puedan ser obtenidas por un tercero con fines maliciosos (en caso de hurto del dron por ejemplo), se decidió encriptar los datos que se encuentran en la base de datos ubicada en el sistema de Accsys.

#### 6.3.4. Performance

##### **Objetivos**

Se plantea la necesidad de aumentar la eficiencia y el manejo de algunos recursos del sistema que puedan resultar en un desempeño lento de la aplicación, haciendo énfasis en flujo más importante del sistema, que es la recorrida para recolectar lecturas de los animales en el campo y el procesamiento posterior de las mismas [\[84\]](#).

##### **Implementación**

Se optimizó y se dedicó mucho tiempo al algoritmo del plan de vuelo del sistema, el cual a partir de una zona, calcula la forma óptima para recorrerla. Debido a su importancia se dedica el [capítulo 9](#) para detallar la solución.

Además se procesan en forma concurrente las lecturas recolectadas por el dron en su recorrida, en la API REST para poder enviar las notificaciones pertinentes a los usuarios del sistema.

#### 6.3.5. Testeabilidad

##### **Objetivos**

El objetivo principal que se busca en este atributo de calidad es poder probar las distintas partes del sistema en algo similar a un ambiente de producción final [\[85\]](#).

##### **Implementación**

Para lograr el objetivo se trabajó con dos ambientes separados, un ambiente productivo que fue el que se utilizó para realizar las entregas continuas del producto al cliente, y otro de *sandbox* para poder probar las integraciones del sistema entre sí, en ambos se utilizó el PaaS de Heroku, con *continuous delivery*.

El ambiente de *sandbox* en Heroku arma un artefacto y lo despliega, partiendo de la rama *develop* del repositorio del proyecto, mientras que el ambiente productivo hace lo mismo desde la rama *master*, por lo que al finalizar cada *release* se hizo un *merge* con los cambios de *develop* a *master* para poder realizar las entregas al cliente.

Además se abstrajeron los datos de prueba debido a que la base de datos en el Heroku de *sandbox* era distinta a la del ambiente productivo, pudiendo contar así con datos de prueba distintos a los datos productivos que usaba el cliente.

### 6.3.6. Interoperabilidad

#### **Objetivos**

Los servicios web expuestos por los *backend* deben ser claros y seguir un estándar para que los equipos que los consumen puedan realizarlo con facilidad, pensando que posiblemente sean distintos los equipos que colaboren y consuman estos servicios entre las distintas aplicaciones del sistema [\[86\]](#).

#### **Implementación**

Se decidió implementar todos los servicios web utilizando una arquitectura REST, que establece ciertos estándares para realizar y comunicarse a través del protocolo HTTP, haciendo más sencillo e intuitivo el uso de los mismos, ya sea por personas internas a la aplicación o futuros desarrolladores de la misma.

### 6.3.7. Portabilidad

#### **Objetivos**

Es necesario que el sistema pueda usarse tanto desde dispositivos Android como iOS, y en la mayor cantidad de variedad de estos dispositivos, además la aplicación web debe verse bien independientemente de las dimensiones del monitor en la que se visualice.

## **Implementación**

Para ello se eligieron tecnologías para que facilitaran la codificación de un sistema que funcionará en iOS y Android, así como el armado de los artefactos que luego podrán ser descargados de las tiendas de los teléfonos móviles, en ésta etapa no se publicó ninguna versión a una tienda.

También se utilizó un diseño responsive en el armado de la aplicación web con el fin de reordenar los elementos de la web según varía el tamaño de la pantalla, pudiendo así utilizarse en pantallas de distintas dimensiones con el fin de adaptarnos a las computadoras de escritorio que pudieran haber en el campo (en caso de contar con las mismas previo a la instalación del sistema).

### **6.4. Tecnologías**

Las decisiones tenidas en cuenta a la hora de elegir las tecnologías a utilizar fueron las siguientes:

- Curva de aprendizaje por parte de todos los integrantes del equipo.
- Requerimientos funcionales y no funcionales.
- Conocimientos previos de los integrantes del equipo en las tecnologías elegidas.
- Recomendaciones del cliente.

Debajo se encuentra un diagrama de las tecnologías elegidas y cómo interactúan entre ellas. El análisis de cada tecnología se detalla en las siguientes secciones del capítulo.

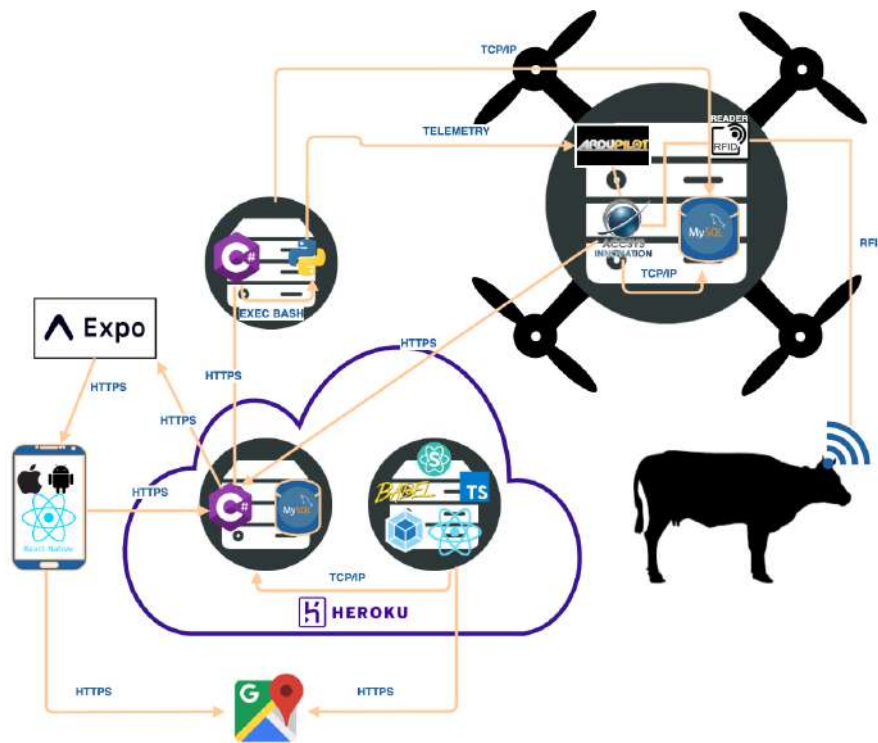


Ilustración 42: Diagrama de tecnologías

#### 6.4.1. Web API REST

- *C#* sobre el *framework* .NET Core 3.0. Esta decisión, si bien no fue un requerimiento impuesto por el cliente, está fuertemente influenciada por el hecho de que el equipo de desarrollo de Todosoft está compuesto principalmente por ingenieros con conocimientos en *C#* y el *framework* .NET Standard. A su vez el equipo contaba con sólidos conocimientos en esta tecnología, por lo que no representó un inconveniente.

Por otro lado, dado que Todosoft pretende entregar el producto como SaaS, se decidió utilizar .NET Core en vez de .NET Standard ya que el primero es multiplataforma y de esa forma es posible aprovechar la enorme cantidad de herramientas y plataformas de DevOps que sirven solo para Unix, además de que permite al cliente ahorrar en costos derivados del cómputo en la nube.

## 6.4.2. Aplicaciones de *frontend*

Se decidió utilizar TypeScript en lugar de JavaScript en todas las aplicaciones de *frontend*, para que sirva como auto documentación para el cliente y la curva de aprendizaje disminuya en futuros integrantes nuevos al equipo de desarrollo del cliente.

### 6.4.2.1 Aplicaciones móviles

- React Native, permite reutilizar código escrito en la aplicación web, reduciendo el tiempo invertido en el desarrollo, además que genera código nativo, tanto en Android como en iOS. Esto beneficia a la performance, e incluso es posible codificar una particularidad en el lenguaje nativo para un dispositivo o sistema operativo específico y es por esto que se elige esta tecnología.
- Expo, facilitó el armado de artefactos para desplegar en los teléfonos móviles, poder probar las funcionalidades y brindar entregables al cliente luego de finalizar las iteraciones, ya que es una tecnología que compila React Native en lenguaje nativo Android e iOS, además de armar un APK en caso de querer una aplicación Android o un IPA en caso de iOS.
- Expo server se seleccionó como solución al desarrollo de las *push notifications*, ya que provee una librería para generar un *token* de Expo desde el celular (previo permiso del usuario para enviarle notificaciones), y con ese *token* consumir una API REST de Expo para publicar notificaciones en el celular que generó el *token*.

### 6.4.2.2 Aplicación web

- Se utilizó React para el desarrollo de la aplicación web, debido a que es altamente performante, dando al usuario una sensación de fluidez al utilizarla y que no presenta complicaciones en el equipo ya que cuatro de los integrantes ya han utilizado esta tecnología.

Además brinda la posibilidad de reutilizar código entre las aplicaciones móviles y la web.

Se descartó el uso de otras tecnologías como Angular, por temas de *performance* y reuso de código entre las distintas aplicaciones del *frontend*, y se descartó Vue debido

a que contaba con una comunidad menor y con menos ejemplos que React, además de que varios integrantes del equipo ya contaban con conocimientos previos de ésta tecnología y eso reducía los tiempos de desarrollo.

- Babel, es un compilador (transpilador) de JavaScript - TypeScript que permite codificar en TypeScript y llevarlo a una versión antigua de JavaScript que soporten *browsers* menos modernos o desactualizados, para poder reproducir la página web en una mayor cantidad de dispositivos, cubriendo el primer requerimiento no funcional del sistema (ver [capítulo 4.2.3](#)).
- Webpack, permite armar un *bundle* del código que se ejecuta en cada cliente web, y seleccionar cómo se va a armar, minimizando los tamaños de los archivos a desplegar, dejando el código JavaScript resultante minificado.

Además permite activar o desactivar sencillamente *logs* de *debug* o producción en la consola de los *browsers*.

También reduce el tamaño de las imágenes en el *browser*, y por este y los anteriores beneficios se eligió utilizarlo.

- Semantic UI, es una librería de componentes para React, con iconos y estilos que permitió desarrollar componentes visuales agradables y responsive sin la necesidad de invertir mucho tiempo en el maquetado de la aplicación web, se descartó el uso de Material UI, una librería de similares características de los creadores de Material Design, debido a que Semantic UI contaba con mejor documentación para utilizar la librería, y contaba con todos los componentes que se necesitaban para la solución.

#### 6.4.3. Aplicación del dron (SkyGuard-Copter)

- Python y el *framework* DroneKit [\[87\]](#). Este es el artefacto de *software* que se comunica con la controladora ArduPilot que lleva el dron. Si bien la SDK oficial del componente está escrita en C++, existe un proyecto de código abierto llamado DroneKit que provee una API escrita en Python que permite utilizar este lenguaje para comunicarse con la controladora. Con el fin de maximizar la mantenibilidad y minimizar la complejidad del sistema se decidió utilizarla.

#### 6.4.4. Aplicación de instrucciones al dron (SkyGuard-Pathfinder)

- C# y Bash. Este es el artefacto de *software* que hace de intermediario entre la API web y la aplicación del dron. También es el que a partir de la información que recibe de la API web determina el plan de vuelo del dron y se lo envía a la aplicación del dron para que esta lo ejecute.

C# fue elegido por la misma razón que la API web, facilitar la adopción y el entendimiento del equipo de Todosoft.

Bash fue elegido porque es necesario ejecutar la aplicación del dron utilizando el intérprete de Python (por cómo está construida) y además hacerlo de forma asíncrona, para que el cliente no tenga que esperar hasta que la misión termine (una misión puede demorar horas). Ejecutar un proceso en el entorno del sistema operativo ofrece ambas cosas (acceso al intérprete de Python y capacidades de ejecución asíncronas) por lo que en este caso Bash era el camino a seguir.

#### 6.4.5. Base de datos

- Se optó por usar MySQL debido a que es una tecnología que ya usa el cliente, y además el equipo tenía experiencia previa usándola, también se tuvo en cuenta que son bases de datos de código abierto con una performance adecuada para los datos que maneja el sistema.

## 7. Desarrollo de *software*

El 38,8 % de las horas dedicadas al proyecto por el equipo fueron destinadas a actividades de desarrollo. Este capítulo describe la organización adoptada por el equipo para distribuir las tareas de desarrollo.

### 7.1. Herramientas de desarrollo

#### 7.1.1. *Frontend*

El *frontend* fue desarrollado utilizando React en el caso de las aplicaciones web y React Native para las aplicaciones móviles. Junto a estas librerías se decidió utilizar el lenguaje TypeScript ya que aporta muchos beneficios a la hora de claridad y mantenibilidad de código, entre otros (los cuales son mencionados en el [capítulo 6](#)).

Se utilizó el IDE más recomendado por la comunidad para el desarrollo JavaScript que es Visual Studio Code. El mismo es un editor de código muy eficiente con soporte para operaciones típicas de desarrollo como *debugging*, ejecución de tareas, y versionado de código.

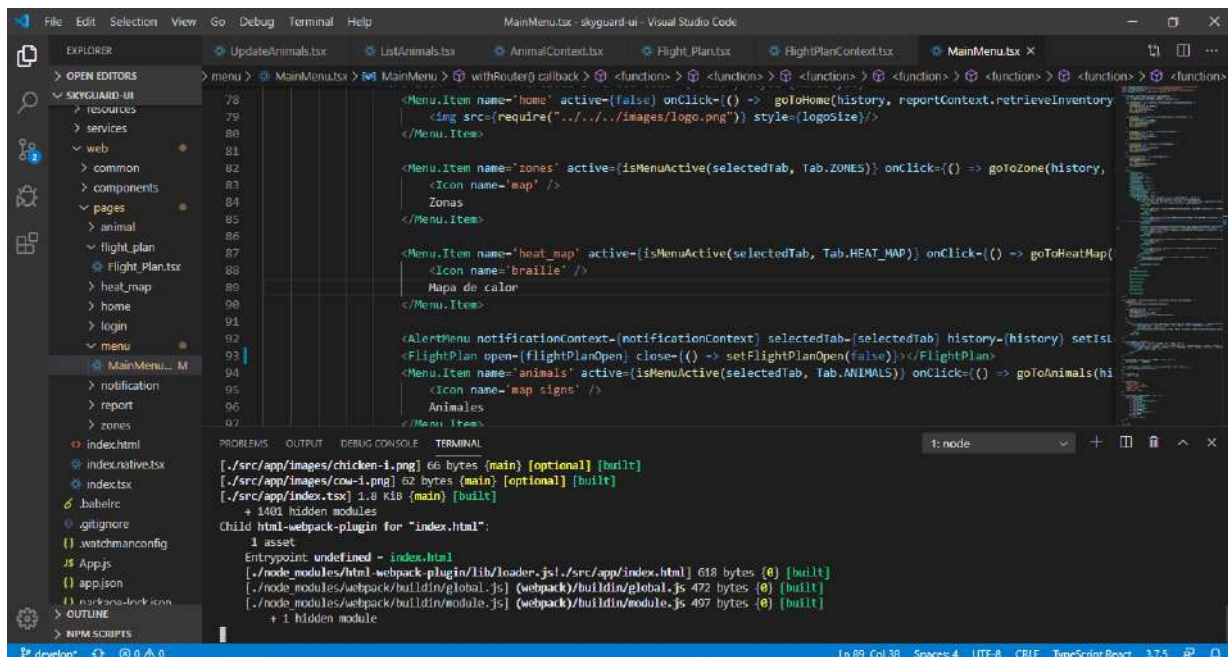


Ilustración 43: Visual Studio Code, React

### 7.1.2. Backend

El *backend* fue desarrollado utilizando .NET Core 3.0 en parte por los beneficios que otorga al ser un *framework* multiplataforma y por otro lado por un acuerdo con el cliente. Se utilizó el IDE Visual Studio Community ya que es la versión gratuita que ofrece Microsoft.

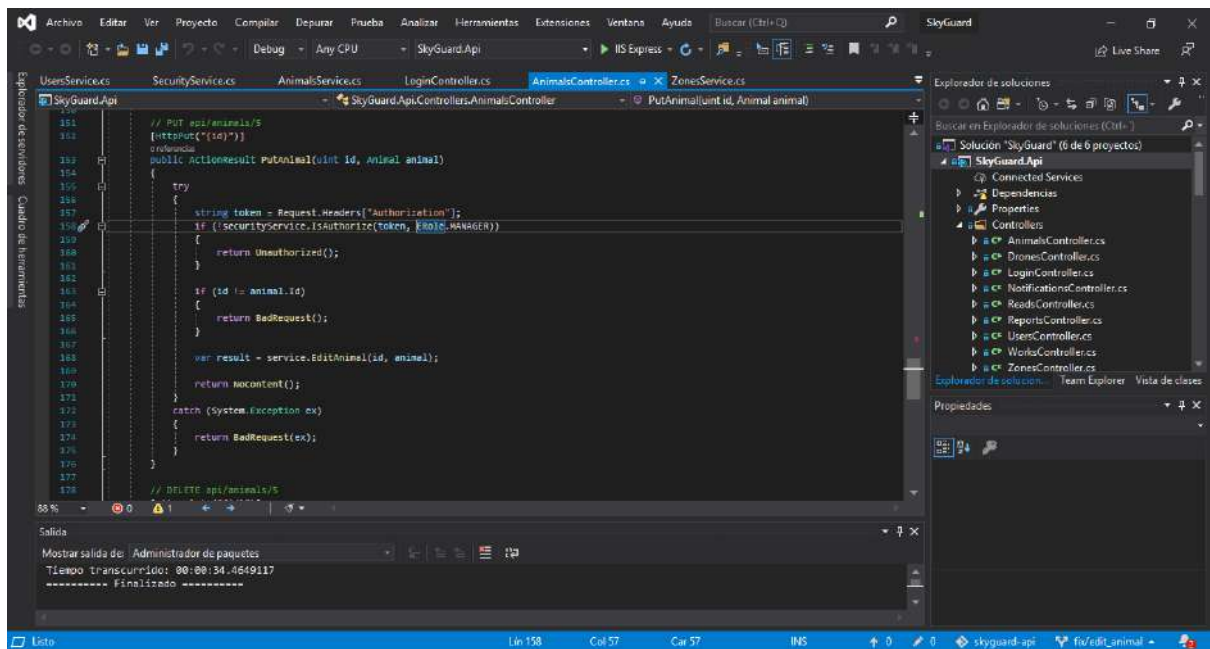


Ilustración 44: Visual Studio Community

### 7.1.3. Integración con el dron

Para el desarrollo de *scripts* que interactúan con el dron, se utilizó Python ya que brinda muchas librerías que facilitan la comunicación con el dron.

En este caso se utilizó el IDE Visual Studio Code nuevamente ya que cubría las necesidades del equipo a la hora de desarrollar.

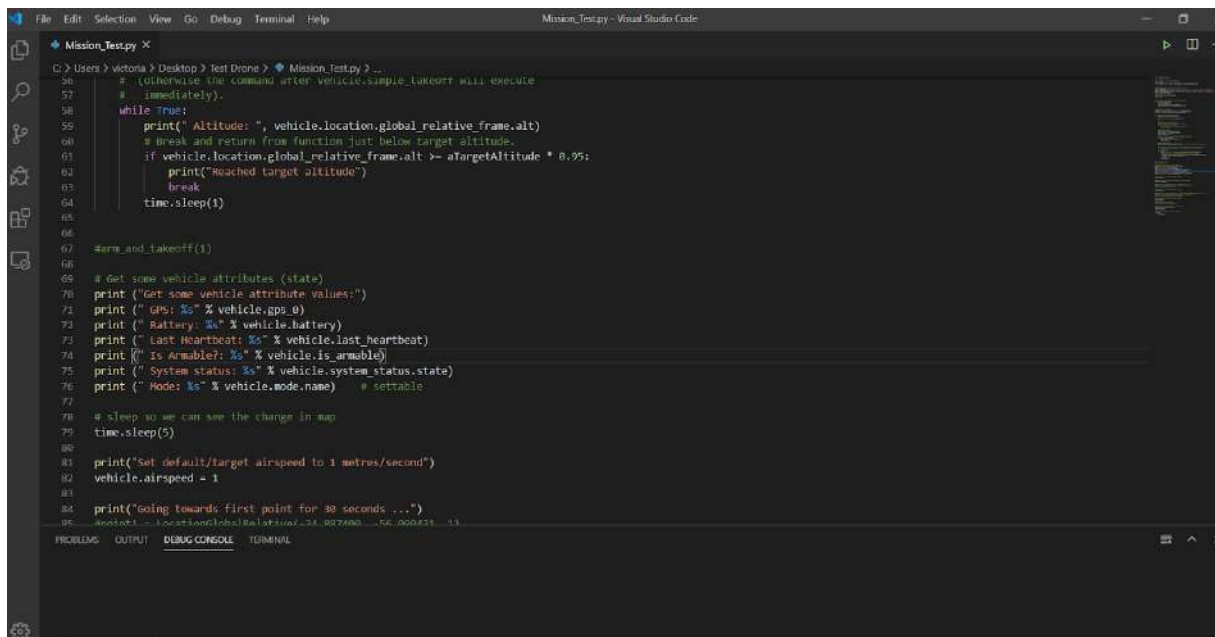


Ilustración 45: Visual Studio Code, Python

## 7.2. Flujo de trabajo

La herramienta utilizada para el versionado de código fue Git, y los hosts de los repositorios fue GitHub, tal como se explica en el [capítulo 11.8](#).

A continuación se detallan los conceptos básicos que conforman el flujo de trabajo.

### 7.2.1. Modelo de ramas: GitFlow

El equipo utilizó el flujo de trabajo GitFlow [\[88\]](#) para la organización del desarrollo en ramas con significado semántico asociado. De esta forma se establecen dos ramas principales:

- *master*: Refleja el estado del código aprobado para estar en producción.
- *develop*: Refleja el estado de código en desarrollo, por lo tanto tiene las nuevas *features* desarrolladas y estables.

Por otra parte, se establecieron las siguientes ramas de soporte:

- *feature*: Rama dedicada para el desarrollo de una nueva funcionalidad. En el proyecto se utilizó una rama *feature* para cada tarea perteneciente a una historia de usuario. Al finalizar la iteración, la rama *feature* es integrada a la rama *develop*.
- *hotfix*: Rama creada exclusivamente para solucionar problemas detectados en producción. En esta rama se realiza una corrección rápidamente y luego es integrada tanto a *master* como a *develop*.
- *release*: Rama creada para armar versiones productivas con nuevas funcionalidades. La misma parte de *develop* y es integrada con *master* y *develop* respectivamente (podría sufrir algún cambio que no se encuentra en *develop*, por eso es importante integrarla con el mismo también)

### 7.2.2. Pull Requests

Se trabajó utilizando *pull requests*, lo que se define como un proceso de desarrollo que provee una base para la discusión y revisión de una *feature* completada.

Su propósito es notificar a los miembros del equipo que una *feature* ha sido completada y que está abierta para ser discutida y modificada en caso de ser necesario. La idea es mejorar la calidad y consistencia del código.

En un caso de uso típico, un desarrollador que ha terminado con su trabajo en una rama *feature*, registra desde la web de GitHub un *pull request* con la intención de integrar dicha rama a la rama *develop*. Otro miembro del equipo actúa como revisor, pudiendo aprobar, agregar comentarios, o solicitar cambios en el *pull request*. Una vez aprobado este, la rama *feature* estará pronta para ser integrada en la rama *develop*.

El equipo se definió que para integrar una rama cualquiera a la rama *develop*, esta rama debería ser aprobada por al menos una persona. Usualmente los revisores del código generado para las aplicaciones *frontend* fueron los responsables del código *frontend*, como se detalla en el [capítulo 7.4](#). Lo mismo ocurre con *backend* e integraciones.

### 7.3. Despliegue

Actualmente se cuenta con un único ambiente de desarrollo para cada una de las aplicaciones. A futuro será necesario contar con al menos dos más (uno de *testing* y uno de producción).

El despliegue se realizó en Heroku, una plataforma como servicio que cubre de forma satisfactoria las necesidades para la ejecución del sistema.

Ambiente	Aplicación	URL	Descripción
Dev	Web	<a href="https://skyguard-webapp-dev.herokuapp.com">https://skyguard-webapp-dev.herokuapp.com</a>	Ambiente privado de desarrollo. Aquí se despliega la rama <i>develop</i> automáticamente.
	API REST	<a href="https://skyguard-api-dev.herokuapp.com">https://skyguard-api-dev.herokuapp.com</a>	

Tabla 4: Despliegue de aplicaciones

### 7.4. Distribución de responsabilidades

En esta sección se detalla cómo fue la distribución de responsabilidades entre los integrantes del equipo sobre los diferentes componentes, durante el transcurso de la etapa de desarrollo.

En una primera instancia, el equipo decidió la asignación de los roles teniendo en cuenta los gustos y preferencias de cada uno de los integrantes.

Para poder brindar una mayor calidad de código se decidió focalizar el trabajo de cada uno de los integrantes en una única aplicación en particular y evitar que cada uno de ellos deba intervenir en varias aplicaciones.

Es por ello que en una primera instancia la distribución de trabajo fue como se detalla en la siguiente tabla.

Subsistema	Componente	Desarrollador
<i>Backend</i>	API REST	Emiliano
	Servicios web que intervienen entre el dron y la API REST	Agustín

<i>Frontend</i>	Web	Gastón
	Móvil	Dan
Dron	Scripts de integración con dron	Victoria

Tabla 5: Distribución de trabajo

Luego de algunas iteraciones, el equipo tuvo que hacer una pequeña modificación debido a que detectó que las integraciones con el dron necesitaban de un mayor esfuerzo que el esperado, por lo que se decidió que Agustín pase a ayudar en dicho rol abandonando su rol anterior de *backend*.

## 8. Usabilidad

La experiencia de usuario es un punto importante para el éxito del sistema, debido a que los usuarios del sistema son personas relacionadas con el campo, y en muchos casos no son muy cercanos a la tecnología, por lo que el uso del sistema tiene que ser intuitivo para que efectivamente le facilite su trabajo y no lo haga más complicado. Por eso se decidió hacer un capítulo explicando el trabajo que se realizó en torno a la usabilidad del sistema a nivel de experiencia de usuario, en el cual no se hace mención del atributo de calidad usabilidad a nivel de arquitectura ya que no se tomaron decisiones en esta área específicas para favorecer a la usabilidad del sistema.

Para la creación de todas las pantallas que tuvieran interacción entre un usuario y el sistema se utilizaron las heurísticas de Nielsen y Material Design, que se detallan a continuación en profundidad.

### 8.1. Heurísticas de Nielsen

Las heurísticas de Nielsen [\[89\]](#) son un conjunto de diez reglas generales para identificar problemas de usabilidad que se desprenden del estudio de varios problemas de usabilidad que realizó Jakob Nielsen.

Estas reglas generales o heurísticas son utilizadas para analizar las pantallas de las aplicaciones de *frontend* con el fin de mejorarlas y encontrar posibles problemas de usabilidad.

### 8.2. Material Design

Material Design [\[90\]](#) cuenta con guías claras para el armado de pantallas tanto web como móviles, y estudios relacionados al comportamiento de los usuarios y sus interacciones no solo con los sistemas sino con su entorno. De eso se utilizó puntualmente las guías para el armado de las pantallas de nuestro sistema.

Se autodenomina como un lenguaje visual que sintetiza los principios clásicos de un buen diseño con la innovación de la ciencia y la tecnología.

Es decir, ayuda con guías en la construcción y el armado de las pantallas de nuestro sistema mezclando los principios tradicionales de lo que es considerado un buen diseño, con herramientas nuevas que les permite saber cómo generar las mismas de una manera que resulte intuitivo para el usuario, reduciendo así el tiempo de aprendizaje de los mismos, sin la necesidad de documentación adicional como manuales, utilizando así las propias pantallas como material de guía para la interacción y el correcto uso del sistema.

Además se plantea como objetivos desarrollar un único sistema subyacente que unifique a la experiencia de usuario entre plataformas, dispositivos y métodos de entrada.

### 8.2.1 Principios

Material Design se plantea varios principios, de los cuales se destacan los siguientes:

#### **Audaz, gráfico e intencional**

Se guía por métodos de diseño de impresión (tipografía, cuadrículas, espacio, escala, color e imágenes) para crear jerarquía, significado y enfoque que sumerjan a los usuarios en una experiencia.

#### **Movimiento provea significado**

El movimiento enfoca la atención y mantiene una continuidad a través de comentarios sutiles y transiciones coherentes. A medida que los elementos aparecen en la pantalla, los mismos se transforman y reorganizan el entorno con interacciones que generan nuevas transformaciones.

### 8.3. Resultados

Se muestra a continuación cómo se beneficiaron las pantallas de las aplicaciones de *frontend* por el uso de Material Design y las heurísticas de Nielsen.

### 8.3.1. Análisis de las heurísticas de Nielsen

Se realizó un pequeño análisis utilizando las heurísticas sobre las aplicaciones móviles y web con el fin de evaluar si dichas aplicaciones están dentro de un rango aceptable en lo que respecta a usabilidad.

Algunos de los problemas detectados y mejorados durante el análisis fueron.

- **Visibilidad del estado del sistema:**

Esta heurística propone brindar retroalimentación al usuario para que sepa en todo caso lo que está pasando.

Para ello se trabajó en la retroalimentación obtenida cuando un usuario ejecuta un plan de vuelo como podemos ver a continuación.

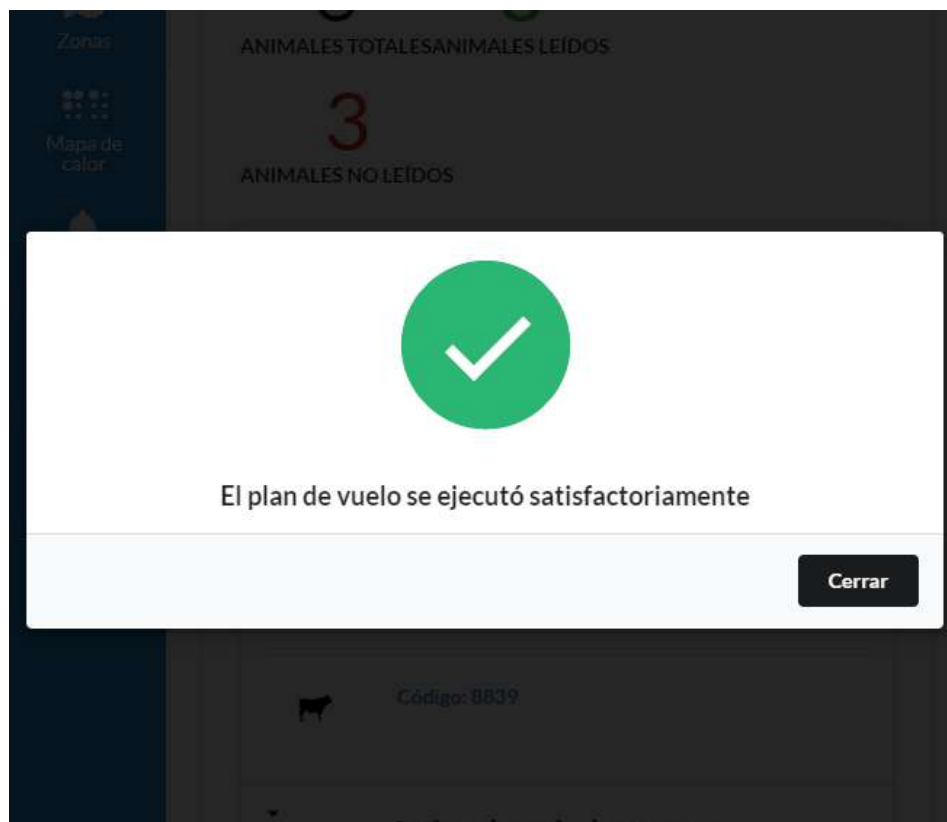
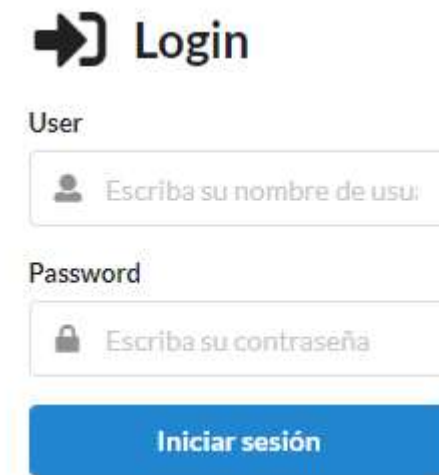


Ilustración 46: Pantalla de ejecución plan de vuelo

- **Utilizar el mismo lenguaje que el usuario**

Para esta heurística se encontraron mejoras debido a que en principio se utilizaron algunos nombre en inglés ya que se dió por asumido que el usuario entendería.

Como podemos ver a continuación.



The image shows a login interface. At the top, there is a header with a right-pointing arrow icon and the text 'Login'. Below this, there are two input fields. The first is labeled 'User' and contains the placeholder text 'Escriba su nombre de usu:'. The second is labeled 'Password' and contains the placeholder text 'Escriba su contraseña'. Below the input fields is a blue button with the text 'Iniciar sesión'.

Ilustración 47: Pantalla login

- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores**

Esta heurísticas propone que el usuario sea capaz de solucionar los errores emitidos gracias a los mensajes que muestra el sistema. Basándose en esto fue que se definieron los mensajes como se muestra a continuación.



The image shows two error messages. The first is for the birth month field, labeled '\* Mes de nacimiento', which contains the value '222'. Below the input field is a red error message box that says 'El mes debe ir del 1 al 12'. The second is for the birth year field, labeled '\* Año de nacimiento', which contains the value '11'. Below the input field is a red error message box that says 'El año deben ser 4 dígitos y empezar con 1 ó 2'.

Ilustración 48: Pantalla mensajes de error

- **Diálogos estéticos y diseño minimalista**

Esta heurística propone que el usuario no debe perder de vista el contenido importante pero teniendo al alcance otro tipo de información. Por este motivo se decidió realizar un menú lateral.

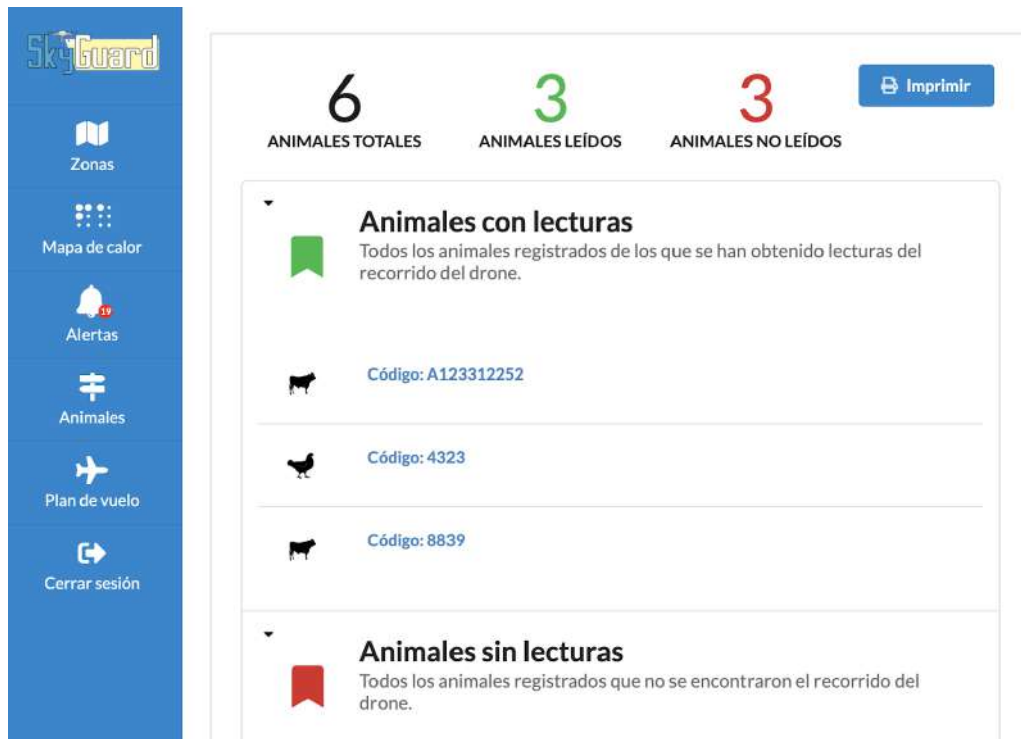


Ilustración 49: Diseño minimalista

### 8.3.2. Análisis de Material Design

Dentro de las guías de material design se encuentran varias secciones a utilizar, se indican debajo algunas de las que se utilizaron para la construcción de las pantallas.

#### **Color > Aplicar color a la UI > Iconografía y tipografía**

Dentro del análisis de colores, y cómo aplicarlos a la UI tenemos la tipografía y la iconografía que dice qué íconos del sistema deben ayudarnos a identificar acciones y proveer información, que el color debe contrastar sobre el fondo para asegurarse de que son legibles e identificables los íconos.

Por lo que se eligió utilizar formas que permitan identificar las acciones de los menús, de forma tal de que sea intuitivo para el usuario la acción que realiza cada ítem del menú y a su vez se utilizó un color que destaque sobre el fondo de los mismos para que sean más legibles e identificables, como se puede ver a continuación.

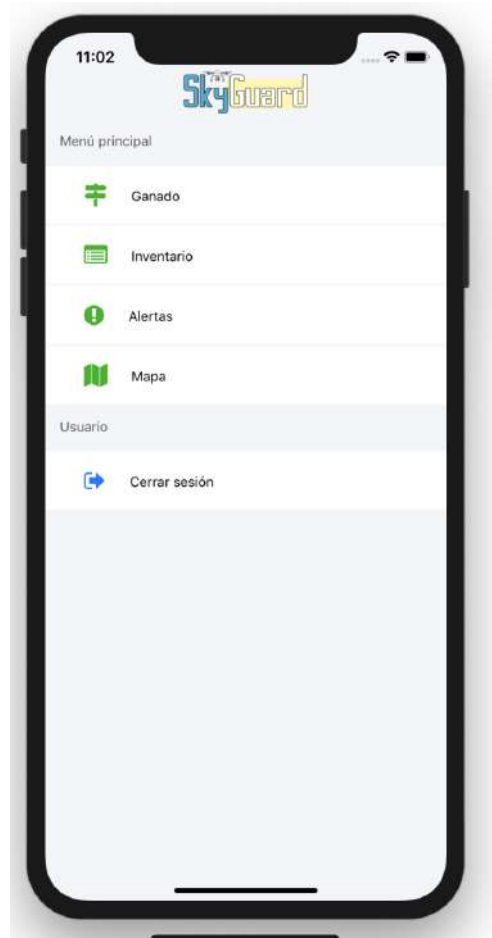


Ilustración 50: Iconografía y tipografía

### **Navegación > Entendiendo la navegación > Navegación reversa**

Material design habla de que la navegación reversa se refiere al movimiento hacia atrás entre pantallas, que puede mover a los usuarios cronológicamente a través de su historial de pantalla reciente o hacia arriba a través de la jerarquía de una aplicación.

Por lo que siempre que se envía al usuario a una nueva pantalla, le ofrecemos un botón intuitivo en la esquina superior izquierda (en caso de las aplicaciones móviles) para que pueda regresar al punto anterior donde se encontraba, como es posible observar en la ilustración:

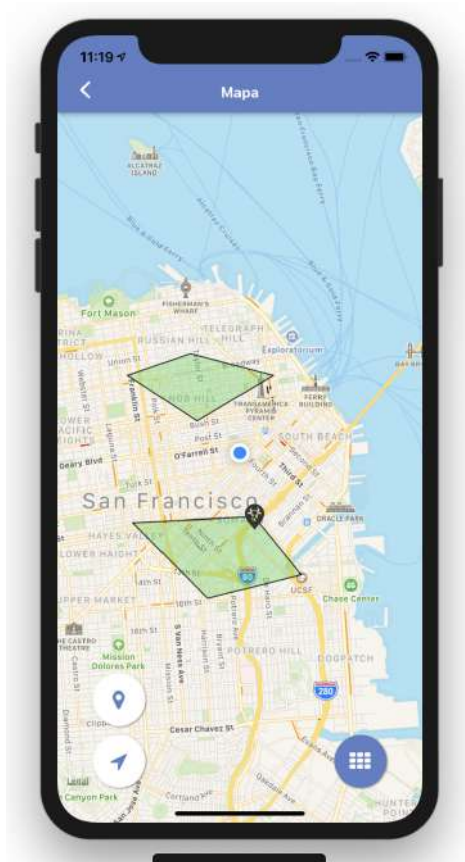


Ilustración 51: Navegación reversa

### ***Layout* > Grilla de *layout* responsive > Customización de grilla**

La guía dice que la grilla de *layout* se puede ajustar para satisfacer las necesidades de su producto y varios tamaños de dispositivo.

Cómo se puede observar, si cambia el tamaño de la ventana, la grilla de alertas se reajustan y redistribuyen para acomodarse en la pantalla de forma natural.

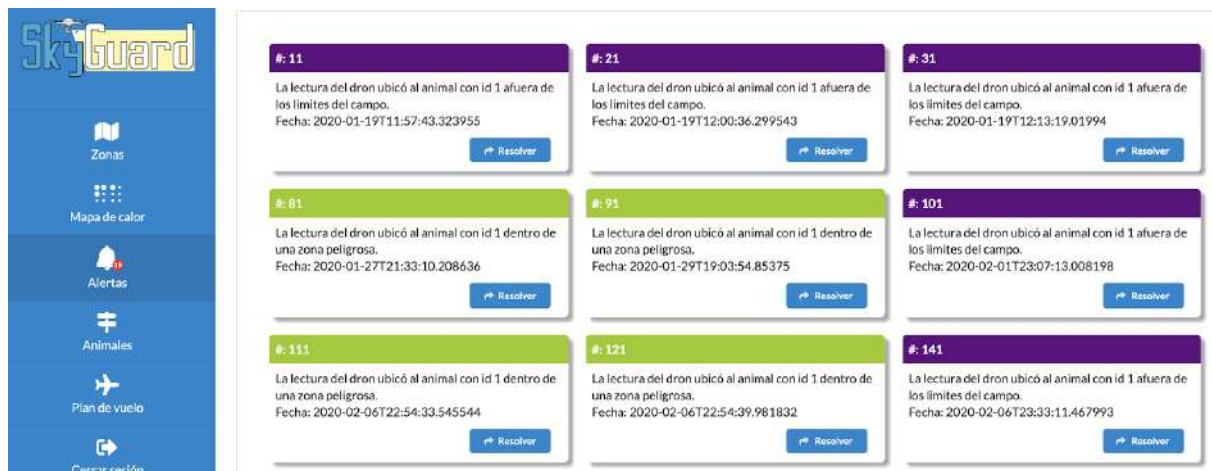


Ilustración 52: Customización de grilla - pantalla grande

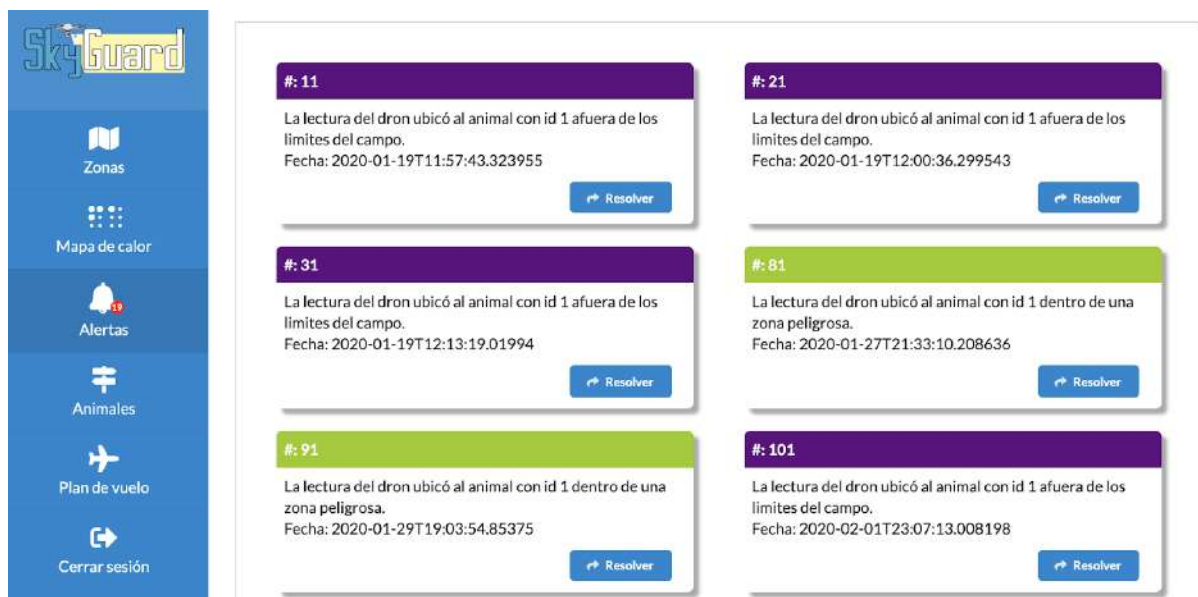


Ilustración 53: Customización de grilla - pantalla pequeña

## Layout > Métodos de espaciado > Contenedores y ratios

Se define contenedor como una forma utilizada para representar un área cerrada. Los contenedores pueden incluir elementos de la interfaz de usuario, como imágenes, iconos o superficies.

Ésta guía es utilizada a lo largo de prácticamente todas las pantallas, ya que en todas se pueden ver contenedores que agrupan elementos como se puede observar en la siguiente imagen.

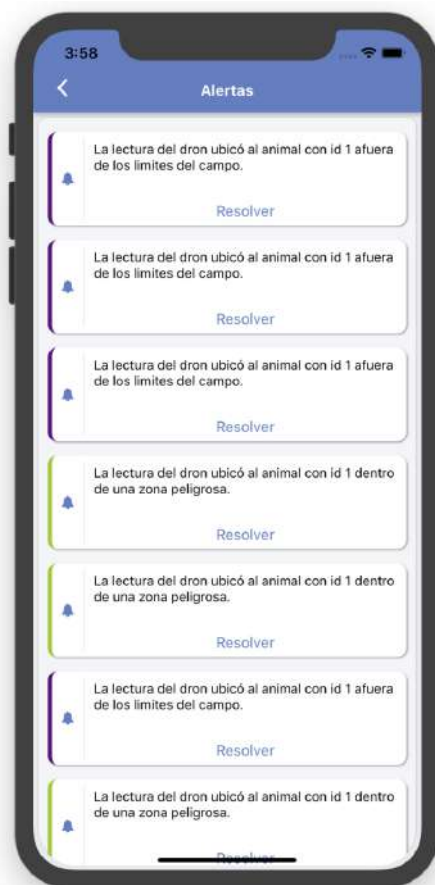


Ilustración 54: Contenedores y ratios

### **Comunicación > Confirmación y reconocimiento > Confirmación**

Las guías de confirmación y reconocimiento dicen que las comunicaciones deben solicitar confirmación antes de tomar una acción y reconocer las acciones exitosas.

Se puede observar que antes de tomar una acción el sistema solicita confirmación de los usuarios para proseguir.



Ilustración 55: Confirmación eliminar animal

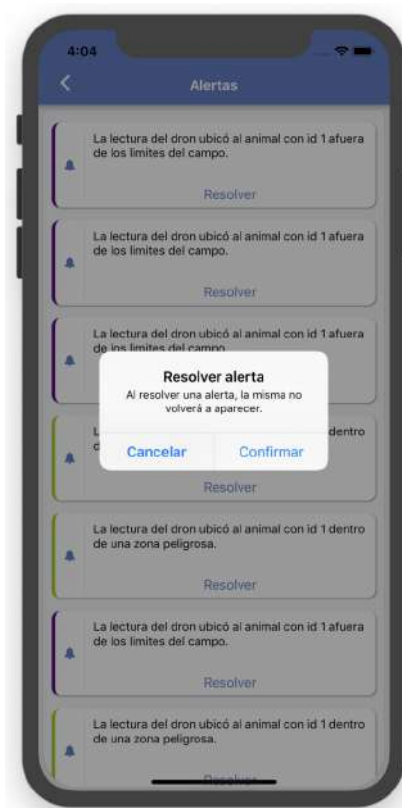


Ilustración 56: Confirmación resolver alerta

## 9. Plan de vuelo del dron

El plan de vuelo, como se mencionó anteriormente, es el listado de instrucciones que recibe y ejecuta el dron para llevar a cabo una recorrida completa del campo.

Las recorridas realizadas por el dron sobre el campo se hacen de forma automática en determinados días y horas pero el proceso de carga de la batería del dron se hace de forma manual por un peón de campo.

Debido al hecho de que los drones tienen un tiempo de duración de la batería limitada, es importante realizar la recorrida del campo en el menor tiempo y de la forma más eficiente posible con el fin de maximizar el área recorrida y así la entrega de valor del producto.

A continuación se describe el algoritmo utilizado, que contempla los puntos geográficos por los que el dron debe pasar, así como la altitud y velocidad a la que el mismo debe volar para realizar una recorrida completa del campo.

### 9.1. Algoritmo

#### 9.1.1. Problema

Algunas de las funcionalidades principales del sistema consisten en un dron recorriendo de forma automática una porción de un campo. Dado el hecho de que los drones tienen batería limitada, es de vital importancia que las mismas se realicen en el menor tiempo y con el menor uso de energía posible, con el fin de maximizar la duración de la batería permitiendo un mayor recorrido y por ende la entrega de valor de la solución.

#### 9.1.2. Métricas de éxito

Un promedio de más de 100 hectáreas recorridas sin interrupción (Dron yendo a base de carga, por ejemplo) por vuelo.

#### 9.1.3. Entradas

- Polígono correspondiente a la sección del campo a recorrer (*\$field*).

- Posición inicial del dron ( $\$droneStart$ ).
- Radio de escaneo del dron, en metros ( $\$droneRadius$ ).
- Resolución del algoritmo, en metros ( $\$flightPlanRes$ ). La resolución se define como la distancia máxima entre las posiciones del plan de vuelo. A mayor resolución (menor distancia) el camino trazado por el dron será más preciso pero también crecerá el costo computacional de calcularlo. A continuación una gráfica que muestra la relación entre la resolución y el tiempo que le toma al algoritmo terminar para un campo de tamaño promedio.

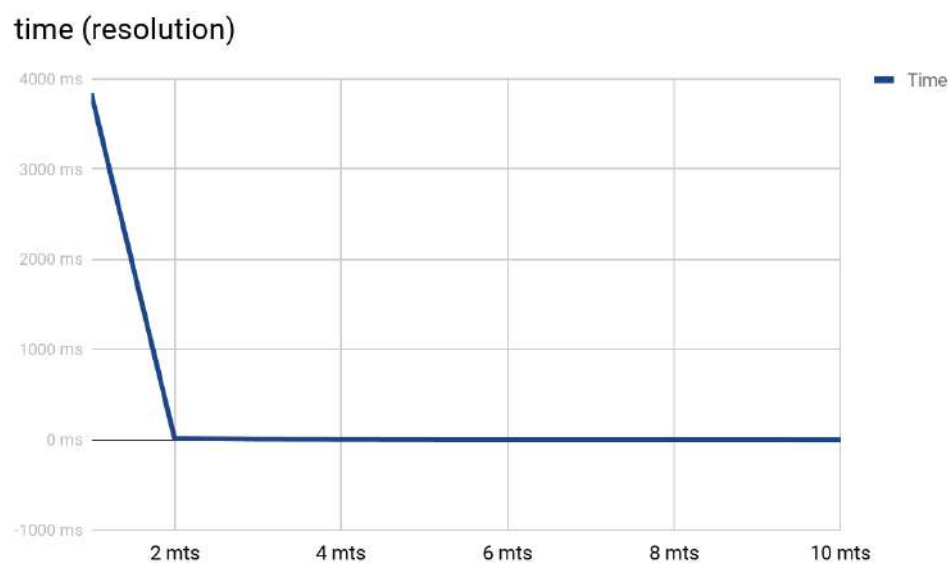


Ilustración 57: Relación entre la resolución del algoritmo (eje X) y el tiempo necesario para completar el cálculo (eje Y).

#### 9.1.4. Salida

- Lista de coordenadas geográficas a recorrer ( $\$flightPlan$ ).

#### 9.1.5. Consideraciones

1. El algoritmo está inspirado en el artículo “*Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones*” publicado por Chien-Ming Tseng, Chi-Kin Chau, Khaled Elbassioni, y Majid Khonji [\[91\]](#), principalmente en el apartado

en el que se describe el proceso de planeamiento de una misión de vuelo en el caso de un dron que vuela a una velocidad uniforme con condiciones de viento estables. Allí se describen las relaciones entre el largo del camino, el tiempo que toma recorrerlo y el consumo de energía con el fin de encontrar la relación que hace más eficiente al plan. La conclusión más relevante para nuestro proyecto es que es de vital importancia minimizar la distancia recorrida por el dron.

2. El diseño del camino a recorrer por el dron está inspirado en la técnica “*Grid Search Pattern*” [92] -frecuentemente utilizada en el campo de la investigación de escenas de crimen- por proveer una buena relación entre el área cubierta y la distancia recorrida, como se menciona en el artículo “*Optimal grid pattern model for search and rescue operation in dipterocarp forest research methodology*” publicado por Alsagoff [93]. Para minimizar aún más los metros que el dron debe moverse se separan las líneas paralelas de la grilla a  $\$droneRadius$  de distancia, de forma de aprovechar al máximo el rango de escaneo de la aeronave. La siguiente figura representa el camino computado (línea amarilla) para un polígono que representa una zona de un campo típico:



Ilustración 58: Representación gráfica del resultado del algoritmo

3. Los cálculos de geometría espacial necesarios para la computación del plan de vuelo son delegados a la librería de código abierto NetTopologySuite [94]. La misma es rápida, confiable y está bien mantenida ya que es utilizada en el núcleo de Microsoft EntityFramework [95].

#### 9.1.6. Proceso

- Calcular el rectángulo que delimita a  $\$field$  ( $\$envelope$ ).
- Calcular el punto más al suroeste de  $\$envelope$  ( $\$start$ ).
- Establecer la posición actual ( $\$current$ ) =  $\$start$ .
- Mientras  $\$current$  esté dentro de  $\$envelope$ 
  - Agregar  $\$current$  a  $\$flightPlan$ .
  - Calcular el punto a  $\$flightPlanRes$  de distancia hacia el norte a partir de  $\$current$  ( $\$nextToNorth$ ).
  - Si  $\$nextToNorth$  está dentro  $\$envelope$ 
    - establecer  $\$current = \$nextToNorth$ .
  - Sino
    - Calcular el punto a  $\$droneRadius$  de distancia hacia el este a partir de  $\$current$  ( $\$nextToEast$ ).
    - establecer  $\$current = \$nextToEast$ .
- Eliminar todos los puntos que no están dentro del polígono correspondiente a la sección del campo a recorrer.

#### 9.1.7. Implementación

El algoritmo está implementado como un módulo del componente SkyGuard-Pathfinder (ver [capítulo 6.4.4](#)) y se ejecuta a demanda cuando un cliente invoca el *endpoint* correspondiente. La salida es luego enviada al componente de integración con el dron (SkyGuard-Copter), quien a su vez lo envía a la placa ArduPilot para que finalmente la aeronave lo ejecute.

La implementación para el alcance de este proyecto está parametrizado de esta forma:

- Polígono: elegido por el administrador a través de la aplicación web.
- Posición inicial del dron: esquina más al suroeste del polígono.

- Radio de escaneo del dron: 100 metros.
- Resolución del algoritmo: 5 metros.

A continuación un extracto del código fuente:.

#### PathfinderAlgorithm.cs - Compute()

```
var path = new List<Coordinate>();

var polygon = CreatePolygon(coordinates);
var polygonEnvelope = polygon.EnvelopeInternal;

var start = new Coordinate(polygonEnvelope.MinX, polygonEnvelope.MinY);
var end = new Coordinate(polygonEnvelope.MaxX, polygonEnvelope.MaxY);
var current = start

.Translate(DIRECTION_RIGHT, pathResolution)

.Translate(DIRECTION_UP, pathResolution);

for (;;)
{
    path.Add(current);

    current = NextPointInsideEnvelope(current, polygonEnvelope);

    if (current == null) break;
}

return path.FindAll((coordinate) =>
polygon.Contains(coordinate.Point()));
```

### PathfinderAlgorithm.cs - NextPointInsideEnvelope()

```
var next = current.Translate(direction, pathResolution);

if (!envelope.Contains(next))
{
    next = current.Translate(DIRECTION_RIGHT, pathWidthMts);
    direction = direction == DIRECTION_UP ? DIRECTION_DOWN : DIRECTION_UP;
}

if (!envelope.Contains(next))
{
    return null;
}

return next;
```

#### 9.1.8. Futuros pasos

- Agregar las estaciones de carga como otro de los factores del algoritmo de forma que este pueda prever el recorrido hacia las mismas como parte del plan de vuelo. Este cambio puede basarse en la implementación del algoritmo propuesto en “*Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones*”.
- Agregar condiciones climáticas como otro de los factores del algoritmo de forma que este pueda considerar la dirección e intensidad del viento para poder decidir con qué dirección recorre el camino computado. Algunas consideraciones con respecto a la dirección e intensidad del viento también pueden encontrarse en “*Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones*”.

## 10. Gestión del proyecto

Este capítulo describe la planificación, ejecución y seguimiento del proyecto. En él se detallan la gestión del alcance, la gestión del tiempo, la gestión de los riesgos y la gestión de la comunicación.

### 10.1. Gestión del alcance

La gestión del alcance le permitió al equipo tener un control sobre el trabajo que debía realizar para poder desarrollar un producto funcional, dentro del tiempo académico establecido y cumpliendo con los objetivos planteados.

Además de la planificación inicial se realizó un continuo seguimiento y varios ajustes a lo largo del proyecto que se detallarán más adelante.

#### 10.1.1. Definición del alcance

La definición del alcance fue realizada en su totalidad por el equipo luego de varias investigaciones, diversas actividades de relevamiento de requerimientos, reuniones con el cliente y análisis de plazos. Entre los requerimientos que la empresa proveyó se encuentran la utilización de drones y la necesidad de solucionar las problemáticas que tienen los estancieros en la actualidad.

Fue acordado entre el equipo y el cliente que el desarrollo del sistema sería comenzado por nosotros y que tanto el código como las investigaciones serían entregadas a Todosoft una vez finalizada la tesis. Todosoft continuará con el desarrollo para en un futuro disponer de un sistema completo, funcional, pulido y por ende comercializable a los clientes de la empresa. El detalle de los entregables se encuentra en el [capítulo 4.4](#).

Desde un primer momento la empresa Todosoft manifestó que lo que ellos pretendían del proyecto era construir una versión del sistema lo más completa posible, pero sabiendo que el tiempo requerido para completar un producto totalmente funcional, pulido y vendible era muy superior al otorgado por la Universidad para los proyectos finales de carrera. Por esta razón el cliente considera valioso todo avance que les permita reducir tiempos de desarrollo e

investigación una vez que el equipo les haga entrega del código. Consideran tan valiosa las investigaciones de drones como la implementación de cualquiera de las funcionalidades más importantes del sistema.

Basados en esta premisa definimos el alcance estableciendo la necesidad de realizar las investigaciones necesarias previamente al comienzo del desarrollo de *software*.

El *hardware* final del proyecto (dron de la empresa Accsys y el lector de RFID activo) **no llegó a tiempo para poder ser probado**, por lo que se decidió adquirir un *hardware* provisorio para realizar el desarrollo y las pruebas. Por este motivo se decidió limitar la realización de pruebas en escenarios reales (campos con ganado), realizando las pruebas en lugares urbanos, limitando el alcance en ese sentido.

El plazo con el que se contaba para la realización del proyecto fue una restricción muy importante a la hora de definir el alcance, como también lo fue el objetivo planteado de aprobar la tesis con una calificación de excelencia, por lo que redactar una documentación apropiada al nivel requerido no era negociable. El alcance por ende incluyó la documentación formal y por escrito de cada aspecto que el equipo consideraba importante.

Por lo antedicho el equipo, junto al director de Todosoft, definió un conjunto de funcionalidades necesarias para poder cumplir con los objetivos planteados. Dichas funcionalidades fueron obtenidas luego de realizar la etapa de ingeniería de requerimientos que se detalló en el [capítulo 4](#).

Las funcionalidades que el equipo consideró necesarias para poder cumplir con el objetivo principal se detallan en el siguiente *Product Breakdown Structure*.



Ilustración 59: Product Breakdown Structure

Por lo tanto el alcance total del proyecto está formado por todas las funcionales del sistema previamente mencionadas más la documentación, investigaciones, elementos de calidad, soporte y demás entregables adicionales al producto, los cuales se especificaron en el [capítulo 4.4](#).

### 10.1.2. Plan de *release*

Definimos un plan de entregas para presentar los avances a Alejandro Martínez, director de Todosoft.

Para realizar el plan se priorizaron los requerimientos, se estimó la complejidad, esfuerzo e incertidumbre, se calculó la velocidad del equipo y se planificó los sprints de cada *release*.

Se detallan a continuación las fechas y contenido de los *releases*:

#### 10.1.2.1. Número de release: 1

**Fecha:** 28 de setiembre de 2019

#### **Funcionalidades planificadas:**

- **RF1.** Como dueño de campo quiero poder agregar y eliminar las zonas a escanear por el ejército de drones para poder armar las rutas que recorrerán los drones.
- **RF2.** Como dueño de campo quiero poder definir zonas peligrosas para los animales, para poder controlar que los mismos no se encuentren en dichas zonas.
- **RF4.** Como dueño de campo quiero poder agregar, modificar y eliminar ganado para poder llevar el control de cada animal asociándolo al *tag*.
- **RF8.** Como dueño de campo quiero poder ver en un mapa la posición de cada animal en el momento en que fue leído su chip para saber dónde está ubicado.
- **RF13.** Como dueño de campo quiero poder buscar por un animal y que me muestre la última posición.

**Grado de cumplimiento:** se completaron la mayoría de las funcionalidades, aunque el RF8 fue realizado en las aplicaciones móviles pero no así en la aplicación web; no se llegó a tiempo y se añadió esa funcionalidad en el *release 2*.

#### 10.1.2.2. Número de release: 2

**Fecha:** 07 de enero de 2020

#### **Funcionalidades planificadas:**

- **RF3.** Como dueño del campo quiero poder asignar zonas a uno o más drones para que el dron recorra dicha zona mientras escanea a los animales.
- **RF6.** Como dueño de campo quiero poder definir alertas en caso de que un animal ingrese a una zona peligrosa para evitar que los mismos mueran sin ser detectados.
- **RF7.** Como dueño de campo quiero recibir una notificación si un animal sale de la zona definida para evitar robos.
- **RF11.** Como dueño de campo quiero poder *loguearme* al sistema para tener acceso a las funcionalidades de la aplicación.
- **RF14.** Como usuario administrador de la plataforma quiero poder agregar, modificar y eliminar dueños de campos para poder añadir nuevos clientes a nuestro sistema.
- **RF15.** Como dueño de campo quiero poder definir referencias en el mapa de mi campo para poder utilizarlas luego para ubicar animales.

**Grado de cumplimiento:** no se completaron los requerimientos funcionales 3 (faltó terminar el plan de vuelo), 7 (faltó poder abrir la notificación), 11 (faltaron detalles en el *login web*), 14 y 15 (fueron despriorizados por el equipo luego de una reunión de alineamiento con Todosoft).

### 10.1.2.3. Número de release: 3

**Fecha:** 28 de febrero de 2020

**Funcionalidades planificadas:** Terminar todos los RF pendientes

- **RF5.** Como dueño de campo quiero poder ver la información cacheada de las zonas, animales y lecturas para tener acceso a información sin tener conexión a Internet.
- **RF9.** Como dueño de campo quiero poder ver un reporte en forma de mapa de calor con la trazabilidad del ganado.
- **RF10.** Como dueño de campo quiero poder ver un reporte de cada una de las recorridas que hizo cada dron sobre las zonas definidas para tener un control de los mismos.
- **RF12.** Como dueño de campo quiero poder marcar como leída una alerta para tener a la vista solo las alertas sin leer.

**Grado de cumplimiento:** se cumplió con todo lo planificado.

La distribución de los *releases*, junto a los hitos más importantes del proyecto, pueden verse en la línea de tiempo documentada en el [capítulo 10.4.2](#).

## 10.2. Gestión del esfuerzo

El equipo decidió registrar las horas invertidas en cada una de las tareas ejecutadas a lo largo del proyecto. En un comienzo se registraron solamente las horas trabajadas en tareas de desarrollo, pero luego de analizar las métricas se llegó a la conclusión de que se debía contabilizar el esfuerzo de todas las tareas realizadas.

Se analizó cómo y dónde era mejor hacer el registro: se evaluó la herramienta Toggl, las planillas de Google Sheets y la función de trackeo de ClickUp, la cual finalmente se eligió. En el [anexo 5](#) se detallan las herramientas analizadas por el equipo previo a la elección de ClickUp.

ClickUp permite tener en una misma interfaz las tareas, las estimaciones y el esfuerzo. Al analizar las métricas de esfuerzo se corrigieron varios aspectos:

- En la segunda revisión, junto a la tutora y al revisor Gastón Mousques, se detectó que se estaban creando tareas muy grandes, por lo que las mismas se trabajaban en varios sprints y eso no permitía calcular la velocidad del equipo adecuadamente.
- Luego de algunas semanas se comenzó a trazar todas las tareas y no solamente las relacionadas con el desarrollo de *software*.

A continuación se muestra cómo expone ClickUp el esfuerzo para las diferentes tareas (columna “*TIME TRACKED*”).



TASK	ASSIGNEE	TIME TRACKED
Planning, retro, alineamientos <i>reuniones</i>	AR, BR, RR, P-2	3h
Login en BD <i>backend desarrollo shirt-m</i>	ER	11h
Crear una misión con DroneKit para subir al dron y que se ejecute en el dron usando la función upload = <i>desarrollo drone</i>	YK	3h
Backend - Implementar llamada a Expo <i>backend desarrollo shirt-m</i>	ER	2h
Alta y modificación de alertas en BD <i>backend desarrollo shirt-s</i>	ER	1h
Login en frontend <i>desarrollo frontend shirt-s</i>	DB	8h
Coordinar con Luis García el envío de lecturas por HTTP <i>backend calidad-y-pruebas shirt-s</i>	ER	1h
Conseguir raspberry y configurar para ver si se puede usar con el dron <i>drone investigación</i>	GD, R	19h
Frontend - Investigar y empezar a implementar push notifications <i>desarrollo frontend investigación</i>	DB	10h

Ilustración 60: Registro de horas en la herramienta ClickUp

### 10.2.1. Métricas de esfuerzo

Para gestionar el esfuerzo se definieron ocho métricas cuyo seguimiento se consideró de interés para el equipo:

- Esfuerzo por actividad en el proyecto.
- Total de horas trabajadas.
- Dedicación de horas por iteración por persona.
- Horas de desarrollo por iteración.
- Horas de desarrollo por persona.

- Horas de documentación por persona.
- Cumplimiento de los objetivos por iteración.
- Desviación entre esfuerzo estimado y esfuerzo real en las tareas de ingeniería.

#### 10.2.1.1. Esfuerzo por actividad en el proyecto

Se consideró muy importante medir y controlar la cantidad de horas que el equipo dedicaba a las diversas áreas del proyecto. Se separaron todas las actividades realizadas en diez categorías:

- **Investigación:** todas las tareas que requieran búsqueda de información, contactos con expertos en la materia, búsqueda de mejores soluciones, etcétera.
- **Dron:** actividades referidas al *hardware* del dron tales como ensamblado, pruebas, instalación.
- **Reuniones:** horas dedicadas a las reuniones entre el equipo -ya sean planificadas dentro de la metodología de trabajo o reuniones esporádicas-, reuniones con expertos, reuniones con la tutora y reuniones con el cliente.
- **Gestión:** actividades relacionadas a la gestión del equipo y cumplimiento de los requisitos del proyecto de grado por parte de la Universidad (entregas, ceremonias, etcétera).
- **Desarrollo:** horas dedicadas al desarrollo de *software*.
- **Documentación:** horas dedicadas a la elaboración de la documentación que será entregada al finalizar la tesis.
- **Calidad y pruebas:** actividades de creación y ejecución de pruebas, tanto de *software* como de *hardware* (pruebas con el dron y RFID).
- **Requerimientos:** actividades relacionadas con el relevamiento de los requerimientos del sistema.
- **Definición de proceso:** horas dedicadas a definir la metodología de trabajo y cómo se iba a implementar.
- **Diseño de arquitectura:** actividades de análisis, creación y discusión de la arquitectura del sistema.

El total de horas dedicadas fue 1894.

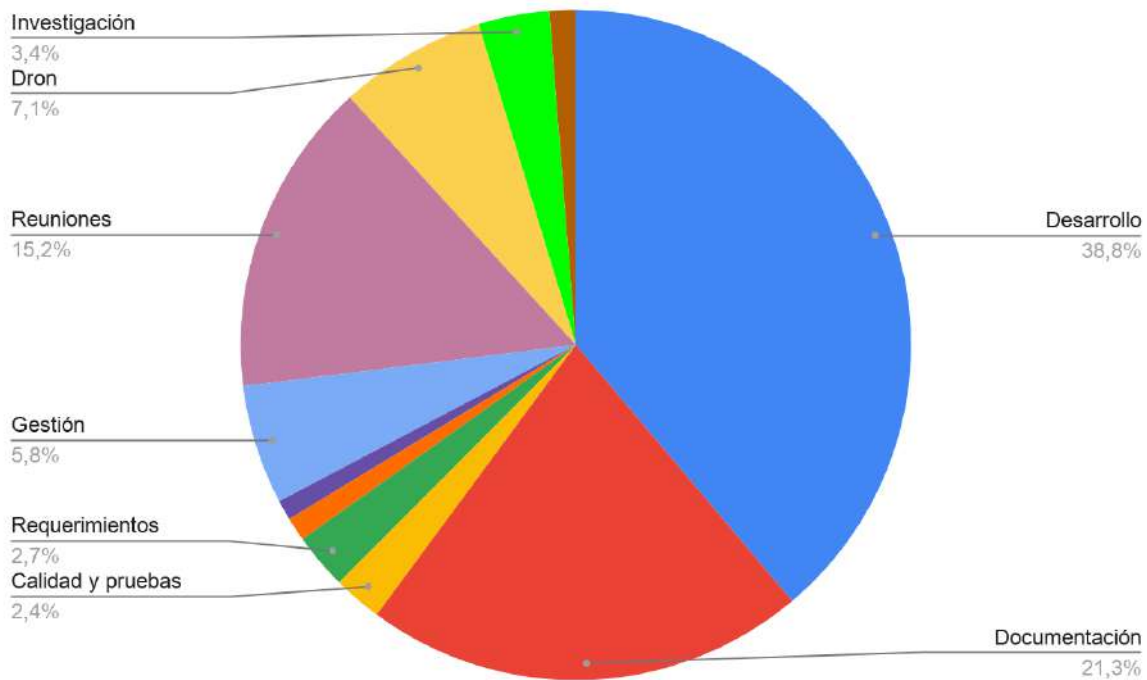


Ilustración 61: Gráfica de dedicación de horas por cada área del proyecto

Analizando el gráfico se puede observar que el área que llevó mayor dedicación fue el desarrollo de *software*, algo que es lógico debido a que se desarrolló un sistema de *backend*, una aplicación para iOS, una aplicación para Android, una web y el *software* del dron, junto a las integraciones entre todas estas aplicaciones.

A su vez, en un proyecto de estas características en el que el mayor desafío era el uso de drones manejados por sistemas propios, el área “Dron” y “Calidad y pruebas” fueron fundamentales y por ende se le dedicó muchas horas.

El área “Documentación” requirió mucha dedicación debido a que uno de los objetivos era entregar un documento de alta calidad, por lo que a lo largo de todo el proyecto se dedicaron horas para cumplirlo. La documentación de calidad fue un requisito académico en primer lugar, pero como la misma será entregada a la empresa Todosoft y será importante para ellos, también es imprescindible dotarla de la mayor calidad posible.

En el área “Investigación” se invirtió una una gran cantidad de horas que no se ven reflejadas en la gráfica. Esto se debe a que a los efectos de graficar las horas dedicadas, en algunas tareas que compartían diferentes áreas (dron e investigación y desarrollo e investigación, por ejemplo) fue necesario elegir una única área para no sumar por duplicado (ver [capítulo 5](#) para conocer las investigaciones realizadas).

Por último, algo que demandó mucho tiempo y quizás no fue lo ideal, fue el área de “Reuniones”. Si bien ayudó a la gestión del equipo, la metodología que se usó tuvo una carga alta de horas de reuniones (véase [capítulo 3](#)). Quizás reduciendo la cantidad de tiempo invertido en esta área se podría haber aprovechado el tiempo para avanzar más rápido en el desarrollo o hacer más pruebas.

A continuación se presenta otra gráfica que muestra la evolución del esfuerzo por área a lo largo de las iteraciones.

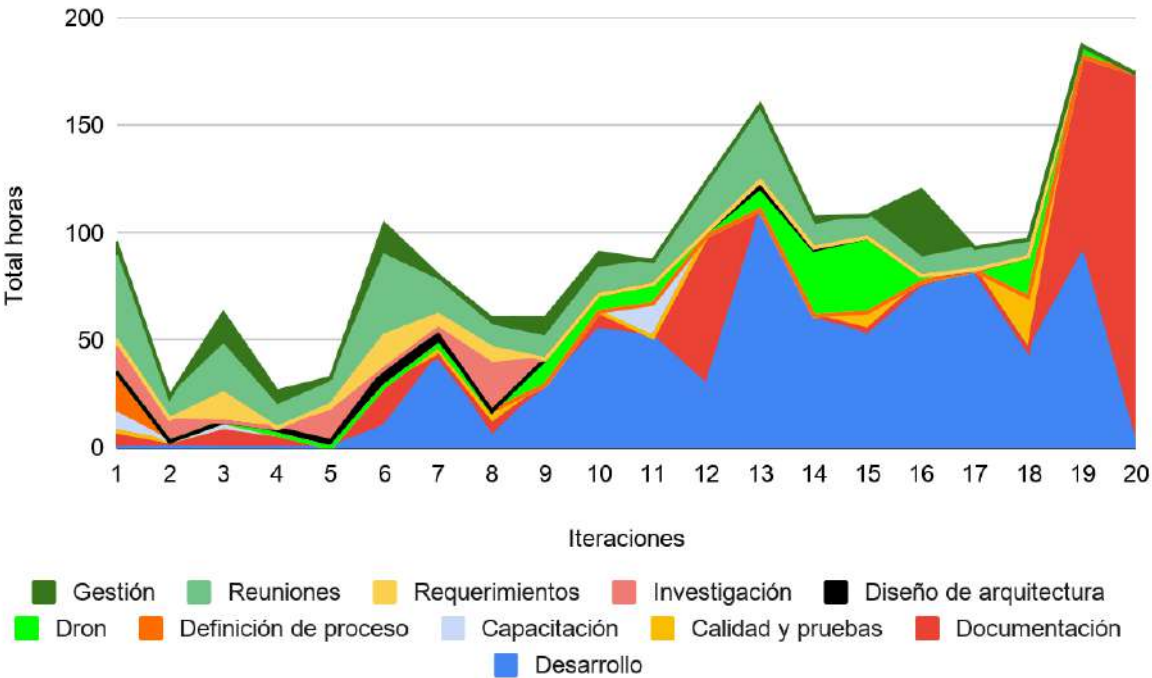


Ilustración 62: Horas dedicadas por área

Al principio del proyecto, la gestión del mismo y las reuniones representaban la mayor parte del esfuerzo invertido, algo que fue decayendo con el paso de las iteraciones (sobre todo la gestión), mientras iba creciendo exponencialmente la dedicación en el área del desarrollo de *software*.

La definición de proceso y el diseño de arquitectura son actividades que se realizaron al comienzo del proyecto, donde se encuentran la mayor cantidad de horas dedicadas a las mismas. Con el correr de la tesis se hicieron ajustes pero no se invirtió demasiado tiempo.

Desde la iteración 13 se empieza a ver cómo las pruebas comienzan a tomar protagonismo. Esto es debido a que por esas fechas se adquirió el dron de pruebas (ver [capítulo 12](#)) por lo que, al mismo tiempo que se iba avanzando con el desarrollo de los diferentes componentes, se iban realizando las pruebas necesarias.

#### 10.2.1.2. Total de horas trabajadas

Se presenta a continuación la distribución de horas por iteración.

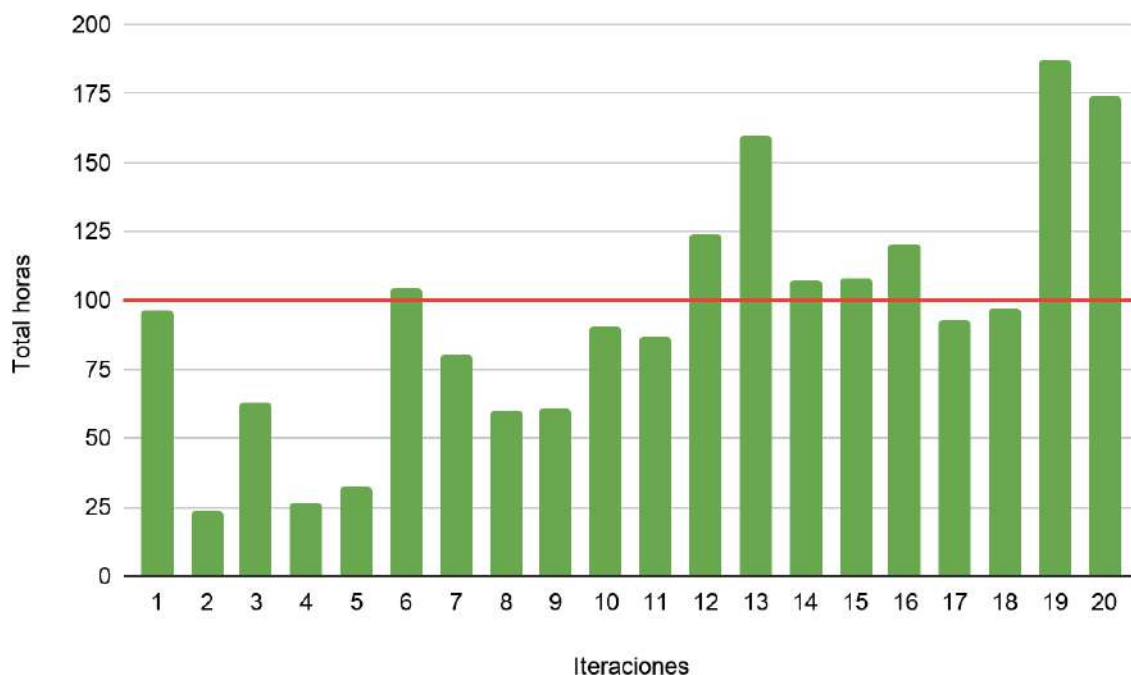


Ilustración 63: Horas dedicadas por iteración

El equipo se había propuesto realizar un estimado de 10 horas por semana (20 horas por iteración) por persona.

Podemos ver que a lo largo de todo el proyecto no se mantuvo un valor constante de horas dedicadas por iteración, sobre todo al principio. En la iteración 11 se analizaron las métricas y se realizó una reunión para sacar conclusiones y accionables debido a que se notó que no se estaba llegando con los objetivos planteados en las diferentes iteraciones. A partir de ese momento la cantidad de horas dedicadas comenzó a subir (tabla de trackeo de horas en [anexo 2](#)).

Sin contar las primeras iteraciones donde todavía no se podía avanzar demasiado rápido ya que se dependía de factores externos (coordinación con el cliente, cuestiones administrativas, coordinación con la tutora) y algunas iteraciones de rendimiento bajo (7, 8 y 10), el ritmo de dedicación del equipo fue satisfactorio. Cuando uno de los integrantes, por diferentes razones, no pudo completar sus horas por iteración, el resto aumentó su dedicación para alcanzar los objetivos planteados.

Es importante tener en cuenta que el desarrollo se comenzó en la iteración 5 pero el grueso del mismo se construyó a partir de la iteración 9.

### 10.2.1.3. Dedicación de horas por persona

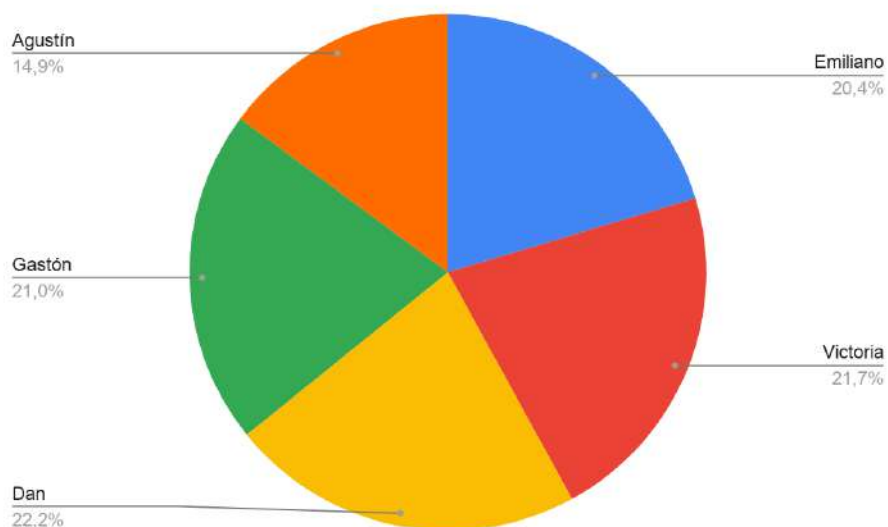


Ilustración 64: Dedicación de horas por integrante durante todo el proyecto

En esta gráfica se muestra cómo se distribuyó, entre los integrantes del equipo, el tiempo dedicado. Podemos observar que todos han dedicado un porcentaje muy similar de horas al proyecto.

La buena ejecución de la metodología permitió que el equipo pudiera trabajar coordinado y de manera constante.

Es importante remarcar que los 5 miembros, durante todo el proyecto, alinearon el tiempo de la tesis junto a sus trabajos (9 horas diarias en todos los casos) y al tiempo académico invertido en otras materias y evaluaciones curriculares.

#### 10.2.1.4. Horas de desarrollo por iteración

A continuación se repasa cómo fue la dedicación de horas al desarrollo de *software* en las diferentes iteraciones.

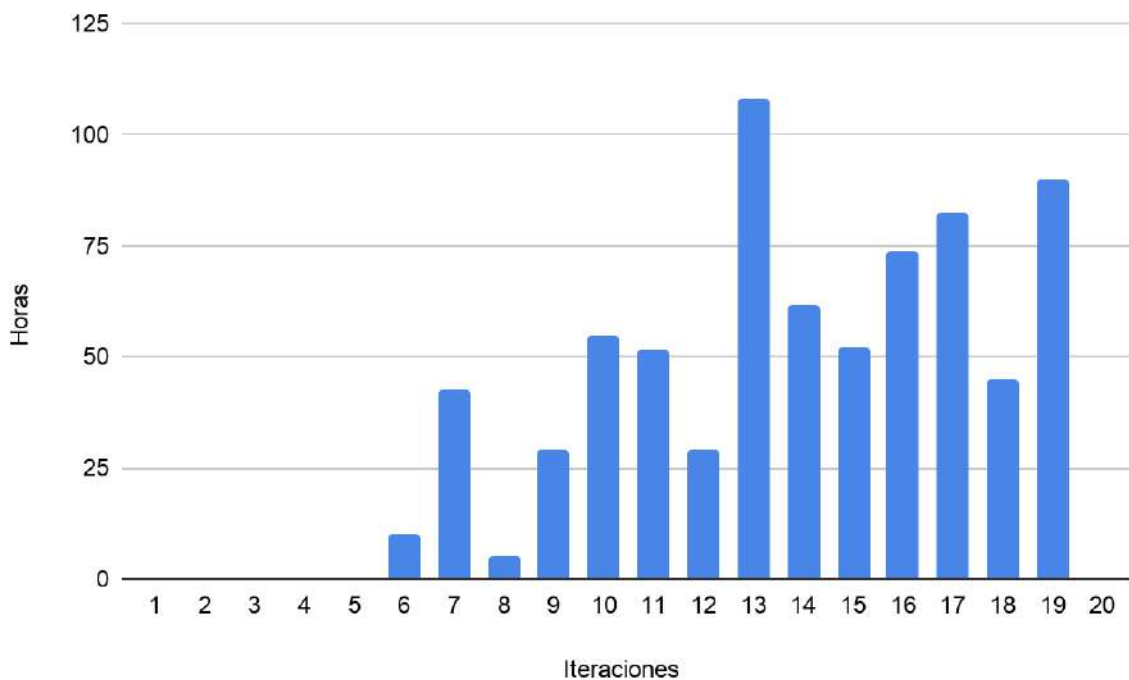


Ilustración 65: Dedicación de horas en desarrollo por iteración

El desarrollo comenzó recién en la iteración 5 donde se empezó de a poco en los diferentes componentes.

Como todo proyecto nuevo, sobre todo cuando se trata de tecnologías que los desarrolladores no dominan a la perfección, existe una curva de adaptación y aprendizaje. Se considera que en la iteración 13 se llegó a un punto de fluidez que permitió avanzar sin demoras. Esto, sumado a los ajustes descritos en el punto anterior, llevó a incrementar la dedicación motivados por el hecho de que las funcionalidades iban saliendo y se podía desarrollar en paralelo.

#### 10.2.1.5. Horas de desarrollo por persona

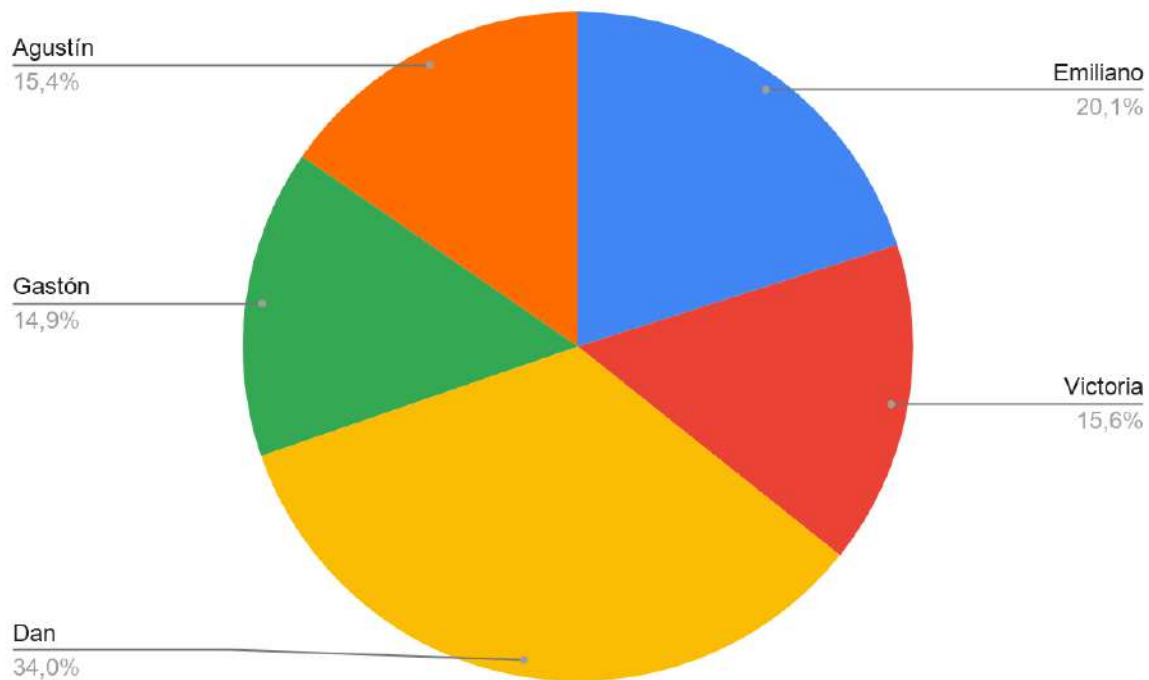


Ilustración 66: Dedicación horaria en desarrollo por persona

Podemos observar que Dan y Emiliano fueron las personas que más tiempo dedicaron a la programación. Esto se dio debido a que estaban encargados principalmente del *frontend* y el *backend* respectivamente. El resto de los integrantes equilibraron su dedicación en desarrollo con pruebas con el dron y otras actividades.

#### 10.2.1.6. Horas de documentación por persona

A continuación se observa cómo fue la dedicación de horas para la documentación, por persona.

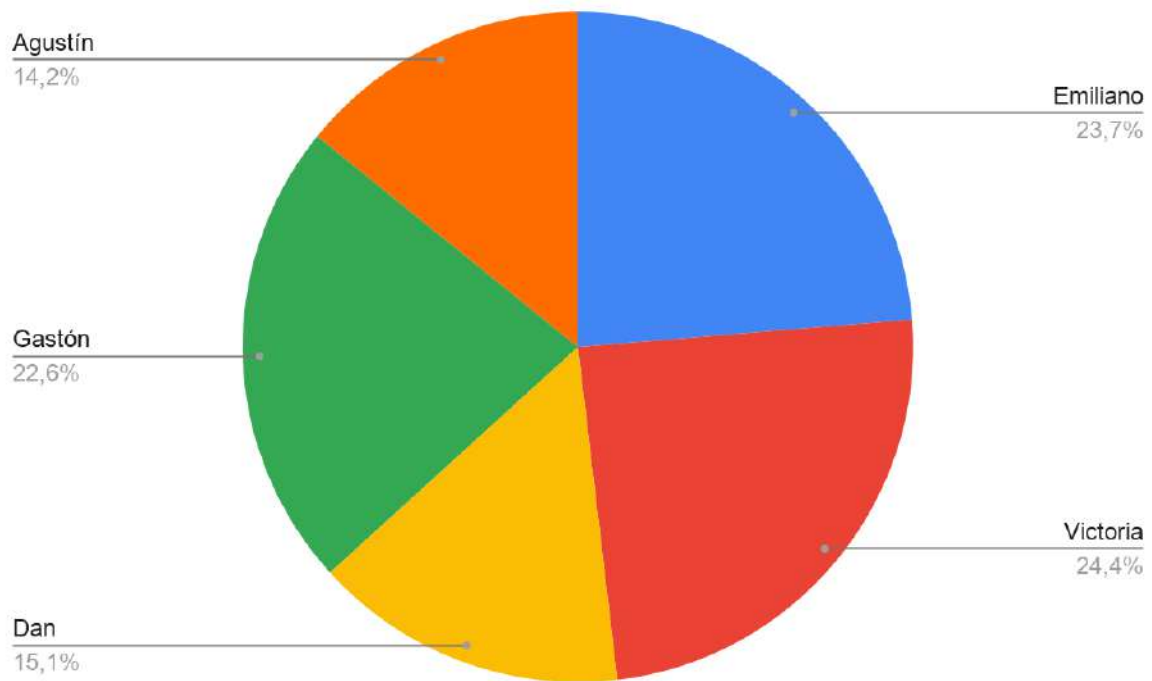


Ilustración 67: Dedicación horaria en documentación por persona

La gráfica muestra únicamente la distribución de las horas dedicadas a la documentación final del proyecto. Cabe destacar que el equipo trabajó durante todo el proyecto realizando los documentos pertinentes para cada área del mismo, en los cuales se basó la presente documentación.

Podemos ver que la gráfica anterior y esta se complementan, siendo quienes más horas dedicaron a la documentación los integrantes con menor carga horaria en el desarrollo.

### 10.2.1.7. Cumplimiento de los objetivos por iteración

Para conocer cómo era el grado de cumplimiento individual en cada iteración se desarrolló una métrica por porcentaje de tareas, la cual enviaba un informe a Slack al finalizar cada iteración.

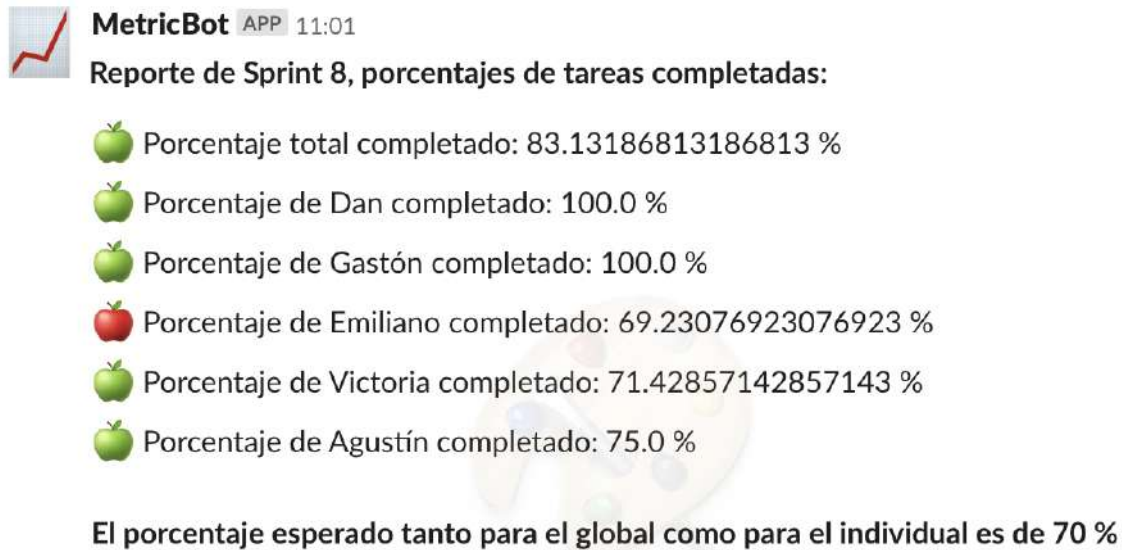


Ilustración 68: Porcentaje de tareas completadas por iteración

Se automatizó esta métrica creando un *script* en Python que consume los datos directamente de la API de ClickUp, la herramienta que se utilizó para gestionar las tareas del proyecto, y publicar la información obtenida en un canal de Slack que se creó para este propósito.

Fue útil para ver un resumen de cada iteración sin necesidad de analizar los datos manualmente.

### 10.2.1.8. Desviación entre esfuerzo estimado y esfuerzo real en las tareas de ingeniería

A continuación se muestra el *Burndown Chart* [96] para conocer qué tanta desviación existía en cada momento, con respecto a lo planificado.

Burndown chart

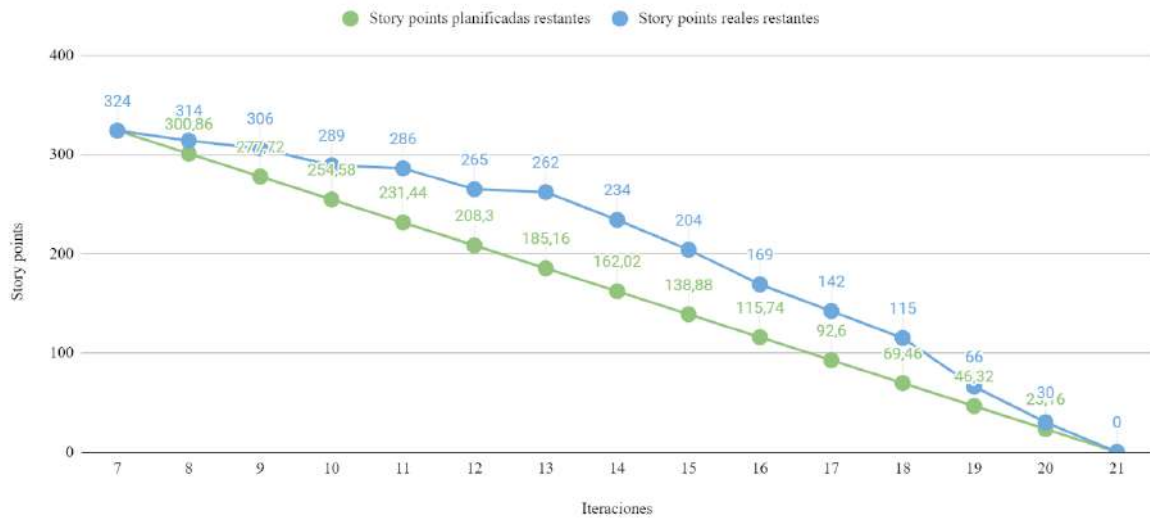


Ilustración 69: Desviación en las tareas de ingeniería entre lo planeado y lo realizado

Una de las razones por las cuales se nota un desfase es porque hubo errores de estimación, de tareas que se pensaban que eran más sencillas y que finalmente resultaron más complicadas.

Un claro ejemplo en la iteración 9, fue tener que re investigar el dron que se necesitaba para la solución del proyecto, debido a que el dron que más servía resultó tener un precio muy por encima del publicado, ya que este era engañoso y no incluía muchos de los complementos que eran requisito fundamental para que funcionara la solución.

### 10.3. Gestión de riesgos

La gestión del riesgo tiene como objetivo reducir la probabilidad e impacto de los eventos adversos en un proyecto. Si bien existen riesgos que pueden derivar en efectos positivos, se decidió gestionar solamente aquellos cuya consecuencia fuera negativa para el proyecto.

Debido a la inexperiencia del equipo en un proyecto de estas características, se decidió dedicar tiempo y esfuerzo a esta gestión, para controlar potenciales problemas que pudiesen comprometer los objetivos planteados.

En esta sección se desarrolla la estrategia seleccionada para gestionar los riesgos, así como su impacto a lo largo del proyecto. Dicha estrategia se basó en la definición brindada por el PMBOK [97]. Se incluye la identificación de los riesgos, su análisis cualitativo y seguimiento.

### 10.3.1. Identificación

Si bien se utilizó una metodología de trabajo ágil y por ende no correspondía construir un plan de gestión de riesgos, se gestionaron los riesgos más importantes para poder efectuar acciones preventivas y tener planes de contingencia en caso de que se confirmen los escenarios negativos.

Se creó en consecuencia la siguiente tabla de riesgos, los cuales fueron monitoreados iteración a iteración para actualizar su probabilidad e impacto. Para ver más información sobre cada riesgo observar el [anexo 3](#).

Id	Riesgo	Probabilidad (1-99)	Impacto (1-5)	Valor (1-5)
R1	No conseguir financiación para la compra de los drones, <i>tags</i> , y reader de RFID.	50,00 %	4	2
R2	Que el lector de RFID junto con la altura y el movimiento del dron no obtenga los resultados que se esperan.	40,00 %	4	1,6
R3	Que los requerimientos varíen mucho y atrasen el proyecto.	40,00 %	4	1,6
R4	Mala organización de los tiempos para cumplir con el proyecto impidiendo que este sea funcional y de calidad.	30,00 %	5	1,5
R5	Que se rompa el hardware.	30,00 %	5	1,5
R6	Que las especificaciones de los drones y/o de las otras	20,00 %	5	1

	herramientas no se ajusten a futuros requerimientos.			
R7	Dificultad para reunirnos con el experto	30,00 %	3	0,9
R8	Tener el dron, tags y lector RFID para probar apenas empezemos a desarrollar.	30,00 %	3	0,9
R9	Que el o los referentes de negocio no estén disponibles para relevar requerimientos y validarlos.	40,00 %	2	0,8
R10	Falta de comunicación en el equipo.	20,00 %	4	0,8
R11	Mala estimación de las tareas que genere retrasos	40,00 %	2	0,8
R12	Problema de alguno de los integrantes retrase el avance del proyecto.	20,00 %	3	0,6
R13	Desconocimiento de las tecnologías.	30,00 %	2	0,6
R14	Mala definición de los requerimientos.	30,00 %	2	0,6
R15	Mala división de tareas.	30,00 %	2	0,6
R16	Conseguir el campo para probar, y que esté disponible cuando necesitemos probar	40,00 %	1	0,4
R17	Falta de compromiso de alguno de los integrantes retrase el avance del proyecto.	10,00 %	3	0,3

Tabla 6: Riesgos asociados al proyecto

### 10.3.2. Actividades preventivas

A continuación se detallan las actividades preventivas realizadas para cada riesgo:

**Riesgo 1: No conseguir financiación para la compra de los drones, tags, y lectora de RFID**

Se realizó una solicitud profesional y dentro de los parámetros correspondientes para poder obtener la financiación.

**Riesgo 2: Que el lector de RFID junto con la altura y el movimiento del dron no obtenga los resultados que se esperan**

Se realizaron las investigaciones correspondientes para confirmar que esta tecnología cumple satisfactoriamente con los objetivos.

**Riesgo 3: Que los requerimientos varíen mucho y atrasen el proyecto**

Se buscó claridad en la definición de los requerimientos al iniciar el proyecto, así como realizar un exhaustivo relevamiento junto al cliente y los expertos de negocio.

**Riesgo 4: Mala organización de los tiempos para cumplir con el proyecto impidiendo que este sea funcional y de calidad**

Se llevó un estricto control sobre los tiempos, y los objetivos a cumplir. Se siguió el burndown chart iteración a iteración para saber qué tan alejados de lo comprometido se estaba, y así poder corregir a tiempo.

**Riesgo 5: Que se rompa el hardware**

Se estudiaron los manuales de uso de los drones, placas ArduPilot y lectores de RFID, de forma que su utilización sea como el proveedor espera. Aún así tuvimos muchos problemas de este tipo a lo largo de la tesis, los cuales fuimos mitigando según el caso (ver [capítulo 12](#)).

**Riesgo 6: Que las especificaciones de los drones y/o de las otras herramientas no se ajusten a futuros requerimientos**

Se realizaron muchas investigaciones para obtener el dron con las características que se adaptan mejor a las necesidades. Se estudiaron decenas de alternativas y se decidió por la utilización de drones compatibles con la placa ArduPilot, como se detalló en el capítulo

**Riesgo 7: Dificultad para reunirnos con el experto**

Se coordinaron las reuniones con mucha anticipación para evitar la falta disponibilidad

**Riesgo 8: Tener el dron, tags y lector RFID para probar apenas empecemos a desarrollar**

Estuvimos pendientes de los movimientos de financiación para la compra de los instrumentos. Se buscó obtener la respuesta sobre la financiación de la CI<sup>2</sup> y luego hicimos las investigaciones y averiguaciones para que Todosoft haga las compras.

**Riesgo 9: Que el o los referentes de negocio no estén disponibles para relevar requerimientos y validarlos**

Se pautaron reuniones con los referentes del negocio, de forma que estos se comprometieran a asistir.

**Riesgo 10: Falta de comunicación en el equipo**

Se planificaron reuniones frecuentes del equipo entero. Se usó una metodología de trabajo la cual fue cumplida estrictamente. El equipo estuvo permanentemente comunicado por lo cual las dudas y problemas fueron solventados de inmediato.

**Riesgo 11: Mala estimación de las tareas que genere retrasos**

Se utilizó una metodología ágil para poder mitigar los riesgos de tener malas estimaciones. Se fue analizando la velocidad del equipo iteración a iteración para ajustar en caso de ser necesario.

**Riesgo 12: Problema de alguno de los integrantes retrase el avance del proyecto**

Se contemplaron horas para enfermedades y otros problemas que puedan ocurrir en el desarrollo del proyecto. Se creó un Google Calendar para registrar estas ausencias y poder planificar en base a ellas.

**Riesgo 13: Desconocimiento de las tecnologías**

Se contempló un tiempo para la capacitación en determinadas tecnologías, y cada integrante trabajó con la que se sentía más a gusto. Al comienzo de los desarrollos se estimó teniendo en cuenta el desconocimiento de las tecnologías.

#### **Riesgo 14: Mala definición de los requerimientos**

Se realizaron reuniones con el equipo entero para definir los requerimientos de forma que el riesgo de equivocarse sea menor.

#### **Riesgo 15: Mala división de tareas**

Se utilizó la experiencia previa de los integrantes del equipo para dividir las tareas, es así que Agustín y Victoria, quienes contaban con más experiencia en lenguajes de bajo nivel, se encargaron del desarrollo del *software* del dron; Gastón y Dan eran los más expertos en aspectos *frontend* y por ende tomaron esas tareas, y Emiliano se dedicó al *backend* que era su especialidad.

#### **Riesgo 16: Conseguir el campo para probar, y que esté disponible cuando necesitemos probar**

Se establecieron contactos con varios dueños de campo previamente para coordinar la disponibilidad. Se consiguió un campo para realizar las pruebas.

#### **Riesgo 17: Falta de compromiso de alguno de los integrantes retrase el avance del proyecto**

No se encontraron tareas preventivas para este caso, más que motivar y hacer notar lo importante que era cada uno para el éxito del proyecto.

### 10.3.3. Seguimiento y evolución

En la siguiente gráfica se muestra cómo fue el avance (impacto por probabilidad) del valor de los cinco riesgos más importantes, iteración por iteración.

#### **Se agruparon los principales riesgos en cinco categorías:**

- Financiación: Que la financiación solicitada a CI<sup>2</sup> no sea aprobada.
- Falla en el *hardware*: Que falle, se rompa o no sea compatible algún artefacto de hardware.
- Compra del dron: Realizar la compra del dron adecuado.

- Integración *software-hardware*: Realizar la integración entre el dron y el lector RFID.
- Riesgos tecnológicos.

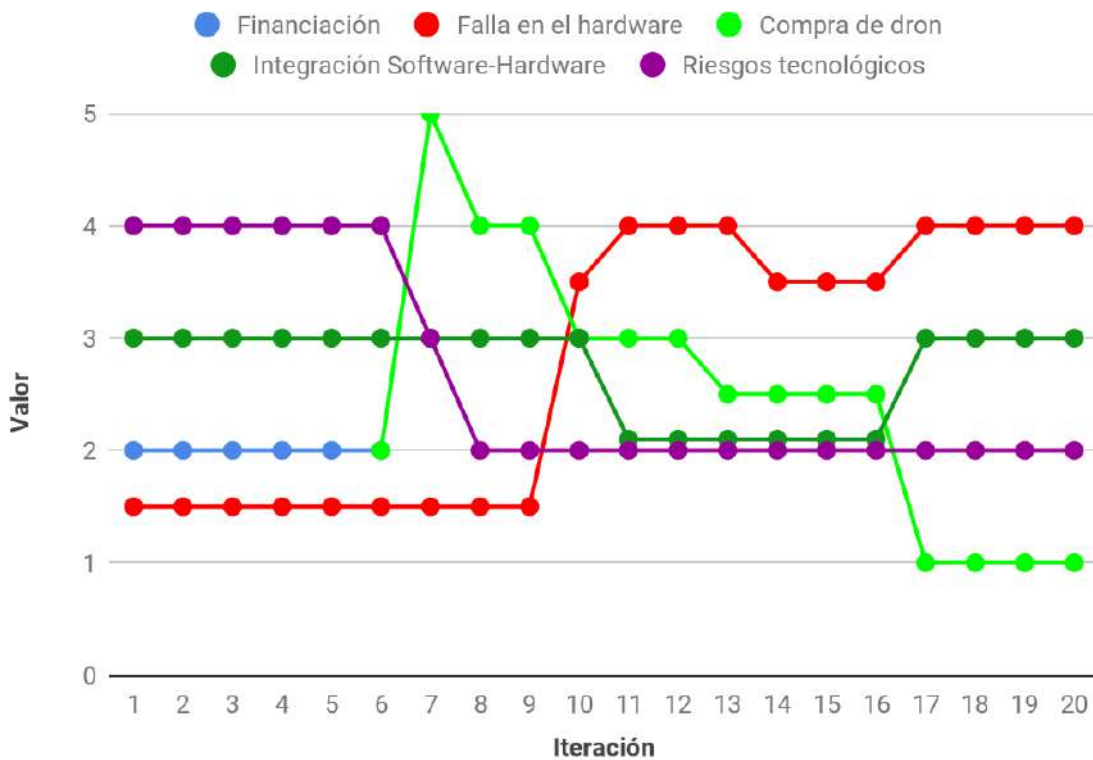


Ilustración 70: Evolución de los riesgos principales a lo largo del proyecto

### Que la financiación solicitada a CI<sup>2</sup> no sea aprobada

Al comenzar el proyecto se sabía que había que comprar un dron, pero que debido a su costo (superior a los 6.000 dólares) se debía buscar una manera de financiarlo.

La primera opción fue presentarse a la financiación ofrecida por la CI<sup>2</sup> para tesis de grado (ver [anexo 4](#)).

Sobre la iteración 6 se recibió la noticia de que la financiación había sido rechazada, por lo que en ese momento finalizó este riesgo y comenzó el riesgo “Realizar la compra del dron adecuado”.

### **Realizar la compra del dron adecuado**

Tal como se detalla en el [capítulo 5.2](#), conseguir el dron adecuado no fue tarea sencilla. Los requerimientos, el precio, el envío y el margen de modificabilidad fueron factores determinantes para la elección, la cual requirió mucho tiempo y esfuerzo.

Este riesgo se disparó al máximo en la iteración 5, ya que si no se lograba conseguir rápidamente el dron, todo el proyecto iba a fracasar.

A medida que se fue investigando y aclarando el panorama el riesgo fue disminuyendo su valor.

### **Que falle, se rompa o no sea compatible algún artefacto de hardware**

Desde el momento en que se compró el dron propio apareció este riesgo, y durante todo el proyecto tuvo un valor alto. De hecho, uno de los principales problemas que se tuvo a lo largo del proyecto fueron los inconvenientes en el hardware, tal y como se detalla en el [capítulo 12](#).

Si bien se implementaron acciones preventivas como comprar un dron de repuesto (que ayudó en varias oportunidades), surgieron muchos imprevistos que retrasaron de manera importante.

Hasta el final del proyecto este riesgo estuvo presente.

### **Realizar la integración del *software* con el *hardware* (dron y lector RFID)**

Ya sea por errores del dron, por falta de información, por la dificultad de realizar las pruebas o por otros motivos, realizar la integración entre el *software* y el dron fue un riesgo que estuvo presente durante todo el proyecto.

Las pruebas, por ejemplo, fueron siempre complicadas de hacer. Luego de probar el *software* en el simulador, se tenía que hacer lo mismo en el dron. Para probar en el dron había que ir a un lugar abierto, grande, donde no haya muchos árboles. Además debía ser a la luz del día y había que llevar la laptop, por lo que se dependía de la batería de la misma. Llevó mucho tiempo, y por ende el riesgo de que la integración no quede bien hecha fue grande.

## Riesgos tecnológicos

Inicialmente, cuando aún no estaban definidas las tecnologías, estaba presente el riesgo de elegir las mal. A medida que se fueron realizando las diferentes investigaciones, que se definió la arquitectura y que finalmente se seleccionaron las tecnologías más acordes este riesgo disminuyó hasta que quedó estable en un valor bajo.

## 10.4. Ejecución de iteraciones

### 10.4.1. Velocidad

Utilizando el concepto del *framework* Scrum, el esfuerzo, es decir, la dedicación de horas de los integrantes debe ser siempre constante, midiendo la cantidad de *story points* por iteración. A esto se le llama velocidad del equipo.

Al principio de cada iteración, se evaluó el tiempo disponible entre todos los integrantes del equipo para ver cuántos *story points* se podrían hacer en la misma. En las iteraciones que el equipo contaba con la misma cantidad de horas disponibles que en otra iteración ya finalizada, se tomaban historias de usuario que sumasen la misma cantidad de *story points* que dicha iteración previa.

Tal como indica Scrum, se tomó siempre la última velocidad para estimar la velocidad de la siguiente iteración, siempre y cuando esta no disminuyera con respecto a la anterior. En ese caso no se considera dicha velocidad y se toma la anterior.

A continuación se muestra una gráfica que indica la velocidad que hubo según la iteración:

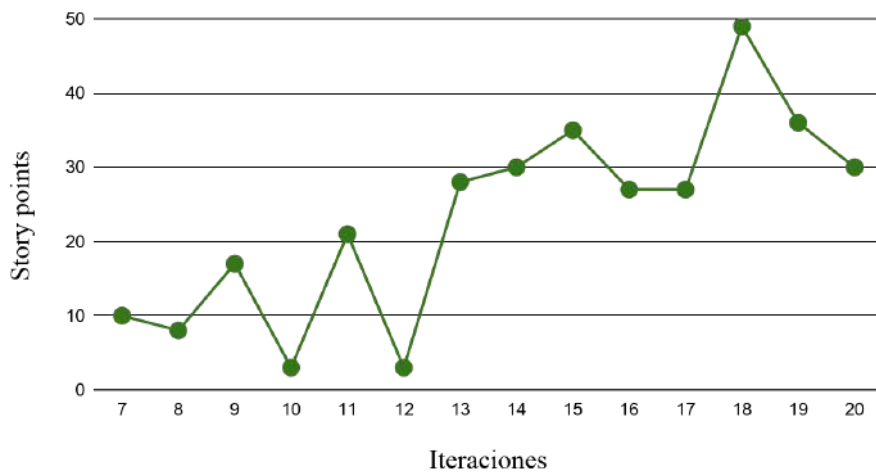


Ilustración 71: Velocidad de desarrollo a lo largo del proyecto

Para la velocidad solamente se toma en cuenta historias de desarrollo, razón por la que la gráfica comienza en la iteración 7.

Se puede ver que la velocidad del equipo por momentos no fue pareja. Esto se debió principalmente a que al comienzo se trabajó sobre tareas muy grandes, y se contaban como realizados los *story points* solamente cuando el 100 % de la tarea estaba finalizada (esto generó un sistema muy binario que no permitía medir el avance real).

Esto se corrigió a partir de la iteración 14, y desde ahí los valores fueron más estables.

## 10.4.2. Hitos del proyecto



Ilustración 72: Diferentes hitos que se atravesaron a lo largo del proyecto

**Marzo:** Se realizó el relevamiento de los requerimientos.

**Abril:** Se definió la metodología de trabajo y se entregó el documento para solicitar la financiación de la tesis al CI<sup>2</sup>.

**Junio:** Se definió la primera versión de la arquitectura del sistema.

**Julio:** Se validó el prototipo y los requerimientos con los expertos de negocio, y se realizó la primera revisión del proyecto de grado.

**Agosto:** Se puso en marcha la gestión de la calidad y SCM. A su vez se decidió comprar y utilizar un dron de la empresa DJI junto con una lectora de RFID de la empresa Marktrace.

**Setiembre:** Se inició el desarrollo de *software* y se definió que finalmente se compraría un dron de la empresa Accsys. También se realizó la segunda revisión académica.

**Octubre:** Se lanzó el primer *release* de la plataforma y se comenzó con las pruebas del dron.

**Noviembre:** Se presentó la tercera y última revisión de la tesis.

**Enero:** Se lanzó el segundo *release* de la plataforma, en el cual el dron comenzó a volar automáticamente (con el sistema desarrollado).

**Febrero:** Se liberó el último y definitivo *release*, donde se completaron todos los requerimientos establecidos previamente junto a la empresa Todosoft.

## 10.5. Gestión de la comunicación

En la siguiente sección se detalla cómo se gestionó la comunicación interna del equipo, así como con los demás interesados en el proyecto.

El equipo consideró desde un inicio que la comunicación sería algo fundamental para el proyecto ya que una buena comunicación ayuda a evitar problemas de interpretación y mejorar la calidad del proyecto.

El objetivo a nivel de comunicación fue que las herramientas seleccionadas permitan a todos los integrantes del equipo mantener una comunicación fluida y adecuada.

La herramienta principal para la comunicación dentro del equipo fue Slack, dentro del *workspace* “kobauy.slack.com”, en el canal correspondiente dependiendo del tópico de la comunicación. También se utiliza WhatsApp para las comunicaciones de carácter urgente.



Para la comunicación con la tutora del equipo, cliente y demás involucrados del proyecto se utilizó WhatsApp y correo electrónico.

Para comunicarnos con la tutora se coordinaron reuniones en la Universidad ORT con cierta frecuencia.

Con el cliente, por su parte, se hicieron varias reuniones presenciales en las oficinas de la empresa Todosoft.

### 10.5.1. Espacio de trabajo en Slack

Con la finalidad de mantener una comunicación ordenada se establecen algunas pautas para el correcto manejo de los canales:

- El canal #tesis se utiliza para discutir tópicos generales y compartir información de interés para todos los miembros del equipo.
- El canal #importante-novedades se utiliza para anunciar decisiones o novedades importantes (que impactan a todo el equipo o que afectan el tiempo, alcance o costo del proyecto).
- Los mensajes compartidos en los canales que empiezan con la palabra #importante deben ser leídos y respondidos por todos los miembros del equipo reaccionando al mismo con el ícono .
- Los canales que empiezan con la palabra #reunión son para registrar las agendas y minutas de las reuniones. Todos los miembros del equipo deben leer y reaccionar con el ícono  la minuta.

### 10.5.2. Calendario de Google

Tiene la finalidad de compartir los eventos que conciernen a todos los miembros del equipo.

Es un calendario compartido lo que permite que cualquiera pueda crear eventos nuevos o modificar los existentes.

También sirve para marcar los períodos de ausencia de los miembros del equipo, de forma de conocer la disponibilidad de cada uno.

## 10.6. Conclusiones

El equipo considera que la gestión del alcance fue correcta ya que se pudieron implementar los 324 *story points* comprometidos (100 % de los *story points* de requerimientos de prioridad alta y media), cumpliendo así con el objetivo inicial propuesto por el equipo que era desarrollar en su totalidad los requerimientos de máxima prioridad. A su vez, los entregables que no son de producto, definidos en el [capítulo 4.4](#) se completaron satisfactoriamente y serán entregados al cliente una vez finalizado el proyecto.

Las herramientas seleccionadas para la gestión del tiempo y de la comunicación fueron las adecuadas ya que cumplieron su objetivo.

El equipo consideró que a la hora de medir la velocidad de este proyecto, la forma de definir cuándo una historia de usuario estaba finalizado no fue la mejor y por ende se corrigió sobre el transcurso del desarrollo.

En proyectos a futuro se consideraría cambiar los criterios de finalización de una historia de usuario, dividiendo en subtareas más pequeñas para poder ir finalizándolas más regularmente y así medir correctamente la velocidad.

Por último, se considera que se debería organizar mejor la disponibilidad del equipo de trabajo para que el esfuerzo de todos los integrantes sea constante a lo largo del proyecto y así mejorar la productividad y obtener una velocidad más uniforme.

## 11. Calidad

Este capítulo describe los objetivos de calidad planteados para este proyecto y las actividades para lograrlos.

Debido a que en el sistema SkyGuard intervienen componentes de alto costo, el equipo consideró que controlar la calidad tanto de *software* como de *hardware* es vital para el éxito del proyecto.

### 11.1. Objetivos de calidad

Con el fin de asegurar la calidad, se definieron objetivos a nivel de proceso, de producto y académicos.

#### 11.1.1. Objetivos de calidad del proceso

Con el objetivo de realizar el proyecto de forma apropiada, se decidió llevar adelante un proceso que cuente con las actividades que le permitieran al equipo mantener un orden y con el cual los integrantes se sientan cómodos de forma de poder cumplir los objetivos generales del proyecto de manera eficiente.

Para esto se decidió llevar adelante una metodología de desarrollo ágil, inspirada en Scrum, dándole la importancia no solo a las ceremonias sino que también a los artefactos definidos en el [capítulo 3](#).

#### 11.1.2. Objetivos de calidad del producto

Asegurar el cumplimiento del 100 % de los requerimientos funcionales de prioridad alta y media así como también de los requerimientos no funcionales del sistema, validando los mismos con el cliente, de forma de aportar el mayor valor agregado a los usuarios finales.

Además, se buscó asegurar solidez y robustez en la integración de *hardware* con *software* de forma de evitar fallos que podrían terminar en la pérdida total de componentes fundamentales para el funcionamiento del sistema.

Ante cualquier inconveniente, el dron está preparado para volver a la base y quedarse a la espera de la ejecución de la próxima misión.

En caso de falta de conexión la información no se pierde, sino que queda almacenada internamente para su posterior procesamiento.

### 11.1.3. Objetivos de calidad académicos

Realizar un documento final cumpliendo con los estándares exigidos por la Universidad ORT Uruguay que se especifican en los documentos nro. 302 y nro. 303.

## 11.2. Plan de calidad

Se definió al comienzo del proyecto un plan de calidad para poder alcanzar los objetivos propuestos.

Si bien el equipo decidió utilizar una metodología ágil para la fase de desarrollo, a la hora de definir el plan de calidad se optó por seguir un proceso tradicional ya que un plan de estas características utiliza procesos más formales, definiendo con mayor nivel de detalle los elementos que lo componen.

Para definir el plan de calidad lo que se hizo fue dividir los ciclos de vida del proyecto en las distintas áreas que lo conforman y listar las diversas actividades de aseguramiento de calidad que se debían llevar a cabo en cada una de ellas.

Como parte de la estructura del plan de calidad, se incluye el producto consumido, el producto resultante así como los roles responsables y participantes.

A continuación se detalla el plan de calidad definido:

Fase	Actividad	Entrada	Salida	Herramienta	Rol encargado	Rol participante
Investigación	Estudio del problema	de potencial problema y confirmación de su existencia	Problema del Proyecto	Documentos online, reuniones con expertos en el tema,	Equipo	Equipo y expertos

	Buscar soluciones	Problema del Proyecto	Solución del Proyecto	bench marking, documentos online, reunión con el cliente		Equipo y cliente (Alejandro)
	Estudio de las tecnologías	Problema Definido	Documento con datos de investigación	Documentos online, reuniones con expertos de las tecnologías		Equipo y expertos de tecnología
	Capacitación	Tecnologías Definidas	Conocimientos sobre Tecnologías	Documentación, información online		Equipo
Ingeniería de requerimientos	Identificación de requerimientos	Información relevada por el equipo	Lista de requerimientos.	Reunión con expertos, reunión con el cliente, brainstorming, benchmarking.	Equipo	Equipo
	Validación de requerimientos	Lista de requerimientos	Lista de requerimientos validada.	Reunión con expertos.		
	Priorización de requerimientos	Lista de requerimientos	Lista de requerimientos priorizados	Reunión de equipo, costo - beneficio.		
	Estimación de requerimientos	Lista de requerimientos priorizados	Lista de requerimientos priorizados y estimados	Técnica Shirts, Poker planning.		
	Definición inicial del alcance	Lista de requerimientos procesados y estimados	Project Backlog	Reunión de equipo, reunión con el cliente, ClickUp.		
	Validación del alcance	Project Backlog	Requerimientos validados	Consultar con tutor y con el cliente.	IR	IR

	Especificación de requerimientos	Requerimientos validados	Especificación de requerimientos	ClickUp.		
	Validación de especificación	Especificación de requerimientos	Especificación de requerimientos validados	Cliente y expertos. <i>Mockups</i>		
Arquitectura y Diseño	Análisis de restricciones y/o requerimientos no funcionales	Restricciones y requerimientos no funcionales	Atributos de calidad y tácticas de arquitectura	Artículos y experiencia propia	Arquitecto	Arquitecto, Equipo
	Análisis de posibles soluciones de arquitectura y diseño	Restricciones, requerimientos NF, atributos de calidad y tácticas	Primeros diagramas de arquitectura y diseño	<a href="#">Draw.io</a>		
	Diseño de la arquitectura	Primeros diagramas	Documento de arquitectura	Consulta con expertos		
	Revisión y validación de la arquitectura	Documento de arquitectura	Arquitectura validada y revisada	Revisión con expertos y cliente		
	Diseño de la solución	Requerimiento Funcionales	Documento de diseño	UML		
Construcción	División de Tareas	Arquitectura, Diseño y Tecnologías	Responsabilidades Asignadas	Reunión de equipo y evaluación de habilidades	Equipo	Equipo
	Desarrollo de Tareas	Diseño, Arquitectura y Project Backlog	Código Fuente	C#, Python, React, React Native		
	Pruebas unitarias y de integración	Plan de prueba, Código fuente y Requerimientos	Pruebas Implementadas y ejecutadas correctamente	C#, Python, React, React Native y simuladores de vuelo		
	Pruebas Funcionales	Pruebas a ejecutar	Pruebas ejecutadas correctamente	Postman		

Pruebas	Pruebas de Usabilidad	Diseño y Especificación de Requerimientos	Resultados de Test y de Evaluación	Test con usuarios	Equipo	Equipo
	Pruebas de integración	Diseño y Especificación de Requerimientos	Resultados de pruebas de integración	Conexión <i>hardware - software</i>		
	Pruebas de dron en entorno real	Diseño y Especificación de Requerimientos	Resultados de pruebas de campo	Dron, campo		

Tabla 7: Plan de calidad

### 11.3. Aseguramiento de la calidad

Para poder asegurar que el producto y el proceso fueron de calidad, el equipo se propuso cumplir con los objetivos de calidad previamente definidos y el seguimiento de estándares.

A continuación se describen los estándares que fueron seguidos por parte de todo el equipo a lo largo del desarrollo del proyecto con el fin de lograr consistencia.

#### 11.3.1. Aplicación de estándares

Una de las primeras actividades llevadas a cabo fue la definición de estándares de codificación y documentación.

Dado que el sistema seguirá siendo desarrollado por el cliente, el equipo se apegó fuertemente al cumplimiento de los estándares definidos.

##### 11.3.1.1. Estándares de codificación

Para lograr un código mantenible y entendible se decidió utilizar los siguientes estándares:

- Código en idioma Inglés, por ser el idioma universal para el desarrollo de *software*.
- El código de los distintos componentes seguirá las prácticas de Clean Code [\[98\]](#).
- El *backend* seguirá los estándares de .NET [\[99\]](#) que la comunidad del *framework* definió.

- El *frontend*, desarrollado React seguirá los estándares de ECMAScript [\[100\]](#).

#### 11.3.1.2. Estándares de documentación

Para los estándares de documentación se siguieron las normas provistas por la Universidad ORT Uruguay para los proyectos finales de carrera.

- **Documento 302:** Normas específicas para la presentación de trabajos finales de carrera [\[101\]](#).
- **Documento 303:** Hoja de verificación. Pautas de presentación de trabajos finales de carreras [\[102\]](#).
- **Documento 304:** Normas para el desarrollo de trabajos finales de carrera [\[103\]](#).
- **Documento 306:** Orientación para títulos, resúmenes, abstracts e informes de corrección de trabajos finales de carrera [\[104\]](#).

#### 11.3.2. Pruebas de *software*

En conjunto con el cliente se decidió que, dado que el sistema una vez entregado por parte del equipo aún estará en fase de desarrollo, se haría foco en brindar pruebas sólidas en lo que respecta a la integración de los distintos componentes que forman parte de la solución y a las pruebas de vuelo del dron.

Se decidió no dedicar tiempo a la realización de pruebas unitarias, priorizando los otros tipos de pruebas debido a lo siguiente:

- La empresa Todosoft no suele realizar pruebas unitarias en sus proyectos y prefiere que el equipo se enfoque en las de integración.
- Se puso mucho empeño en la realización de pruebas manuales de integración del *hardware* y *software*.
- Se dio prioridad máxima a las pruebas manuales con el dron, debido a que es el componente más importante y delicado del sistema

Como plan a futuro, la empresa Todosoft se encargará de crear las pruebas unitarias del *backend* a medida que lo vea necesario.

A continuación se especifican las pruebas realizadas por el equipo.

#### 11.3.2.1. Pruebas cruzadas

Dado que el sistema SkyGuard está formado por varios componentes, y no todos los integrantes del equipo participaron del desarrollo de cada uno, se decidió que era necesario realizar pruebas cruzadas para cada funcionalidad desarrollada ya que si bien cada integrante prueba su funcionalidad, también es importante que otro lo haga debido a detalles que pueden no ser vistos por una sola persona.

#### 11.3.2.2. Pruebas de integración entre el *hardware* y el *software*

Dada la complejidad de integración que tiene la solución y los tipos de componentes que intervienen, el equipo decidió hacer pruebas de integración en donde participen los componentes de *hardware* y los componentes de *software* relacionados con el plan de vuelo y la ejecución de una ruta, es decir, SkyGuard-Copter y SkyGuard-Pathfinder.

Las pruebas iniciales durante la fase de desarrollo se hicieron utilizando simuladores de vuelo ya que en ese momento el equipo no disponía del *hardware*. Las mismas permitieron simular entornos reales y comprobar que el funcionamiento del plan de vuelo era acorde a lo esperado.

Una vez obtenido el *hardware* de pruebas el equipo visitó varios lugares abiertos y despejados como plazas y campos, de manera de realizar las mismas pruebas hechas en el simulador pero en un entorno real, y también detectar si en los campos una vez el dron volando sobre los animales, los mismos se asustan o tienen alguna reacción en base al ruido generado por el mismo.

Se realizaron un total de 14 sesiones de prueba en entornos reales, en donde cada sesión tuvo aproximadamente 7 vuelos.

Los vuelos realizados fueron de corto alcance debido a que una mala configuración podría terminar estrellando el dron y dejándolo inutilizable para futuras pruebas. Para estas primeras sesiones, el vuelo del dron se realizó sujetándolo con una cuerda de largo máximo de 6 metros

para evitar que se aleje más de lo deseado. Igualmente por más que se tomaron los recaudos necesarios, el primer dron sufrió roturas importantes debido a un aterrizaje imperfecto.



Ilustración 73: Dron roto tras una caída

Los resultados acerca del comportamiento del dron llevaron a realizar varios ajustes para poder tener un vuelo más preciso. Se detectaron 3 tipos de errores que se detallan a continuación.

- En primera instancia el dron recibía los comandos y rutas a través de cable USB. Esto generó muchos problemas ya que el dron comenzaba a volar sin tener toda la información procesada por lo cual se realizaba el vuelo de forma correcta hasta que no tenía más información y se descontrolaba. Gracias a la ayuda obtenida por parte de expertos pertenecientes al foro de Ardupilot [\[105\]](#) encontramos la solución que consistió en utilizar un módulo de telemetría para poder tener una conexión al dron inalámbrica.
- Una vez solucionado el punto anterior se detectó que la precisión del GPS utilizado es de un rango de 5 metros aproximadamente por lo que en primera instancia cuando se realizaban pruebas con una distancia de hasta 6 metros (distancia de la cuerda) el margen era muy pequeño para obtener buenos resultados. Una vez detectado esto, se realizaron pruebas con un mayor alcance pudiendo obtener mejores resultados.
- Por último, la calibración de los componentes del dron, como por ejemplo los motores, en caso de estar mal configurados, hacían que el dron volara con comportamientos

extraños. La solución a esto fue que cada vez que se realizara algún cambio de componente, se reiniciaría la calibración y se hiciera nuevamente desde cero.

En cuanto al sentimiento de los animales hacia el ruido generado por el dron, se demostró que no es un problema para ellos. Los mismos no se ven afectados al volar sobre ellos (a una distancia de aproximadamente 5 metros).



Ilustración 74: Dron conectado a una PC ejecutando el plan de vuelo



Ilustración 75: Despegue de dron atado a una cuerda

#### 11.3.2.3. Pruebas con usuarios

Si bien la usabilidad no se atacó en los atributos de calidad a nivel de arquitectura de *software*, el equipo consideró importante tener la aprobación de los usuarios, tanto de usuarios expertos en el negocio como del cliente Todosoft, ya que un descontento de los mismos podría llevar una reestructuración del proyecto impactando en los costos.

Obtener retroalimentación de los usuarios también sirvió para poder desarrollar una mejor solución y tener una visión de la facilidad de uso del sistema.

Las pruebas de usabilidad se realizaron con el cliente, la empresa Todosoft. En el [anexo 8](#) se detallan los cambios sugeridos por el mismo. Así como también se realizaron pruebas con expertos de negocio, en particular con José Luis Callero en Facultad de Veterinaria.

Jose Luis probó la aplicación web y la aplicación móvil, le pareció intuitiva y fácil de usar, no tuvo ninguna objeción al respecto, pero nos realizó dos sugerencias referentes al mapa:

- Los colores del mapa de calor le gustaría que sean más intensos.
- El zoom debe ser dependiendo del tamaño del campo. Si un campo es muy grande estaría bueno que se pueda ver todo en la primera instancia.

#### 11.3.2.4. Pruebas de conectividad

Para cumplir con uno de los objetivos que consiste en la integridad de los datos, se realizaron pruebas de conectividad pudiendo detectar que en caso de pérdida de conexión los datos no se pierden y son almacenados para su posterior sincronización con los servicios de *backend*.

Se realizaron varios tipos de pruebas para comprobar que el funcionamiento sea el esperado. Vuelos con conexión; sin conexión; y con conexión intermitente.

El dron que se eligió fue específicamente pensando en la problemática que puede haber dada la falta de conectividad que existe hoy en los campos. Esto quiere decir que la infraestructura del dron permite almacenar las lecturas leídas por el lector RFID en caso de falta de conexión.

#### 11.3.2.5. Pruebas de satisfacción con el cliente

A lo largo del desarrollo del proyecto, se realizaron reuniones con el cliente para ir chequeando el correcto avance del proyecto, y entre ello ir viendo la satisfacción del mismo en lo que respecta a cada uno de los componentes que intervienen en la solución final.

Una vez que se tuvo una estructura sólida y bien definida de lo que se quería, se publicó el componente web con entrega continua y el componente móvil generando un instalable y entregándolo al cliente, de forma tal que pueda ver los avances en cualquier momento sin necesidad de una reunión presencial o virtual. Esto aportó una retroalimentación rápida y continua, lo cual ayudó a ir modificando algunos puntos sobre la marcha. Más adelante en el [capítulo 11.9](#) se verá con mayor detalle.

### 11.4. Gestión de incidentes

El objetivo de la gestión de incidentes fue registrar los defectos y mejoras que fueron detectadas en el producto, para poder ir las corrigiendo en orden a su prioridad.

Se decidió llevar un registro de incidentes en la herramienta ClickUp de igual forma que se registraban las historias de usuarios y tareas a realizar, pudiendo asignarle un responsable, y una prioridad.

En las distintas reuniones de planificación, se analizaban los incidentes y se veía cual podría entrar en la iteración para ser desarrollado y cuáles se dejarían para las siguientes iteraciones.

En caso de ser un incidente que bloqueara el desarrollo de alguna tarea perteneciente a la iteración, se comunicaba con el posible responsable, y en caso de ser algo sencillo, se intentaba resolver en el momento para no bloquear a los demás integrantes.

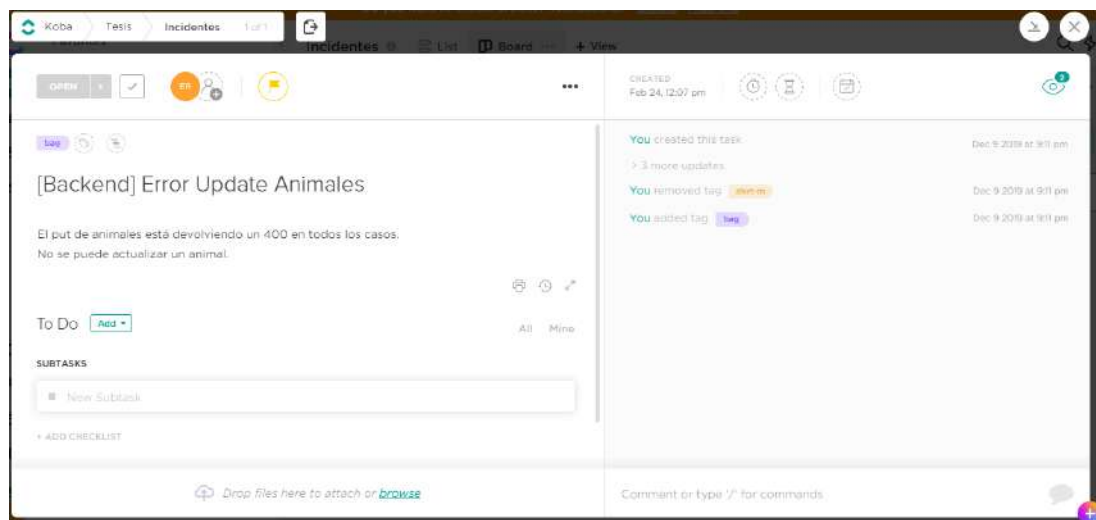


Ilustración 76: Tarjeta de incidente

## 11.5. Revisiones

Se realizaron revisiones tanto dentro del equipo como con expertos fuera del mismo.

Dentro del equipo se realizaron revisiones de código con al menos un integrante que no haya participado en la porción de código a revisar.

Estas prácticas sirvieron para mejorar la implementación, asegurar el uso de nomenclatura uniforme, buenas prácticas de Clean Code y mantenimiento de los estándares de codificación definidos. En el [anexo 9](#) se encuentra información más detallada acerca de la revisión de código.

También se realizaron revisiones de documentación. El equipo decidió utilizar herramientas de Google en la nube para el almacenamiento y creación de los mismos. Esto permitió trabajar de manera simultánea entre todos los integrantes pudiendo modificar o agregar comentarios de forma de que todos puedan aportar o brindar sus opiniones.

Además, se realizaron revisiones con expertos externos que ayudaron brindando una visión diferente y oportunidades de mejora, como por ejemplo se consultó al Lic. Gastón Mousques acerca de aspectos de Arquitectura de *Software*. También al Ing. Nicolás Cremona en lo que refiere a electrónica.

Y por último se llevaron a cabo tres revisiones formales con miembros del laboratorio ORT Software Factory. Ing. Álvaro Ortas, Lic. Gastón Mousques, Ing. Darío Macchi. En cada una de estas instancias se expusieron los avances del proyecto al revisor, con el objetivo de recibir retroalimentación y poder tomar acciones correctivas y preventivas. Si bien estas revisiones fueron impuestas por la Universidad ORT, el equipo las consideró de gran importancia y aporte de calidad al proceso.

En el [anexo 7](#) se encuentran los informes de las revisiones hechas con los docentes.

## 11.6. Retrospectivas

A lo largo del proyecto, se mantuvieron múltiples reuniones entre los integrantes del equipo al finalizar cada iteración en las que se fue evaluando el trabajo realizado, los incidentes ocurridos, e identificando los aciertos y errores del equipo con el fin de poder atacar de forma temprana los problemas identificados.

Este proceso de retrospectivas se detalla en el [capítulo 3.4.3](#).

## 11.7. Validación

Como se mencionó anteriormente, el cliente tenía una idea, pero no había un estudio sólido por detrás que brindara la seguridad de que el proyecto fuera a ser exitoso.

Es por esto que el equipo decidió tener reuniones con los usuarios finales, y expertos del negocio, para asegurar la viabilidad del producto y tener una mejor idea de lo que se espera para el mismo.

Para ello se utilizaron prototipos que permitieron ir definiendo los requerimientos y construir el *backlog* más estable posible.

Además se realizaron validaciones del producto a medida que se fue desarrollando gracias al despliegue continuo. Esto nos permitió la posibilidad de que el cliente viera constantemente los nuevos cambios pudiendo brindarnos feedback temprano reduciendo así el costo de cambio.

## 11.8. Gestión de la configuración

A continuación se describe cómo se gestionó la configuración del *software*, con el objetivo de lograr un orden respecto a las versiones del código para maximizar la productividad y evitar horas de re trabajo.

También se define cómo se trabajó en la organización de los documentos utilizados de forma de poder guardar y obtener información de manera eficiente.

### 11.8.1. Identificación de los elementos de configuración

Los elementos de la configuración del *software* son piezas de *software* y documentos generados a lo largo del proyecto que deben ser gestionados y controlados.

Se analizaron las diferentes áreas del proyecto con el objetivo de determinar cuáles debían ser versionadas y con qué herramienta.

Una vez definido y realizado el análisis, se armó la siguiente definición:

Área	Elemento
<i>Software</i>	Código fuente <i>backend</i>
	Código fuente aplicación web
	Código fuente aplicación móvil
	Código fuente <i>scripts</i> de interacción con el dron
Documento	Plan de calidad
	Gestión de riesgos
	Prototipación
	Documento de arquitectura
	Metodología de trabajo
	Retrospectivas
	Relevamiento de requerimientos
	Reuniones con expertos

	Métricas de avance
	Informes de revisión
	Formularios de financiación
	Estándares de codificación

Tabla 8: Elementos de configuración de *software*

## 11.8.2. Elección de las herramientas

En esta sección se presenta la elección de herramientas de apoyo para gestionar los productos generados durante el proyecto.

### 11.8.2.1. *Software*

Al momento de determinar la herramienta a utilizar para la gestión, versionado y seguimiento del código fuente se optó por un repositorio distribuido, en particular el *software* de control de versiones Git.

Motivos por los que se eligió Git:

- Permite trabajar con ramas locales obteniendo una gran flexibilidad en la forma de trabajo.
- Facilidad para trabajar independientemente.
- Fácil colaboración entre miembros del equipo.
- *Workflow* basado en funcionalidades.
- Resolución de conflictos de forma sencilla.
- Conocimiento previo por parte de los integrantes.
- Herramienta gratuita

Como servicio de alojamiento para los repositorios se utilizó GitHub. La elección de esta plataforma respecto a otras competidoras fue debido a que no tenía costos para obtener repositorios privados gracias a convenios que tiene la Universidad ORT con ellos, el equipo

está acostumbrado a trabajar en esta plataforma, y permite integrarse con herramientas de integración continua lo cual era necesario para entregarle versiones constantemente al cliente.

## **Repositorios**

El equipo decidió desde el inicio repartir los diferentes componentes en distintos repositorios pudiendo así mantener una independencia en el desarrollo de los mismos. También si se quería brindar una integración continua, la cual era más fácil si se hacía foco en cada uno de estos componentes por separado.

Por lo tanto la división de los repositorios quedó de la siguiente manera:

SkyGuard-UI: Es el repositorio utilizado para el desarrollo *frontend*. Este repositorio contiene la aplicación web desarrollada en React, y las aplicaciones para dispositivos móviles desarrolladas con React Native.

SkyGuard-Api: Es el repositorio utilizado para el desarrollo *backend*. Este repositorio contiene una API REST desarrollada en .NET Core la cual es consumida por la aplicación web y las móviles.

SkyGuard-Copter: Es el repositorio utilizado para el desarrollo de *scripts*. Este repositorio contiene *scripts* desarrollados en Python los cuales se encargan de comunicaciones e interacciones con el dron.

SkyGuard-PathFinder: Es el repositorio utilizado para el desarrollo *backend* que actúa como intermediario entre la API REST y el dron. Este repositorio contiene servicios web que indican la ejecución del *script* en Python, y disponibiliza un *endpoint* para obtener las lecturas del dron en forma manual.

## **Política de trabajo**

Se utilizó GitFlow como flujo de trabajo, ya que define un modelo de ramificaciones muy organizado y apropiado para la gestión de proyectos.

En todos los repositorios se aplicó esta política de trabajo, teniendo dos ramas principales, una rama *master* que contiene el código listo para estar en producción, y una rama *develop* que contiene código funcionando con las últimas actualizaciones.

Los integrantes del equipo trabajan en una rama individual referente al *feature* que están desarrollando, y una vez que decidan pasar este código a la rama *develop*, debe pasar por la revisión de otro integrante.

A continuación se muestra una ilustración del manejo de ramas.

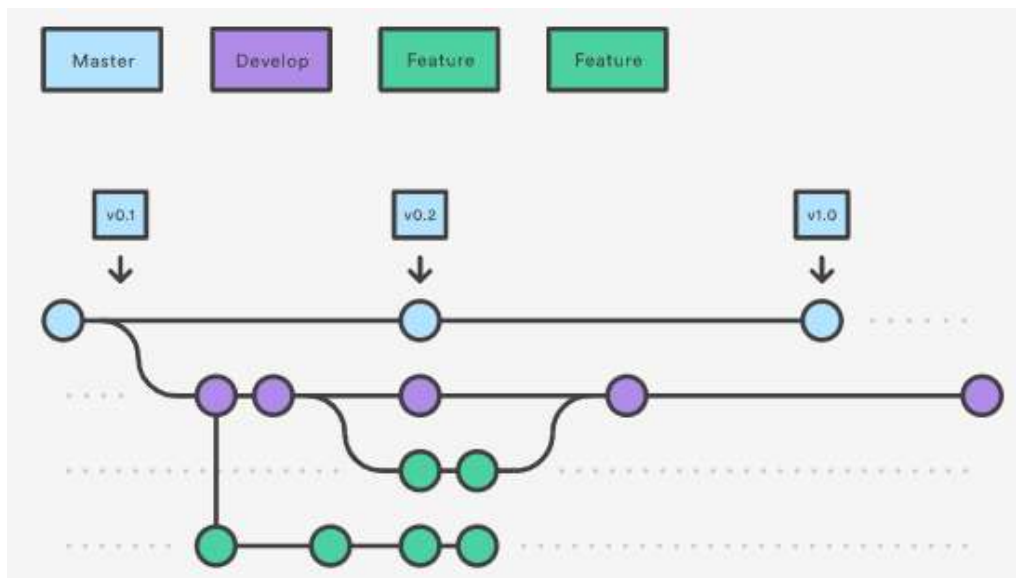


Ilustración 77: Manejo de ramas

#### 11.8.2.2. Documentación

El equipo decidió que para la documentación, las herramientas que más se adecuaban eran las provistas por Google. Por lo tanto para el almacenamiento se utilizó Google Drive, y para la creación y mantenimiento de documentos se utilizó Google Docs, Sheets y Presentation.

La decisión fue debido a las ventajas que Google brinda con sus herramientas para poder trabajar de forma colaborativa en documentos, ya sea agregando comentarios, notas, viendo modificaciones en tiempo real, almacenamiento de históricos, etcétera. Y también brinda un almacenamiento de 15 GB lo cual fue más que suficiente para el proyecto.

Para la organización de los documentos se trabajó con una estructura de carpetas como se observa a continuación:

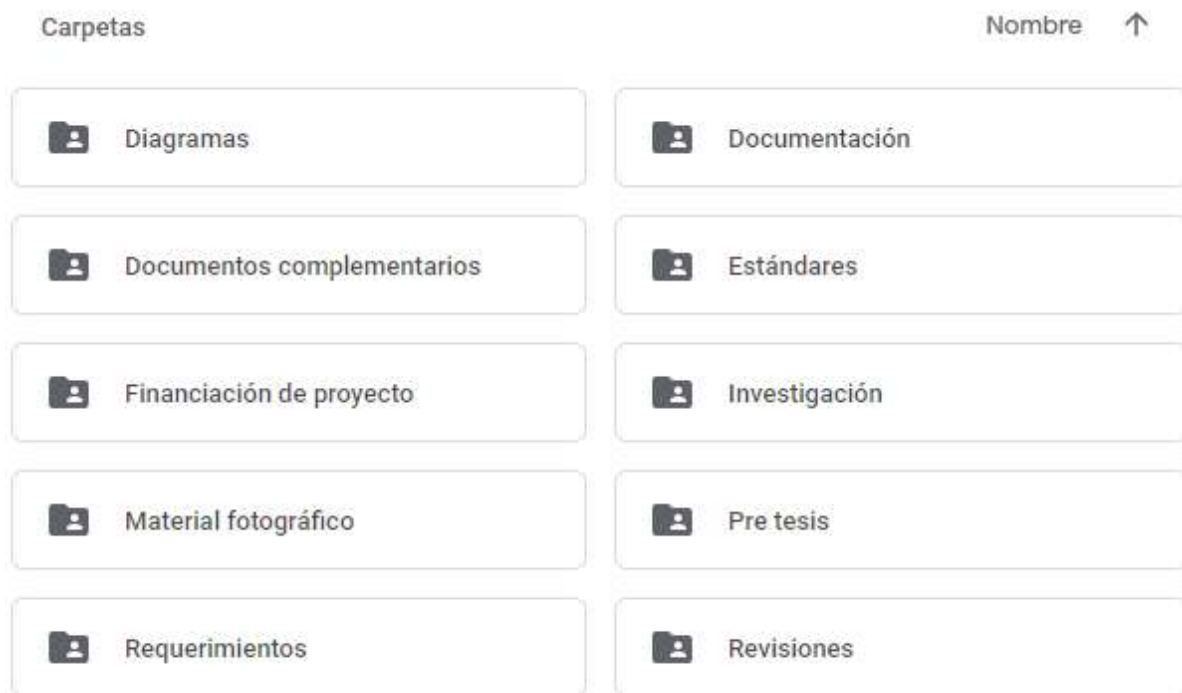


Ilustración 78: Estructura de carpetas

A continuación se describe el contenido de las carpetas.

**Diagramas:** Almacena todos los diagramas realizados, ya sean diagramas de paquetes, de clases, de flujo, *mockups*, entre otros.

**Documentación:** Almacena todos los documentos formales que intervienen en la documentación final como por ejemplo documentos de arquitectura, métricas, plan de calidad, riesgos, etcétera.

**Documentos complementarios:** Almacena los documentos no formales que son utilizados por el equipo, como por ejemplo formularios de preguntas para las entrevistas con expertos, historial de retrospectivas, mejoras propuestas, etcétera.

**Estándares:** Almacena documentos con la definición de estándares a seguir por el equipo durante el desarrollo del producto. Por ejemplo los documentos 302, 303, 304 y 306 brindados por la Universidad ORT.

**Financiación de proyecto:** Almacena los documentos que se presentaron para la financiación del dron. Por ejemplo formularios y carta de motivación.

**Investigación:** Almacena la información obtenida en la etapa de investigación. Por ejemplo comparativa entre diferentes drones, información de RFID, entre otros.

**Material fotográfico:** Aquí se almacenan imágenes y videos de varios tipos. Imágenes y videos que se realizaron cuando se hicieron pruebas en el campo, imágenes de lluvia de ideas en papel, etcétera.

**Pre tesis:** Aquí se almacenan documentos presentados previo a la selección de la tesis, donde se presenta ante la Universidad ORT el proyecto para ser aprobado.

**Requerimientos:** En esta carpeta está la información obtenida en las reuniones con expertos, lluvias de ideas y la definición de los requerimientos final.

**Revisiones:** Aquí se almacenan las presentaciones y la retroalimentación obtenida en las tres revisiones acordadas por la Universidad ORT.

## 11.9. Métricas de calidad

Previo al comienzo del desarrollo, se establecieron diferentes métricas para evaluar la gestión de calidad y tomar decisiones orientadas a cumplir con los objetivos. Las mismas se detallan a continuación.

### 11.9.1. Métricas del proceso

Se tomaron métricas del proceso con el fin de evaluar y realizar mejoras en cada una de las iteraciones asegurando la calidad ellas.

Dentro de las métricas de proceso se destacaron:

- Desviación entre el esfuerzo estimado y el realizado, pudiendo detectar errores a la hora de estimar las historias.
- Cumplimiento de objetivos por iteración, pudiendo detectar bloqueos o falta de compromiso.
- Total de horas trabajadas, pudiendo detectar si el objetivo planteado sobre la cantidad de horas está dando resultados.

Para tener información más detallada acerca de estas métricas mencionadas, ir al [capítulo 10.2](#).

### 11.9.2. Métricas del producto

Las métricas del producto se utilizaron con el fin de asegurar la calidad del mismo y validar que se cumplieran los atributos de calidad.

Dentro de las métricas que se utilizaron están:

#### **Tiempo de respuesta**

Si bien el equipo no hizo especial foco en el atributo de calidad referente a la performance, igualmente se realizaron acciones para contemplarlo esperando tener tiempos de respuestas en un rango aceptable.

Se definió que un rango aceptable es un tiempo menor a 0,5 segundos, a excepción de la operación que agrega lecturas a la base de datos, la cual tiene procesamiento y envío de alarmas a los dispositivos móviles por lo que se permite una flexibilidad de hasta 3 segundos promedio.

Los datos obtenidos por el sistema son:

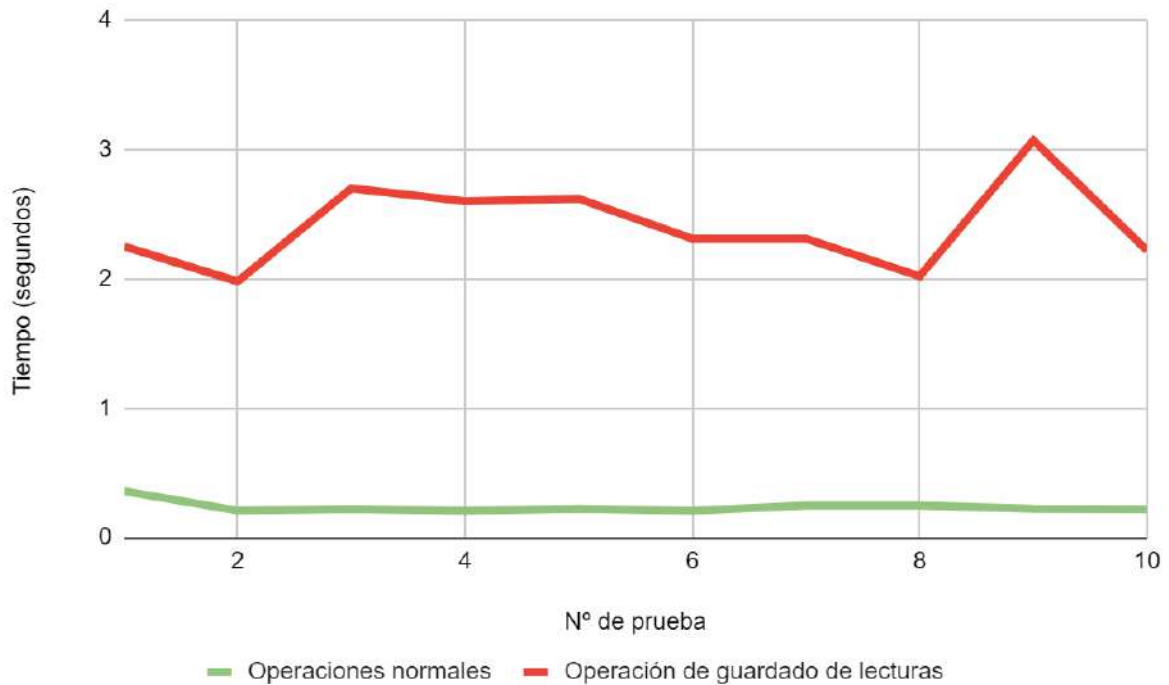


Ilustración 79: Tiempo de respuesta

- Operaciones normales: 0,24 segundos promedio
- Operación de guardado de lecturas con alerta: 2,41 segundos promedio

Se concluyó que los tiempos de procesamiento son aceptables y están dentro del promedio esperado.

### Heurísticas de Nielsen

Si bien el equipo resolvió no trabajar en el atributo de calidad referente a la usabilidad en relación a la arquitectura de *software*, si se decidió tener un foco en la experiencia de usuario, uno de los elementos que se tuvieron en cuenta a la hora de hacerlo, son las heurísticas de Nielsen para realizar un análisis a lo largo del proceso de desarrollo tanto en las aplicaciones móviles como la web.

Esta métrica permitió detectar oportunidades de mejora las cuales se registraron para posterior realización.

Se decidió dar un pequeño margen para llevar a cabo las mejoras propuestas que surgieron del análisis de estas heurísticas concluyendo en un total de 4 una vez finalizado el desarrollo. Para ver este análisis con mayor detalle, ver el [capítulo 8.3.1](#)

Hecho el análisis se concluyó que dichas aplicaciones son consistentes y cuentan con un nivel aceptable de usabilidad.

### **Satisfacción del cliente**

Con el objetivo de medir la satisfacción del cliente, se le entregó constantemente nuevas versiones con los avances del producto de forma de obtener retroalimentación y mejoras propuestas.

Estas mejoras fueron documentadas y realizadas, siendo para el equipo de alta prioridad. Se documentaron 19 mejoras por parte del cliente, con un cumplimiento del 95 % de las mismas.

Dado que el 5 % restante consiste en sugerencias que luego de discutir las se decidió no realizar, se puede concluir que la satisfacción del cliente es de un 100 %.

Para ver en detalle las mejoras sugeridas por el cliente ver el [anexo 8](#).

## 12. Dificultades del proyecto

A lo largo del proyecto hubo diferentes obstáculos o dificultades no planeadas, los cuales obligaron a implementar planes de contingencia y/o modificar ciertas características del trabajo.

Las dificultades más importantes que existieron fueron:

- Rechazo de la solicitud de financiación
- Necesidad de cambio de proveedor de dron
- Demora en la compra y construcción del dron final
- Instalación y configuración de placa al dron de prueba
- Cortocircuito de placa durante la instalación
- Necesidad de repuestos debido a la rotura de componentes
- Necesidad de buen clima para realizar pruebas

### 12.1. Rechazo de la solicitud de financiación

Desde el momento en el que el equipo tomó conocimiento y decidió presentarse a la financiación ofrecida por el CI<sup>2</sup> se sabía que existía la posibilidad de que la solicitud fuera rechazada, y por eso se realizaron algunas acciones preventivas como se especifica en el [capítulo 10.3](#).

Aunque se tenía planeado qué hacer en caso de que la financiación fuera rechazada, en el momento que se confirmó hubo que sentarse a negociar con la empresa Todosoft para que ellos se hagan cargo de la compra del dron.

La negociación incluyó ajustarse al presupuesto requerido por Todosoft y buscar proveedores de drones que se adapten a esto.

La investigación, los contactos con los proveedores y la presentación del dron elegido demandó un tiempo importante en esa etapa del proyecto.

## 12.2. Necesidad de cambio de proveedor de dron

En primera instancia el equipo había decidido que el dron a utilizar sería el DJI Matrice 200. Cuando la empresa Todosoft fue a realizar la compra, los diferentes proveedores les indicaron que además del precio publicado había que comprar varios añadidos para que el dron sea usable. Esto elevó considerablemente el costo del vehículo no tripulado y el mismo dejó de ser viable económicamente.

En ese momento se volvió a iniciar la fase de investigación de drones debido a que se necesitaba un reemplazo.

Con menos tiempo, con tareas bloqueadas por no tener el dron y con nuevos requerimientos de dinero se tuvo que buscar una rápida solución, invertir más horas y retrasar otras tareas.

## 12.3. Demora en la compra y construcción del dron final



Ilustración 80: Dron Cheerson CX-20

El dron final (Sky Track de Accsys) demoró mucho en venir (el proveedor demoró en enviar información y en la construcción del dron) y como se sabía que la placa ArduPilot es utilizada por muchos fabricantes de drones, se compró un dron Cheerson CX-20, de menor tamaño que el definitivo pero que sirvió para realizar el desarrollo y las pruebas.

Con este vehículo no tripulado se pudo desarrollar el 100 % de las funcionalidades de control, ruteo y seguridad del dron.

Además del dron el equipo compró una placa ArduPilot y realizó la instalación al mismo. Hubo que realizar también esta compra debido a que el Cheerson CX-20 por defecto viene con una placa diferente y no era lo que se necesitaba.

En Uruguay no se vende esta placa, por lo que hubo que comprarla en Estados Unidos y esperar a que la misma llegue al país.

Al contar con un dron con la placa madre de ArduPilot, se pudo realizar pruebas de vuelo y programación tal como si fuera el dron final.

Como el código es desarrollado para la placa madre, el *hardware* del dron es indiferente para propósito de este proyecto. Acá se le sacó provecho a la ventaja de haber elegido un ecosistema de drones genérico y no haberse limitado a trabajar con un proveedor específico.

## 12.4. Instalación y configuración de placa al dron de prueba

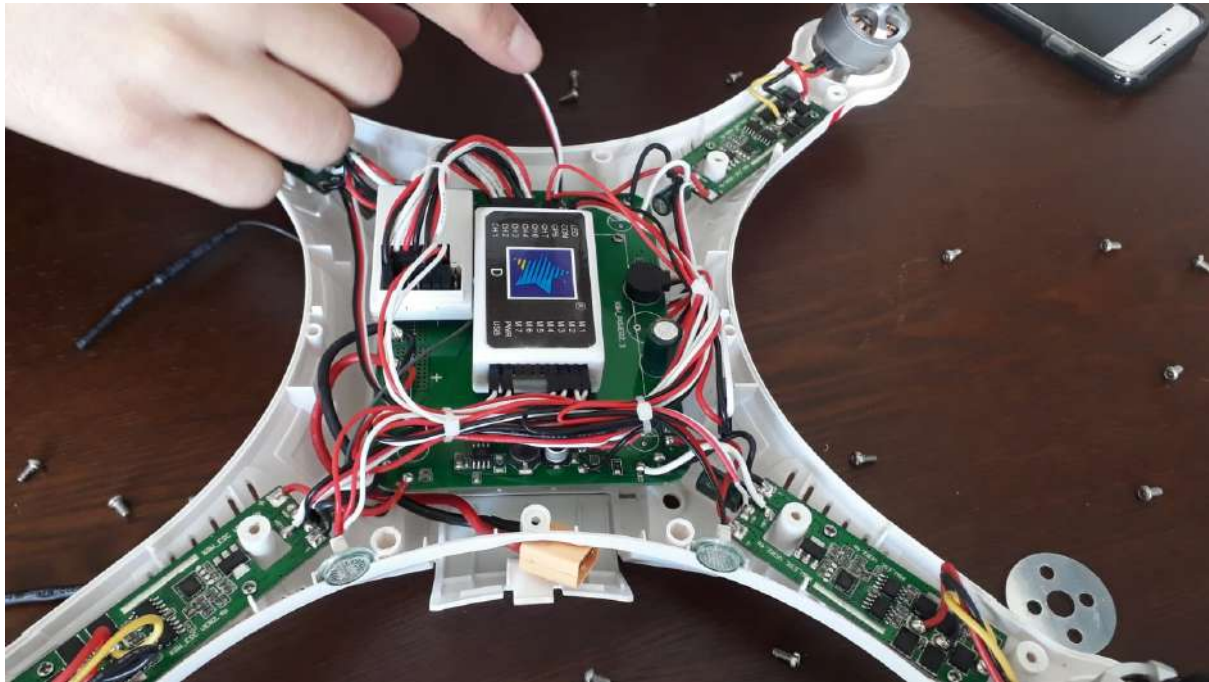


Ilustración 81: Momento en el que el equipo desarmaba el dron Cheerson CX-20 para montarle la placa ArduPilot

El dron que se compró para realizar pruebas, Cheerson CX-20, vino por defecto con una placa madre fabricada por la propia empresa Cheerson, por lo que el equipo tuvo que realizar la compra de una placa ArduPilot y montarla en el dron. Esto fue algo desconocido hasta luego de realizada la compra, debido a la poca información que se encontró previamente.

El tener que montar la placa demandó un trabajo extra que no se había planificado, y sobre todo muy riesgoso debido a que el equipo no contaba con los conocimientos electrónicos necesarios para hacerlo.

Para disminuir los riesgos se buscó ayuda en la Universidad; se dialogó con varios profesores de la Cátedra de Física. Los docentes mostraron voluntad e intentaron ayudar, pero al desconocer del tema no fue suficiente, por lo que el montaje tuvo que hacerlo el propio equipo.

## 12.5. Cortocircuito de placa durante la instalación

Lamentablemente, mientras se intentaba montar la placa al dron, hubo un cortocircuito que la quemó. En un principio no se sabía qué le había pasado a la placa, hasta que utilizando un tester para medir el voltaje se pudo comprobar que la placa se había quemado.

Hubo que comprar una nueva placa en Estados Unidos, lo que supuso una nueva demora en el proyecto.

Cuando se decidió comprar el dron Sky Track, se hizo teniendo en cuenta que al traer la placa y el lector de RFID montado, no se iban a tener que hacer instalaciones eléctricas, pero al demorarse su venida obligó a hacerlas de todas formas.

## 12.6. Necesidad de repuestos debido a la rotura de componentes

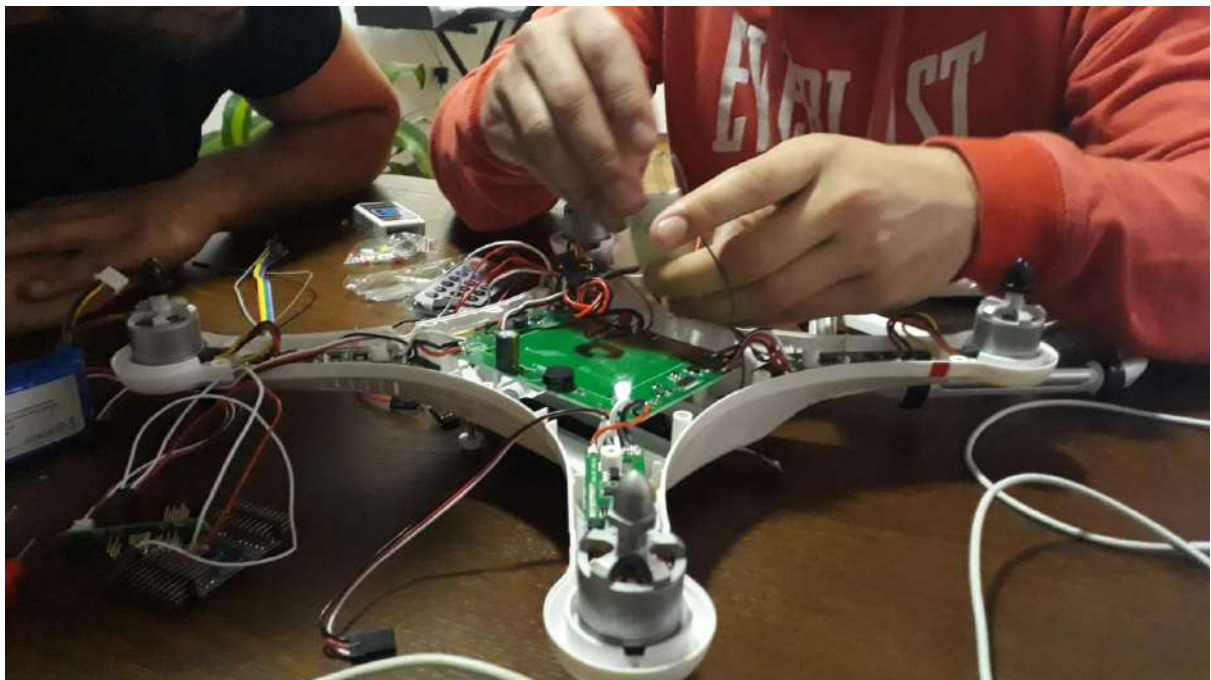


Ilustración 82: Equipo intentando reparar Cheerson CX-20

Una vez instalada la placa de ArduPilot en el dron, se comenzaron a realizar las pruebas volándolo.

Esta etapa trajo nuevos problemas, ya que por diferentes motivos se hicieron pruebas que en los simuladores funcionaban bien pero al volar el dron real no funcionaban de igual manera.

Que una prueba con el dron no funcione significaba que el aparato sufriera golpes y caídas, por lo que se comenzaron a sufrir roturas de diferentes piezas: se rompió una hélice y se quemó el motor de otra.

Comprar esas piezas en Uruguay no era posible ya que no se vendían, por lo que se tuvo que comprar un nuevo dron Cheerson CX-20 para continuar realizando las pruebas.

Analizándolo una vez terminado el proyecto, fue una muy buena decisión debido a que los problemas con las piezas continuaron ocurriendo, pero al tener 2 drones se logró intercambiarlas cuando fue necesario y se evitó de perder tiempo por esta razón.

## 12.7. Necesidad de buen clima para realizar pruebas



Ilustración 83: Equipo volando el dron en un espacio abierto

Por último, pero no menos importante, algo que era imprescindible para realizar las pruebas era contar con un buen clima debido a que con lluvia, mucho viento, poca luz o mucha humedad no era posible volar el dron.

Al ser un dron de pruebas y por ende menos robusto y resistente, muchas veces hubo que posponer las pruebas debido a las condiciones climáticas.

## 13. Conclusiones

En este capítulo se expone el estado actual del proyecto y las conclusiones generales, analizando el cumplimiento de los objetivos definidos al comienzo, las lecciones aprendidas y el trabajo a futuro.

### 13.1. Estado actual

Actualmente el sistema está en una versión beta cumpliendo con el alcance acordado con el cliente. Tanto la aplicación web como las aplicaciones móviles cuentan con todas las funcionalidades definidas.

Las pruebas de campo y la integración del dron se realizaron con componentes físicos prototipados, los cuales no son los que utilizará el cliente final, pero sí son 100 % compatibles con ellos. Esto se dio debido a los inconvenientes mencionados en el [capítulo 12](#). De igual forma, no debería tener un costo extra hacer el cambio de componentes por los comercializables; debería ser transparente tanto para los clientes finales como para la empresa Todosoft.

### 13.2. Conclusiones generales

La construcción del proyecto significó un gran desafío para el equipo, teniendo que trabajar en áreas en las que ningún integrante tenía conocimiento, como lo fue el área electrónica. Si bien varios de los integrantes se conocieron durante el transcurso de la carrera, también fue un desafío el trabajar en conjunto, algo que requirió un proceso de adaptación, tolerancia y respeto a la forma de trabajar de cada uno. La investigación fue una gran parte del proyecto y obligó al equipo a recurrir a expertos en diferentes áreas para poder comprender bien los problemas que se presentaron.

En cuanto a la gestión del proyecto, el equipo no tuvo un buen comienzo debido a la falta de experiencia en el área de gestión, pero gracias a la ayuda que brindó tanto la tutora Mariel Feder, como los revisores asignados por la Universidad ORT, se pudo realizar una corrección a tiempo, pudiendo culminar con un final satisfactorio.

En cuanto a los objetivos planificados al comenzar el proyecto, se puede decir que los mismos fueron alcanzados con éxito. La experiencia y el conocimiento adquirido durante la carrera, permitió que se superaran todos los desafíos presentados, e incluso desafíos que estuvieron por fuera del área de conocimiento del equipo.

En cuanto al producto, se puede decir que se desarrolló un producto innovador, que tiene mucho potencial con mucho interés por parte de las personas pertenecientes al rubro ganadero. Permitted que el equipo conociera y formara vínculos con personas externas al rubro tecnológico. En cuanto a la solución, se creó una solución escalable gracias a las decisiones tomadas, que brindará bajo costo a la empresa Todosoft para continuar con el desarrollo.

Como conclusión final, el equipo se sintió motivado durante todo el proceso, superó ampliamente las expectativas en cuanto al conocimiento y experiencia adquirida, y se siente con ganas de ser tenido en cuenta por la empresa Todosoft para continuar con el desarrollo del producto y poder ver el producto final funcionando. Esta motivación no viene solo por el lado del aprendizaje que los integrantes se llevan, sino también por el aporte que puede hacer un producto como este a la sociedad.

### 13.3. Lecciones aprendidas

A lo largo del proyecto, el equipo adquirió muchos conocimientos y habilidades pero también se cometieron errores los cuales se fueron resolviendo a lo largo del mismo. Es por eso que las lecciones aprendidas vienen tanto de los resultados positivos como de los negativos.

Como primer desafío, el trabajar en equipo resultó clave para lograr que el proyecto fuera exitoso. Por este lado se adquirió un gran aprendizaje para todos los integrantes pudiendo resolver conflictos o asperezas generadas en el transcurso del proyecto, logrando una gran tolerancia y apoyo para con el resto de los integrantes. En este punto cabe destacar el compromiso y la confianza de todos los integrantes del equipo.

En cuanto a la gestión del proyecto se puede decir que fue donde se cometieron más errores en la primera etapa del proyecto, pero se pudieron resolver rápidamente, lo cual deja un gran aprendizaje para futuros proyectos.

Por otro lado, otra gran lección aprendida fue la de gestionar los riesgos. En este punto si bien se cometieron errores, los mismos fueron descubiertos y solucionados a tiempo, y hoy el equipo es consciente de que si no se hubieran gestionado de forma correcta los riesgos, y aplicando los planes de acción y soluciones alternativas identificadas, el proyecto hubiera sido un fracaso, ya que incluso hasta la fecha de entrega de la tesis, no se pudo contar con los componentes que integrarán la solución final como se esperaba inicialmente. Es por esto que queda el aprendizaje de lo importante y valioso que es gestionar los riesgos adecuadamente en un proyecto.

Y por último queda el aprendizaje de las tecnologías, los componentes, y la importancia de realizar una buena etapa de investigación. El equipo se lleva conocimiento de electrónica, de nuevas tecnologías, de integración *hardware-software*, drones, entre muchos otros.

#### 13.4. Próximos pasos

- En los próximos pasos se deberá ensamblar el dron final, el cual integra un lector de RFID. Esto implicará realizar pruebas en campo y ajustes del mismo. En cuanto al desarrollo debería tener costo muy bajo ya que se trabajó con un dron 100 % compatible con el definitivo.
- Se deberán realizar pruebas con un lector de RFID activo que brinda un mayor alcance. Actualmente se realizaron pruebas con un lector pasivo. Igualmente que con el dron, esto debería tener un costo mínimo.
- Para mejorar la calidad de la aplicación *backend*, se sugiere realizar pruebas unitarias para verificar la ausencia de errores que pudieran haberse omitido.
- Se podrán agregar nuevos componentes al sistema. Algunos de los que mencionaron los expertos en las entrevistas fueron:
  - Sensores para la humedad de la tierra.
  - Fotografiar a los animales para un posterior análisis.
  - Sensores para detectar si un bebedero está vacío.
- Desarrollar una funcionalidad para la detección de jauría de perros. Por las investigaciones realizadas, este problema se podría resolver con reconocimiento de imágenes.

- El cliente planteó la idea de poder extender el sistema a lecturas de insumos en depósitos, para esto se deben realizar pequeñas modificaciones a nivel de *frontend*.
- Y por último, el sistema queda libre para ser extendido a cualquier otra funcionalidad.

## 14. Referencias bibliográficas

[1] Wikipedia, “Abigeato” [Online]. Available: <https://es.wikipedia.org/wiki/Abigeato>.

Accessed on: Mar., 01, 2020.

[2] Wikipedia, “Telefonía móvil 3G” [Online]. Available:

[https://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_3G](https://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_3G). Accessed on: Mar., 01, 2020.

[3] Flimper, “¿Qué es una API y para qué se usan?” [Online]. Available:

<https://www.flimper.com/blog/es/que-es-una-api-y-para-que-se-usan-en-flimper>. Accessed on Feb., 25, 2020.

[4] Wikipedia, “Arduino” [Online]. Available: <https://es.wikipedia.org/wiki/Arduino>.

Accessed on: Mar., 01, 2020.

[5] Wikipedia, “Arquitectura ARM” [Online]. Available:

[https://es.wikipedia.org/wiki/Arquitectura\\_ARM](https://es.wikipedia.org/wiki/Arquitectura_ARM). Accessed on: Mar., 01, 2020.

[6] Wikipedia, “Bash” [Online]. Available: <https://es.wikipedia.org/wiki/Bash>. Accessed on:

Feb., 19, 2020.

[7] Wikipedia, “Benchmarking” [Online]. Available:

<https://es.wikipedia.org/wiki/Benchmarking>. Accessed on: Feb., 19, 2020.

[8] Wikipedia, “Bluetooth” [Online]. Available: <https://es.wikipedia.org/wiki/Bluetooth>.

Accessed on: Mar., 01, 2020.

[9] CII, “¿Quiénes somos?” [Online]. Available: <http://www.cii.uy/quienes-somos/>. Accessed

on: Feb., 23, 2020.

[10] Wikipedia, “CRUD” [Online]. Available: <https://es.wikipedia.org/wiki/CRUD>. Accessed

on Feb., 25, 2020.

[11] Wikipedia, “DevOps” [Online]. Available: <https://es.wikipedia.org/wiki/DevOps>.

Accessed on: Mar., 01, 2020.

- [12] Wikipedia, “DynDNS” [Online]. Available: <https://es.wikipedia.org/wiki/DynDNS>. Accessed on: Mar., 01, 2020.
- [13] Wikipedia, “Ethernet” [Online]. Available: <https://es.wikipedia.org/wiki/Ethernet>. Accessed on: Mar., 01, 2020.
- [14] Expo [Online]. Available: <https://expo.io/>. Accessed on: Mar., 01, 2020.
- [15] Wikipedia, “Fail-safe” [Online]. Available: <https://en.wikipedia.org/wiki/Fail-safe>. Accessed on: Mar., 01, 2020.
- [16] Wikipedia, “Firmware” [Online]. Available: <https://es.wikipedia.org/wiki/Firmware>. Accessed on: Feb., 22, 2020.
- [17] Wikipedia, “Pilotaje con visión remota” [Online]. Available: [https://es.wikipedia.org/wiki/Pilotaje\\_con\\_visi%C3%B3n\\_remota](https://es.wikipedia.org/wiki/Pilotaje_con_visi%C3%B3n_remota). Accessed on: Mar., 01, 2020.
- [18] Wikipedia, “Framework” [Online]. Available: <https://es.wikipedia.org/wiki/Framework>. Accessed on: Feb., 22, 2020.
- [19] Wikipedia, “GPIO” [Online]. Available: <https://es.wikipedia.org/wiki/GPIO>. Accessed on: Mar., 01, 2020.
- [20] Wikipedia, “GPS” [Online]. Available: <https://es.wikipedia.org/wiki/GPS>. Accessed on: Mar., 01, 2020.
- [21] Wikipedia, “GRASP” [Online]. Available: <https://es.wikipedia.org/wiki/GRASP>. Accessed on: Mar., 01, 2020.
- [22] Wikipedia, “Alta definición” [Online]. Available: [https://es.wikipedia.org/wiki/Alta\\_definici%C3%B3n](https://es.wikipedia.org/wiki/Alta_definici%C3%B3n). Accessed on: Mar., 01, 2020.
- [23] Ecured, “IDE de Programación” [Online]. Available: [https://www.ecured.cu/IDE\\_de\\_Programaci%C3%B3n](https://www.ecured.cu/IDE_de_Programaci%C3%B3n). Accessed on: Mar., 01, 2020.

- [24] Comunidad Leroy Merlin, “Estándar IP43” [Online]. Available: <https://comunidad.leroymerlin.es/t5/Bricopedia-Iluminaci%C3%B3n-y/Qu%C3%A9-es-el-est%C3%A1ndar-IP43-de-un-LED/ta-p/11278>. Accessed on: Mar., 01, 2020.
- [25] Wikipedia, “.ipa” [Online]. Available: <https://es.wikipedia.org/wiki/.ipa>. Accessed on: Mar., 01, 2020.
- [26] Wikipedia, “JavaScript” [Online]. Available: <https://es.wikipedia.org/wiki/JavaScript>. Accessed on: Mar., 01, 2020.
- [27] Wikipedia, “LTE (telecomunicaciones)” [Online]. Available: [https://es.wikipedia.org/wiki/LTE\\_\(telecomunicaciones\)](https://es.wikipedia.org/wiki/LTE_(telecomunicaciones)). Accessed on: Mar., 01, 2020.
- [28] Wikipedia, “Material Design” [Online]. Available: [https://es.wikipedia.org/wiki/Material\\_Design](https://es.wikipedia.org/wiki/Material_Design). Accessed on: Mar., 01, 2020.
- [29] Wikipedia, “Mockup” [Online]. Available: <https://es.wikipedia.org/wiki/Mockup>. Accessed on: Mar., 01, 2020.
- [30] GitHub, “ngrok - Introspected tunnels to localhost” [Online]. Available: <https://github.com/inconshreveable/ngrok#what-is-ngrok>. Accessed on: Mar., 05, 2020.
- [31] Wikipedia, “No-IP” [Online]. Available: <https://en.wikipedia.org/wiki/No-IP>. Accessed on: Mar., 05, 2020.
- [32] Wikipedia, “Código abierto” [Online]. Available: [https://es.wikipedia.org/wiki/C%C3%B3digo\\_abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto). Accessed on: Mar., 01, 2020.
- [33] Wikipedia, “Computación en la nube - plataforma como servicio” [Online]. Available: [https://es.wikipedia.org/wiki/Computaci%C3%B3n\\_en\\_la\\_nube#Plataforma\\_como\\_servicio](https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube#Plataforma_como_servicio). Accessed on Feb., 25, 2020.
- [34] Wikipedia, “Product breakdown structure” [Online]. Available: [https://en.wikipedia.org/wiki/Product\\_breakdown\\_structure](https://en.wikipedia.org/wiki/Product_breakdown_structure). Accessed on: Mar., 01, 2020.

[35] Wikipedia, “RFID” [Online]. Available: <https://es.wikipedia.org/wiki/RFID>. Accessed on Feb., 25, 2020.

[36] Wikipedia, “Secure Digital” [Online]. Available: [https://es.wikipedia.org/wiki/Secure\\_Digital](https://es.wikipedia.org/wiki/Secure_Digital). Accessed on Feb., 25, 2020.

[37] Wikipedia, “Script” [Online]. Available: <https://es.wikipedia.org/wiki/Script>. Accessed on Feb., 25, 2020.

[38] Wikipedia, “SOLID” [Online]. Available: <https://es.wikipedia.org/wiki/SOLID>. Accessed on: Mar., 01, 2020.

[39] Wikipedia, “Telemetría” [Online]. Available: <https://es.wikipedia.org/wiki/Telemetría>. Accessed on Feb., 29, 2020.

[40] Wikipedia, “TypeScript” [Online]. Available: <https://es.wikipedia.org/wiki/TypeScript>. Accessed on: Mar., 01, 2020.

[41] Wikipedia, “Unix” [Online]. Available: <https://es.wikipedia.org/wiki/Unix>. Accessed on: Mar., 01, 2020.

[42] Wikipedia, “Interfaz de usuario” [Online]. Available: [https://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario). Accessed on Feb., 25, 2020.

[43] Wikipedia, “Waypoint” [Online]. Available: <https://es.wikipedia.org/wiki/Waypoint>. Accessed on Mar., 01, 2020.

[44] CUTI [Online]. Available: <https://www.cuti.org.uy/portada>. Accessed on Mar., 08, 2020.

[45] Búsqueda, “Ruralistas advierten aumento del delito de abigeato, que genera pérdidas de millones de dólares” [Online]. Available: <https://www.busqueda.com.uy/nota/ruralistas-advierten-aumento-del-delito-de-abigeato-que-genera-perdidas-de-millones-de-dolares>. Accessed on: Mar., 02, 2019.

[46] El Observador, “Casos de abigeato consolidaron crecimiento en 2015” [Online]. Available:

<https://www.elobservador.com.uy/nota/casos-de-abigeato-consolidaron-crecimiento-en-2015-2016531500>. Accessed on: Abr., 02, 2019.

[47] Toptal, “Los drones comerciales están revolucionando las operaciones comerciales”

[Online]. Available:

<https://www.toptal.com/finance/market-research-analysts/los-drones-comerciales-estan-revolucionando-las-operaciones-comerciales>. Accessed on: Feb., 25, 2020.

[48] ProyectosAgiles, “Ejecución de la iteración (Sprint)” [Online]. Available:

<https://proyectosagiles.org/ejecucion-iteracion-sprint/>. Accessed on Feb., 24, 2020.

[49] Proyectos Ágiles, “Scrum” [Online]. Available: <https://proyectosagiles.org/que-es-scrum>.

Accessed on: Mar., 04, 2020.

[50] Agileando, “Artefactos” [Online]. Available:

<http://agileando.com/que-son-los-artefactos-de-scrum>. Accessed on: Mar., 04, 2020.

[51] SmartFarm [Online]. Available: <https://www.smartfarm.ag/>. Accessed on Feb., 25, 2020.

[52] G&D Argentina [Online]. Available: <https://www.gddevelopers.com/>. Accessed on Feb., 25, 2020.

[53] ACGDrone [Online]. Available: <https://www.acgdrone.com/>. Accessed on Feb., 25,

[54] DJI, “Inspire 2” [Online]. Available: <https://www.dji.com/inspire-2/info#specs>. Accessed on: Jun., 09, 2019. 2020.

[55] MundoDrone, “Inspire 2” [Online]. Available:

<http://www.mundodrone.com.uy/sitio/img/inspire-two-detalles.pdf>. Accessed on: Jun., 10, 2019.

[56] Developer DJI, “Onboard SDK” [Online]. Available:

<https://developer.dji.com/products/#!/onboard>. Accessed on: Jun., 09, 2019.

[57] MundoDrone, “DJI Mavic 2 Enterprise” [Online]. Available:

<http://mundodrone.com.uy/sitio/index.php/producto/dji-mavic-2-enterprise/>. Accessed on: Jun., 09, 2019.

- [58] Sigis, “Matrice 100” [Online]. Available: <http://www.sigis.com.ve/index.php/matrice-100>. Accessed on: Jun., 10, 2019.
- [59] DJI, “Mobile-SDK” [Online]. Available: <https://developer.dji.com/mobile-sdk/>. Accessed on: Jun., 09, 2019.
- [60] UAV Coach, “Top 100 Drone Companies to Watch in 2020” [Online]. Available: <https://uavcoach.com/drone-companies/>. Accessed on: Jun., 10, 2019.
- [61] Drones Baratos Caseros, “DJI Matrice 200 Review En Español” [Online]. Available: <https://www.dronesbaratoscaseros.com/dji-matrice-200-review-en-espanol/1220/>. Accessed on: Jun., 10, 2019.
- [62] Caddy geomatics, “DJI Matrice 200 m200” [Online]. Available: <https://www.caddy-geomatics.de/es/producto/dji-matrice-200-m200/>. Accessed on: Jun., 10, 2019.
- [63] GitHub, “ArduPilot” [Online]. Available: <https://github.com/ArduPilot/ardupilot>. Accessed on: Feb., 23, 2020.
- [64] ArduPilot, “dev” [Online]. Available: <http://ardupilot.org/dev/>. Accessed on: Feb., 23, 2020.
- [65] ArduPilot [Online]. Available: <https://ardupilot.org/>. Accessed on: Jun., 12, 2019.
- [66] Drone Casero, “Qué es ArduPilot” [Online]. Available: <http://dronecasero.blogspot.com/p/que-es-ardupilot.html>. Accessed on: Jun., 13, 2019.
- [67] MAVlink [Online]. Available: <https://mavlink.io/en/>. Accessed on: Feb., 22, 2020.
- [68] GitHub, “MissionPlanner” [Online]. Available: <https://github.com/ArduPilot/MissionPlanner>. Accessed on: Feb., 23, 2020.
- [69] QGroundControl [Online]. Available: <http://qgroundcontrol.com/>. Accessed on: Feb., 23, 2020.

- [70] GitHub, “apm\_planner” [Online]. Available: [https://github.com/ArduPilot/apm\\_planner](https://github.com/ArduPilot/apm_planner). Accessed on: Feb., 23, 2020.
- [71] Accsys [Online]. Available: <https://accsys.mx/index.php/en/>. Accessed on: Jul., 03, 2019.
- [72] Google Cloud Vision [Online]. Available: <https://cloud.google.com/vision>. Accessed on: Feb., 23, 2020.
- [73] Google Cloud Vision, “Installing the Client Libraries” [Online]. Available: <https://cloud.google.com/vision/docs/libraries#client-libraries-install-nodejs>. Accessed on: Feb., 23, 2020.
- [74] Catastro, “Catalogodatos” [Online]. Available: <https://catalogodatos.gub.uy/dataset/direccion-nacional-de-catastro-dxf-del-parcelario-urbano-y-rural>. Accessed on Mar., 01, 2020.
- [75] ArcGis [Online]. Available: <https://www.arcgis.com/index.html>. Accessed on Mar., 01, 2020.
- [76] P. Clements, *Documenting Software Architectures: Views and Beyond*, Westford, Massachusetts, USA: Pearson Education, 2011.
- [77] Modificabilidad, “Software architecture in practice” [Online]. Available: <http://etutorials.org/Programming/Software+architecture+in+practice,+second+edition/Part+Two+Creating+an+Architecture/Chapter+5.+Achieving+Qualities/5.3+Modifiability+Tactics/> Accessed on: Mar., 04, 2020.
- [78] Medium, “Los 10 patrones comunes de arquitectura de software” [Online]. Available: <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>. Accessed on Feb., 25, 2020.
- [79] Codingame, “React higher order components” [Online]. Available: <https://www.codingame.com/playgrounds/8595/reactjs-higher-order-components-tutorial>. Accessed on Feb., 25, 2020.

- [80] Confiabilidad, “Software architecture in practice” [Online]. Available: <http://etutorials.org/Programming/Software+architecture+in+practice,+second+edition/Part+Two+Creating+an+Architecture/Chapter+5.+Achieving+Qualities/5.2+Availability+Tactics/> Accessed on: Mar., 04, 2020.
- [81] Heroku [Online]. Available: <https://www.heroku.com/>. Accessed on Feb., 25, 2020.
- [82] Wikipedia, “Programación defensiva” [Online]. Available: [https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_defensiva](https://es.wikipedia.org/wiki/Programaci%C3%B3n_defensiva). Accessed on Feb., 25, 2020.
- [83] Seguridad, “Software architecture in practice” [Online]. Available: <http://etutorials.org/Programming/Software+architecture+in+practice,+second+edition/Part+Two+Creating+an+Architecture/Chapter+5.+Achieving+Qualities/5.5+Security+Tactics/> Accessed on: Mar., 04, 2020.
- [84] Performance, “Software architecture in practice” [Online]. Available: <http://etutorials.org/Programming/Software+architecture+in+practice,+second+edition/Part+Two+Creating+an+Architecture/Chapter+5.+Achieving+Qualities/5.4+Performance+Tactics/> Accessed on: Mar., 04, 2020.
- [85] Testabilidad, “Software architecture in practice” [Online]. Available: <http://etutorials.org/Programming/Software+architecture+in+practice,+second+edition/Part+Two+Creating+an+Architecture/Chapter+5.+Achieving+Qualities/5.6+Testability+Tactics/> Accessed on: Mar., 04, 2020.
- [86] L. Bass, P. Clements, R. Kazman, *Software Architecture in Practice*, 3rd. ed., New Jersey, USA: Pearson Education, 2012.
- [87] GitHub, “DroneKit-Python library for communicating with Drones via MAVLink” [Online]. Available: <https://github.com/dronekit/dronekit-python>. Accessed on: Feb., 23, 2020.
- [88] Atlassian, “Gitflow Workflow” [Online]. Available: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>. Accessed on Feb., 24, 2020.

- [89] Braintive, “10 reglas heurísticas de usabilidad de Jakob Nielsen” [Online]. Available: <http://www.braintive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/>. Accessed on Mar., 1, 2020.
- [90] Material Design [Online]. Available: <https://material.io/>. Accessed on Mar., 1, 2020.
- [91] Chien-Ming Tseng, Chi-Kin Chau, Khaled Elbassioni, and Majid Khonji, “Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones,” 1, 1, Article 1, January, 2017. [Online serial]. Available: <https://arxiv.org/pdf/1703.10049.pdf>.
- [92] “Crime Scene Search Patterns,” weebly.com. Available: <https://crimescenesearchpatterns.weebly.com/types-of-search-patterns.html>. [Accessed Feb., 25, 2020].
- [93] Alsagoff, S. N., “Optimal grid pattern model for search and rescue operation in dipterocarp forest research methodology,” 2011. [Online serial]. Available: [https://www.researchgate.net/publication/271423086\\_Optimal\\_grid\\_pattern\\_model\\_for\\_search\\_and\\_rescue\\_operation\\_in\\_dipterocarp\\_forest\\_research\\_methodology](https://www.researchgate.net/publication/271423086_Optimal_grid_pattern_model_for_search_and_rescue_operation_in_dipterocarp_forest_research_methodology).
- [94] “A .NET GIS solution that is fast and reliable for the .NET platform.”, github.com. Available: <https://github.com/NetTopologySuite/NetTopologySuite>. Accessed Feb., 25, 2020.
- [95] “Spatial Data”, microsoft.com. Available: <https://docs.microsoft.com/en-us/ef/core/modeling/spatial>. Accessed Feb., 25, 2020.
- [96] Wikipedia, “Burn down chart” [Online]. Available: [https://es.wikipedia.org/wiki/Burn\\_down\\_chart](https://es.wikipedia.org/wiki/Burn_down_chart). Accessed on Feb., 29, 2020.
- [97] B. Klaus and P. Horn, *Guía de los FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (GUÍA DEL PMBOK)*. Newtown Square, Pensilvania, USA: Project Management Institute, Inc., 2013.
- [98] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. NJ, USA: Prentice Hall, 2008.

- [99] Microsoft, “C# Coding Conventions” [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>. Accessed on: Feb., 23, 2020.
- [100] ECMA International, “Standard ECMA-262” [Online]. Available: <https://www.ecma-international.org/publications/standards/Ecma-262.htm>. Accessed on Feb., 23, 2020.
- [101] “Normas específicas para la presentación de trabajos finales de carreras”, ort.edu.uy. Available: <https://fi.ort.edu.uy/innovaportal/file/30964/3/documento-302-et-26102018.pdf>. Accessed on Mar., 05, 2020.
- [102] “Hoja de verificación. Pautas de presentación de trabajos finales de carreras”, ort.edu.uy. Available: <https://fi.ort.edu.uy/fi/documentos/documento-303-biotecnologia.pdf>. Accessed on Mar., 05, 2020.
- [103] “Normas para el desarrollo de trabajos finales de carrera”, ort.edu.uy. Available: <http://www.ort.edu.uy/varios/pdf/documento304.pdf>. Accessed on Mar., 05, 2020.
- [104] “Orientación para títulos, resúmenes, abstracts e informes de corrección de trabajos finales de carrera”, ort.edu.uy. Available: <http://www.ort.edu.uy/varios/pdf/documento306.pdf>. Accessed on Mar., 05, 2020.
- [105] ArduPilot Discourse [Online]. Available: <https://discuss.ardupilot.org/>. Accessed on Mar., 05, 2020.
- [106] Trello [Online]. Available: <https://trello.com/>. Accessed on Mar., 05, 2020.

## **Anexo 1: Investigación de RFID**

### **¿Qué es RFID?**

RFID son las siglas inglesas de *Radio Frequency IDentification* lo que en español significa Identificación por radiofrecuencia.

La idea es que el dispositivo de lectura (antena o *handheld*) emite una sonda de radiofrecuencia y los chips (*tags*) le responden, logrando así la identificación del mismo.

### **Propósito**

El propósito fundamental de la tecnología RFID es identificar mediante un lector, sin contacto y a distancia, una tarjeta o etiqueta (*tag*) portada por una persona, un animal, un vehículo en movimiento o cualquier producto que se encuentra en un almacén o en una cadena de producción automatizada.

### **Tags**

Existen diferentes tipos de chip

- Algunos con más alcance
- Otros más pequeños (por ende más baratos)
- Otros de materiales todavía más baratos (papel)
- Otros específicos para trabajar sobre materiales metálicos

El código de identificación que contienen es único y puede ser personalizado durante la fabricación de la etiqueta.

### **¿Cómo son los tags?**

Los *tags* incluyen en su interior un pequeño chip y una antena impresa o bobinada para comunicarse con el lector. El chip tiene grabado un número (ID) de serie único que lo identifica entre los demás, y puede disponer de una pequeña memoria para guardar datos, que los lectores son capaces de leer y escribir.

## Tipos de tags

- Pasivos: No necesitan alimentación interna, toman la energía de la propia emisión de las antenas y solo se activan cuando se encuentran en el campo de cobertura de la misma
- Activos: Utilizan alimentación propia de una pequeña batería y por ende pueden comunicarse con el lector a una distancia mucho mayor y procesar una cantidad de datos superior

## Alcance de los tags

Banda de frecuencias	Descripción	Rango	Ejemplo
125 kHz	LF (Baja Frecuencia)	Hasta 50 cm.	Garrafas de la empresa Megal
13,56 MHz	HF (Alta Frecuencia)	De 8 cm.	Tarjetas de proximidad
400 MHz – 1.000 MHz	UHF (Ultra Alta Frecuencia)	De 3 a 10 m.	Telepeaje, parkings
2,45 GHz – 5,4 GHz	Microondas	Más de 10 m.	SkyGuard con tags activos

Tabla 9: Tabla de alcance de los tags

## Frecuencias

La banda de 125 Khz era utilizada por las tarjetas para los primeros lectores de proximidad. Lo bueno de estas frecuencias es el buen alcance conseguido para leer las tarjetas con lectores técnicamente poco complejos, y lo malo, la baja seguridad debido a la facilidad de copiar las tarjetas. Hoy en día prácticamente no se utilizan.

La frecuencia de 13,56 MHz es la utilizada por las modernas tarjetas de identificación, lo bueno de esta tecnología es la alta seguridad de las aplicaciones y la cantidad de información que se puede guardar en su memoria.

Las bandas 400 MHz - 1.000 MHz son los tags de alta frecuencia que se comercializan normalmente. Dependiendo del fabricante, tamaño y costo se puede obtener mejores capacidades de lectura en diferentes escenarios.

### **Frecuencias en Uruguay**

No todos los dispositivos de RFID son aceptados para trabajar en Uruguay. Cada país tiene unas bandas de radiofrecuencia habilitadas.

## Anexo 2: Registro de dedicación de horas por iteración

En base a los registros de las horas trackeadas fueron calculadas las métricas de esfuerzo presentadas en el capítulo de gestión. A continuación se presentan los registros.

Iteración	Desarrollo	Documentación	Calidad y pruebas	Capacitación	Definición de proceso	Dron	Diseño de arquitectura	Investigación	Requerimientos	Reuniones	Gestión	Total horas
1	0	8	0	0	10	0	0	12	4	20	7	68
2	0	3	0	0	0	0	0	10	1	10	0	24
3	0	10	0	0	2	0	0	0	13	22	15	63
4	0	6	0	0	0	0	3	0	0	10	7	26
5	0	0	0	0	0	0	0	3	14	2	10	33
6	10	18	0	0	0	0	8	1	16	37	15	104
7	43	0	0	0	0	0	5	3	7	16	3	80
8	5	6	6	6	0	0	0	22	8	10	4	60
9	20	0	0	0	0	12	0	0	0	10	0	60
10	55	8	0	0	0	8	0	0	0	12	8	81
11	52	0	0	16	0	9	0	0	0	11	0	87
12	29	70	0	0	0	2	0	0	0	23	0	124
13	108	3	0	0	0	11	0	3	0	32	4	160
14	82	0	0	0	0	31	0	1	0	10	4	107
15	52	3	8	0	0	35	0	0	0	10	0	108
16	74	3	0	0	0	3	0	0	0	8	32	120
17	80	0	0	0	0	0	0	0	1	10	0	83
18	45	0	25	0	0	19	0	0	0	8	0	87
19	80	82	0	0	0	5	0	0	0	0	0	167
20	0	174	0	0	0	0	0	0	0	0	0	174
<b>Total de horas</b>	<b>738</b>	<b>403</b>	<b>45</b>	<b>21</b>	<b>22</b>	<b>134</b>	<b>18</b>	<b>65</b>	<b>51</b>	<b>288</b>	<b>100</b>	<b>1.684</b>

Ilustración 84: Registros de dedicación de horas

### Anexo 3: Tabla de análisis de riesgos

<b>Id</b>	<b>Riesgo</b>	<b>Probabilidad (1-99)</b>	<b>Impacto (1-5)</b>	<b>Valor (1-5)</b>	<b>Plazo</b>	<b>Acción preventiva</b>	<b>Plan de contingencia</b>	<b>Disparador del plan de contingencia</b>
R1	No conseguir financiación para la compra de los drones, tags, y reader de RFID.	50,00 %	4	2	Inmediato	Se realizará una solicitud profesional y dentro de los parámetros correspondientes para poder obtener la financiación.	En caso de no conseguir financiación, los materiales serán proporcionados por la empresa.	Pasado el mes de junio si no tenemos novedades de la financiación se disparará el plan de contingencia.
R2	Que el lector de RFID junto con la altura y el movimiento del dron no obtenga los resultados esperados.	40,00 %	4	1,6	Mediano	Realizar investigaciones para saber que esta tecnología cumplirá satisfactoriamente con nuestros objetivos.	Se probarán otras tecnologías que se adapten mejor y se obtengan los resultados esperados.	Una vez comenzadas las pruebas, se dispondrá de 1 mes para obtener datos coherentes, y en caso de que no se consiga, se disparará el plan de contingencia.
R3	Que los requerimientos varíen mucho y atrasen el proyecto.	40,00 %	4	1,6	Mediano	Se buscará claridad en la definición de los requerimientos al iniciar el proyecto.	Se utilizará una metodología ágil para poder mitigar los riesgos de tener malos requerimientos o variaciones de los mismos	El plan de contingencia se disparará en conjunto con el desarrollo del proyecto desde el primer momento.
R4	Mala organización de los tiempos para cumplir con el proyecto logrando algo funcional y de calidad.	30,00 %	5	1,5	Mediano	Se llevará un estricto control sobre los tiempos, y los objetivos a cumplir.	Se reorganizarán la distribución de horas en el equipo para poder llegar a un producto funcional y de calidad.	En las reuniones de roadmap el equipo decidirá si disparar el plan de contingencia.
R5	Que se rompa el hardware.	30,00 %	5	1,5	Mediano	Se estudiará el <i>hardware</i> de forma que su utilización sea como el proveedor espera.	Se tendrá <i>hardware</i> de repuesto	En el momento que se detecte una falla en el hardware, de inmediato se disparará el plan

								de contingencia.
R6	Que las especificaciones de los drones y/o de las otras herramientas no se ajusten a futuros requerimientos.	20,00 %	5	1	Largo	Se realizará una investigación correspondiente para obtener el dron con las características que se adaptan mejor a nuestras necesidades.	Se intentarán adaptar los requerimientos a las herramientas, y en caso de no poderse conseguirán otras herramientas que se ajusten mejor.	En el momento que se detecte que las herramientas no se ajustan, se evaluará y se disparará el plan de contingencia.
R7	Dificultad para reunirnos con el experto	30,00 %	3	0,9	Mediano	Se coordinarán reuniones con anticipación para evitar la falta de disponibilidad.	Se ofrecerán alternativas, como reuniones online, telefónicas, etc, donde participen la mayoría en caso de que no puedan todos.	Dada la primera dificultad para reunirnos con el experto, se disparará el plan de contingencia.
R8	Tener el dron, tags y lector RFID para probar apenas empezamos a desarrollar.	30,00 %	3	0,9	Inmediato	Estaremos pendientes de los movimientos de financiación para la compra de los instrumentos.		
R9	Que el o los referentes de negocio no estén disponibles para relevar requerimientos y validarlos.	40,00 %	2	0,8	Mediano	Se pautarán reuniones con los referentes del negocio, de forma que estos se comprometerán a asistir.	Se consultará con otros referentes del negocio.	En caso de detectar la falta de disponibilidad se disparará de inmediato el plan de contingencia.
R10	Falta de comunicación en el equipo.	20,00 %	4	0,8	Mediano	Se planificarán reuniones frecuentes del equipo entero.	Se intentará hablar con el equipo para que no vuelva a ocurrir.	En caso de detectar la falta de disponibilidad se disparará de inmediato el plan de contingencia.
R11	Mala estimación de las tareas que genere	40,00 %	2	0,8	Mediano	Se utilizará una metodología ágil para poder mitigar los riesgos de tener malas	Se reestimarán las tareas y el equipo se esforzará más	En las reuniones de alineamiento se evaluará la opción de disparar el plan

	retrasos					estimaciones	para compensar el retraso.	de contingencia.
R12	Problema de alguno de los integrantes retrase el avance del proyecto.	20,00 %	3	0,6	Mediano	Se contemplarán horas para enfermedades o problemas que puedan ocurrir en el desarrollo del proyecto	Se espera que las horas contempladas sean suficientes para no retrasar el proyecto.	
R13	Desconocimiento de las tecnologías.	30,00 %	2	0,6	Mediano	Se contemplará un tiempo para la capacitación en determinadas tecnologías, y cada integrante trabajará con la que se sienta más a gusto.	Se realizará un esfuerzo mayor por parte del equipo para aprender sobre las tecnologías de manera más rápida.	En cuanto se detecte la falta de conocimiento por algún integrante del equipo, este disparará el plan de contingencia.
R14	Mala definición de los requerimientos.	30,00%	2	0,6	Mediano	Se realizarán reuniones con el equipo entero para definir los requerimientos de forma que el riesgo de equivocarse sea menor.	Se utilizará una metodología ágil para que en caso de tener una mala definición de requerimientos el impacto sea menor.	El plan de contingencia se disparará en conjunto con el desarrollo del proyecto desde el primer momento.
R15	Mala división de tareas.	30,00 %	2	0,6	Mediano	Se utilizará la experiencia previa de los integrantes del equipo para dividir las tareas.	Se planea una metodología de trabajo la cual se irá ajustando a medida que avancemos. Por lo tanto podremos ajustar una mala división de tareas en el correr del desarrollo	El plan de contingencia se disparará en conjunto con el desarrollo del proyecto desde el primer momento.
R16	Conseguir el campo para probar, y que esté disponible cuando necesitemos probar	40,00 %	1	0,4	Mediano	Se contactará al campo previamente para coordinar la disponibilidad.	Se tendrá el contacto de más de un campo para tener varias opciones en el caso de que alguno no esté disponible.	Se evaluará la urgencia para realizar las pruebas, y en caso de ser necesario se disparará el plan de contingencia.

R17	Falta de compromiso de alguno de los integrantes retrase el avance del proyecto.	10,00 %	3	0,3	Mediano		Se utilizarán herramientas para gestionar tareas y horas, de forma que el trabajo sea parejo, y en caso de que no lo sea poder tomar acciones.	El plan de contingencia se disparará en conjunto con el desarrollo del proyecto desde el primer momento.
-----	--	---------	---	-----	---------	--	--	--

Tabla 10: Tabla de riesgos

## Anexo 4: Documentos de financiación entregados a CI<sup>2</sup>



Formulario para la presentación de Proyectos Fin de Carrera - 2019



1

Ilustración 85: Formulario de financiación

<b>INFORMACIÓN SOBRE LOS POSTULANTES</b>	<b>3</b>
Datos del llamado	3
Resumen publicable del proyecto:	4
Personas involucradas	5
Empresa/Institución Contraparte	11
<b>ESPECIFICACIONES DEL PROYECTO</b>	<b>13</b>
Descripción del Problema a solucionar	13
Descripción de la solución propuesta	14
Descripción del público/área objetivo	14
<b>PLAN DE TRABAJO</b>	<b>16</b>
<b>PRESUPUESTO</b>	<b>20</b>
PRESUPUESTO GENERAL DEL PROYECTO	20
CRONOGRAMA DE EJECUCIÓN FINANCIERA	21
<b>CRONOGRAMA DE EJECUCIÓN DE ACTIVIDADES</b>	<b>22</b>
<b>DOCUMENTOS ADJUNTOS</b>	<b>23</b>

**INFORMACIÓN SOBRE LOS POSTULANTES**
**1. Datos del llamado**

UNIVERSIDAD	Universidad ORT
CONTRAPARTE	TodoSoft Uruguay

TÍTULO DEL PROYECTO	Trazabilidad de animales con tecnología RFID y Drones
---------------------	---

SECTOR PRIORITARIO	Sector agrícola
--------------------	-----------------

ÁREA DE CONOCIMIENTO	Área tecnológica
----------------------	------------------

PROBLEMA A RESOLVER (describa en menos de 100 palabras el problema que propone resolver el proyecto)	<p>Actualmente los trabajadores del campo dedican muchas horas a tareas manuales de control y gestión de ganado que incluso muchas veces se ven interrumpidas o postergadas por situaciones climatológicas. Como consecuencia muchos animales salen de los límites de su campo, lo que genera que a veces se pierdan, queden atrapados en zonas de cañadas, inundaciones o lugares peligrosos, e incluso a veces hasta sean robados.</p> <p>Al mismo tiempo, es muy difícil tener un control de qué zonas de su campo tienen suelo propicio para el pastoreo y cuáles no, qué zonas están inundadas, cuáles son focos de incendio, o si en general son peligrosas.</p> <p>Si un animal desaparece es complicado detectarlo y más aún saber cómo buscarlo.</p> <p>Parte de los terrenos son difíciles de transitar y bajo determinadas situaciones climatológicas es hasta peligroso hacerlo.</p>
DURACIÓN DEL PROYECTO en meses)	12 meses

**Nota:** Tiene que controlar que la duración No supere los 12 meses.

FECHA PREVISTA DE INICIO:	05/2019
---------------------------	---------

DEDICACIÓN HORARIA SEMANAL (estimada por alumno)	15 horas
--	----------

## 2. Resumen publicable del proyecto:

*Exponer en un máximo de 300 palabras, los objetivos y aspectos más relevantes del proyecto. Se debe considerar que el texto de este ítem podrá ser utilizado en documentación pública y de difusión, a diferencia del resto del contenido del presente documento, que es de carácter confidencial.*

El sistema se basa en la gestión totalmente remota y automatizada del inventario de ganado en grandes hectáreas. Constará de una aplicación Android, una aplicación iOS, una aplicación web, un dron con lector de RFID y etiquetas de RFID que irán colocadas en el ganado.

Desde el sistema se crearán las rutas que deba recorrer el dron y se elegirá en qué horarios se realizarán las mismas. El dron volará por dicha ruta e irá captando las señales de RFID del ganado gracias a los chips que el mismo poseerá.

Una vez finalizado el inventario, el dron regresará a su base y sincronizará la información. Con la información sincronizada se realizará y mostrará el conteo y mapa de calor del ganado, se mostrará estadísticas, posición GPS de cada animal, fotografías a lo largo del trayecto, mediciones de altura de los pastizales a través de un sensor láser.

El usuario del sistema manejará todo de manera remota, tanto en la antesala del recorrido como en el posterior análisis.

Se podrán configurar diversas alertas al notar faltante de ganado, chips con poca batería (las baterías de los chips duran un aproximado de 9 años), ganado que hace tiempo no es leído, ganado en posiciones GPS sospechosas, etcétera.

El sistema permitirá controlar los robos y pérdida de ganado, el largo de los pastizales, la calidad del césped (gracias al mapa de calor). Se pueden agregar otras funcionalidades, sensores, reportes, etcétera. El sistema se basa en la gestión totalmente remota del inventario de ganado en grandes hectáreas. Constará de una aplicación Android, una aplicación iOS, una página web, un dron con lector de RFID y etiquetas de RFID que irán colocadas en el ganado.

Desde el sistema se crearán las rutas que deba recorrer el dron y se elegirá en qué horarios se realizarán las mismas. El dron volará por dicha ruta e irá captando las señales de RFID del ganado gracias a los chips que el mismo poseerá.

Una vez finalizado el inventario, el dron regresará a su base y sincronizará la información. Con la información sincronizada se realizará y mostrará el conteo y mapa de calor del ganado, se mostrará estadísticas, posición GPS de cada animal, fotografías a lo largo del trayecto, mediciones de altura de los pastizales a través de un sensor láser.

El usuario del sistema manejará todo de manera remota, tanto en la antesala del recorrido como en el posterior análisis.

Se podrán configurar diversas alertas al notar faltante de ganado, chips con poca batería (las baterías de los chips duran un aproximado de 9 años), ganado que hace tiempo no es leído, ganado en posiciones GPS sospechosas, etcétera.

El sistema permitirá controlar los robos y pérdida de ganado, el largo de los pastizales, la calidad del césped (gracias al mapa de calor). Se pueden agregar otras funcionalidades, sensores, reportes, etcétera.

**3. Personas involucradas**

a. Estudiantes		
	Nombre	Rol en el proyecto
1.	Agustín Rodríguez	Desarrollador iOS
2.	Gastón Donadio	Desarrollador Android
3.	Victoria Rocha	Desarrollador Backend
4.	Dan Blanco	Arquitecto de Software
5.	Emiliano Rodríguez	Gerente de Proyecto

b. Tutores	
Tutor/a Facultad	Maríel Feder
Otros tutores (Si corresponde):	N/C

**a. Estudiantes**

(En caso de ser más de uno, copiar la planilla y completar con los datos de todos los integrantes del equipo)

NOMBRES:	Emiliano Javier		
APELLIDOS:	Rodríguez Naguil		
Cédula de Identidad - Nro.:	4.924.561-5	SEXO: (F/M)	M
FECHA DE NACIMIENTO:	03/12/1993		
PAÍS DE NACIMIENTO:	Uruguay	DEPARTAMENTO DE NACIMIENTO:	Montevideo
INSTITUCIÓN A LA QUE ASISTE:	TIPO (Pública, Privada o Mixta):	Privada	
	Grado de avance en la carrera de grado (Porcentaje de materias aprobadas):	88 %	
	Escolaridad:	91 %	
FACULTAD:	Universidad ORT		
DEPARTAMENTO DE RESIDENCIA	Montevideo		
CIUDAD DE RESIDENCIA:	Montevideo		
DIRECCIÓN DE CONTACTO	Ferrer Serra 2318		
TELÉFONO DE CONTACTO	099326468		
MAIL DE CONTACTO:	emirona07@gmail.com		

Vinculación a la temática propuesta (No más de 50 palabras) (En caso de contar con experiencia, describa bajo qué condiciones desarrolló la experiencia y si mantiene la vinculación en la actualidad.)	Ex trabajador de la empresa Todosoft Uruguay por 5 años y medio. En dicho período trabajó en multitud de proyectos basados en la tecnología RFID. Fue también participante de la investigación sobre drones para el desarrollo del proyecto.  Integrante del equipo que desarrollará el proyecto..
Desea recibir notificaciones del CII (Si/No)	Sí

NOMBRES:	Dan		
APELLIDOS :	Blanco Ochoa		
Cédula de Identidad - Nro.:	4.651.249-1	SEXO:	M (F/M)
FECHA DE NACIMIENTO:	01/04/1989		
PAÍS DE NACIMIENTO:	Uruguay	DEPARTAMENTO DE NACIMIENTO:	Montevideo
INSTITUCIÓN A LA QUE ASISTE:	TIPO (Pública, Privada o Mixta):	Privada	
	Grado de avance en la carrera de grado (Porcentaje de materias aprobadas):	84 %	
	Escolaridad:	86 %	
FACULTAD:	Universidad ORT		
DEPARTAMENTO DE RESIDENCIA	Montevideo		
CIUDAD DE RESIDENCIA:	Montevideo		
DIRECCIÓN DE CONTACTO	Av. Brasil 2986 ap. 206		
TELÉFONO DE CONTACTO	+598 99 750 526		
MAIL DE CONTACTO:	dan.blanco8a@gmail.com		
Vinculación a la temática propuesta (No	Integrante del equipo que desarrollará el proyecto.		

más de 50 palabras) (En caso de contar con experiencia, describa bajo qué condiciones desarrolló la experiencia y si mantiene la vinculación en la actualidad.)	
Desea recibir notificaciones del CII (Si/No)	No

NOMBRES:	Victoria		
APELLIDOS:	Rocha		
Cédula de Identidad - Nro.:	4.608.018-7	SEXO:	F (F/M)
FECHA DE NACIMIENTO:	06/01/1991		
PAÍS DE NACIMIENTO:	Uruguay	DEPARTAMENTO DE NACIMIENTO:	Montevideo
INSTITUCIÓN A LA QUE ASISTE:	TIPO (Pública, Privada o Mixta):	Privada	
	Grado de avance en la carrera de grado (Porcentaje de materias aprobadas):	89 %	
	Escolaridad:	93 %	
FACULTAD:	Universidad ORT		
DEPARTAMENTO DE RESIDENCIA	Montevideo		
CIUDAD DE RESIDENCIA:	Montevideo		
DIRECCIÓN DE CONTACTO	Pedro Campbell 1407		
TELÉFONO DE CONTACTO	+598 91 037 491		
MAIL DE CONTACTO:	victoria.rocha.lapachian@gmail.com		

Vinculación a la temática propuesta (No más de 50 palabras) (En caso de contar con experiencia, describa bajo qué condiciones desarrolló la experiencia y si mantiene la vinculación en la actualidad.)	Integrante del equipo que desarrollará el proyecto.
Desea recibir notificaciones del CII (Si/No)	No

NOMBRES:	Federico Agustín		
APELLIDOS:	Rodríguez Giuliani		
Cédula de Identidad - Nro.:	4.784.742-5	SEXO:	M (F/M)
FECHA DE NACIMIENTO:	03/01/1991		
PAÍS DE NACIMIENTO:	Uruguay	DEPARTAMENTO DE NACIMIENTO:	Montevideo
INSTITUCIÓN A LA QUE ASISTE:	TIPO (Pública, Privada o Mixta):	Privada	
	Grado de avance en la carrera de grado (Porcentaje de materias aprobadas):	73 %	
	Escolaridad:	82 %	
FACULTAD:	Facultad de Ingeniería, Universidad ORT Uruguay		
DEPARTAMENTO DE RESIDENCIA:	Montevideo		
CIUDAD DE RESIDENCIA:	Montevideo		
DIRECCIÓN DE CONTACTO	Defensa 1827 apto 322		
TELÉFONO DE CONTACTO	+598 96 052 713		

MAIL DE CONTACTO:	me@agurodriguez.net
Vinculación a la temática propuesta (No más de 50 palabras) (En caso de contar con experiencia, describa bajo qué condiciones desarrolló la experiencia y si mantiene la vinculación en la actualidad.)	Integrante del equipo que llevará adelante el proyecto.
Desea recibir notificaciones del CII (Si/No)	Sí

NOMBRES:	Gastón		
APELLIDOS:	Donadio Aprile		
Cédula de Identidad - Nro.:	4.440.748-B	SEXO:	M (F/M)
FECHA DE NACIMIENTO:	04/02/1992		
PAÍS DE NACIMIENTO:	Uruguay	DEPARTAMENTO DE NACIMIENTO:	Montevideo
INSTITUCIÓN A LA QUE ASISTE:	TIPO (Pública, Privada o Mixta):	Privada	
	Grado de avance en la carrera de grado (Porcentaje de materias aprobadas):	91 %	
	Escolaridad:	82 %	
FACULTAD:	Universidad ORT		
DEPARTAMENTO DE RESIDENCIA:	Montevideo		
CIUDAD DE RESIDENCIA:	Montevideo		
DIRECCIÓN DE CONTACTO	Orinoco 5315 apto 103		

TELÉFONO DE CONTACTO	094181012
MAIL DE CONTACTO:	tongadonadio@gmail.com
Vinculación a la temática propuesta (No más de 50 palabras) (En caso de contar con experiencia, describa bajo qué condiciones desarrolló la experiencia y si mantiene la vinculación en la actualidad.)	Integrante del equipo que desarrollará el proyecto.
Desea recibir notificaciones del CII (Si/No)	Sí

**b. Tutor y otros participantes (contraparte del proyecto en la empresa, asesores, etc).**

En el caso de participar más de un tutor, copiar la planilla y completarla.

NOMBRES:	Maríel
APELLIDOS:	Feder Szafir
ROL en el proyecto:	Tutora
Cédula de Identidad - Nro.:	1.767.515-5
Institución en la que se desempeña	Universidad ORT Uruguay
País:	Uruguay
TIPO (Pública, Privada o Mixta):	Privada
Disciplinas	Ingeniería de Software Otras.
¿Tiene ingresado su CVUy en los registros de ANII? En caso afirmativo incluir el link público al formulario. En caso negativo, adjuntar el CV al formulario. (Si/No)	No
CARGO en la organización a la cual representa:	Docente
DEPARTAMENTO DE RESIDENCIA	Montevideo
CIUDAD DE RESIDENCIA:	Montevideo
DIRECCIÓN DE CONTACTO	Juan Jacobo Rousseau 4724
TELEFONO DE CONTACTO	099 511 222

MAIL DE CONTACTO:	mfeder@netgate.com.uy	
Desea recibir notificaciones del CII (Si/No)	No	

#### 4. Empresa/Institución Contraparte

Deben completarse los datos de la empresa o institución involucrada en el proyecto.

##### Datos de la empresa

EMPRESA / INSTITUCION	TS Group SRL				
RAMA ACTIVIDAD	Tecnología				
TIPO	Pública	<input checked="" type="checkbox"/>	Privada	<input type="checkbox"/>	Otras (indicar)
PRODUCTOS PRINCIPALES	Control de Personal, Control de Acceso, Control de Activos con RFID				
Tamaño	1 - 50 empleados				
DIRECCION EMPRESA	Presidente Giró 2523				
TELEFONO	2486 2523				
MAIL EMPRESA	comercial@todosoft.com.uy				
PAGINA WEB	www.todosoft.com.uy				
REPRESENTANTE ANTE EL PROYECTO(*)	Alejandro Martínez				
ROL EN EL PROYECTO	Experto del negocio				
CARGO	Director				
MAIL DE CONTACTO:	a.martinez@todosoft.com.uy				
CIUDAD DE RESIDENCIA:	Montevideo				
DIRECCIÓN DE CONTACTO	Presidente Giró 2523				
TELEFONO DE CONTACTO	099194494				
Desea recibir notificaciones del CII (Si/No)	No				

(\*) Si actúa como Cotutor, adjuntar CV

## II. ESPECIFICACIONES DEL PROYECTO

### 5. Descripción del Problema a solucionar

Describir en un máximo de 1200 palabras, el problema que se pretende solucionar a través del proyecto de innovación presentado. Expresar brevemente el sector de incidencia y las áreas específicas donde se aplicará el proyecto. Asimismo, exponga si existen experiencias previas, la vinculación del proyecto con las mismas y de ser posible, experiencias internacionales en la temática propuesta.

El problema que se pretende solucionar es la dificultad que encuentran hoy los estancieros para controlar y manejar su ganado y obtener la siguiente información de su campo:

- Detectar la posición en que se encuentra el ganado, ya sea porque está perdido, para saber si está en alguna zona peligrosa como por ejemplo zonas inundadas, o detectar si está atrapado en una cañada o sitio similar.
- Detectar estado del campo, ya sea para saber si es bueno para el pastoreo, encontrar zonas inundadas, focos de incendio o humedad de la tierra.

Hoy en día un trabajador del campo realiza un control diario a pie para obtener la información mencionada anteriormente. Esta tarea puede llevar varios días, dependiendo el tamaño del campo. A su vez depende de la situación climatológica ya que los días en que hay lluvias y rayos estas personas no pueden llevar a cabo dichas tareas.

Otro de los problemas detectados es que hay estancias con mucha forestación y zonas peligrosas, las cuales son de difícil acceso, lo que dificulta los controles manuales que se realizan actualmente.

Animales mueren o desaparecen por no ser detectados a tiempo por parte del personal que realiza los controles, esto se debe a los inconvenientes previamente mencionados al momento de llevarse a cabo los controles manuales.

El proyecto busca ofrecer una solución a estos problemas utilizando la tecnología RFID junto con drones de manera que la tarea sea automatizada de acuerdo a las necesidades de la estancia. El dron es capaz de hacer estos controles aún en malas condiciones climáticas y reportar a los estancieros de los resultados recolectados. El sistema también alertará buscando prevenir la pérdida de los animales.

### 6. Descripción de la solución propuesta

Exponer, en un máximo de 1500 palabras, una descripción de los aspectos técnicos asociados a la solución y los principales aportes a desarrollar en el sector. Describir en este punto tecnologías a emplear, metodología, grado de innovación de la propuesta, impacto en la empresa, indicadores de éxito del Proyecto, entre otros y toda otra información que considere relevante para reforzar la presentación.

La solución propuesta se llevará a cabo utilizando drones y tecnología RFID. Los drones hoy en día son cada vez más accesibles por su precio y facilidad de uso y genera incrementos en la productividad, reduciendo los costos y errores que cometen las personas que realizan los controles manuales de ganado en los campos.

Los drones vuelan sobre el campo lo que permite una mirada de aproximadamente 100 metros utilizando un sistema anticollisiones. Esto permite detectar situaciones que a simple vista no se ven.

De acuerdo a la velocidad máxima que alcanza el dron, es capaz de recorrer un campo de 500 hectáreas en 25 minutos con una buena definición de imagen. Esto permite optimizar tiempos evitando el recorrido manual.

Se desarrollarán aplicaciones móviles soportadas por IOS y Android de manera de permitir una supervisión remota de los trabajos realizados por el dron a distancia.

Interactuando con la cámara de alta resolución que ofrece el dron podemos tomar fotografías al campo de manera que con un procesamiento a estas imágenes se logre detectar inundaciones u otros siniestros ocurridos.

El dron tendrá incorporado GPS de manera de volver a la base de carga cuando lo necesite, ya sea que se quede sin batería, que pierda señal o simplemente para marcarle una ruta a seguir.

Etiquetas RFID se colocarán en el ganado así como una lectora RFID en el dron para permitir la identificación del ganado logrando la gestión del inventario que se necesita en los campos.

Con RFID se evita que el ganado sea ubicado fácilmente ya que la información resultante es mucho más confiable y útil que los controles manuales. También la identificación en tiempo real de ganado facilita la prevención de robos y muerte en animales por no ser detectados a tiempo.

La integración de este sistema ofrece beneficios y ventajas a la hora de tomar decisiones en tiempo real con datos fiables logrando de esta manera aumentar la productividad y la competitividad en el mercado.

### 7. Descripción del público/área objetivo

Exponer, en un máximo de 1000 palabras, el público/área objetivo de su proyecto.

Ejemplo: si el mercado es una industria, especificar en qué áreas de esa industria y las principales características de los agentes con potencial interés y capacidad para la apropiación del producto o servicio. Describir la posición de la industria frente a sus competidores y si el desarrollo supone una sustitución de importaciones.

El público objetivo son los dueños de campos que posean una cantidad de hectáreas superior a 400 y que por ende destinen actualmente muchas horas y trabajo para lograr controlar su ganado.

La empresa Todosoft cuenta con interesados en el proyecto una vez el mismo se concrete.

En la actualidad existen sistemas similares, pero los mismos son de producción extranjera y económicamente inaccesibles para el mercado sudamericano.

**Sección no obligatoria:** De ser posible, desarrollar los usos alternativos de la innovación propuesta, sea en el mismo sector de actividad o en otros.

### III. PLAN DE TRABAJO

Describe brevemente el plan de trabajo a seguir para el desarrollo del proyecto, los resultados esperados y los indicadores de cumplimiento:

Ejemplo:

Actividad	Descripción	Hito (Sí/No)	Resultado Esperado	Indicador de cumplimiento
(Ej1: Testear la necesidad del producto o servicio en el mercado)	(Ej1: Presentar el proyecto a referentes en el área y analizar las posibilidades de comercialización del mismo)	(Si)	(Ej 1: Testeo de la necesidad de mercado del producto)	(Ej 1: Cantidad de empresas y/o referentes entrevistados)
(Ej 2: Estudio de alternativas tecnológicas posibles para el desarrollo)	(Ej 2: describir las alternativas estudiadas)		(Ej 2: alternativa seleccionada para aplicar)	(Ej 2: Validación de la alternativa)

Actividad	Descripción	Hito (Sí/No)	Resultado Esperado	Indicador de cumplimiento
1. Investigación de drones a utilizar.	Identificar que drones son los más adecuados para utilizar.	Si	Drones más adecuados identificados.	Nombre, modelo, y presupuesto de los drones.
2. Investigación de tags RFID a utilizar.	Identificar que caravanas RFID son las más adecuados para utilizar.	Si	Caravanas RFID más adecuadas identificadas.	Nombre, modelo, y presupuesto de caravanas RFID.

3. Investigación de lector RFID a utilizar.	Identificar que lectores RFID son los más adecuados para utilizar.	Si	Lectores RFID más adecuados identificados.	Nombre, modelo, y presupuesto de lector RFID.
4. Relevamiento de requerimientos.	Relevar todos los requerimientos tanto funcionales como no funcionales que entran dentro del alcance del proyecto.	Si	Requerimientos funcionales y no funcionales identificados.	Lista de requerimientos funcionales, y no funcionales.
5. Investigación de tecnologías a utilizar.	Identificar las tecnologías más adecuadas a utilizar para el desarrollo de las aplicaciones según las características del proyecto.	Si	Tecnologías más adecuadas a utilizar identificadas.	Listado de tecnologías a utilizar para cada componente del sistema.
6. Búsqueda de potenciales clientes.	Buscar interesados que deseen comprar y usar el producto.	No	Contar con interesados en el proyecto.	Nombres, mails y teléfonos de los interesados del proyecto.
7. Búsqueda de campo de pruebas.	Identificar un lugar para poder hacer pruebas de campo de la solución.	Si	Contar con un campo en el cual hacer pruebas.	Dirección del campo, y horarios para el uso del proyecto.

8. Identificación y mitigación de riesgos.	Identificar los riesgos más relevantes del proyecto y que podemos hacer para que no impacten en el proyecto y si impactan no sean críticos para la realización del mismo.	Si	Análisis y gestión de los riesgos.	Documento con listado de riesgos junto con las acciones preventivas, planes de contingencia y disparadores de cada uno de ellos.
9. Investigación de metodología de trabajo a utilizar.	Identificar la metodología de trabajo más adecuada para la realidad del proyecto.	Si	Metodología de trabajo definida.	Nombre de la metodología a utilizar.
10. Implementación de Web App.	Implementar la Web App de la solución del sistema.	Si	Web App implementada.	Web App releseada.
11. Implementación de Android App.	Implementar la Android App de la solución del sistema.	Si	Android App implementada.	Android App releseada.
12. Implementación de iOS App.	Implementar la iOS App de la solución del sistema.	Si	iOS App implementada.	iOS App releseada.
13. Implementación de los servicios web.	Implementar los servicios web de la solución del sistema.	Si	Servicios web implementados.	Servicios web releseados.

14. Testeo de web app.	Buscar el correcto funcionamiento de la web app y en caso de encontrar anomalías corregirlas.	Si	Todos los bugs encontrados en la web app corregidos.	Tickets de los bugs encontrados en la web app con su descripción, y su estado actual, si se arreglaron o no.
15. Testeo de Android app.	Buscar el correcto funcionamiento de la Android app y en caso de encontrar anomalías corregirlas.	Si	Todos los bugs encontrados en la Android app corregidos.	Tickets de los bugs encontrados en la Android app con su descripción, y su estado actual, si se arreglaron o no.
16. Testeo de iOS app.	Buscar el correcto funcionamiento de la iOS app y en caso de encontrar anomalías corregirlas.	Si	Todos los bugs encontrados en la iOS app corregidos.	Tickets de los bugs encontrados en la iOS app con su descripción, y su estado actual, si se arreglaron o no.
17. Testeo de integración con drones en campo.	Corroborar que los distintos componentes del sistema interactúan de forma correcta entre ellos, encontrar y solucionar anomalías encontradas.	Si	Todos los bugs encontrados en integración de todos los componentes del sistema corregidos.	Tickets de los bugs encontrados en la integración de los sistemas con su descripción, y su estado actual, si se arreglaron o no.

18. Documentación académica.	Documentar lo exigido por la universidad para la entrega del proyecto.	Si	Proyecto con los requisitos académicos documentado.	Documento académico a presentar como proyecto de fin de carrera.
19. Documentación y manuales técnicos.	Documentar las características técnicas de relevancia que pueda necesitar el cliente para su correcto entendimiento y futuro mantenimiento del sistema.	Si	Documentación y manuales técnicos necesarios creados.	Documentos técnicos y manuales técnicos.

## IV. PRESUPUESTO

### 8. PRESUPUESTO GENERAL DEL PROYECTO

Presupuesto estimado en Rubros agregados.

El presupuesto detallado debe completarse en la planilla excel que forma parte del presente formulario y debe coincidir con este cuadro. Los ítems incluidos en el presupuesto deben ser pertinentes al desarrollo y se recomienda la justificación de los gastos más importantes.

Presupuesto General			
Título del Proyecto:			
Duración (meses)			
Rubro	Aporte CII (UYU)	Aporte Empresa (UYU)	Total (UYU)
Adecuación Edilicia	\$0,00	\$0,00	\$0,00
Equipamiento	\$0,00	\$0,00	\$0,00
Actualización de instrumental	\$0,00	\$0,00	\$0,00
Servicios y seguros	\$0,00	\$0,00	\$0,00
Personal Incremental	\$0,00	\$0,00	\$0,00
Consultores	\$0,00	\$0,00	\$0,00
Promoción y Difusión	\$0,00	\$0,00	\$0,00
Materiales e Insumos	\$199.122,00	\$138.686,00	\$337.808,00
Otros costos	\$0,00	\$0,00	\$0,00
<b>Total</b>	<b>\$199.122,00</b>	<b>\$138.686,00</b>	<b>\$337.808,00</b>
%	0,5894531805	0,4105468195	

(NOTA: debe coincidir con el Formulario Presupuesto Detallado)



### V. CRONOGRAMA DE EJECUCIÓN DE ACTIVIDADES

Especificar el Cronograma de Ejecución de Actividades en forma cronológica, señalando el tiempo estimado

de cada una de ellas y los resultados parciales esperados.

Nº	Actividad	Resultado esperado asociado	Mes 1	2	3	4	5	6	7	8	9	10	11	12	Obs.
1	Investigación de drones a utilizar.	Drones más adecuados identificados.	X												
2	Investigación de tags RFID a utilizar.	Caravanas RFID más adecuadas identificadas.	X												
3	Investigación de lector RFID a utilizar.	Lectores RFID más adecuados identificados.	X	X											
4	Relevamiento de requerimientos.	Requerimientos funcionales y no funcionales identificados.	X	X											
5	Investigación de tecnologías a utilizar.	Tecnologías más adecuadas a utilizar identificadas.	X	X											
6	Búsqueda de potenciales clientes.	Contar con interesados en el proyecto.	X										X	X	
7	Búsqueda de campo de pruebas.	Contar con un campo en el cual hacer pruebas.	X												
8	Identificación y mitigación de riesgos.	Análisis y gestión de los riesgos.	X	X	X	X	X	X	X	X	X	X			
9	Investigación de metodología de trabajo a utilizar.	Metodología de trabajo definida.	X												
10	Implementación de web app.	Web App implementada.				X	X	X	X	X	X				
11	Implementación de Android app.	Android App implementada.				X	X	X	X	X	X				
12	Implementación de iOS app.	iOS App implementada.				X	X	X	X	X	X				
13	Implementación de los servicios web.	Servicios web implementados.				X	X	X	X	X	X				
14	Testeo de web app.	Todos los bugs encontrados en la web app corregidos.										X	X	X	



**VI. DOCUMENTOS ADJUNTOS**

1. Presupuesto detallado (Excel)
2. <i>Curriculum vitae</i> * de los involucrados en el proyecto.
3. Escolaridad actualizada*.
4. Carta aval de participación firmada por el Tutor seleccionado.
5. Carta aval de la empresa u organización participante en el proyecto con compromiso de aporte de contrapartida
6. Acuerdo de Propiedad Intelectual (en el caso que corresponda)
7. Otros documentos adjuntos relevantes para el proyecto

**Nota para el usuario:** (\*) Estos documentos deben adjuntarse en formato PDF al formulario.

De no ser adjuntados, la presentación no será admitida para su evaluación por no cumplir con las Bases y no se procederá a la evaluación de la Solicitud.

En caso de tratarse de CVUy podrá incluirse el link a la versión pública del documento (sin ser necesario adjuntar la versión PDF al formulario).

## **Anexo 5: Análisis de herramientas para seguimiento de horas**

Con el objetivo de mantener un historial de las horas dedicadas al proyecto y en qué tareas fueron invertidas por cada uno de los integrantes del equipo es que se buscó una herramienta para ayudar en esta gestión del tiempo.

Se comenzó utilizando una tabla dentro de un documento en Google Drive pero la misma se volvió muy larga y no era una forma práctica de visualizar el tiempo dedicado. Además de que no es fácil generar reportes en base a esa información.

A continuación se analizan 4 de las herramientas más utilizadas y que tienen una buena recomendación por parte de sus usuarios.

### **ClickUp**

Es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos.

La misma incluye una funcionalidad para el seguimiento de horas dedicadas a una tarea específica permitiendo medir el tiempo gastado y el tiempo estimado de la misma.

La versión gratuita contiene:

- Ordenamiento y filtro de las tareas con la visualización del tiempo dedicado y estimado.
- Editar el tiempo estimado y tiempo dedicado.
- Agregar dedicación de tiempos asociados a una fecha.

La versión paga incluye a demás reportes generados que muestran el tiempo dedicado por integrante del equipo, por iteración y por tipo de tarea.

### **Toggl**

Herramienta que permite monitorear el tiempo dedicado al proyecto volviendo más productivo al equipo.

La versión gratuita contiene:

- Timer para iniciar el conteo de tiempo de una determinada tarea.
- Soporta hasta 5 usuarios.
- Reportes en base a la información obtenida.
- Usable desde cualquier dispositivo ya sea web o móvil.
- Visualización gráfica de a dónde se fueron las horas de cada integrante del equipo.
- Se puede agregar como una extensión y funciona con Trello [\[106\]](#).

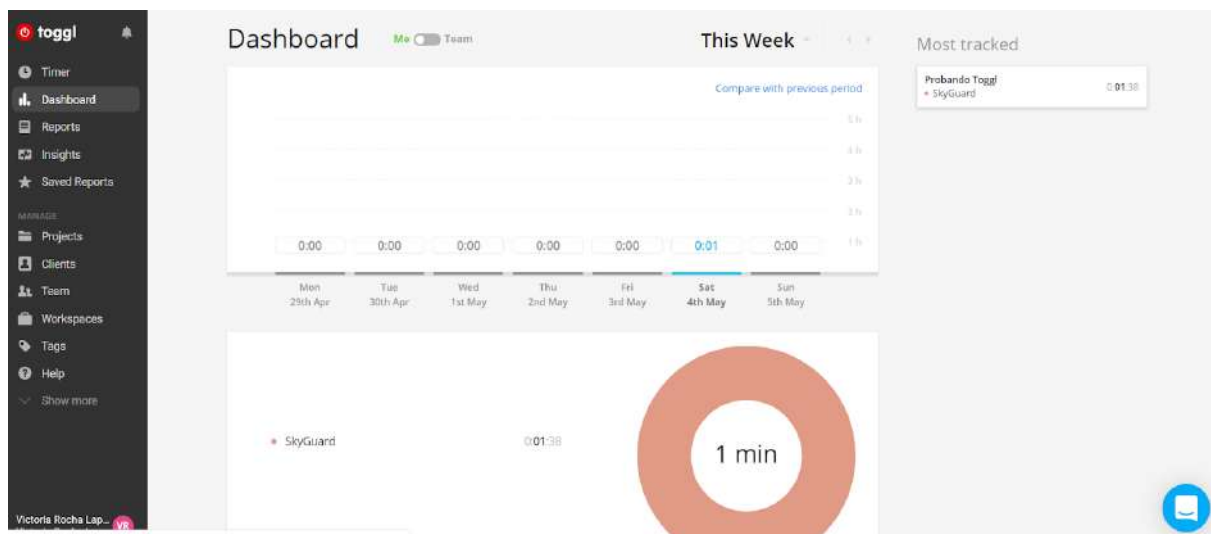


Ilustración 109: Pantalla principal de Toggl

## Bitrix24

Herramienta que para la gestión del proyecto así como también del tiempo. Características:

La versión gratuita contiene:

- Soporta hasta 12 usuarios.
- Ilimitada cantidad de proyectos.
- Definición de tareas y de dependencia entre tareas.
- Gráficas Gantt.

- Versión web o móvil.

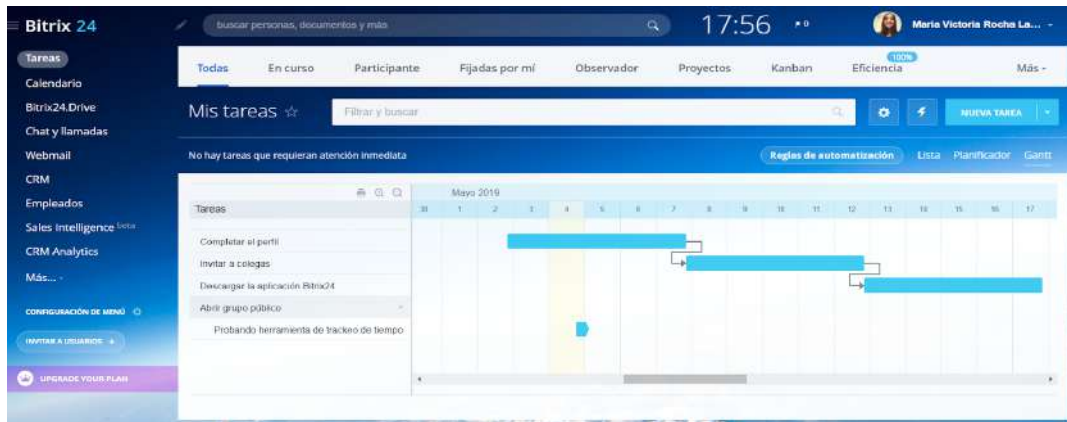


Ilustración 110: Pantalla principal de Bitrix24

## Clockify

Herramienta 100 % gratuita que permite trackear el tiempo dedicado al proyecto por cada uno de los integrantes del equipo.

La versión gratuita contiene:

- Tracking de horas usando un timer.
- Categorización del tiempo por proyecto y por actividad.
- Agregar a miembros del equipo.
- Reportes de tiempos en diferentes formatos: pdf, csv, excel.
- Aplicación web soportada por Chrome o Firefox.

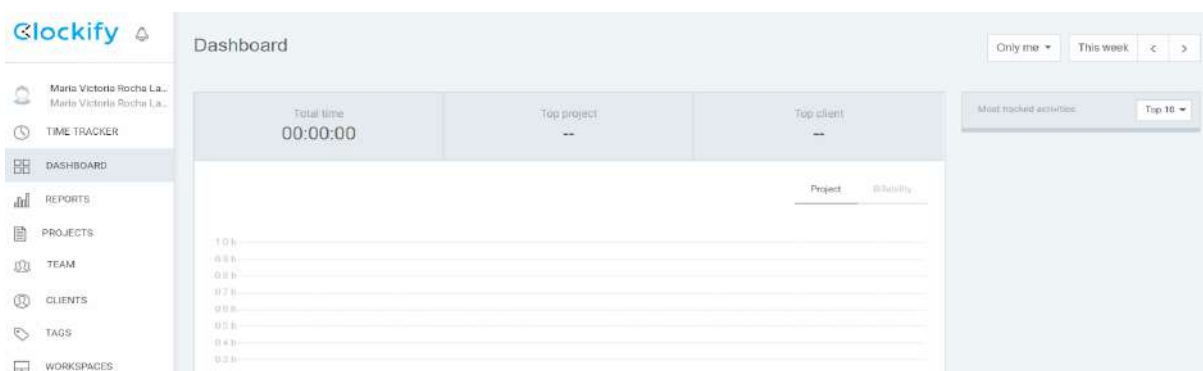


Ilustración 111: Pantalla principal de Clockify

## Conclusión

Después de probar cada una de las herramientas mencionadas se concluye:

Bitrix no tiene una interfaz tan agradable como las demás herramientas ni permite contabilizar el tiempo dedicado a cada tarea. Es más bien una herramienta para gestión del proyecto como lo es Trello.

Clockify si tiene una buena interfaz gráfica, es usable y tiene un timer para contabilizar el tiempo dedicado a cada tarea. También permite definir proyectos y tareas pero no es un gestor de proyectos.

Toggl tiene una buena interfaz gráfica y tiene un timer para contabilizar el tiempo para cada tarea así como Clockify. Se integra con Clickup.

**ClickUp:** Tiene una buena interfaz gráfica, intuitiva y fácil de usar. Ya que ClickUp es la herramienta que se utilizará la gestión de las tareas, es conveniente realizar el monitoreo de las horas con la misma herramienta ya que proporciona todas las funcionalidades necesarias.

## Anexo 6: Historial de retrospectivas

### Retrospectiva 28 de mayo de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	5	6	6	7	6
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Finalizamos la metodología que venía pendiente	División en trabajos en equipo.	El equipo se mostró un poco más adecuado a la metodología de trabajo	Nos sirvió para seguir adentrándonos en cómo vamos a trabajar	Trabajo en equipo
	Concretamos reunión con Alejandro y sacamos bastantes cosas en concreto	Comunicación con Alejandro	Incremento en el compromiso y participación de los integrantes		
		Bajar más a tierra el proyecto con el formulario de financiación y la lista de requerimientos			
		Conocer más sobre las tecnologías de drones y RFID			
<b>Aspectos negativos u Oportunidade</b>					

s de mejora:					
	Emiliano	Victoria	Agustín	Gastón	Dan
	Pocas historias, no avanzamos mucho	Muchos canales en Slack +4	Las iteraciones deberían tener más carga horaria +3	Desprolijidad con las fechas de las iteraciones	no llenar en tiempo y forma lo de reunión alineamiento +3
	Varias tareas super triviales	Más documentos y menos en canales de Slack así en el doc queda como la conclusión y todos podemos leer ahí y las discusiones quedan en el canal de slack	Poner más foco en los deadlines y en las cosas importantes	Desprolijidad en reuniones	no se cumplieron todos los objetivos
	No terminé de completar mi historia de requerimientos				
	Quedaron cosas pendientes que se pasaron a I3				
	Más compromiso personal +2				
	Poner más historias				
	Elegir objetivo de la iteración				
	Deadlines más presentes +9				
	Cumplir con el reporte de alineamiento				

	Desprolijidad en las reuniones  +3				
	Muchos canales de Slack, priorizar docs				
	Pasar cosas conversadas a documentos				
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Poner más foco en los deadlines y en las cosas importantes	A partir del 28/05/2019 las iteraciones van a tener un objetivo que quedará escrito. En el caso de la iteración actual es "Enviar el documento de financiación y definir la lista de requerimientos para un MVP"	Moderador de la reunión		
	Muchos canales en Slack	Tonga y Vicky enviarán un plan de acción que será evaluado por el equipo para luego tomar una decisión. El canal #producto-reqs será migrado a un GDocs por Agustín.	Definidos en el plan de acción		
	Las iteraciones deberían tener más carga horaria	Hacer una revisión de la carga de la iteración en las reuniones de alineamiento y agregar tarjetas según corresponda	Moderador de la reunión de alineamiento		
	No se llenó en tiempo y forma la reunión-alineamiento	MdT, no hay necesidad de CTA	N/A		
	Desprolijidad en reuniones	MdT, no hay necesidad de CTA	N/A		

Tabla 11: Tabla de retrospectiva 28 de mayo de 2019

## Retrospectiva 15 de junio de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	8	6	7	7
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Se cumplieron objetivos importantes (reunión con Alejandro, con experto, envío de financiación)	Modificación a los días de las reuniones.	Me gusta el orden con el que estamos trabajando	Mejoramos la metodología que era la idea para estas primeras iteraciones	Mayor compromiso
	Terminamos de definir y empezar a usar la metodología	Entregar el documento de financiación.	Buena ejecución. Las tareas se ejecutan y, en general, llegamos al resultado esperado	Mejoramos las reuniones (prolijidad y cumplimiento)	Venimos cumpliendo la metodología que nos planteamos
	Se avanzó en investigaciones importantes (roadmap, investigación de drones)	Reunión de Tonga con el experto.	10 puntos la reunión social	Obtuvimos información importante al reunirnos con expertos, y pudimos hacer el contacto con más experto para futuras reuniones	
		Compromiso de cada uno.	La mayoría de los miembros participa en la mayoría de los temas. Eso es bueno IMO		
		Reunión en la casa de Emi	Está habiendo buena recepción de la		

			retroalimentación		
		Cambios propuestos a la metodología de trabajo.	La comunicación es buena y fluida		
		Retroalimentación de Mariel.			
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Más carga horaria (podríamos investigar qué temas podemos ir avanzando desde ahora para el documento final)  +3 Dan  +3 Emi  +2 Vicky  +2 Agu  +2 Tonga	No tenemos cargadas las horas que le dedicamos a cada tarea.  +2 Vicky  +1 Tonga  +1 Emi	Siento que podemos hacer más cosas	Varios integrantes nos perdimos reuniones importantes	Conseguir una herramienta de board para la retro que sea online, porque si se repiten cosas negativas o positivas no podemos agruparlas de manera sencilla en el excel, y votar no es del todo cómodo
	Ausencia de integrantes en las reuniones  +2 Dan  +1 Tonga  +3 Agu	No conseguimos la reunión con el experto que nos pasó el CIE.  +1 Vicky	Algunos miembros del equipo no participaron de la reunión de alineamiento (Al igual que en la iteración pasada). Este punto pesa mucho para mí	Confusión en las reuniones de alineamiento	

	Algunos cambios en la metodología quedaron confusos o no quedan totalmente claros de inmediato  +1 Emi  +1 Tonga				
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Las iteraciones deberían tener más carga horaria	Hacer una revisión de la carga de la iteración en las reuniones de alineamiento y agregar tarjetas según corresponda. Agregar tareas adicionales que nos permitan adelantar con el documento final.	Moderador de la reunión de alineamiento		
	Mayor compromiso en las reuniones pactadas	Tener el horario reservado, hangout e internet	Cada uno		
	Reporte automático	Configurar el bot para que nos recuerde cargarlas, avisarnos entre nosotros si vemos que alguno no lo hizo.	Agus para configurar bot, todo el equipo para el resto.		

Tabla 12: Tabla de retrospectiva 15 de junio de 2019

## Retrospectiva 25 de junio de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	7	7	7	6
<b>Aspectos positivos:</b>					

	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Se avanzaron en varios aspectos (arquitectura, análisis de negocio, deadlines, objetivos, prototipo)	Cargamos las horas a las tareas en Clickup.	Primer iteración donde todos participamos de todas las reuniones	Todos pudimos participar de todas la reuniones	Usamos más fluida la metodología que planteamos
	Hubo muchos debates	Discusiones que surgieron de la arquitectura que sirvieron para bajar más a tierra la solución.		Se avanzó con el documento de arquitectura	Buen documento de objetivos
	Hubo menos dudas en cuanto a la metodología	Compromiso de todos en las reuniones		Parece haber quedado claro el tema de las weekly	
	Mejoramos en todos los aspectos que en la retro pasada consideramos negativos y por ende definimos acciones	Trabajo en equipo y disposición de todos por ayudar.			
		Reuniones coordinadas con Alejandro.			
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	No mostramos/validamos todo lo que hicimos (documentos, prototipo)	No conseguir reuniones con expertos. +1 Vicky	Seguimos sin entender completamente el problema que buscamos solucionar (Paradójicamente)	Algunas respuestas en slack demoraron más de 24 horas en ser respondidas	Se movieron reuniones que planeabamos no mover

	+2 Vicky +2 Emi +2 Tonga	+3 Agu +2 Tonga	)  +2 Agu +1 Emi		
	No cumplimos los objetivos de la iteración +2 Emi +3 Dan	Cerramos una tarea sin tener retroalimentación de los demás.	Mejorar foco en los objetivos de la iteración	Estamos trancados con el tema expertos y dji +2 Dan	No llegamos con el objetivo que era tener la arquitectura pronta +1 Vicky
		Muchos comentarios abiertos que se podrían resolver. +1 Vicky	Más proactividad a la hora de ayudar al compa +1 Tonga	Mala repartición de tareas	Aún sin respuesta de DJI
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>		<b>Responsable</b>	
	Mostrar y solicitar retroalimentación de lo que llevamos desarrollado para X tarea	Fomentar las micro reuniones espontáneas para avanzar más rápido		Todos los que las necesiten	
	Priorizar las tareas imprescindibles para cumplir el objetivo de una iteración	Asignar como mínimo 2 personas a cada tarea de estas		Moderador de la planning	
	Conseguir más expertos de negocio	Asignar la tarea a todos y que cada uno consiga al menos un candidato a experto con el que comience un contacto		Todos	

Tabla 13: Tabla de retrospectiva 25 de junio de 2019

## Retrospectiva 09 de julio de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	5	5	5	7	7
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Conseguimos varios expertos	Reunión de arquitectura		Se consiguieron más posibles expertos	Se consiguieron contactos para validar requerimientos
	Se definieron owner's de los componentes	Contactos de expertos, el de Zafrales y el que consiguió Dan		Se redondeo lo de arquitectura	Se avanzó en lo que va a ser la aplicación (arquitectura)
				Se avanzó con la ppt por más que hay que modificarla	
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Muy poca dedicación	Poca dedicación en horas.	Creo que no llegamos a las 15 horas c/u	Falta de compromiso Tonga +3 Vicky +2	Nos fue mal con la primera versión de la ppt

	Poca participación en reunión con Mariel	Falta de cierre de comentarios que están hace mucho.	Se puede ser más proactivo +2 Dan +2 Agu		
	Se avisó el mismo día que 3 personas no iban a la reunión con Mariel		Se puede mejorar en la calidad del trabajo que entregamos (Este comentario me surge por 3era vez)  Emi +3 Agu +1 Vicky +1		
	Pocas tareas finalizadas		Me pa que el sistema de que todos hagamos todo nos tranca un poco  Agu +2		
	No conocemos la disponibilidad de los integrantes  +3 Dan				
	Mala ejecución de la tarea de la PPT				
	Comentarios en los documentos mucho tiempo sin ser finalizados  Emi +2 Tonga +2 Vicky +2				

<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Comentarios en los documentos con mucho tiempo sin ser finalizados	Hacer relevamiento de los comentarios y ver como seguir	Emi		
	Se puede mejorar en la calidad del trabajo que entregamos	Dedicar al menos una hora a la semana para revisar y corregir los documentos. Agregar esta tarea en clickup	Todos		
	Falta de compromiso	Hablarlo entre los integrantes para concientizar	Todos		

Tabla 14: Tabla de retrospectiva 09 de julio de 2019

## Retrospectiva 25 de julio de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	7		8	7
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Se tuvo reuniones con expertos	Contactar a expertos de negocio		Pudimos contactar más experto	Se arrancó el código
	Se empezó con el código	Resultado de la financiación		Realizamos la primer revisión	Validamos requerimientos

	Se confirmó quién compra el dron	TodoSoft compra el Dron		Se confirmó que hay simuladores lo cual nos va a permitir avanzar sin el dron	TodoSoft compra el Dron
	Cumplimos los objetivos de la iteración	Comenzar a bajar a tierra cada uno de los componentes		Obtuvimos respuesta de la financiación	
	Se cumplieron con las acciones de la retro pasada	Equipo dispuesto a contestar dudas, juntarse, etc.			
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	No tenemos definido el dron que vamos a comprar	<b>No tener claro qué dron comprar.</b>		Idem a los demás	A pesar de las investigaciones sigue sin estar claro qué dron nos serviría más
		<b>No tener validados los requerimientos.</b>			
		<b>Pocas reuniones con expertos.</b>			
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>		<b>Responsable</b>	
	No saber qué dron comprar	Investigar alternativas al DJI Matrice 100 y 200		Dan	
	No tener validados los requerimientos	Coordinar reuniones con los expertos que tenemos		Todos	

	Pocas reuniones con expertos	Coordinar reuniones con los expertos que tenemos	Agus, Dan, Tonga y Emi	
--	------------------------------	--	------------------------	--

Tabla 15: Tabla de retrospectiva 25 de julio de 2019

## Retrospectiva 06 de agosto de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	6	5	5	3	5
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Reunión con experto de hoy	Reunión social		Entender la digitalización de planos	Tenemos los prototipos
	Análisis de digitalización e identificación de perros en fotos	Reunión con Alejandro		Nuevo contacto de experto	Tenemos los requerimientos
	Definición del dron a comprar	Reunión con el experto de hoy			Reunión con Alejandro
	Prototipos				Reunión social
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>

	<p>No sabemos qué reader comprar y no compramos el dron</p> <p>Tonga +3</p> <p>Dan +3</p> <p>Vicky +2</p> <p>Agu +2</p>	<p>Pocas reuniones presenciales</p> <p>Vicky +2</p> <p>Tonga +2</p> <p>Dan +2</p> <p>Emi +1</p>	<p>Falta de foco en los objetivos</p> <p>Agu +2</p> <p>Emi +3</p>	<p>No se mejoraron las cosas planteadas de la iteración pasada</p>	<p>No tenemos lectora definida</p>
	<p>No coordinamos reuniones con varios expertos como habíamos hablado</p>	<p>No tener el reader de RFID y el dron definidos</p>	<p>Gradualmente nos vamos alejando de la metodología de trabajo</p> <p>Agu +1</p>	<p>No pudimos cerrar el tema de la compra del dron</p>	<p>Mariel cada vez nos presiona más por no tener dron ni pruebas hechas con drones en campo y tiene razón</p>
		<p>Falta de dedicación</p>	<p>Esta iteración dejó ver fallas en la ejecución de tareas de iteraciones pasadas</p>	<p>No pudimos reunirnos con ningún experto</p>	<p>No compramos dron todavía</p>
		<p>Falta de aviso a los demás cuando no se llega con una tarea.</p> <p>Vicky +1</p>		<p>No pudimos validar los prototipos</p>	
				<p>La carga de trabajo estuvo mal repartida</p> <p>Emi +1</p>	
				<p>Desempeño de Agustín</p>	
				<p>Los owners de producto no están del todo metidos con el producto</p>	
				<p>Mensajes colgados o que se demora mucho en</p>	

				responder	
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	No sabemos qué reader comprar	Decidir el reader para el domingo y plantearle a Alejandro la opción. Confirmarle el dron para el domingo a Alejandro	Todos		
	Falta de foco en los objetivos	En la planning primero definir los objetivos y luego las tareas en base a eso	Todos		
	Pocas reuniones presenciales	Nos juntamos después de la reunión con la tutora los martes.	Todos		

Tabla 16: Tabla de retrospectiva 06 de agosto de 2019

## Retrospectiva 12 de setiembre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	4	3	5	5	5
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Tenemos dron y la placa llegando	Compramos el Dron		Conseguimos un dron para ir avanzando (si bien aún estamos trancados)	Compramos dron y placa
	Comenzaron a construir el dron los mexicanos	Compramos la nueva placa APM			

Aspectos negativos u Oportunidades de mejora:					
	Emiliano	Victoria	Agustín	Gastón	Dan
	Muy poca dedicación personal	Poca comunicación en especial para ver como va cada uno y el board no sirvió para saber porque no se movió en todo el sprint.  Emi +3  Dan +2  +2 Tonga	Desorden en todos los frentes  +2 Tonga  +5 Agus  Emi +1	Perdimos un poco el contacto con alejandro	Pocas horas dedicadas (sobretudo hablo por mi que tuve examen y no me di cuenta que iba a estar tan complicado)
	Reuniones que no se hicieron/cambiaron de fecha, muy desordenado  Emi +1  Dan +3  Tonga +1	Tareas que agregó Agus y se puso a trabajar que no lo hablamos en la Planning.	Las cosas/tareas siguen sin tener responsable. Anarquía	Desprolijidad en reuniones	No anduvo la placa que traía el drone
	No se cumplieron todos los objetivos	Poca dedicación en horas a la tesis o no se actualiza el board. Las tareas siguen en TODO y es el último día del sprint.		No cumplimos los objetivos	
		Cambiamos mucho las reuniones y nos perdimos un sprint.		La placa del dron no sirvió como esperábamos	

		No hicimos la reunión con Alejandro.			
		No creamos el grupo de Whatsapp con Alejandro.			
		No hicimos la reunión social.			
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Nos está faltando orden y comunicación	Seguir la metodología de trabajo Domingo a las 11 nos juntamos a ver el tema de la metodología	Tonga		

Tabla 17: Tabla de retrospectiva 12 de setiembre de 2019

## Retrospectiva 26 de setiembre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	5	6	6	6	5
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Tenemos la placa y funciona	Asistencia a las reuniones.	Métricas!	Tenemos placa y dron para trabajar	Tenemos la placa y el dron, todo para la tesis

	Mejoramos en el uso de la tecnología	Placa que funciona.		Ya podemos trabajar fluidamente	
	Empezamos a medir con métricas	Métricas			
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Muy poca dedicación personal	Poca dedicación en horas. +3 Dan +2 Emi +3 Vicky +5 Tonga +2 Agu	Uso de ClickUp Agu +2 Emi +1	Necesitamos meter más horas	Usamos poco Clickup
		No hicimos pruebas en campo.	Estaría bueno que el trabajo que cada uno hace sea accesible para todo el equipo (Repos de front, placa, prototipos, etc) +2 Vicky +2 Dan		Metimos poco tiempo dedicado
					No somos muy prolijos a la hora de poner el alineamiento antes de la reunión, y leer los canales de slack Agu +1

<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Poca dedicación de horas	Medir la cantidad de horas que ponemos, intentar de hacer más o menos 15 horas semanales	Agu		
	Mal uso de click-up	Empezar a reflejar el trabajo real en el board de click up Registrar las horas a medida que avanzamos y no al final	Todos		
	Trabajo de tesis no accesible para todos	Documento con credenciales y lugares en donde están publicadas las cosas	Agu		

Tabla 18: Tabla de retrospectiva 26 de setiembre de 2019

## Retrospectiva 10 de octubre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	7	6	7	7
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Incrementamos la cantidad de horas dedicadas	Más comunicación en el equipo.	Más % de completitud	Estoy conforme con el funcionamiento del equipo	Mejoramos % de avance
		Más dedicación.			Reunión con Alejandro
		Reunión con Mariel y			Salida a la casa mia :p.

		Alejandro.			
		Salida a la casa de Dan.			
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	<p>No estamos avanzando a la velocidad que deberíamos</p> <p>+1 Agu</p> <p>+2 Tonga</p> <p>+2 Vicky</p> <p>+2 Emi</p>	<p>No cumplimos con las tareas que nos planificamos. Esto viene pasando hace muchos sprints.</p>	<p>Discutir y DEFINIR el problema que estamos solucionando con la TESIS (A Mariel le dijimos que estamos solucionando el abigeato cuando no estamos ni sacando fotos, ni obteniendo datos en tiempo real -el dron va a volar una vez cada x tiempo-). Al menos definir un sólo discurso para contarle lo mismo a todas las partes (Alejandro y Mariel principalmente). (Quizás sea una buena discusión para el armado de la presentación)</p> <p>+3 Vicky</p> <p>+1 Emi</p> <p>+2 Agu</p>	<p>Necesitamos meter más horas</p>	<p>No pudimos meter todas las tareas que pensábamos en la sprint</p> <p>+2 Tonga</p>

		<p>No nos estamos juntando a definir cómo vamos a resolver las cosas. Hay un montón de cosas que no están claras.</p> <p>+2 Dan</p>	<p>Siento que falta un plan a más largo plazo (Lo que al principio intentábamos hacer con el roadmap). En la fecha X vamos a cumplir con tal hito, en la fecha Y vamos a cumplir con tal otro, etc.</p> <p>+2 Agu</p> <p>+2 Emi</p>		
		<p>No documentamos los flujos de nuestras principales funcionalidades. Tipo, esto pasa primero por front, luego a <i>backend</i> se le pide esto etc...</p>			
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	No avanzamos a la velocidad que deberíamos.	Asegurarnos de dedicar 15 horas semanales cada uno. En todas las reuniones de alineamiento vamos a repasar si cada uno cumplió con las 15 horas, y si no lo hizo se explicará por qué	Emi		
	Plan a largo plazo + definiciones de problemas a solucionar	Coordinar para el fin de semana una reunión para alinearnos.	Vicky		

Tabla 19: Tabla de retrospectiva 10 de octubre de 2019

## Retrospectiva 24 de octubre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	7	6	7	7
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Bajamos a tierra muchas cosas: errores, problemas a resolver, requerimientos	Tenemos la nueva placa	Reaccionamos ante la retroalimentación de la revisión	Tuvimos la revisión	Charla entre todos, introspectiva
	Aprendimos cómo manejar las tareas de acá al futuro	Retroalimentación de la revisión 2	Conversamos los temas incómodos	Charla de incomodidades	Segunda placa APM
	Tenemos bastante claro cómo seguir de acá en adelante	Salida a Tierra de osos			Charla con gastón
	La revisión 2 fue mucho mejor que la 1				Revisión 2 dejó ver muchos problemas que vamos atacando de a poco
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>

	Casi nulo avance en tema desarrollo +2 Vicky	Seguimos sin dedicar las horas que deberíamos	Poco avance en el desarrollo de la plataforma (problema de horas?)	Problemas en la gestión de proyecto +3 tonga +3 Agu +5 Dan +2 Emi	Seguimos sin poder cerrar bien las iteraciones en cuanto a tareas completadas  +2 Agu +2 tonga
	No dedicamos las 30 horas por sprint salvo Dan +3 Emi +3 Vicky	No tenemos una velocity	No estamos haciendo una buena gestión de proyecto (Reaccionamos ante la revisión y estamos trabajando en eso, pero creo que vale la mención).		
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Problemas en la gestión de proyecto	Coordinar reunión para bajar a tierra y ver planes de mejora	Emi		
	No dedicamos las 30 horas por sprint	Por el momento no se hace nada, se espera por los resultados de cambios aplicados anteriormente			

Tabla 20: Tabla de retrospectiva 24 de octubre de 2019

## Retrospectiva 07 de noviembre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	8	7	7	7	6

<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Aumentó la dedicación horaria	Avanzamos con la integración de back y front.	Deploy de back	Buena dedicación de horas	Front integrado con back (finally!)
	Integración <i>backend - frontend</i>	Reunión con Alejandro para la estimación.	Alineamiento con Ale	Reunión con alejandro	Viene avanzando bien la codificación
	Alineamiento con Alejandro				Empezamos a darle más importancia a la gestión
	Sin ser el dron, no parece haber blockers en otros ámbitos				Reunión con alejandro
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Seguimos trancados con el dron	No tenemos un Dron funcionando por lo tanto tampoco pudimos hacer pruebas de campo.	Necesitamos el dron ASAP +3 Dan +2 Tonga +3 Agus +1 Emi +1 Vicky	No llegamos al objetivo de cerrar la historia de zonas	Seguimos sin tener dron funcionando
			No pudimos completar una US en el sprint	El dron sigue sin funcionar	No podemos hacer pruebas de campo

			+2 Emi +1 Tonga		
					Mucho código del dron terminado con incertidumbre de si va a funcionar igual que en el simulador  +2 Vicky
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	No tener el dron funcionando	Asignar owner y dedicar muchas horas a solucionarlo	Agus		
	Terminar las US	Aumentar la coordinación y trabajar para completar los objetivos (que tienen que estar especificados)	Emi		

Tabla 21: Tabla de retrospectiva 07 de noviembre de 2019

## Retrospectiva 24 de noviembre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	8	8		8	9
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	El dron funciona	Tenemos el dron funcionando		Seguimos trabajando de forma fluida	Hay muchas épicas parcialmente

					cerradas
	3 historias cerradas	Avanzamos un montón con el back y el front		Destrancamos lo del dron	Funciona el drone!!
	El <i>backend</i> está deployado y ya no da problemas. Se puede avanzar más libres	Estamos enfocados en terminar las US		Destrancamos lo de zonas	
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Todavía Alejandro no puede validar +2 Emi +2 Dan +1 Vicky +2 Tonga	No hicimos pruebas de campo +1 Emi +2 Vicky +1 Dan		Personalmente no pude dedicar muchas horas por las entregas	No validamos con alejandro
	No hemos ido al campo	No validamos con Alejandro las cosas de front			
	Poca dedicación personal				
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>		<b>Responsable</b>	
	No hemos ido al campo	Tonga va a coordinar una ida al campo		Tonga	
	Alejandro no ha	Ver de deployar en heroku		Dan	

	validado las cosas			
--	--------------------	--	--	--

Tabla 22: Tabla de retrospectiva 24 de noviembre de 2019

## Retrospectiva 09 de diciembre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	8	N/A	6	6
<b>Aspectos positivos:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	Tenemos el dron funcionando (la parte de hardware)	Nuevo dron con la APM conectada y calibrado.	Pudimos ensamblar, volar y configurar el dron	Pudimos ensamblar, volar y configurar el dron	Drone volando
	Se avanzó bastante en <i>backend</i>	Pruebas con el dron.			
	Arrancamos con las pruebas en campo con el drone	Buen trabajo en equipo y que en general todos estamos en contacto y sabemos en que estamos. Buena onda de todos :)			
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>

	<p>No tenemos cerrado el plan de vuelo y me preocupa el algoritmo de ejecutar una misión a determinada hora si el dron está apagado</p> <p>+2 Vicky</p> <p>+1 Emi</p> <p>+3 Agu</p>	<p>Más compromiso para dedicarle tiempo y ser responsables en que si nos comprometemos a una tarea está bueno terminarla o avisar con tiempo como habíamos dicho.</p> <p>+3 Dan</p> <p>+1 Vicky</p>	<p>Poca dedicación</p> <p>+2 Emi</p> <p>+3 Tonga</p>	Poca dedicación	Poca dedicación
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	Plan de vuelo no cerrado	Fue incluido en el sprint para terminarlo en él	Agu		
	Ejecutar misión a determinada hora no cerrado	Agu le pasa el <i>script</i> a Vicky y ella lo prueba	Agu y Vicky		

Tabla 23: Tabla de retrospectiva 09 de diciembre de 2019

## Retrospectiva 23 de diciembre de 2019

<b>Lugar:</b>	Hangouts				
<b>Puntuación:</b>	<b>Emiliano</b>	<b>Victoria</b>	<b>Agustín</b>	<b>Gastón</b>	<b>Dan</b>
	7	8	8	8	7
<b>Aspectos positivos:</b>					

	Emiliano	Victoria	Agustín	Gastón	Dan
	En la revisión nos corrigieron pocas cosas, parece que estamos más estables que antes	Tenemos todo más encaminado!	La presentación y la revisión salió bien, IMO	Dimos una buena revisión	Buena revisión
	Definimos cómo se va a comunicar el <i>backend</i> con el dron		Tenemos un plan y parece bueno	Seguimos avanzando con la configuración del dron y tenemos pa dónde encarar	
	El vuelo del dron está encaminado				
<b>Aspectos negativos u Oportunidades de mejora:</b>					
	Emiliano	Victoria	Agustín	Gastón	Dan
	No accionamos el plan de probar el arranque de la misión a X hora	No pude dedicarle suficiente cantidad de horas.	Le falta a la integración sistema <-> dron y. eso es core	Le falta a la integración sistema <-> dron y. eso es core +3 Tonga +2 Emi +3 Dan	No pude dedicarle lo que quería por temas personales  +2 Vicky
	No cumplimos 2 de los 3 objetivos del sprint (Poder cargar una misión al dron y que funcione bien para filmar un video, tener una primera versión de las Push Notifications)  +3 Emi		Dron definitivo está cojido	Dron definitivo está cojido  +2 Tonga +2 Agu +2 Dan	

	+3 Vicky +3 Agu				
<b>Conclusiones:</b>					
	<b>Problema</b>	<b>Plan</b>	<b>Responsable</b>		
	No cumplimiento de algunos objetivos del sprint	Prestar más atención a la hora de comprometernos con los objetivos en cada planning	Emi		
	Falta de integración sistema-drone	Lo comprometimos para el sprint 17	Tonga y Agu		

Tabla 24: Tabla de retrospectiva 23 de diciembre de 2019

## Anexo 7: Informe de revisiones

### Informe revisión 1

<b>Grupo:</b>		<b>Fecha:</b>	11/07/2018
<b>Carrera:</b>	Ingeniería en Sistemas	<b>Revisores:</b>	Álvaro Ortas

**Importante:** Este informe debe ser elaborado por el Equipo y enviado al Tutor en un plazo máximo de 10 días corridos a partir de la fecha de la Revisión.

#### Fortalezas del Grupo

- Tenemos un muy buen proyecto.
- Tenemos un equipo capaz de obtener la nota máxima en el proyecto.

#### Oportunidades de Mejora

- Mejorar la Presentación.
- Realizar más entrevistas con expertos de negocio para validar los requerimientos y el prototipo.
- Volver a pensar los objetivos de forma que sean SMART.
- Identificar el segmento de cliente al que nos vamos a enfocar.

### Acciones a Realizar

#	Descripción de la Acción	Fecha estimada
1	Modificar la PPT para contar con mucho más detalle la solución.	Setiembre 2019
2	Modificar la PPT para mostrar con más detalle el negocio haciéndose énfasis en datos del mismo en Uruguay (cantidad de ganado, porcentaje de representación de la ganadería en la economía uruguaya, etcétera).	Setiembre 2019
3	Modificar la PPT para hacer énfasis en quién es nuestro cliente en la parte de objetivos.	Setiembre 2019
	Modificar la PPT para explicar los problemas actuales que tiene nuestro cliente.	Setiembre 2019
4	Modificar la PPT para borrar los objetivos que no se adapten a una metodología ágil y poner más datos de los que mostramos (demostrar que son SMART).	Setiembre 2019
5	Modificar la PPT para borrar los nombres Plan de Gestión de Riesgos (cambiar por Gestión de Riesgos), Plan de Calidad (buscar palabra para identificar los estándares de calidad a seguir) y Plan de SCM (Gestión de Repositorios).	Setiembre 2019
6	Modificar la PPT para contextualizar más: a qué campos apuntaremos?, a qué parte de la gestión de ganado?, a qué comercio atacaremos? (ganadería, de carnes, etcétera).	Setiembre 2019
7	Modificar la PPT para mostrar qué requerimientos teníamos para elegir el dron y por qué elegimos el DJI Matrice 200.	Setiembre 2019
8	Modificar la PPT para explicar el papel que juega el peón de campo con nuestra solución.	Setiembre 2019
9	Modificar la PPT para incluir la relación entre RFID y las caravanas actuales que poseen los animales.	Setiembre 2019
10	Mirar al público cuando presentamos y cuando presentan	Setiembre 2019

	nuestros compañeros.	
11	Modificar la PPT para mostrar con más detalles los riesgos: impacto, probabilidad, plan de contingencia.	Setiembre 2019
12	Modificar la PPT para sacar las PPTs que no agregan valor como las métricas de cantidad de páginas escritas, mensajes enviados, etcétera.	Setiembre 2019
13	Modificar la PPT para explicar el porqué de la utilización de drones como parte de la solución y las características del mismo.	Setiembre 2019
14	Explicar en la PPT que la técnica de validación es el prototipado,	Setiembre 2019
15	Hacer prototipo de la aplicación móvil.	Agosto 2019
16	Definir quién es nuestro cliente más importante (estanciero o peón).	Agosto 2019
17	Hacer entrevistas a peones.	Agosto 2019
18	Gestionar la calidad usando Scrum + BDD.	Hecho
19	Estudiar la presentación para hablar con más propiedad, trancarse menos y usar menos muletillas.	Setiembre 2019
20	Solucionar la compra del dron cuanto antes.	Agosto 2019
21	Explicar qué incluye la versión 1.0 del sistema.	Setiembre 2019
22	Definir de qué se va a encargar la aplicación móvil y de qué la aplicación web.	Agosto 2019
23	Incluir en la PPT solo los riesgos más importantes con la información detallada de plan de acción y plan de contingencia.	Setiembre 2019
24	Definir releases: el contenido de cada uno y la fecha.	Agosto 2019
25	Definir la gestión del repositorio.	Agosto 2019

Tabla 25: Tabla de acciones a realizar - revisión 1

## Informe revisión 2

<b>Grupo:</b>		<b>Fecha:</b>	26/10/2019
<b>Carrera:</b>	Ingeniería en Sistemas	<b>Revisores:</b>	Gastón Mousques

**Importante:** Este informe debe ser elaborado por el Equipo y enviado al Tutor en un plazo máximo de 10 días corridos a partir de la fecha de la Revisión.

### Fortalezas del Grupo

- “Creo que están haciendo un proyecto muy interesante, complejo y con muchos desafíos de todo tipo. Es un proyecto con varias oportunidades para lucirse en el contexto de un proyecto académico. Creo que este proyecto sería muy desafiante incluso para un equipo de gente con años de experiencia y uds lo van llevando bien”

### Oportunidades de Mejora

- Mejorar la Presentación, ser más explicativos.
- Realizar más entrevistas con expertos de negocio para validar los prototipos.
- Priorizar los objetivos y el alcance para definir bien qué pueden hacer en función de los objetivos de la solución que definan

### Acciones a Realizar

#	Descripción de la Acción	Fecha estimada
1	Modificar la PPT para contar con más detalle la solución.	Diciembre 2019
2	Dejar claro que el cliente nos planteó una idea y les encargó definir una solución y evaluar la factibilidad técnica.	Diciembre 2019
3	Sería bueno profundizar (o definir bien con el cliente) en los objetivos de la solución que uds. derivaron del desafío planteado. Definir qué parte es un MVP para validar el atractivo de la idea para potenciales clientes de la empresa (funcionalidades web, móvil, <i>backend</i> , etc.), y qué aspectos es necesario validar para ver la factibilidad tecnológica (Drones, imágenes, conectividad, obtención de datos en el campo, etc.).	Noviembre 2019
4	En base a los objetivos de la solución, del proyecto y académicos sería bueno describir la estrategia que definieron para llegar a	Diciembre 2019

	buen puerto.	
5	Presentar antes el concepto de la solución, cómo va a funcionar entrando en más detalle en las tecnologías. Tal vez presentando algún escenario de uso para alguno de los usuarios.	Diciembre 2019
6	Enumerar las investigaciones que decidimos hacer: sobre drones, reconocimiento de imágenes, las pruebas de concepto para validar tecnologías, etc. Destacar el esfuerzo invertido en esto.	Diciembre 2019
7	Describir mejor cómo su release plan apunta a lograr los objetivos de la solución. Qué va a entregar en cada release 0.x y por qué en ese orden. Qué valor aporta cada release.	Diciembre 2019
8	En relación a Scrum revisar cómo definimos las historias de usuario y cómo las "quemamos" en los sprints.	Octubre 2019
9	Definir mejor los atributos de calidad a nivel del sistema en general ( <i>software</i> , hardware, conectividad)	Diciembre 2019
10	Estamos manejando el problema como dos subproyectos (el de drones y el de las apps), aclararlo a lo largo de la presentación.	Diciembre 2019
11	¿Qué pasa con la conectividad en el campo? No explicamos esto. Explicar mejor que va a haber una base con wifi (sí o sí) en el campo. Explicar que el dron no necesita internet para realizar su trabajo y que no se va a perder ya que vuelve a la base ante determinadas circunstancias.	Diciembre 2019
12	Explicar lo fácil que va ser escalar el <i>hardware</i> gracias a la placa ardupilot (Ardupilot compliance) y hasta qué nivel programamos esta placa.	Diciembre 2019
13	Contar varias cosas que hicimos para minimizar riesgos, como probar rfid, utilizar simuladores de drones, probar el procesamiento de imágenes, etc...  Hay riesgos de arquitectura y tecnología que son grandes. Qué puntos de tecnología pueden fallar? Qué puntos de conectividad pueden fallar? Etc.	Diciembre 2019

Tabla 26: Tabla de acciones a realizar - revisión 2

## Informe revisión 3

<b>Grupo:</b>		<b>Fecha:</b>	17/12/2019
<b>Carrera:</b>	Ingeniería en Sistemas	<b>Revisores:</b>	Darío Macchi

---

**Importante:** Este informe debe ser elaborado por el Equipo y enviado al Tutor en un plazo máximo de 10 días corridos a partir de la fecha de la Revisión.

### Fortalezas del Grupo

- Proyecto muy interesante que integra *Hardware* y *Software*.
- El proyecto tiene muchas complejidades que las hemos resuelto sobre todo de *hardware* que no es lo que más sabe el equipo.
- El problema que resolvemos está claro.
- La investigación de RFID está bien bajada a tierra.
- Las funcionalidades en alto nivel quedaron claras.
- Los riesgos y la calidad están claros.

### Oportunidades de Mejora

- Colocar la parte de investigación de RFID después de explicar la solución al problema en alto nivel y luego de explicar RFID bajar a más detalle la solución con RFID.
- Agregar en un anexo en la documentación final la investigación de RFID.
- Explicar cómo gestionamos el tiempo que dedicamos a la investigación para que no parezca que hubo una gran parte del tiempo dedicada a investigación no gestionada.
- No quedó claro si la parte de procesamiento de imágenes es un entregable del producto.
- Explicar cómo son las ondas radiales de RFID que utilizamos para evitar que si una vaca se mueve el dron pueda no leerla.
- Explicar cómo le permitimos a los usuarios del sistema delimitar su terreno en la aplicación. Contar la investigación hecha sobre los archivos digitalizados que actualmente existen y que pueden utilizarse para delimitar automáticamente el terreno.
- Agregar al agrónomo a los expertos en la presentación.
- Aclarar que un animal en peligro quiere decir que un animal se encuentra en una zona

definida como peligrosa y que no hay más variables en juego que esa.

- No utilizar la frase “en tiempo real” porque realmente no es en tiempo real, sino que la sincronización se hace mientras el dron está volando o en caso de no tener señal se hará cuando el mismo llegue a la base.
- El plan de vuelo es una muy buena funcionalidad que no está explicada en la presentación.
- La imagen de brainstorming no está acorde a lo que es un brainstorming porque ya tiene cosas definidas en formato de historia de usuario.
- No queda del todo claro por qué usamos Typescript más allá del tipado.
- Aclarar cómo reemplazamos las Dailys y las Reviews que propone Scrum.
- Estamos llegando tarde a las pruebas con usuarios, es decir, a las pruebas de campo con las personas que utilizarán el sistema a diario.
- Agregar el número de diapositiva.

#### Acciones a Realizar

#	Descripción de la Acción	Fecha estimada
1	Mover en la presentación la investigación de RFID a después de explicar la solución.	Diciembre 2019
2	Explicar que de la investigación de RFID generamos un entregable con la investigación documentada.	Diciembre 2019
3	Explicar que la gestión del dron tiene que ser automática y que no puede ser una opción que la persona controle el dron con el control en el campo por el tamaño del mismo	Noviembre 2019
4	Explicar lo que se hizo de procesamiento de imágenes pero aclarar que no es un entregable del producto. Agregarlo en un anexo en la documentación final.	Diciembre 2019
5	Agregar a la presentación cómo son las ondas radiales de RFID.	Diciembre 2019
6	Contar la investigación hecha sobre los archivos digitales que existen con la delimitación del campo.	Diciembre 2019
7	Agregar a todas las personas consultadas como expertos en la presentación (Estancieros, peones, agrónomo, Juan, etc.)	Diciembre 2019
8	Explicar que quiere decir que un animal esté en peligro y	Enero 2020

	mencionar cómo lo manejamos con las alertas.	
9	Mostrar alguna imagen de cómo se ven las alertas.	Enero 2020
10	Aclarar “en tiempo real” porque el dron no vuela 24 horas y no tiene señal todo el tiempo	Enero 2020
11	Agregar a la presentación el plan de vuelo que estamos armando.	Enero 2020
12	Cambiar la imagen de brainstorming de la presentación por una acorde a lo que hicimos antes de bajarlo a historias de usuario.	Enero 2020
13	Explicar por qué usamos Typescript: somos 5 integrantes del equipo y nos permite definir un acuerdo para los tipos, evitar errores en tiempo de ejecución y dejarle algo más documentado al cliente.	Enero 2020
14	Explicar que diseñamos para reutilizar componentes entre React y Reac native.	Enero 2020
15	Explicar que reemplazamos las Dailys por comunicación por Whatsapp/Slack.	Enero 2020
16	Explicar que con Continuous Delivery reemplazamos las Reviews de Scrum.	Enero 2020
17	Agregar al diagrama que explica nuestra metodología las Retrospectivas que realizamos.	Enero 2020
18	Hacer pruebas de campo con quienes realmente utilizarán el sistema.	Enero 2020
19	Agregar un gráfico indicando el tiempo dedicado a cada tipo de investigación.	Enero 2020
20	En las fotos del equipo agregar a Alejandro como Product Owner.	Enero 2020
21	Agregar la técnica de “Observación” en la etapa de Ingeniería de requerimientos.	Enero 2020
22	Agregar a la presentación: logros, a futuro y lecciones aprendidas	Enero 2020
23	Agregar números a las diapositivas	Enero 2020

Tabla 27: Tabla de acciones a realizar - revisión 3

## Anexo 8: Mejoras propuestas por el cliente

Gracias a la liberación constante de versiones, el cliente pudo dar su retroalimentación de forma temprana sobre las funcionalidades desarrolladas, la cual ha sido documentada y desarrollada a lo largo del proyecto.

A continuación se listan las sugerencias obtenidas, divididas por tópicos:

### 1. Home

- a. Estaría bueno que sea una especie de *dashboard* con la siguiente información.
  - i. Cantidad de ganado activo
  - ii. Información de los últimos inventarios
    1. Total de ganado encontrado
    2. Total de ganado no encontrado
  - iii. Cantidad de vuelos realizados
  - iv. Animales ubicados en zonas peligrosas
  - v. Animales ubicados fuera de zonas definidas

### 2. Zonas

- a. Estaría bueno editar zona
- b. Visualizar nombre de zona
- c. Modificar el menú de acciones que queda poco intuitivo
- d. Poder extender el mapa a pantalla completa

### 3. Alertas

- a. Texto más descriptivo en las notificaciones

### 4. Animales

- a. Agregar editar animales
- b. Agregar los campos definidos por ARU
- c. La eliminación de animales no debería poder hacerla un usuario normal
- d. La baja de animales le asociaría un motivo

- i. Fallecimiento
- ii. Venta
- iii. Permuta
- iv. Desaparición
- e. Agregar reportes de animales

## **5. Seguridad**

- a. Agregar roles de usuario

## **6. Plan de vuelo**

- a. Agregar un botón para ejecutar el plan de vuelo

## Anexo 9: Revisión de código

El equipo decidió realizar revisiones de código para mejorar la calidad de código generado en el proceso de desarrollo de *software*.

En esta práctica participó al menos un integrante que no participó en el desarrollo de la porción de código a revisar.

Para llevar a cabo esto, cada vez que se quiso integrar un nuevo código a la rama de desarrollo *develop*, se realizó un pull request asignando un revisor. El revisor es el encargado de revisar el código y verificar que se esté cumpliendo con la nomenclatura y estándares de calidad entre otras cosas.

A continuación podemos ver un ejemplo.

The screenshot shows a GitHub pull request titled "Estructura del proyecto y configuración básica de la entidad Drone #1". The pull request was merged by Emirona-07 on 25 Sep 2019, merging 11 commits into the develop branch from the estructura\_del\_proyecto branch. The interface includes a conversation tab with 30 comments, 11 commits, 0 checks, and 17 files changed. A comment from Emirona-07 on 18 Aug 2019 describes the project structure: "Se crea la estructura del proyecto conteniendo un Controller, un Model, el contexto para la base de datos, migraciones para la creación de la base de datos y la tabla drone." The pull request is reviewed by agurodriguez on 29 Aug 2019. The review shows two instances of a file named "Controllers/DronesController.cs" marked as "Outdated". The right sidebar shows the pull request settings, including reviewers (agurodriguez), assignees (No one—assign yourself), labels (None yet), projects (None yet), milestone (No milestone), and linked issues (Successfully merging this pull request may close these issues).

Ilustración 112: Pull request

## **Anexo 10: Metodología de trabajo**

### **Expectativas de un miembro del equipo**

Con el fin de estar alineados en lo que se espera de un miembro del equipo es que se definen las siguientes expectativas. Un miembro del equipo debe:



- Cuando estamos tomando decisiones, hay que contribuir, discutir y opinar con libertad, a la hora de tomar una decisión si no se consensua, aceptar de forma amable y sin encapricharse lo que le parece mejor a la mayoría del equipo. Una vez que una decisión está tomada se debe comprometer a cumplirla.
- Dar retroalimentación constructiva, con amabilidad y respeto. Asimismo, no debe tomarse a mal la retroalimentación entregada por otros miembros del equipo. Es una herramienta fundamental para el crecimiento personal y profesional de los miembros del equipo y por ende el éxito del proyecto.
- Estar disponible en los canales de comunicación definidos y responder los mensajes en menos de 24 horas (siempre y cuando no haya avisado de su ausencia).
- Hacer todo lo posible por cumplir con los objetivos del sprint y por mantener un promedio mayor al 70 % de cumplimiento. Asimismo mostrar empatía con los compañeros que por algún motivo no puedan cumplir con los objetivos planteados.
- Notificar al resto del equipo de los períodos de ausencia o vacaciones agregando un evento compartido en Google Calendar (Tesis - ORT). La notificación debe darse con al menos una semana de anticipación.
- Ser amable y respetuoso al dirigirse a otro miembro del equipo.
- Ser proactivo a la hora de ayudar a otros miembros del equipo.
- Ser puntual al asistir y moderar las reuniones de equipo.
- Tener iniciativa a la hora de proponer mejoras a los procesos de trabajo y al producto.

### **Comunicación**

Toda comunicación del equipo relacionada al proyecto será realizada vía Slack, *workspace* kobauy.slack.com, en el canal que corresponda. Para mensajes de carácter urgente podrá utilizarse el grupo de WhatsApp creado para dicho propósito.

## **Slack workspace**

Agustín Rodríguez será el responsable del *workspace*. Con la finalidad de mantener una comunicación ordenada se establecen algunas pautas para el correcto manejo de los canales:

- El canal #tesis será utilizado para discutir tópicos generales y compartir información de interés para todos los miembros del equipo.
- El canal #importante-novedades será utilizado para anunciar decisiones o novedades importantes (que impactan a todo el equipo o que afectan el tiempo, alcance o costo del proyecto).
- Los mensajes compartidos en los canales que empiezan con la palabra #importante deben ser leídos y confirmados por todos los miembros del equipo reaccionando al mismo con el ícono .
- Los canales que empiezan con la palabra #reunión son para registrar las agendas y minutas de las reuniones (Salvo el #reunión-alineamiento cuyo propósito se describe más abajo). Todos los miembros del equipo deben leer y reaccionar con el ícono  la minuta.

## **Google Calendar**

Victoria Rocha será la responsable del Google Calendar. Tiene la finalidad de compartir los eventos que conciernen a todos los miembros del equipo. Es un calendario compartido lo que permite que cualquiera pueda crear eventos nuevos o modificar los existentes.

También servirá para marcar los períodos de ausencia de los miembros del equipo, de forma de conocer la disponibilidad de cada uno.

## **Herramientas**

- Google Calendar
- Slack Client (de escritorio y móvil)
- WhatsApp

## Reuniones

Las reuniones formales son las definidas en la siguiente tabla y son de carácter obligatorio. Cada reunión es moderada por un miembro del equipo (Emiliano).

La función del moderador es la de llevar adelante la reunión ordenando la participación de los actores, velando por el cumplimiento de la agenda y los tiempos establecidos. Además es quien registra la minuta en el canal que corresponda (de las reuniones con el tutor y con Alejandro Martínez).

Las reuniones serán incluidas en el calendar (Tesis - ORT).

Motivo	Agenda	Fecha	Lugar	Inicio	Fin
#reunión-alineamiento  Reunión de sincronización con el equipo. El objetivo principal de esta reunión es alinearlos en las tareas que está haciendo cada uno, resolver temas que involucren a más de un miembro del equipo y planificar próximas tareas.	* Previo a la reunión los miembros del equipo deben responder el cuestionario de alignment en el canal #reunión-alineamiento, así sea que asistan o no.  15 mins para leer el canal. Los mensajes deben ser ACK con el ícono <input checked="" type="checkbox"/> .  15 mins para resolver dudas de los comentarios.  Si sobra tiempo Q&A abierto.	Cada semana, el día jueves	Online	23:10	30 mins
#reunión-planificación  Revisión de la iteración actual.  Planificación de la siguiente iteración.	Revisión de cómo terminamos la iteración.  Revisión de deadlines.  Planificación de la siguiente iteración. De acuerdo a las tareas definidas en el backlog decidir cuáles se van a	Cada dos semanas, el día jueves	Online	23:40	30 mins

	<p>incluir en la iteración que comienza.</p> <p>Revisar carga equitativa de las tareas para la nueva iteración.</p>				
<p>#reunión-retrospectiva</p> <p>Revisión de cómo terminamos la iteración. El objetivo principal de esta reunión es mirar el trabajo pasado, entender qué hicimos bien y qué podemos mejorar.</p>	<ol style="list-style-type: none"> <li>1. Cada miembro del equipo asigna un puntaje a la iteración pasada (del 1 al 10) [5 mins].</li> <li>2. Cada miembro del equipo plantea las cosas buenas de la iteración, nuevas ideas y oportunidades de mejora [10 mins].</li> <li>3. Cada miembro del equipo reparte 5 votos (pudiendo asignar más de 1 a una misma entidad) entre las ideas y oportunidades de mejora planteadas [5 mins].</li> <li>4. Para las 3 más votadas discutir un plan de acción y asignar un responsable [10 mins].</li> </ol>	<p>Cada dos semanas, el día jueves</p>	<p>Online</p>	<p>22:30</p>	<p>40 mins.</p>
<p>#reunión-social</p> <p>Reunión para estar todos juntos, reírnos un rato, y charlar en líneas generales del proyecto.</p>	<p>Libre.</p>	<p>Una vez por mes, el primer sábado del mes</p>	<p>Presencial</p>	<p>21:00</p>	<p>1 hora</p>
<p>#reunión-tutor</p> <p>Revisar el avance con la tutora.</p>	<ol style="list-style-type: none"> <li>1. Mostrarle las tareas que están bloqueadas y Mariel puede ayudarnos a desbloquearlas.</li> <li>2. Mostrarle los objetivos planteados por el equipo, como los empezamos a resolver,</li> </ol>	<p>Los martes cada 2 semanas</p> <p>(Se desarrollará cuando el equipo y/o el tutor</p>	<p>Cantina ORT</p>	<p>18:30</p>	<p>19:30</p>

	validar si vamos por buen camino, y dudas en caso de haber alguna.	considere necesario)			
--	--	----------------------	--	--	--

Tabla 28: Listado de reuniones

### Calendario de reuniones fijas

Horario	Martes	Jueves	Sábado
18.30 a 19.30	#reunión-tutor		
21.00 a 22.30			#reunión-social
22.30 a 23.10		#reunión-retrospectiva	
23.00 a 23.40		#reunión-alineamiento	
23.30 a 00.10		#reunión-planificación	

Tabla 29: Calendario de reuniones

### Herramientas

- Google Calendar
- Google Hangouts (móvil y web)
- Slack

### Estimaciones

Las estimaremos utilizando la técnica shirts, siendo S - Pequeño, M - Mediano, L - Largo, y XL - Muy largo.

Cuando una tarea es XL se entiende que es demasiado grande y por ende la dividiremos en tareas más chicas utilizando poker planning.

Para poder tomar métricas es necesario hacer el equivalente en horas de trabajo que se definirá como la siguiente:

- Un día de trabajo equivale a 3 horas.
- Tareas S -> 1 a 3 días -> 1 a 9 horas
- Tareas M -> 4 a 7 días -> 10 a 21 horas
- Tareas L -> 8 a 14 días -> 22 a 42 horas
- Tareas XL -> más de 14 días -> más de 43 horas

### Métricas

Las siguientes métricas se utilizarán como apoyo al resto de las métricas del proyecto con el fin de ayudar a evaluar el esfuerzo del equipo por iteración

Métrica	Valor esperado	Valor real	Responsable	Obtención de datos
Cumplimiento de los objetivos por iteración, individuales y grupales.	70 / 100	x / 100	Dan	Los datos se sacarán del canal #reunión-alineamiento de Slack
Dedicación de horas por semana, individuales y grupales.	15	x	Agustín	Los datos se sacarán de click-up
Desviación entre esfuerzo estimado y esfuerzo real en la ingeniería de <i>software</i>	horas dedicadas / horas estimadas <= 1.3	x / y	Gastón	Los datos se sacarán de click-up

Tabla 30: Listado de métricas






El responsable de la métrica tiene como principal tarea la de asegurarse de que los datos medidos están actualizados y que la métrica esté en buen estado.












### Asignación de responsabilidades






- Calendar + imagen en WhatsApp + ejecución de reuniones (Victoria)
- Reuniones sociales (Gastón)
- Documento de arquitectura (Dan)
- Documento final (Victoria)
- Documento de metodología (Agustín)
- Métricas (Dan, Agustín y Gastón)
- Organización Slack (Agustín)
- Presentaciones PPT (Emiliano)
- Cuenta GitHub (Agustín)
- Carpeta documentación (Gastón)

## Anexo 11: Empresas de drones investigadas

Se muestra a continuación la lista de las empresas investigadas a la hora de elegir el dron.

Empresa	Logo	Página web
Accsys		<a href="http://www.accsys.mx">www.accsys.mx</a>
Action Drone USA		<a href="http://www.actiondroneusa.com">www.actiondroneusa.com</a>
AgEagle		<a href="http://www.ageagle.co">www.ageagle.co</a>
AguaDrone		<a href="http://www.aguadrone.com">www.aguadrone.com</a>
Ambarella		<a href="http://www.ambarella.com">www.ambarella.com</a>
Autel Robotics		<a href="http://www.auteldrones.com">www.auteldrones.com</a>
Azur Drones		<a href="http://www.azurdrones.com">www.azurdrones.com</a>

Blade		<a href="http://www.horizonhobby.com/content/bladehelis">www.horizonhobby.com/content/bladehelis</a>
Cheerson		No tiene
Delair		<a href="http://www.delair.aero">www.delair.aero</a>
DJI		<a href="http://www.dji.com">www.dji.com</a>
Draganfly Innovations		<a href="http://www.draganfly.com">www.draganfly.com</a>
EHANG		<a href="http://www.ehang.com">www.ehang.com</a>
Embention		<a href="http://www.embention.com">www.embention.com</a>
FLIR		<a href="http://www.flir.com">www.flir.com</a>
Flyability		<a href="http://www.flyability.com">www.flyability.com</a>
Freefly		<a href="http://www.freeflysystems.com">www.freeflysystems.com</a>
Hubsan		<a href="http://www.hubsan.com">www.hubsan.com</a>

Impossible Aerospace		<a href="http://www.impossible.aero/">www.impossible.aero/</a>
Insitu		<a href="http://www.insitu.com">www.insitu.com</a>
Kespry		<a href="http://www.kespry.com">www.kespry.com</a>
Parazero		<a href="http://www.parazero.com">www.parazero.com</a>
Parrot		<a href="http://www.parrot.com">www.parrot.com</a>
Pixpro		<a href="http://www.pix-pro.com">/www.pix-pro.com</a>
Planck Aerosystems		<a href="http://www.planckaero.com">www.planckaero.com</a>
senseFly		<a href="http://www.sensefly.com">www.sensefly.com</a>
Skycatch's		<a href="http://www.skycatch.com">www.skycatch.com</a>
Skydio		<a href="http://www.facebook.com/skudio">www.facebook.com/skudio</a>

Syma Toys		<a href="http://www.symatoys.com">www.symatoys.com</a>
Wingtra		<a href="http://www.wingtra.com">www.wingtra.com</a>
Workswell		<a href="http://www.workswell-thermal-camera.com">www.workswell-thermal-camera.com</a>
Yuneec		<a href="http://www.yuneec.com">www.yuneec.com</a>

Tabla 31: Listado de empresas de drones investigadas