



INGENIA SE | CURSO 2022-2023

Informe Final de Proyecto

Equipo Hell-ix

23 de mayo de 2023



Índice general

1. Introducción y descripción del proyecto	3
1.1. Objetivos	3
1.2. Duración	3
1.3. Alcance	3
1.4. Entregables	4
2. Estructura del equipo	4
3. Distribución de tareas y ejecución del proyecto	5
4. Costes del proyecto	7
5. Resultados	8
5.1. Algoritmo planificador de trayectorias	8
5.2. Programa de seguimiento de trayectorias asistido por visión	9
5.3. Entorno de simulación de la competición	10
6. Problemas encontrados	10
7. Conclusiones y lecciones aprendidas	13

1. Introducción y descripción del proyecto

El presente documento describe el proyecto durante su ejecución hasta llegar a la construcción y operación del sistema. En primer lugar, se incluye información sobre la estructura del equipo, la distribución de las tareas y la ejecución del proyecto. A continuación se comentan los resultados obtenidos, los problemas encontrados, los costes del proyecto y la gestión de la documentación y la comunicación. Finalmente, se detallan las conclusiones obtenidas tras la finalización del proyecto, así como las lecciones aprendidas que podrían ser de utilidad para futuros proyectos.

El proyecto consiste en la concepción, diseño, implementación y operación de un sistema de una competición de drones autónomos. Se trata de una competición de drones entre dos equipos: Droning y Hell-ix. La competición comprende tres pruebas diferentes: una prueba de velocidad atravesando cuatro puertas cuadradas en línea recta, una prueba de velocidad atravesando cuatro puertas cuadradas dispuestas en un semicírculo y una prueba de freestyle.

1.1. Objetivos

El objetivo fundamental del proyecto consiste en ganar la competición frente al otro equipo. Dada la naturaleza académica del proyecto, el segundo objetivo fundamental consiste en la adquisición de competencias técnicas, personales y organizacionales por parte de los estudiantes.

1.2. Duración

El proyecto comenzó el 12 de septiembre de 2022 y terminó el 23 de mayo de 2023, teniendo de esta forma una duración aproximada de 36 semanas.

1.3. Alcance

La Figura 1.1 muestra la división del proyecto en paquetes de trabajo, siguiendo la metodología en V, típica en ingeniería de sistemas. En el esquema no se muestran los paquetes de trabajo correspondientes a la elaboración de la documentación final (8).

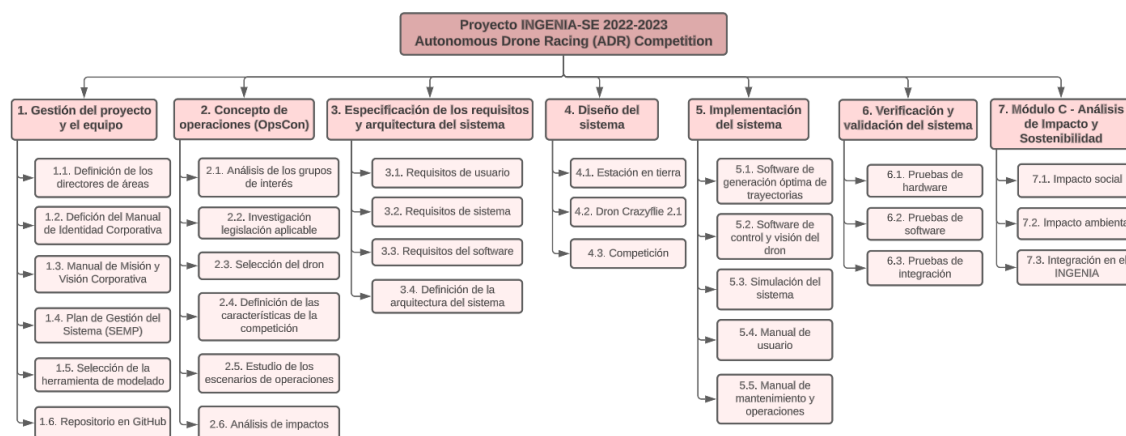


Figura 1.1: EDT del proyecto (falta el apartado 8: Elaboración de la documentación final).

1.4. Entregables

A lo largo del proyecto se han presentado numerosos entregables, como el Manual de Identidad Corporativa, la lista de directores por áreas del equipo, el Documento de Misión y Visión Corporativa, y varias presentaciones de revisión del proyecto con el cliente. Los entregables finales del proyecto comprenden: el OpsCon, el SyRS, el diseño del sistema, el modelo del sistema, la implementación del sistema, el informe de pruebas, el análisis de sostenibilidad e impacto y un póster para el evento final de INGENIA, junto con el presente documento.

2. Estructura del equipo

A continuación, en la Figura 2.1 se muestra el diagrama organizacional final del equipo de 12 personas que conforman Hell-ix junto con su cargo de dirección de un área concreta. Estos cargos de dirección han sufrido modificaciones con respecto a los inicialmente concebidos debido a diferentes motivos. Por un lado, el desarrollo del proyecto ha derivado en la necesidad de eliminar ciertos roles y añadir otros, y por otro, el grado de especialidad de cada uno de los miembros del equipo en las distintas tareas a realizar, el cual no se había analizado en tanta profundidad como debería desde un comienzo. Además, el equipo sufrió una baja al comienzo del INGENIA que hizo necesaria la retribución del cargo de “Director de Comunicación” a otra persona, en este caso al DPM.

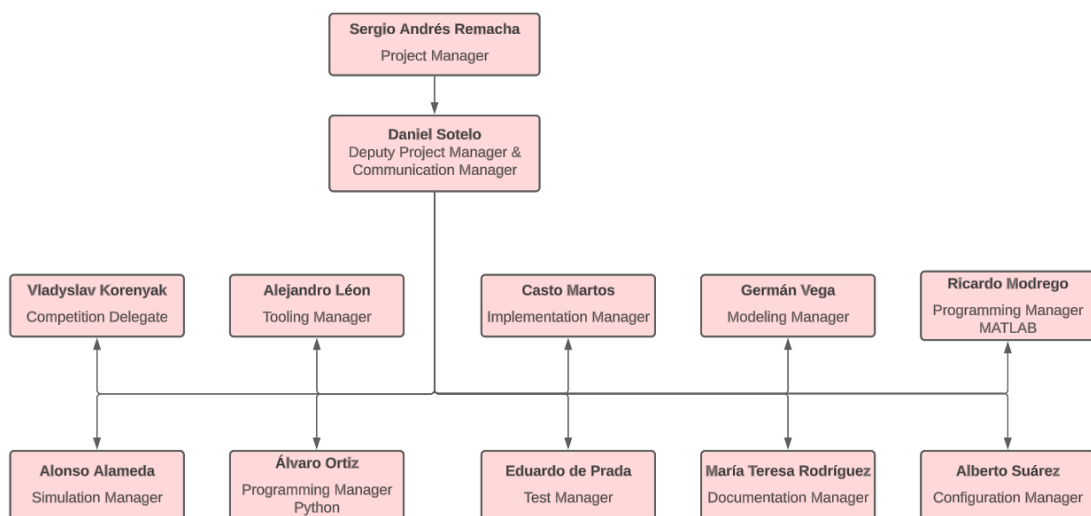


Figura 2.1: Organigrama del equipo Hell-ix.

A continuación, se resumen brevemente las responsabilidades de cada uno de los líderes de las distintas áreas. Nótese que además de dirigir estas áreas, cada uno de los miembros del equipo ha contribuido en las distintas áreas de trabajo del proyecto tal y como se muestra en la sección 3.

- **Project Manager:** Es responsable de la planificación general, coordinación y supervisión del proyecto. Establece objetivos, asigna tareas, coordina los recursos y supervisa el progreso general del proyecto.
- **Deputy Project Manager:** Apoya al PM en todas las actividades de planificación, coordinación y supervisión del proyecto.

- **Communication Manager:** Gestiona la comunicación interna y externa del equipo del proyecto, facilitando la comunicación efectiva entre los miembros del equipo y manteniendo informados a los interesados. Además, asume las tareas del responsable de adquisiciones.
- **Competition Delegate:** Actúa como enlace entre los dos equipos del proyecto y como organizador de la competición. Responsable de la comprensión de los requisitos de la competición, coordinar la participación de los equipos y asegurar el cumplimiento de las reglas.
- **Modeling Manager:** Supervisa el desarrollo e implementación del modelo. Supervisa la asignación de requisitos a la distintas arquitecturas y la trazabilidad entre las mismas. Trabaja junto con el equipo de programación para garantizar la precisión del modelo con respecto a la realidad.
- **Simulation Manager:** Supervisa el desarrollo e implementación de las simulaciones del sistema. Trabaja junto con el equipo de programación para garantizar la precisión de las simulaciones con respecto a la realidad.
- **Tooling Manager:** Se encarga de identificar y coordinar las herramientas y software necesarios para el desarrollo del sistema. Asegura que el equipo tenga acceso a las herramientas necesarias.
- **Programming Manager (Python):** Supervisa el desarrollo de software en Python. Coordina el trabajo del equipo de programación, revisa el código y asegura el cumplimiento de los plazos.
- **Programming Manager (MATLAB):** Supervisa el desarrollo de software en MATLAB. Coordina el trabajo del equipo de programación, revisa el código y asegura el cumplimiento de los plazos.
- **Implementation Manager:** Coordina la implementación del sistema. Supervisa la instalación del hardware y el software y garantiza que se solucionen los problemas relacionados con la implementación. Además elabora y revisa el manual de usuario y el manual de mantenimiento y operaciones del sistema.
- **Test Manager:** Es responsable de planificar, coordinar y ejecutar las pruebas del sistema y sus unidades. Define la política de pruebas, las supervisa, y verifica y valida el sistema.
- **Documentation Manager:** Se encarga de la gestión y organización de toda la documentación del proyecto, lo cual incluye la actualización y mantenimiento de la documentación técnica, manuales de usuario, informes y otros documentos relacionados. Garantiza además que la documentación esté completa, actualizada y sea accesible para todos los miembros del grupo.
- **Configuration Manager:** Es responsable de gestionar el control de versiones del código fuente y los recursos del proyecto en GitHub. Establece políticas de control de versiones, colabora con el equipo de programación para resolver posibles problemas de control de versiones, etc.

3. Distribución de tareas y ejecución del proyecto

A continuación, se describe en detalle el cronograma del proyecto (Diagrama de Gantt), el cual difiere bastante de la versión inicial del SEMP, debido a la gran incertidumbre inicial en la

asignación de tiempos e incluso en la propia definición de los paquetes de trabajo. Conforme el proyecto ha ido avanzando dicha incertidumbre se ha ido reduciendo, lo que ha permitido una definición más detallada de las tareas. En el diagrama se muestran las personas responsables de cada una de las tareas.

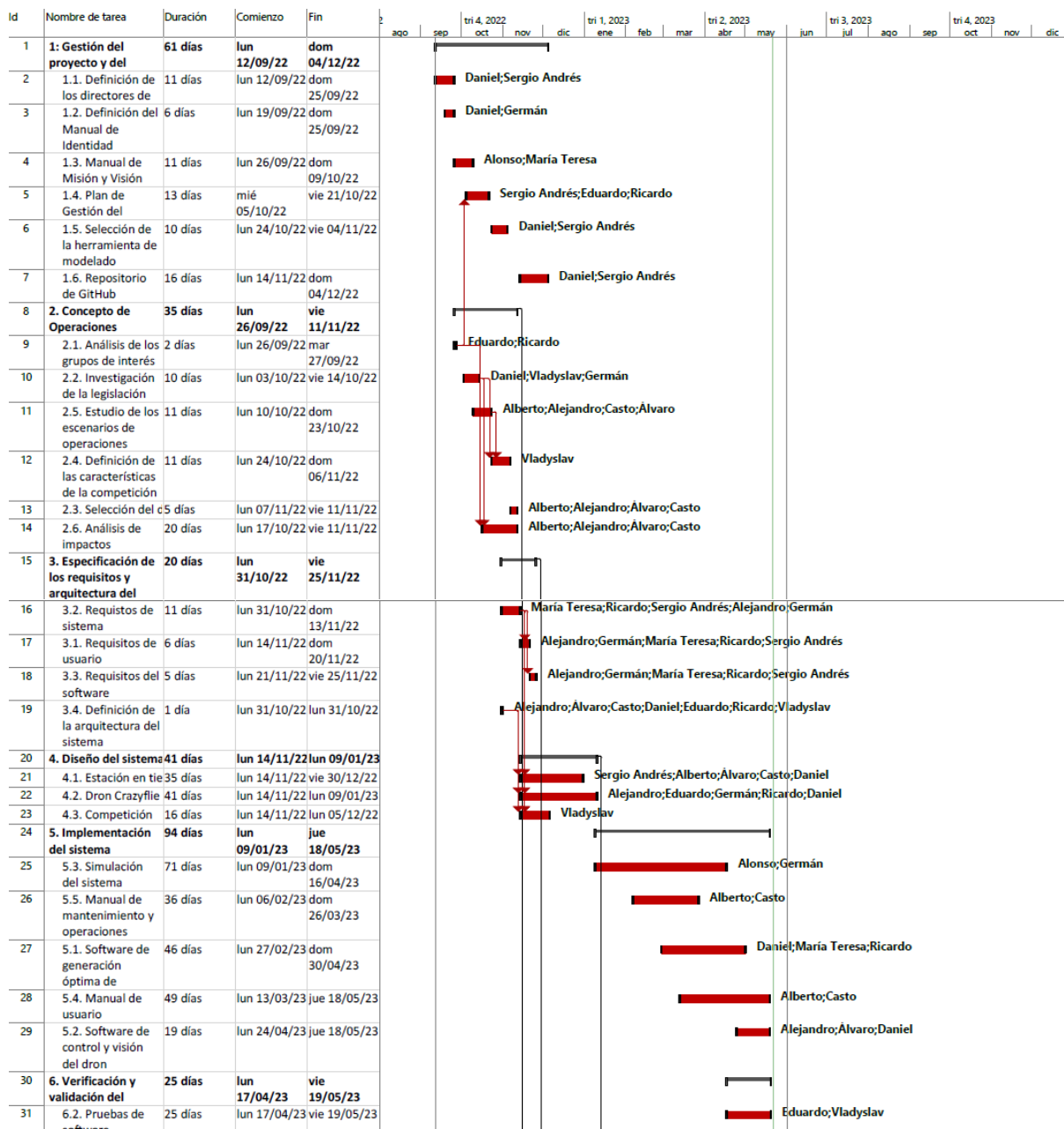


Figura 3.1: Diagrama de Gantt (parte 1).

Para la asignación de las tareas, los responsables de cada área se han mantenido en comunicación con los PM para coordinar las tareas y aprovechar los conocimientos y habilidades previas de los miembros del equipo para mejorar el desempeño en las tareas adjudicadas. La planificación ha sido bastante flexible debido a la incertidumbre existente en el proyecto debido a factores externos principalmente relacionados con la adquisición del hardware.

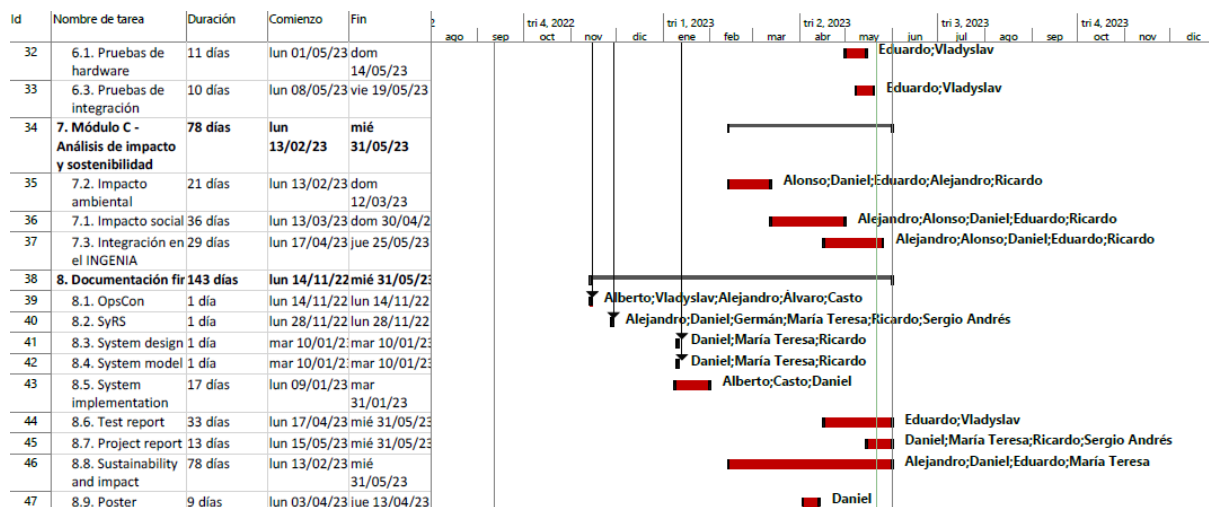


Figura 3.2: Diagrama de Gantt (parte 2).

A continuación, se describe brevemente la ejecución del proyecto. En primer lugar, se realizaron labores de planificación, creación de la imagen corporativa, selección de herramientas, creación del repositorio, etc. A continuación, se comenzó identificando y analizando los grupos de interés, planificando la competición y analizando los impactos y riesgos del proyecto. Se continuó especificando los requisitos y definiendo la arquitectura del sistema y posteriormente con el diseño del sistema desde el alto nivel hasta el nivel de detalle. Para implementar el sistema, se comenzó con la fase de simulación, realizando inicialmente un modelo dinámico en Simulink del dron. No obstante, debido a las dificultades observadas en el control del mismo (posiblemente debido a la necesidad de un filtro Kalman para el control del dron), se decidió pasar a simularlo a través del software de Webots, pese a que este era menos flexible para los objetivos que se tenían en el proyecto y el proveedor no lo tenía suficientemente desarrollado ni documentado. Se procedió al mismo tiempo con el diseño e implementación del software para satisfacer los requisitos del proyecto y se redactaron los manuales de usuario y mantenimiento y operaciones. Finalmente, se realizó la verificación de los requisitos del sistema a través de la planificación y la ejecución de pruebas tanto de software como de hardware e integración. A lo largo del proyecto también se realizó un análisis de impacto y sostenibilidad a través de un análisis del ciclo de vida del dron y un estudio de la diversidad de género en este INGENIA. Finalmente, se procedió a elaborar la documentación final del proyecto.

4. Costes del proyecto

Los costes totales finales del proyecto han sido los debidos a la adquisición del hardware, que han supuesto un **total de 705,73 €**. Debido a la utilización de las licencias que proporciona la universidad para el software empleado, los costes asociados han sido nulos para el equipo Hell-ix. Además, se recurrió al CAR para obtener prestada la Crazyradio PA (la antena) para poder hacer el pedido, debido a que se encontraba fuera de stock, por lo que estos costes también se han evitado. Los costes de envío han sido de 24.48 € netos. A continuación, en la Figura 4.1 se muestra una lista detallada del pedido realizado junto con los precios teniendo en cuenta la tasa impositiva.

Crazyflie 2.1	260.00 €
AI-deck 1.1	260.00 €
Flow deck v2	58.75 €
LED deck	26.25 €
Female deck header	2.50 €
Spare male headers	3.13 €
Battery holder deck	6.25 €
250 mAh battery	7.50 €
250 mAh battery with 500 mA USB charger	11.88 €
4x spare motors mounts	7.50 €
2x Thust upgrade bundle	37.50 €

Figura 4.1: Lista detallada del pedido teniendo en cuenta la tasa impositiva.

5. Resultados

El sistema implementado comprende tres módulos principales para permitir la operación autónoma del dron durante la competición: un **algoritmo planificador de trayectorias** para la estación en tierra, un **programa de seguimiento de trayectorias asistido por visión** que permite al dron seguir una trayectoria pre-calculada por la estación en tierra pero que le ofrece cierta flexibilidad para reorientarse hacia las puertas y corregir así posibles errores de navegación, y un **entorno de simulación** de las pruebas de la competición de drones autónomos.

5.1. Algoritmo planificador de trayectorias

Para la planificación de la trayectoria del dron, se ha empleado un código en MATLAB que utiliza curvas de Bézier suaves en 3D. El algoritmo implementado genera splines cúbicos en 3D, basándose en los puntos de paso requeridos por el dron, que en este caso corresponden a las puertas. El código utiliza el algoritmo de Hobby, que permite no solo especificar los puntos en el espacio, sino también la inclinación deseada al atravesar cada puerta, así como un parámetro que controla la “tensión” de la curva.

Para adaptar el código existente a nuestras necesidades, se ha realizado una modificación con el fin de registrar la trayectoria del dron como una serie de puntos en el espacio. Estos puntos son almacenados en una variable y posteriormente guardados en un archivo con formato .csv. Dicho archivo será importado por el código encargado de controlar el dron, el cual se encargará de traducir estos puntos en el espacio en instrucciones de desplazamiento.

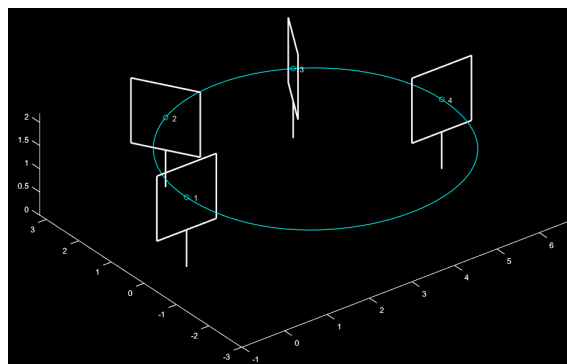


Figura 5.1: Trayectoria que seguirá el dron para pasar por las puertas 1-4 dadas por los puntos. La curva azul muestra la trayectoria del dron (podría optimizarse aumentando el nivel de tensión).

La trayectoria actual es de naturaleza relativamente sencilla; sin embargo, el código empleado proporciona la capacidad de calcular trayectorias cerradas complejas. Mediante la modificación de la "tensión" de las curvas de Bézier para adaptarse a los parámetros de inercia del dron, es posible optimizar la trayectoria con el objetivo de permitir al dron completarla en el menor tiempo posible. Esta flexibilidad brinda la posibilidad de diseñar y planificar rutas más desafiantes y eficientes en el futuro.

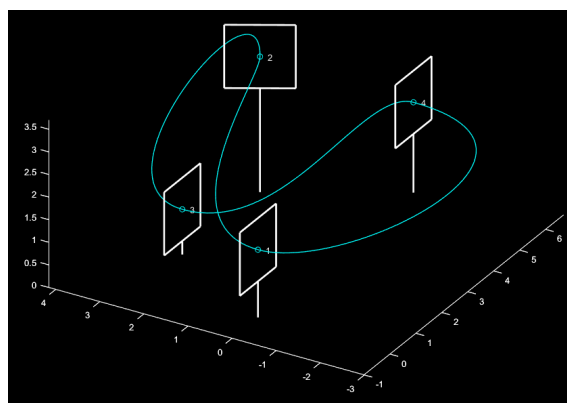


Figura 5.2: Ejemplo de trayectoria compleja que se podría implementar.

5.2. Programa de seguimiento de trayectorias asistido por visión

Se han realizado dos programas para el seguimiento de trayectorias: uno con visión y otro sin visión. El fundamento de estos programas de seguimiento de trayectorias consiste en la lectura a partir de un archivo csv (el generado por el algoritmo generador de trayectorias) de una serie de 'setpoints'. A través de los comandos de la librería de alto nivel del proveedor, es posible que el dron ajuste su movimiento a esta trayectoria. El código con visión implementa además de esta funcionalidad, una corrección de los posibles errores que se pudieran producir por inexactitudes en la navegación del dron a través de la cámara que presenta el AI-deck.

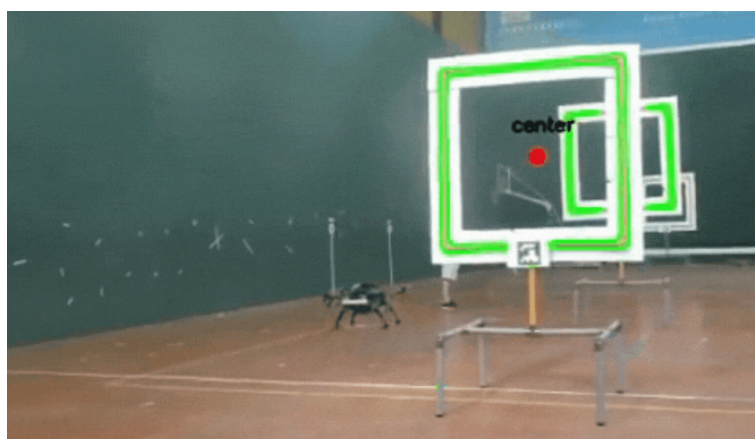


Figura 5.3: Ejemplo de programa de visión para el reconocimiento de los obstáculos.

El funcionamiento es el siguiente: las imágenes que llegan del dron a la estación en tierra se procesan en primer lugar con un difuminado gaussiano para homogeneizar la imagen, y a continuación se cambia el formato de color de RGB a HSV. Luego se aplican máscaras

en las zonas de la imagen para el rango de color definido (blanco y naranja en este caso), y por superposición de ambas máscaras se extraen los contornos de las puertas a atravesar. Finalmente, con el contorno es posible obtener el centro de la puerta, tal y como se muestra en la Figura 5.4. Conocida la posición del centro de la puerta respecto al centro de la imagen, se calculan las distancias en píxeles en ambos ejes y se alimentan a un controlador proporcional para ajustar la altura y el yaw. Pese a que este programa no se ha podido testear sobre el dron debido a la ausencia de cámara, se ha testado en vídeos de los obstáculos.

5.3. Entorno de simulación de la competición

Finalmente, se han implementado las simulaciones de las dos primeras pruebas de la competición: la carrera de cuatro obstáculos en línea recta y la carrera de cuatro obstáculos dispuestos formando un semicírculo. Las simulaciones se han realizado a través de Webots partiendo de las librerías proporcionadas por el proveedor Bitcraze. No obstante, debido a que presentaba un gran número de limitaciones, se implementó un controlador PID para permitir que el dron se ajustara a una posición determinada. Una vez el controlador funcionó correctamente, fue posible leer un archivo csv creado por el algoritmo generador de trayectorias de MATLAB. Debido a la incapacidad de introducir color en los obstáculos, se decidió no implementar la funcionalidad de la cámara sobre el dron en la simulación. La Figura 5.4 muestra las simulaciones correspondientes a ambas pruebas.

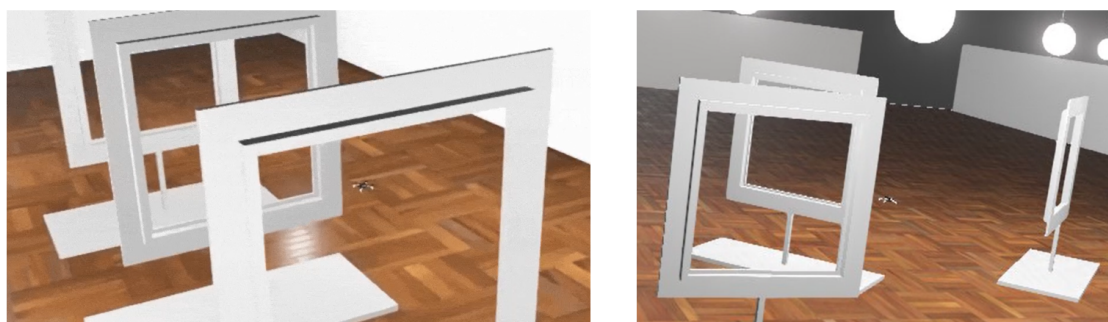


Figura 5.4: Simulaciones de las dos primeras pruebas de la competición.

6. Problemas encontrados

El **principal problema** encontrado durante la ejecución del proyecto estuvo relacionado con los problemas de stock y, en general, de **retrasos en la adquisición del hardware** necesario para llevar a cabo el proyecto. El proceso de compras comenzó a comienzos de marzo. En un primer lugar el PM se comunicó con el responsable de la asignatura, quien le dijo que el proceso de compras debía realizarse en comunicación con Carlos, miembro del CAR. En primer lugar se envió la lista de los componentes necesarios junto con sus enlaces web para su adquisición pero se nos comunicó que era necesario enviar una orden de compra en su lugar. Cuando se envió la orden de compra, ya no se encontraba disponible el módulo de IA, por lo que no era posible realizar el pedido. Se decidió seguir de cerca la disponibilidad del pedido para intentar realizar la compra lo antes posible. No obstante, la comunicación con Bitcraze a través de una persona intermediaria dificultó los procesos de compra debido a la alta demanda del producto solicitado, dado que existían ciertos retrasos entre cuando se detectaba que había disponibilidad y cuando finalmente se intentaba realizar la orden de compra. Este problema se produjo en un par de ocasiones durante marzo, hasta que finalmente se consiguió realizar el pedido la primera semana de abril. No obstante, en la realización de este

pedido se eliminó la compra de la antena Crazyradio PA, debido a que era el único elemento que se encontraba fuera de stock y se nos comunicó desde el CAR que nos la podían prestar. Pensábamos que la entrega del pedido no se demoraría más allá de las dos semanas, pero al parecer, pese a que la empresa mostraba disponibilidad de la compra realizada, se produjeron retrasos significativos en la entrega.

Ante el retardo que presentaba la entrega del dron Crazyflie 2.1 desde Bitcraze en Suecia, se decidió comunicar al responsable de la asignatura la situación debido al escaso tiempo del que se iba a disponer para poder realizar la implantación del sistema y la realización de las pruebas. Se ofreció la posibilidad de intentar adquirir el dron TELLO adquirido por el otro equipo, y se comunicó a Carlos la decisión de adquirir este segundo dron a través de Amazon. Tras una semana, finalmente llegó el dron Crazyflie el 3 de mayo, sin haberse llegado a realizar el pedido del TELLO, teniendo así un plazo muy ajustado para poder llevar a cabo todos los objetivos de la competición que se tenían pensados. Debido a las dificultades del otro equipo para poder cumplir con los requisitos iniciales de la competición, estos se redefinieron de manera conjunta de tal forma que la competición pasó a tener únicamente tres pruebas.

El **segundo problema** encontrado se encuentra relacionado con la **configuración del módulo de IA (AI-deck)** adquirido para el dron Crazyflie, el AI-deck. En primer lugar se realizó la configuración del dron y del módulo del Flow-deck para navegación óptica sin muchas complicaciones. No obstante, el módulo de IA el cual incluye la cámara y permite que el dron pueda transmitir las imágenes al ordenador o emplearlas internamente no pudo ser configurado. Siguiendo las instrucciones ofrecidas por el fabricante se instaló el firmware más reciente en el dispositivo y al intentar cargar el ejemplo proporcionado por Bitcraze para poder generar una red Wi-Fi desde el AI-deck con el que comunicar las imágenes a la estación en tierra, la instalación se quedaba colgada en el 99 %. Tras indagar en los foros de la empresa, se llegó a la conclusión de que era un problema de la revisión que había proporcionado el fabricante del AI-deck, la revisión C.1, una anterior a la última, que es la D (desde la página web no dan opción a elegir la revisión deseada del módulo). De acuerdo con el servicio técnico, para las revisiones de la A a la C es necesario instalar un bootloader en el GAP8 del AI-deck, para lo cual es necesario un cable Olimex ARM-USB-TINY-H JTAG, el cual tiene un precio de aproximadamente 50 euros y no es de fácil adquisición.

Dado el apurado plazo, se decidió acudir al CAR para ver cómo lo podíamos solucionar. Nos comunicaron que ellos no habían tenido ese problema dado que las revisiones de las que ellos disponen eran revisión D, lo que les facilitó en gran medida la configuración con respecto a revisiones anteriores. Tampoco disponían de un cable Olimex ARM-USB-TINY-H JTAG que nos permitiera realizar la instalación. Intentaron ellos mismos configurar nuestro dispositivo sin éxito y nos plantearon la posibilidad de prestarnos un AI-deck revisión D para poder llevar a cabo nuestro proyecto. No obstante, el AI-deck que nos podían prestar tenía una cámara blanco y negro en vez de RGB, por lo que haría necesario cambiar parcialmente el programa de visión. Aún así, se planteó como una opción viable. No obstante, para poder prestárnoslo debían reconfigurar el firmware del dispositivo ya que tenían uno personalizado instalado. Al intentar configurar el dispositivo para generar la red Wi-Fi necesaria para transmitir las imágenes al ordenador, no tuvieron éxito. No llegaron a localizar cuál era el problema y nos dijeron que se lo harían llegar a un contacto en Bitcraze y lo intentarían solucionar el 12 de mayo. Sin embargo, debido a los limitados márgenes disponibles para finalizar el proyecto, se decidió realizar las pruebas del programa de visión a través de simulación ante la imposibilidad de emplear una cámara sobre el dron.

De esta forma, se decidió que en la competición física el dron iría “ciego”, únicamente con el sistema de navegación óptico proporcionado por el Flow-deck y la IMU, siguiendo la trayectoria pre-calculada por el algoritmo de optimización de trayectorias a partir de las posiciones de las puertas. Dadas las características de la competición, consistente en pruebas de

seguimiento de trayectorias una única vez y con puertas muy anchas con respecto al tamaño del dron (1.4 m de lado), se consideró posible el poder superar las pruebas sin la utilización de una cámara en el dron. Si se hubiera dispuesto de más tiempo, se habría podido adquirir el cable y, a través de una máquina virtual, haber instalado el bootloader en el GAP8. Pese a ello, pudimos ver en los foros que a los usuarios les surgían varios problemas en la instalación aún disponiendo del cable, por lo que consideramos que se tomó la decisión adecuada en únicamente testear el código de visión a través de simulación.

Es muy probable que el algoritmo desarrollado no sea válido en la realidad debido a las características reales que tiene la cámara del dron que, según nos comunicaron los trabajadores del CAR, ofrece una latencia muy baja a través del Wi-Fi del AI-deck. Por lo tanto, se trata de una de las principales limitaciones del trabajo desarrollado en este proyecto.

Otro problema que surgió en las últimas semanas del proyecto estuvo relacionado con la **implementación física del módulo LED ring**. Dicho módulo se adquirió para la prueba de freestyle con el fin de realizar una prueba más visualmente atractiva. Además, antes de realizar la adquisición se comprobó la compatibilidad entre los distintos módulos adquiridos y se observó que era posible implementar el Flow-deck v2 y el LED ring a la vez, pese a que ambos están pensados para ubicarse en la parte inferior del dron. Se encontró un tutorial de cómo realizar las conexiones y por tanto se supuso que sería posible implementarlo. No obstante, una vez el dron estuvo en manos del equipo y se comenzó a seguir los pasos del tutorial (sin realizar soldaduras sino con cables Dupont), se pudo observar que el módulo de LED y el Flow-deck v2 tenían un pin de control común que suponía dificultades para la navegación óptica del dron mientras los LED estaban encendidos. Además, los cables Dupont implicaban que el módulo LED hiciera el dron bastante voluminoso y mucho menos robusto, tal y como puede verse en la Figura 6.1. Debido a la falta de tiempo para detectar cómo corregir las conexiones y realizar las soldaduras, se decidió no emplear el módulo de LED para la prueba de freestyle, ya que la navegación óptica proporcionada por el Flow-deck era la prioridad.

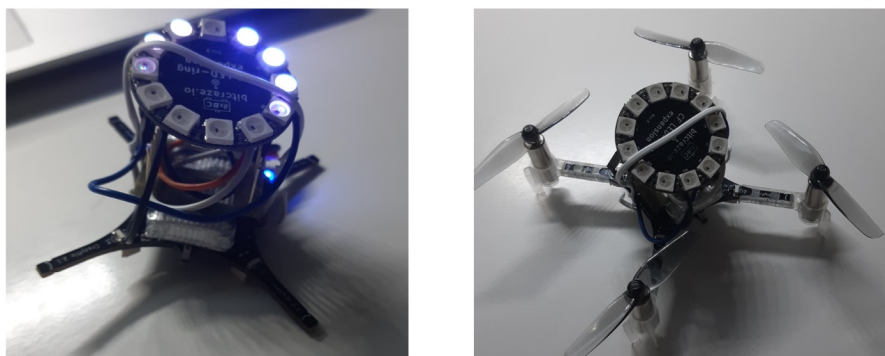


Figura 6.1: LED ring and Flow-deck v2 implementation.

Hubo un **problema** que impidió la verificación del requisito de la **capacidad del dron para atravesar obstáculos** que se detectó durante la fase de pruebas. El problema está relacionado con que el Flow-deck se encuentra constantemente midiendo la altura del dron gracias a un sensor láser, en el momento en el que pasa por un obstáculo, el dron sube rápidamente debido a que detecta que se encuentra a una altura menor que la referencia mientras atraviesa el obstáculo al detectar el marco en vez del suelo. Se intentó paliar este comportamiento atravesando la puerta más rápido pero seguía repitiéndose el mismo problema. Se pensó en cómo solucionarlo a través de un filtrado de la señal proporcionada por el Flow-deck. No obstante, dicha implementación no había sido realizada previamente por Bitcraze y se localizó en varios foros que los usuarios tenían problemas para acceder a los datos del Flow-deck con

el mismo objetivo. Debido a la falta de tiempo para la realización de un estudio detallado de cómo mejorar este comportamiento, finalmente este requisito no se pudo verificar e impidió la validación del sistema para satisfacer las necesidades del cliente.

Otro **problema secundario** fue la incapacidad de poder **testear el algoritmo de visión a través de la simulación de Webots**. Esto se debió a la imposibilidad de incluir color en los obstáculos importados desde Fusion 360° debido al tipo de formato de archivo de geometría aceptado por el simulador de Webots.

7. Conclusiones y lecciones aprendidas

En conclusión, debido a los problemas relacionados con el retraso en la adquisición del hardware y la configuración del mismo, se considera que **el sistema no es válido** dado que no cumple con las necesidades del usuario de poder llevar a cabo la competición de drones autónomos. El sistema no es robusto ante variaciones en las posiciones de las puertas sin haberlo tenido en cuenta en el algoritmo generador de trayectorias, o si la competición implica recorrer en varias ocasiones un mismo circuito. Esto se debe a la imposibilidad que se ha dado en el proyecto para poder configurar la cámara del AI-deck del Crazyflie 2.1. Además, al atravesar los obstáculos el dron se inestabiliza debido al problema relacionado con el sensor láser de altura del Flow-deck previamente comentado.

Se han adquirido varias lecciones valiosas para futuros proyectos a partir de las dificultades surgidas a lo largo de este proyecto de INGENIA-SE 22-23:

- **Es mejor simplificar en proyectos complejos con recursos de tiempo limitados:** Pese a que en la reunión inicial con el CAR se nos hizo saber que entre las distintas opciones posibles (dron customizado, TELLO o Crazyflie 2.1), aquella que a un precio más razonable nos permitía una mayor configuración y un mayor control sobre la programación del dron era el Crazyflie 2.1; haber simplificado el hardware utilizado en el proyecto habría simplificado en gran medida el trabajo desarrollado. Esto se debe a que su adquisición a través de Amazon habría sido mucho más rápida que a través del proveedor Sueco Bitcraze (menos de una semana en vez de un mes) y habría permitido una configuración, aunque más limitada, mucho más sencilla; permitiendo igualmente la satisfacción de los requisitos establecidos para el sistema y las necesidades establecidas por el cliente.
- **Es mejor evitar depender de productos con escasa disponibilidad para la finalización del proyecto:** Tras el primer intento de compra, cuando se nos informó de que parte del pedido que se deseaba realizar se encontraba fuera de stock, se valoró la posibilidad de cambiar de dron. No obstante, en un primer momento no se consideró muy importante debido a que se consideró que podía desarrollarse el proyecto a través de una simulación que permitiera ir avanzando. No obstante, al quedar un mes y medio para la finalización del proyecto y comunicárselo al cliente, éste nos hizo saber que era necesario realizar una competición y no era suficiente con una simulación, fue entonces cuando se valoró adquirir el TELLO si el Crazyflie no llegaba. Haber adquirido directamente el dron TELLO hubiera evitado este problema dado que no presentaba los recurrentes problemas de stock que tiene el Crazyflie y sus distintos módulos adicionales.
- **Planificación de las adquisiciones:** Para evitar los contratiempos que hubo con los problemas de stock, podría haberse realizado la adquisición del dron antes de Navidad, dado que ya se tenía decidido cuál iba a ser el dron que incluiría el sistema. La planificación de las compras desde el principio del proyecto, asegurando un amplio margen de tiempo como plan de contingencia para retrasos, hubiera evitado los problemas sufridos.

- **Importancia de una comunicación fluida con los proveedores:** Mantener una comunicación fluida y constante con los proveedores es esencial. En nuestro caso, la falta de comunicación directa con el proveedor sino a través de una persona intermediaria dificultó las comunicaciones y derivó en un mayor retraso de la entrega.
- **Pruebas y configuración del equipo tempranas:** Es importante realizar ciertas pruebas y configuraciones en los equipos a utilizar en el proyecto cuanto antes, dado que pueden aparecer limitaciones que condicionen la organización del trabajo a desarrollar y la división de las tareas. En nuestro caso, no pudimos configurar la cámara del AI-deck, lo que afectó negativamente a nuestra capacidad para participar plenamente en la competición y nos exigió replantear cómo afrontarla.
- **Importancia de la gestión de riesgos:** Pese a que se trató como un tema formativo durante la impartición de las clases sobre Ingeniería de Sistemas, no se aplicó directamente al proyecto. Se considera que el análisis de riesgos debe ser parte integral del proyecto desde el principio, ya que la identificación y evaluación de los riesgos potenciales como los ocurridos (retrasos en la entrega y problemas de configuración), pueden ayudar a planificar estrategias de mitigaciones adecuadas, incluyendo planes de contingencia.
- **Necesidad de flexibilidad en la planificación:** Debido a los imprevistos que pueden surgir en un proyecto, se extrae como conclusión que es imprescindible que el plan del proyecto este detallado pero que permita cierta flexibilidad para poder reajustar el cronograma y la planificación general.
- **Importancia de documentar los errores cometidos en un proyecto y las lecciones aprendidas:** Mantener un registro de los problemas encontrados y las soluciones tomadas durante el proyecto se considera fundamental para poder evitar repetir errores en futuros proyectos y fomentar el aprendizaje continuo de los miembros del equipo, el cual supone uno de los dos objetivos fundamentales de este proyecto.

Todas estas lecciones aprendidas suponen una fuerte base de conocimiento adquirido a través de la experiencia durante la ejecución de un proyecto complejo real con la implicación de múltiples agentes y grupos de interés. Este conocimiento será de gran utilidad en futuros proyectos para evitar cometer los mismos errores y poder así minimizar los riesgos y las posibilidades de fracaso del proyecto.