



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática
Sistema de Navegación Semiautónomo en
Interiores



Presentado por Mario Bartolomé Manovel
en Universidad de Burgos — 2 de junio de 2018

Tutores: Dr. Alejandro Merino Gómez
Dr. César Ignacio García Osorio
Dr. José Francisco Díez Pastor

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	11
Apéndice B Especificación de Requisitos	19
B.1. Introducción	19
B.2. Objetivos generales	19
B.3. Catalogo de requisitos	19
B.4. Especificación de requisitos	21
Apéndice C Especificación de diseño	31
C.1. Introducción	31
C.2. Diseño de datos	31
C.3. Diseño procedimental	31
C.4. Diseño arquitectónico	31
Apéndice D Documentación técnica de programación	33
D.1. Introducción	33
D.2. Estructura de directorios	33
D.3. Manual del programador	33

D.4. Compilación, instalación y ejecución del proyecto	33
D.5. Pruebas del sistema	33
Apéndice E Documentación de usuario	35
E.1. Introducción	35
E.2. Requisitos de usuarios	35
E.3. Instalación	35
E.4. Manual del usuario	35
Bibliografía	37

Índice de figuras

A.1. Burndown Sprint 1	3
A.2. Burndown Sprint 2	4
A.3. Burndown Sprint 3	5
A.4. Burndown Sprint 4	5
A.5. Burndown Sprint 5	6
A.6. Burndown Sprint 6	7
A.7. Burndown Sprint 7-8	8
A.8. Burndown Sprint 9	8
A.9. Burndown Sprint 11	10
A.10. Burndown Sprint 12	11
B.1. Diagrama de Casos de Uso	22

Índice de tablas

A.1. Costes de personal	11
A.2. Costes de hardware	12
A.3. Costes de software	13
A.4. Costes variados	13
A.5. Costes totales del proyecto	13
A.6. Monetización del proyecto.	14
A.7. Dependencias en backend.	15
A.8. Dependencias en frontend.	16
A.9. Resumen de licencias.	17
B.1. CU-01. Acceso al sistema	23
B.2. CU-02. Cierre de sesión	24
B.3. CU-03. Gestión del sistema	25
B.4. CU-04. Gestión del streaming de vídeo	26
B.5. CU-05. Grabación de vídeo	26
B.6. CU-06. Control manual	27
B.7. CU-07. Gestión de controles automáticos	28
B.8. CU-08. Gestión del backend.	29

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La planificación es, sin duda, una parte fundamental en el ciclo de vida de un proyecto. Con el uso de metodologías ágiles para el seguimiento del proyecto, la planificación inicial se vuelve voluble y cambiante, de forma que la mayor parte de aproximaciones serán precisamente eso, aproximaciones.

En este anexo se detallan los recursos necesarios para el desarrollo del proyecto mediante un estudio de la planificación temporal, y un estudio de la viabilidad del mismo.

El primer apartado, **planificación temporal**, establece un calendario de estimaciones. Cabe destacar que dicho calendario es totalmente flexible. Si que se ha mantenido la estructura habitual estableciendo una fecha de inicio y una fecha de finalización, y se han tenido en cuenta la duración de las tareas, pero estas son extendibles entre diferentes fases, o *sprints*.

El segundo apartado, **estudio de viabilidad**, detalla la viabilidad del proyecto a través de dos subapartados. El primero detallará la viabilidad económica del mismo, teniendo en cuenta los costes y beneficios previstos. El segundo apartado, relacionado con la viabilidad legal, detallará el contexto legal en el que se regula este proyecto.

A.2. Planificación temporal

Para la realización de este proyecto se ha seguido una metodología ágil basada en Scrum. Dicha metodología aporta flexibilidad, muy necesaria durante el desarrollo software. Si bien esta metodología no se ha seguido por completo, ya que el proyecto no ha sido desarrollado por un equipo grande, sí que se ha seguido en esencia la forma que la metodología propone:

- El desarrollo se ha llevado a cabo mediante iteraciones, llamadas *sprints* y revisiones de las mismas.
- La duración de los *sprints* fue de dos semanas. Se consideró hacerlos de una única semana, pero la dificultad del desarrollo de los algoritmos, y sobre todo el amplio estudio que estos requerían, hizo que nos decantásemos por una duración bisemanal.
- Al finalizar cada *sprint* se procedía a realizar una reunión, en la que se comentaba el estado del desarrollo, y se planificaba la siguiente iteración.
- Durante las reuniones de revisión/planificación, se establecían las tareas a realizar en el siguiente *sprint*.
- La monitorización del proyecto se realiza mediante gráficos *burndown*, aunque no son especialmente útiles en fases iniciales, donde el número de *issues* no es alta, y su finalización es muy variable.

Se ha realizado una valoración de la duración de cada *issue* haciendo uso de *story points*, que son una manera de aproximar la duración máxima de las mismas. La estimación temporal dada a los *story points* se ha realizado como una correspondencia a días, es decir, una tarea marcada con 2 *story points* se estima que finalizará al cabo de 2 días de trabajo.

De esta forma se flexibiliza todavía más la metodología, a riesgo de perder precisión. Al no darse una correspondencia absoluta entre los *story points* asignados y la duración, se puede relegar una tarea poco importante para finalizarla en el número de días asignados. Desde luego una tarea sencilla se podrá finalizar antes, y una compleja después, dependiendo de la dedicación, en horas, dedicada a ella.

Sprint 1. 06/11/17 - 20/11/17

Este primer *sprint* fue una toma de contacto, y realizado antes del comienzo del semestre, con la intención de tener claro el desarrollo inicial del proyecto. En él se establecieron los requisitos del mismo, y los diferentes algoritmos a implementar. La descomposición de las tareas puede ser consultada en [Sprint 1](#)



Figura A.1: Gráfico burndown.

Como puede verse en la figura [A.1](#), se relegó el cierre de estas tareas de documentación hasta el final del *sprint*. Se estimaron 4 *story points*, pero se dedicaron al menos 8 días a la búsqueda de información. Dada la carga de trabajo al final del semestre no se pudo dedicar más tiempo al mismo.

Sprint 2. 20/11/17 - 04/12/17

Este segundo *sprint* fue dedicado a continuar con la obtención de información, y realizado antes del comienzo del semestre. El Dr. Alejandro Merino Gómez sugirió la utilización de algoritmos de Campos Potenciales para implementar el sistema de evasión de obstáculos, y proveyó de mucha información relacionada con el mismo. La descomposición de las tareas puede ser consultada en [Sprint 2](#).

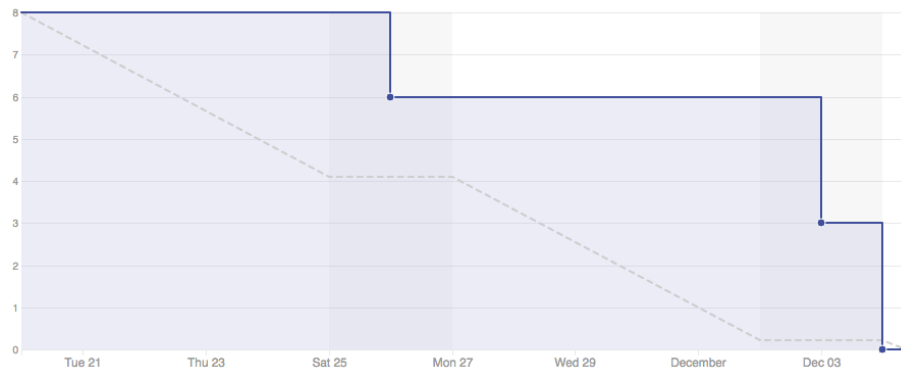


Figura A.2: Gráfico burndown.

Como puede verse en la figura A.2, este *sprint* si que siguió con mayor precisión la forma ideal del gráfico *burndown*. Se estimaron 10 *story points*, pero se dedicaron 9 días a la búsqueda de información. Dada la carga de trabajo al final del semestre no se pudo dedicar más tiempo al mismo.

Sprint 3. 4/12/17 - 15/01/18

Este tercer *sprint*, más largo por la existencia de las vacaciones de navidad, fue dedicado a continuar con la obtención de información, y realizado antes del comienzo del semestre. El Dr. Alejandro Merino Gómez sugirió la búsqueda de algún sistema operativo de tiempo real, en previsión de que el sistema operativo de la RaspberryPi no fuera lo suficientemente rápido en sus cambios de contexto. El Dr. José Francisco Díez Pastor proporcionó información sobre búsqueda de rutas. La descomposición de las tareas puede ser consultada en [Sprint 3](#).

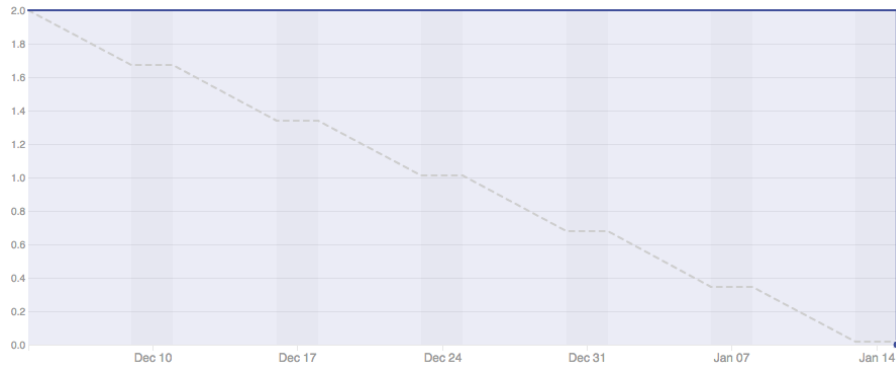


Figura A.3: Gráfico burndown.

Como puede verse en la figura A.3, este *sprint* no siguió la forma ideal del gráfico *burndown*, dadas las pocas tareas creadas. Se estimaron 2 *story points*, pero se dedicaron al menos 10 días a la búsqueda de información. Dada la carga de trabajo al final del semestre, y el periodo vacacional, no se pudo dedicar más tiempo al mismo.

Sprint 4. 17/01/18 - 31/01/18

Este cuarto *sprint*, fue dedicado a conseguir la comunicación entre la RaspberryPi y el drone. Se creó la primera implementación del algoritmo de Campos Potenciales, posteriormente sustituido por *VFH*. En este caso, durante la preparación del *sprint*, expliqué a los tutores las diferentes formas de comunicación que se podían valorar. La descomposición de las tareas puede ser consultada en [Sprint 4](#).



Figura A.4: Gráfico burndown.

Como puede verse en la figura A.4, este *sprint* se ajustó bastante bien a la forma ideal del gráfico *burndown* dada la cantidad de tareas. Se estimaron 9 *story points*, pero se dedicaron al menos 12 días a la consecución del protocolo implementado.

Sprint 5. 31/01/18 - 14/02/18

Este quinto *sprint*, fue dedicado a conseguir la emisión de vídeo desde la RaspberryPi. Durante la preparación de este *sprint*, el Dr. César Ignacio García Osorio sugirió realizar un *refactor* de algunas partes del código ya desarrollado para mejorar su legibilidad y mantenibilidad. La descomposición de las tareas puede ser consultada en [Sprint 5](#).

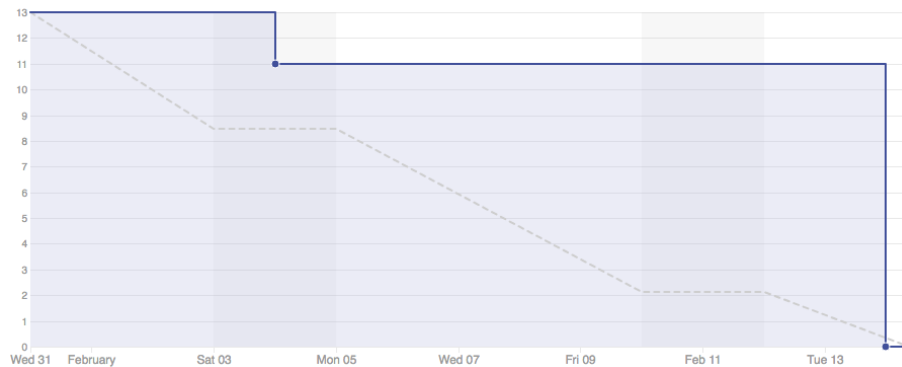


Figura A.5: Gráfico burndown.

Como puede verse en la figura A.5, este *sprint* no se ajustó demasiado a la forma ideal del gráfico *burndown* dada la longitud de algunas de las tareas creadas. Se estimaron 14 *story points*, y se dedicaron 14 días a lograr los objetivos marcados.

Sprint 6. 14/02/18 - 08/03/18

Este sexto *sprint*, fue dedicado a realizar una implementación que sustituyese al algoritmo de Campos Potenciales. Durante la preparación de este *sprint*, se propuso crear otro algoritmo, dados los problemas que presentaba el algoritmo de Campos Potenciales. El Dr. César Ignacio García Osorio mostró preocupación por las latencias existentes en el feed de vídeo proveniente de la RaspberryPi, así que también se comprobó el código, y se buscaron formas de lograr que este fuese más fluido. La descomposición de las tareas puede ser consultada en [Sprint 6](#).



Figura A.6: Gráfico burndown.

Como puede verse en la figura A.6, este *sprint* no se ajustó a la forma ideal del gráfico *burndown* dada la existencia de una única tarea, y su longitud. Se estimaron 8 *story points*, pero realmente se dedicaron 14 días a lograr los objetivos marcados.

Sprint 7-8. 08/03/18 - 12/04/18

Este séptimo-octavo *sprint* se creó con longitud doble, debido a la necesidad de generar una primera release y la cantidad de trabajo requerido para ella. Fue dedicado a finalizar la implementación del VFH, que sustituyese al algoritmo de Campos Potenciales. Se creó una página web que permitía a un usuario iniciar sesión y visualizar el vídeo proveniente del drone asignado, así como su control remoto. Durante este *sprint* se implementó la parte central de la arquitectura de este proyecto.

La descomposición de las tareas puede ser consultada en [Sprint 7-8](#).

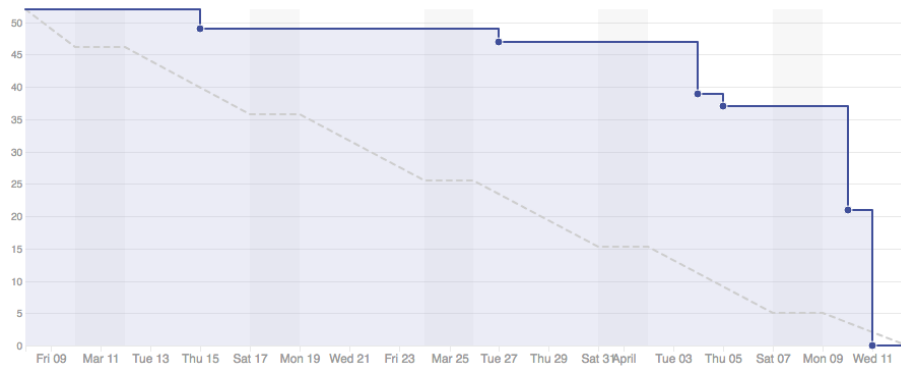


Figura A.7: Gráfico burndown.

Como puede verse en la figura A.7, este *sprint* no se ajustó a la forma ideal del gráfico *burndown* dada la longitud de las tareas finales, que provocan que el escalado del gráfico no sea el apropiado. Se estimaron 52 *story points*, pero realmente se dedicaron 14 días a lograr los objetivos marcados. Obviamente, existió solapamiento de tareas. La release se finalizó correctamente y está disponible para su uso como versión *alpha*.

Sprint 9. 12/04/18 - 26/04/18

Este noveno *sprint* fue dedicado a comenzar con la implementación del Filtro de Partículas, y crear los scripts de instalación de los que carecía la primera release.

La descomposición de las tareas puede ser consultada en [Sprint 9](#).



Figura A.8: Gráfico burndown.

Como puede verse en la figura [A.7](#), este *sprint* se ajustó bastante bien a la forma ideal del gráfico *burndown*. Sin embargo al final del mismo, se aprecia la desaparición de tareas. Esto es debido a pasar la implementación del Filtro de Partículas al siguiente *sprint*. No existe una forma de *extender* a duración de una tarea entre *sprints* de forma que se presente en ambos. Se estimaron 6 *story points*, pero realmente se dedicaron 14 días a lograr los objetivos marcados, además del comenzar con el desarrollo del Filtro de Partículas. En un principio se estimó la duración del desarrollo del Filtro de Partículas en 8 *story points*.

Sprint 10. 26/04/18 - 11/05/18

Este décimo *sprint* fue dedicado a la implementación del Filtro de Partículas.

Tal y como se ha explicado con anterioridad, no existe una forma de establecer la misma *issue* en varios *sprints*, de manera que el gráfico *burndown* de este periodo de tiempo se encuentra vacío. El Filtro de Partículas no se finalizó durante el *sprint*, y por lo tanto fue extendido al siguiente.

La duración del desarrollo del Filtro de Partículas se extendió a 21 *story points*.

Sprint 11. 11/05/18 - 25/05/18

Este undécimo *sprint* fue dedicado a continuar con la implementación del Filtro de Partículas, y establecer el hardware restante en el drone. Además, se implementaron los controladores PID necesarios para la segunda release del proyecto.

La descomposición de las tareas puede ser consultada en [Sprint 11](#).



Figura A.9: Gráfico burndown.

Como puede verse en la figura A.9, este *sprint* se ajustó muy bien a la forma ideal del gráfico *burndown*, dada la gran cantidad de tareas creadas. Sin embargo al final del mismo, se aprecia la desaparición de tareas. Esto es debido a pasar la implementación del Filtro de Partículas al siguiente *sprint*, debido a la necesidad de probarlo correctamente. No existe una forma de *extender* a duración de una tarea entre *sprints* de forma que se presente en ambos. Se estimaron 27 *story points*, pero realmente se dedicaron 14 días a lograr los objetivos marcados, además del continuar con el desarrollo del Filtro de Partículas.

Sprint 12. 25/05/18 - 8/06/18

Este duodécimo *sprint* está siendo dedicado a finalizar la documentación del proyecto, y a las pruebas en un entorno real.

La descomposición de las tareas puede ser consultada en [Sprint 12](#), aunque no se encuentra cerrado ni finalizado.

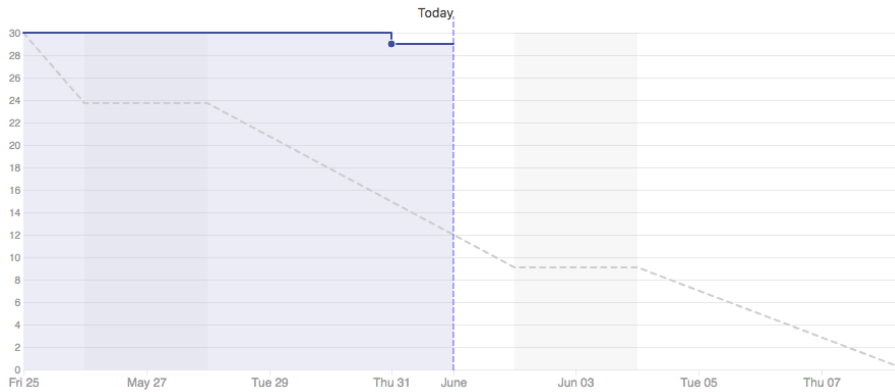


Figura A.10: Gráfico burndown.

Por el momento se han estimado 30 *story points*, entre los que se cuentan los 21 del Filtro de Partículas, y 8 de ellos se corresponden con la adquisición de métricas de SonarCloud.

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se detallan los costes y beneficios que podría suponer el proyecto, de desarrollarse con una intención comercial.

Costes

De personal: Para el desarrollo del proyecto tan solo se ha empleado un desarrollador, empleado a tiempo completo durante cuatro meses, de febrero a mayo.

Concepto	Coste
Salario mensual neto	1102€
Retención IRPF (13.55 %)	271€
Seguridad Social (29.9 %)	598€
Salario mensual bruto	2000€
Total 4 meses	8000€

Tabla A.1: Costes de personal

La retribución a la Seguridad Social se ha calculado como un 23,60 % por contingencias comunes, más un 5,50 % por desempleo de tipo general, más un 0,20 % para el Fondo de Garantía Salarial y má un 0,60 % de formación profesional. En total un 29,9 % que se aplica al salario bruto. Ver [1].

De hardware: Para el desarrollo del proyecto se han empleado numerosos elementos de hardware que se detallan a continuación:

Concepto	Coste	coste amortizado
Ordenador portátil	800€	66,67€
Emisora	177,79€	11,86€
Drone	Desglose	
RaspberryPi 3B	33,74€	2,25€
Tarjeta de memoria	21,99€	1,47€
Carcasa RaspberryPi	10,99€	0,74€
Cámara PiNoIR	28,99€	1,94€
LED IR	9,99€	0,67€
Controladora de vuelo	38,99€	2,6€
Motores EMAX mt2216	47,84€	3,19e
ESC 20A EMAX	30,53€	2,04€
Hélices de repuesto	10,34€	0,69€
Cargador Baterías	18,63€	1,25€
Sensores Ultrasonidos HCSR04	8,79€	0,59€
Chasis drone con PCB	19,99€	1,34€
Tren de aterrizaje	5,89€	0,4€
Cables	7,09€	0,48€
Tornillos	17,15	1,15€
Hilo de estaño	9,9€	0,67€
Soldador JBC 11W	41,95€	2,8€
Total	1340,58€	89,38€

Tabla A.2: Costes de hardware

De software: Para el desarrollo de este proyecto, se valoran los costes del siguiente software, cuya amortización se ha establecido en 2 años.

Concepto	Coste	coste amortizado
PyCharm IDE	79,6€	13,27€
WebStorm IDE	51,6€	8,6€
GitHub Privado	23,43€	3,91€
SonarCloud Privado	40€	6,67€
Total	194,63€	32,45€

Tabla A.3: Costes de software

Varios: Los siguientes, son costes variados:

Concepto	Coste
Alquiler de oficina	600€
Internet	94,4€
Alquiler nave de pruebas	600€
Total	1294,4€

Tabla A.4: Costes variados

Totales: El total de los costes es el siguiente:

Concepto	Coste
Personal	8000 €
Hardware	1340,58 €
Software	194,63 €
Varios	1294,4 €
Total	10829,61€

Tabla A.5: Costes totales del proyecto

Beneficios

Por el momento, se trata de un proyecto con fines académicos, y con unos gastos relativamente elevados para la etapa tan temprana en que se encuentra.

Sin embargo, los beneficios podrían ser muy elevados. Se va a considerar un sistema de suscripción anual al servicio de vigilancia mediante drones. Dicho servicio proporciona vigilancia mediante drones, los cuales estarán supervisados por personal humano, y su mantenimiento. El servicio se detalla en la siguiente tabla:

Entorno	Dimensiones	Drones	Coste
Nave	$> 2000m^2$	1	8000 €/año
Nave grande	$> 4000m^2$	2	15000€/año
Grupo de naves	$> 15000m^2$	4	28000€/año

Tabla A.6: Monetización del proyecto.

La finalidad de establecer las dimensiones y el número de drones, se corresponde con la necesidad de realizar una vigilancia adecuada de un recinto de dimensiones determinadas. Es decir, no es lo mismo cubrir una superficie de $1000m^2$ que una de $2000m^2$, por supuesto teniendo en cuenta la distribución interior de las mismas. El último elemento de la tabla A.6, se corresponde con un grupo de naves ubicadas en el mismo recinto, de forma que los drones tendrían acceso a cada una de las naves, y a un canal de tránsito entre ellas.

Los costes serían rentables para las empresas, dado que contratar personal de seguridad acorde a la dimensionalidad del negocio es más costoso que mantener los drones establecidos en la tabla A.6.

Responsabilidad Social

Con el auge de la mecanización y automatización de los puestos de trabajo, es obvio que los seres humanos iremos perdiendo cabida en empleos que un sistema informatizado podría realizar con mayor eficiencia, y sobre todo de forma más económica.

Este proyecto, tiene un gran potencial de causar un daño social importante. El puesto de vigilante nocturno ha provisto de seguridad en muchos entornos laborales, y ha contribuido a generar riqueza protegiendo los bienes de las diferentes empresas que hacen uso de este personal.

Por tanto, si este proyecto avanza y llega a ser utilizado en sustitución de personal humano, se propone impartir formación en pilotaje de este tipo

de drones a los trabajadores afectados, para que puedan pasar a formar parte de la empresa. De esta forma, el efecto de esta tecnología supondrá una mejora en sus condiciones de trabajo y una evolución en materia de seguridad, tanto para las empresas como para estos trabajadores.

Viabilidad legal

En esta sección se aborda el contexto legal en el que se encuentra el proyecto, así como todo lo relacionado con las licencias.

Dado que el desarrollo de casi la totalidad de las implementaciones de este proyecto ha sido propia, esta sección será breve en lo que a licencias se refiere.

Software

El proyecto consta de las siguientes dependencias software:

Dependencia	Versión	Descripción	Licencia
Numpy	1.14.2	Se trata de la librería de cálculo científico por excelencia en Python.	BSD
SciPy	1.0.1	Librería de múltiples herramientas para cálculo científico.	BSD
PySerial	3.4	Librería que proporciona acceso a puertos serie.	BSD
Bluetin_Echo	0.1.1	Librería que proporciona una clase para usar los sensores de ultrasonidos HC-SR04.	BSD
UV4L	1.14	Conjunto de drivers para captura de vídeo, y framework WebRTC basado en OpenWebRTC.	BSD

Tabla A.7: Dependencias en backend.

Dependencia	Versión	Descripción	Licencia
Flask	1.0	Se trata de un framework para desarrollo de aplicaciones web en Python.	BSD
Eventlet	0.22.1	Librería de comunicación concurrente para Python.	MIT
SQLAlchemy	1.2.6	ORM para Python.	MIT

Tabla A.8: Dependencias en frontend.

De esta forma, el código desarrollado deberá tener una licencia compatible con MIT y BSD, ambas muy permisivas al respecto. La monetización del proyecto se basa en un sistema de suscripción al mismo, pero la existencia de competencia redundante en una mejora de los productos y servicios, ya que fuerza a las empresas a mantenerse al día. Por tanto, se libera todo el código, tanto del backend como del frontend, bajo licencia BSD de 3 cláusulas.

Puede leerse un resumen de la licencia en [BSD-3 Clause license](#)

Documentación, imágenes y vídeos

La documentación generada, así como las imágenes utilizadas (salvo dos presentadas en artículos académicos listados en la bibliografía, y sobre las que se ha aportado el crédito correspondiente) y los vídeos han sido de generación propia.

Por lo tanto se establece una licencia *Creative Commons Attribution 4.0 International* (CC-BY-4.0), que permite compartir, adaptar y distribuir para cualquier propósito la obra, incluso con fines comerciales, siempre y cuando se de la debida atribución de la autoría del trabajo.

Puede leerse un resumen de la licencia en [CC-BY 4.0](#).

Resumen

A modo de resumen de las licencias dadas al material del proyecto, se añade la siguiente tabla:

Recurso	Licencia
Código Backend	BSD
Código Frontend	BSD
Documentación	CC-BY-4.0
Imágenes	CC-BY-4.0
Videos	CC-BY-4.0

Tabla A.9: Resumen de licencias.

Marco legal

Más allá de tratarse de únicamente un proyecto software, el uso de drones entra en un marco legal muy delicado en España.

La legislación al respecto es de reciente creación, y difiere entre países. A continuación se especifica el marco legal en el que se engloba este proyecto, y las diferentes repercusiones que tiene sobre el uso del mismo.

Legislación sobre Aeronaves

Según el Real Decreto 1036/2017, disponible en [BOE - RD1036/2017](#):

- Para operar un dron **de forma profesional**, es necesario ser dado de alta por AESA¹ como operador de drones.
- No está permitido sobrepasar los 120m de altura.
- Se permite el vuelo sobre aglomeraciones de personas y en zonas urbanas, siempre y cuando la aeronave no sobrepase los 10 kg. Se debe guardar una distancia de seguridad de 50 m con edificios y personas. El piloto no podrá encontrarse a más de 100 m de distancia.
- Se permite el **vuelo fuera del alcance visual** con aeronaves de menos de 2 kg, y que cuenten con medios para detectar y evitar a otros usuarios del espacio aéreo.
- La aeronave deberá contar con una placa identificativa, que conste de nombre del fabricante, tipo, modelo, numero de serie, y nombre del operador y datos de contacto.

¹Agencia Estatal de Seguridad Aérea

Dado el marco establecido en la anterior lista, y dado que el drone creado no sobrepasa los 2 kg de peso, tan solo sería necesario contar con personal cualificado, y con una placa identificativa del dispositivo. Además, tal y como se estipula, se hace referencia a drones con medios para evitar colisiones, lo cual enmarca este proyecto.

Pese a no mentarse el uso en recintos privados, y acogiendo el proyecto a la normativa más restrictiva, parece sencillo y razonable cumplir con la legislación vigente.

GDPR

La reciente implantación de la **GDPR**², lleva consigo una serie de restricciones y notificaciones a tener en cuenta:

- Se debe señalar públicamente que la zona cuenta con sistemas de vídeo vigilancia.
- Asimismo se debe informar que los datos pueden ser utilizados con fines de mejora del producto.
- Se informará del no tratamiento de datos biométricos recogidos durante sesiones de grabación.
- Se informará de los derechos de cancelación y de acceso a los datos en cualquier momento.
- La transmisión de vídeo se hará de forma cifrada (es requisito de WebRTC, de manera que no es complejo de cumplir), bien mediante HTTPS o tuneles TLS.
- Derecho al olvido. Permite que una persona captada por el sistema de vigilancia pueda ejercer su derecho a que los datos sean eliminados, de forma que se detenga su procesamiento o compartición.

²Reglamento General de Protección de Datos de la Unión Europea

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este anexo se recogen las especificaciones de requisitos requeridas para este proyecto. Estas definirán el comportamiento esperado del sistema.

B.2. Objetivos generales

En este proyecto se ha tratado de llevar a cabo los siguientes objetivos de carácter general:

- Diseñar un drone capaz de recorrer un espacio, en el que existan obstáculos, de forma segura.
- Diseñar un sistema de acceso al drone de forma segura. Tanto para controlarlo de forma remota, como para activa los mecanismos de control automatizados.
- Diseñar una interfaz web que permita la visualización en tiempo real de la cámara del drone, así como su control remoto por un operador.

B.3. Catalogo de requisitos

Los siguientes requisitos se derivan de los objetivos generales arriba dispuestos:

Requisitos Funcionales

RF-1 Acceso al sistema: El usuario debe poder iniciar sesión en el sistema.

RF-2 Cierre de sesión: El usuario debe poder cerrar sesión en el sistema.

RF-3 Gestión del sistema: El usuario debe ser capaz de gestionar el sistema completo que hay a bordo del drone.

RF-3.1 Gestión del streaming de vídeo: El usuario debe poder iniciar/detener el streaming de vídeo.

RF-3.2 Grabaciones de vídeo: El usuario debe poder iniciar/detener y descargar las grabaciones de vídeo.

RF-3.3 Gestión de controles del drone: El usuario debe poder activar/desactivar el control manual del dispositivo.

RF-3.3.1 Seguridad automatizada: El drone debe presentar mecanismos de autopreservación. Debe protegerse frente a obstáculos.

RF-3.4 Gestión de control automático del drone: El usuario debe poder activar/desactivar los mecanismos de control automatizado.

RF-3.5 Registro de acciones: Las acciones llevadas a cabo por el usuario deben guardarse en un registro.

RF-4 Gestión del backend: El usuario debe poder acceder a la equipo que controla el backend.

RF-4.1 Gestión del sistema operativo: El usuario debe poder llevar a cabo las tareas de gestión del sistema operativo como crea conveniente.

RF-5 Registro de sucesos: Intentos de intrusión deben ser registrados.

RF-6 Reconocimiento de dispositivos: La aplicación web debe ser capaz de reconocer un dispositivo (joystick o emisora) conectado al equipo, para llevar a cabo el control del drone.

Requisitos no funcionales

RNF-1 Usabilidad: El frontend ha de ser sencillo de utilizar e intuitiva. En el caso de producirse errores, estos deben ser claramente explicados, o registrados.

- RNF-2** Rendimiento: Tanto el frontend, como el backend deben funcionar de forma eficiente. Retardos elevados podrían dar lugar a problemas muy serios en la ejecución de las ordenes del usuario.
- RNF-3** Capacidad y Escalabilidad: Tanto el frontend como el backend deben ser escalables. El frontend debe ser capaz de soportar una asiduidad grande de usuarios, así como permitir añadir múltiples agentes a cada usuario. El backend, debe permitir añadir controladores sin que haya que reestructurar todo el sistema.
- RNF-4** Disponibilidad: El sistema al completo debe ser robusto. El frontend debe estar disponible para su uso el mayor tiempo que sea posible. El backend debe ser capaz de recuperarse de errores, y determinar acciones a tomar en el caso de los mismos.
- RNF-5** Seguridad: Tanto el acceso al frontend como al backend deben hacerse de forma segura. En ningún momento el drone debe guardar imágenes de la zona vigilada. El almacenamiento de contraseñas se hará de forma cifrada, y la comunicación entre el backend y el frontend deberá ser segura.

B.4. Especificación de requisitos

Esta sección detallará el diagrama de casos de uso, y detallará cada uno de ellos.

Actores

- **Vigilante:** Actor a cargo de la gestión del sistema en producción. Interactúa con el FrontEnd.
- **Admin:** Actor a cargo de la gestión del sistema en desarrollo o en caso de resolución de problemas en producción. Interactúa con el BackEnd.

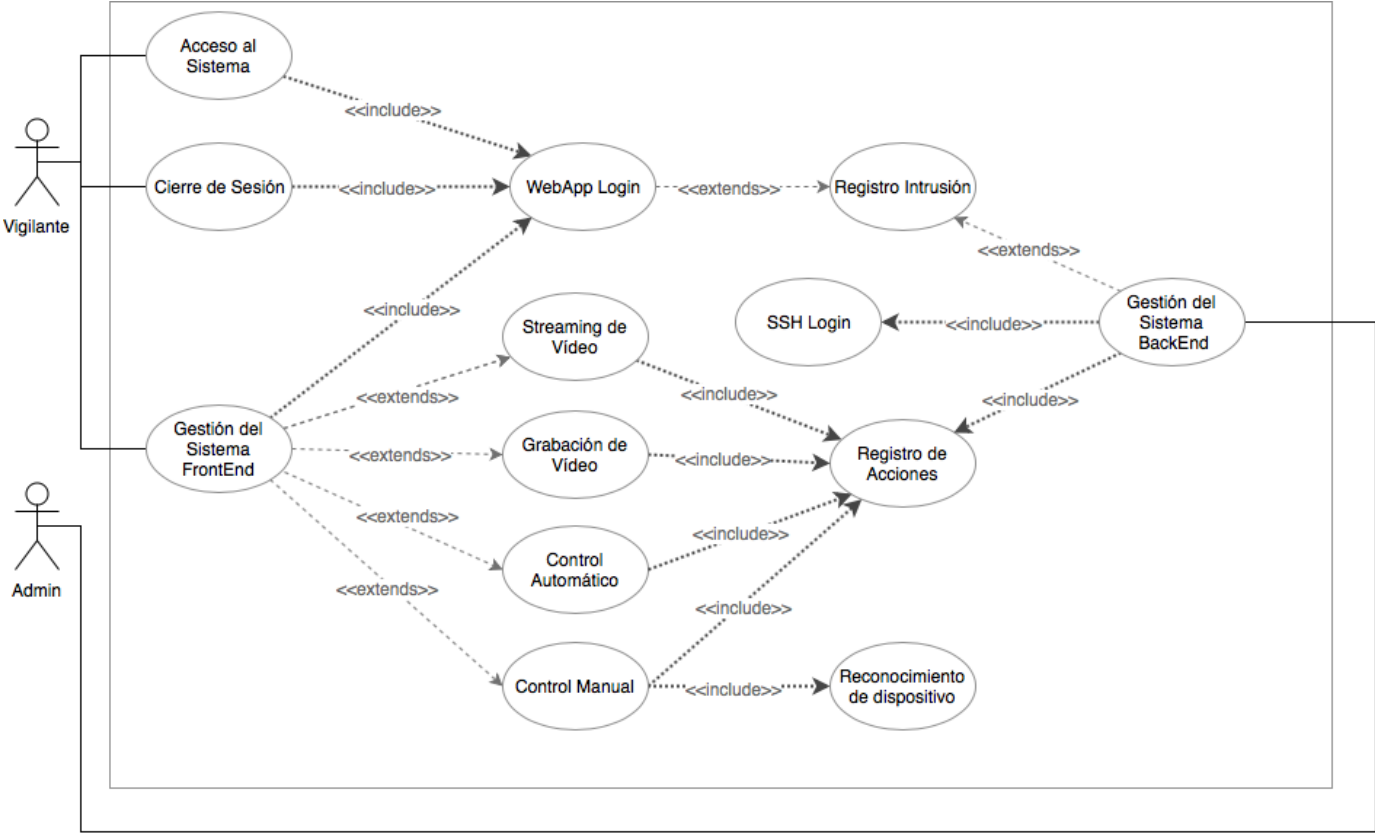


Figura B.1: Diagrama de Casos de Uso.

Casos de Uso

CU-01	Acceso al sistema
Actor	Vigilante
Requisitos asociados	RF-1, RF-5
Descripción	Permite al usuario iniciar sesión en el sistema vía web.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página web. 2. El usuario introduce sus credenciales. 3. Se muestra la página principal con el logo y los botones asociados al streaming de vídeo.
Postcondición	El sistema habrá asignado al usuario el dron correspondiente.
Excepciones	<ul style="list-style-type: none"> ■ Contraseña/Usuario incorrectos (mensaje). ■ No existen drones asociados (mensaje).
Importancia	Alta.

Tabla B.1: CU-01. Acceso al sistema

CU-02	Cierre de sesión
Actor	Vigilante
Requisitos asociados	RF-2, RF-5
Descripción	Permite al usuario cerrar sesión en el sistema vía web.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa. El usuario tiene iniciada una sesión.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación web. 2. El usuario pulsa sobre el enlace 'Logout' situado en la parte superior izquierda. 3. Se muestra la página de inicio de sesión con un mensaje de despedida.
Postcondición	El sistema habrá dado por cerrada la sesión de usuario. Se redirige al usuario a la página de inicio de sesión.
Excepciones	El usuario no tenía sesión activa. (redirección a login).
Importancia	Alta.

Tabla B.2: CU-02. Cierre de sesión

CU-03	Gestión del sistema
Actor	Vigilante
Requisitos asociados	RF-1, RF-3
Descripción	Permite al usuario gestionar las acciones del drone vía web.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa. El usuario tiene iniciada una sesión. El usuario tiene asignado un drone.
Acciones	<ol style="list-style-type: none"> 1. El usuario conecta un dispositivo con el número de canales necesarios. 2. Se informa sobre el reconocimiento de dicho dispositivo y se muestran los valores de los canales activos. 3. Se muestra el botón de activación del control manual.
Postcondición	El sistema habrá asignado al usuario el drone correspondiente. La aplicación web habrá reconocido el dispositivo conectado. Se dará la posibilidad de activar el control manual.
Excepciones	Dispositivo no reconocido.
Importancia	Alta.

Tabla B.3: CU-03. Gestión del sistema

CU-04	Gestión del streaming de vídeo
Actor	Vigilante
Requisitos asociados	RF-3, RF-3.1 RF-3.5
Descripción	Permite al usuario activar/desactivar el feed de vídeo proveniente del drone.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa. El usuario tiene iniciada una sesión. El usuario tiene asignado un drone.
Acciones	El usuario pulsa sobre el botón ‘Start/Stop Streaming’
Postcondición	La aplicación web inicia/detiene el vídeo.
Excepciones	Drone no disponible (mensaje).
Importancia	Alta.

Tabla B.4: CU-04. Gestión del streaming de vídeo

CU-05	Grabación de vídeo
Actor	Vigilante
Requisitos asociados	RF-3, RF-3.2 RF-3.5
Descripción	Permite al usuario grabar el feed de vídeo proveniente del drone.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa. El usuario tiene iniciada una sesión. El usuario tiene asignado un drone.
Acciones	El usuario pulsa sobre el botón ‘Record Video’
Postcondición	La aplicación web comienza con la grabación del feed de vídeo.
Excepciones	Feed de vídeo no disponible (mensaje).
Importancia	Baja.

Tabla B.5: CU-05. Grabación de vídeo

CU-06	Control manual del drone
Actor	Vigilante
Requisitos asociados	RF-3, RF-3.3 RF-3.5, RF-3.3.1, RF-6
Descripción	Permite al usuario controlar el drone de forma manual.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa. El usuario tiene iniciada una sesión. El usuario tiene asignado un drone. El usuario ha conectado un dispositivo adecuado.
Acciones	El usuario pulsa sobre el botón ‘Enable/Disable Manual’
Postcondición	La aplicación web solicita la activación del control manual al sistema. El sistema se conecta al drone y transmite la información proveniente de la aplicación web.
Excepciones	Drone no disponible (mensaje).
Importancia	Alta.

Tabla B.6: CU-06. Control manual

CU-07	Gestión de controles automáticos
Actor	Vigilante
Requisitos asociados	RF-3, RF-3.4 RF-3.5
Descripción	Permite al usuario activar/desactivar los mecanismos de control automatizado del drone.
Precondiciones	La base de datos está activa. El usuario dispone de cuenta activa. El usuario tiene iniciada una sesión. El usuario tiene asignado un drone.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona los mecanismos de control automatizado que desea emplear. 2. El usuario pulsa sobre el botón ‘Enable auto’
Postcondición	La aplicación web solicita la activación de los controles automatizados al sistema. El sistema se conecta al drone y transmite la información proveniente de la aplicación web.
Excepciones	<ul style="list-style-type: none"> ■ Drone no disponible (mensaje). ■ Mecanismo automatizado no reconocido (mensaje).
Importancia	Media.

Tabla B.7: CU-07. Gestión de controles automáticos

CU-08	Gestión del backend
Actor	Admin
Requisitos asociados	RF-4, RF-5
Descripción	Permite al usuario gestionar el equipo que controla el dron.
Precondiciones	El usuario dispone de credenciales de acceso al sistema.
Acciones	<ol style="list-style-type: none"> 1. El usuario inicia sesión en el sistema. 2. Se informa sobre el registro de intentos de acceso no autorizado. 3. Se muestra prompt del sistema a la espera de órdenes.
Postcondición	El sistema operativo habrá registrado el inicio de sesión del usuario. El sistema operativo habrá iniciado sesión con el usuario solicitado.
Excepciones	Acceso denegado, pubkey. (mensaje)
Importancia	Alta.

Tabla B.8: CU-08. Gestión del backend.

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Bibliografía

- [1] S. Social, “Bases y tipos de cotización 2018,” 2018, [Internet; descargado 1-junio-2018]. [Online]. Available: http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm