



Desarrollo de plataforma móvil para SLAM monocular

Autor:

Ing. Gonzalo Gabriel Fernández

Director:

Dr. Ing. Fabio Ardiani (Nimble One)

Codirector:

John Doe (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 25 de abril de 2023 y el 13 de junio de 2023.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar	5
2. Identificación y análisis de los interesados	7
3. Propósito del proyecto	7
4. Alcance del proyecto	7
5. Supuestos del proyecto.	8
6. Requerimientos	8
7. Historias de usuarios (<i>Product backlog</i>).	9
8. Entregables principales del proyecto	11
9. Desglose del trabajo en tareas	12
10. Diagrama de Activity On Node.	13
11. Diagrama de Gantt	16
12. Presupuesto detallado del proyecto	17
13. Gestión de riesgos	17
14. Gestión de la calidad	19
15. Procesos de cierre	21

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	25 de abril de 2023
1	Se completa hasta el punto 4 inclusive	9 de mayo de 2023
2	Se completa hasta el punto 7 inclusive	16 de mayo de 2023
3	Se completa hasta el punto 12 inclusive	25 de mayo de 2023
4	Se completa el plan	30 de mayo de 2023

Acta de constitución del proyecto

Córdoba, 25 de abril de 2023

Por medio de la presente se acuerda con el Ing. Gonzalo Gabriel Fernández que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará “Desarrollo de plataforma móvil para SLAM monocular”, consistirá esencialmente en el desarrollo de una plataforma móvil para la evaluación de algoritmos de SLAM monocular investigados para aplicación en navegación autónoma, y tendrá un presupuesto preliminar estimado de 668 h de trabajo y \$1.801.650 ARS, con fecha de inicio el 25 de abril de 2023 y fecha de presentación pública el 15 de mayo de 2024.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Ing. Leandro Borgnino
Fundación Fulgor

Dr. Ing. Fabio Ardiani
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

La navegación autónoma ha sido un tema de investigación activo durante décadas, pero recientemente ha ganado una atención sin precedentes gracias a los avances en tecnologías como la inteligencia artificial y la robótica.

La navegación autónoma se refiere a la capacidad de un sistema para planificar y ejecutar sus propias acciones de manera autónoma, sin la necesidad de una intervención humana constante.

La capacidad de un sistema para tomar decisiones en forma autónoma en su navegación a través del entorno es crucial para una amplia gama de aplicaciones, desde los vehículos autónomos y los drones, hasta los robots industriales y los sistemas de logística.

La investigación de técnicas de bajo costo es importante para permitir que la tecnología sea accesible para un mayor número de aplicaciones y usuarios. Esto es especialmente importante en países en desarrollo o en áreas con recursos limitados, donde los sistemas de navegación autónoma pueden ser prohibitivamente costosos.

La restricción económica en el desarrollo de nuevas técnicas también puede mejorar la eficiencia y la efectividad de los sistemas de navegación autónoma, al permitir que los recursos sean utilizados de manera más efectiva y maximizar la vida útil de los componentes del sistema.

La odometría visual es una técnica de localización y navegación que se utiliza en robótica para estimar la posición y orientación de un robot en su entorno, en función de la información visual capturada por sus cámaras. La información de las imágenes capturadas permite estimar la cantidad de movimiento y la dirección del vehículo, permitiendo calcular su posición y orientación en relación con su posición inicial.

Las técnicas de odometría visual son particularmente útiles en entornos donde el movimiento del robot es predominantemente lateral. Además, se pueden utilizar en combinación con otras técnicas de localización y navegación, como la odometría inercial o la SLAM (*Simultaneous Localization and Mapping*), para mejorar la precisión y la fiabilidad de la localización del robot.

La SLAM es una técnica de visión por computadora que permite a un robot móvil construir un mapa del entorno en el que se encuentra, mientras estima en simultáneo su propia posición y orientación en ese entorno. En el caso de SLAM monocular, se utiliza una única cámara para capturar imágenes del entorno.

La Fundación Fulgor realiza diversas actividades de investigación. Dentro del campo de navegación autónoma, una de las ramas de interés es la de algoritmos de SLAM monocular en conjunto con la odometría inercial, por las diferentes ventajas que presenta sobre otros métodos. Es por esto que se han realizado experimentos en el marco de la Práctica Profesional Supervisada de la estudiante de grado Evangelina Castellano, implementando y analizando el desempeño de diferentes algoritmos en entornos de simulación.

Actualmente, uno de los objetivos en esta línea de investigación es poder migrar los algoritmos estudiados en un entorno virtual a un sistema físico, para evaluar su desempeño en un entorno real. La experimentación en entornos virtuales se llevó a cabo con el framework ROS 2 (*Robot Operating System 2*). Dada la naturaleza modular de este framework, la migración consiste en el reemplazo de un módulo encapsulado que contenga el algoritmo, como se observa en la figura 1.

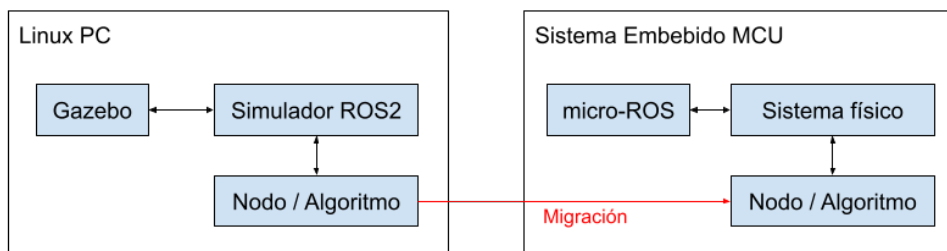


Figura 1. Migración de algoritmos en entorno de simulación a sistema físico.

El entorno ROS es un framework de software libre y de código abierto diseñado para permitir el desarrollo de aplicaciones robóticas distribuidas. Este framework proporciona un conjunto de herramientas para la creación, gestión, depuración y análisis de sistemas robóticos.

La segunda versión de la herramienta, denominada ROS 2, fue desarrollada por la comunidad con el objetivo de mejorar la escalabilidad, la fiabilidad y la seguridad del software. Una de sus principales características es que está diseñado para ser modular y extensible, lo que significa que los desarrolladores pueden elegir los componentes que necesitan para sus aplicaciones y utilizarlos de manera flexible. También se enfoca en proporcionar una abstracción de hardware más clara y permitir el uso de diferentes sistemas operativos y arquitecturas.

La herramienta micro-ROS es una implementación de ROS 2 diseñada específicamente para sistemas embebidos y de tiempo real. A diferencia de ROS 2, que se ejecuta en sistemas operativos de propósito general, micro-ROS se ejecuta en sistemas operativos de tiempo real (como NuttX y FreeRTOS), lo que permite el desarrollo de sistemas robóticos en entornos de baja potencia y recursos limitados.

Como puede observarse en la figura 2, micro-ROS proporciona una capa de abstracción que permite la comunicación entre los sistemas embebidos y los nodos de ROS 2 en otros sistemas. Esto significa que los desarrolladores pueden crear sistemas robóticos distribuidos, que utilicen tanto sistemas embebidos como sistemas de propósito general y que todos ellos puedan comunicarse a través de la misma plataforma.

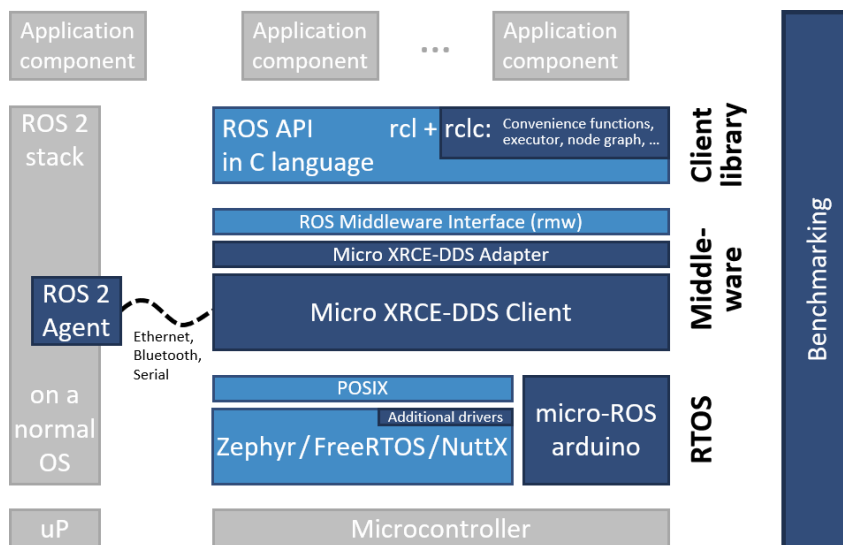


Figura 2. Arquitectura de micro-ROS.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Ing. Leandro Borgnino	Fundación Fulgor	Investigador
Responsable	Ing. Gonzalo Gabriel Fernández	FIUBA	Alumno
Orientador	Dr. Ing. Fabio Ardiani	Nimble One	Director Trabajo final
Equipo	Evangelina Castellano	Universidad Nacional de Córdoba	Colaboradores
Usuario final	Grupos de investigación	Fundación Fulgor	-

- Cliente: el Ing. Leandro Borgnino se encuentra haciendo su doctorado en la Fundación Fulgor y posee un gran conocimiento sobre los algoritmos utilizados para navegación autónoma a alto nivel. Es capaz de resolver dudas específicas sobre los requerimientos a cumplir con el algoritmo a implementar.
- Equipo: Evangelina Castellano se encuentra realizando su Práctica Profesional Supervisada (PPS) en la Fundación Fulgor. Al finalizar su PPS continuará en la institución para realizar su proyecto final de grado. Es necesario planificar teniendo en cuenta que la dedicación es simple, es decir, entre 4 y 5 horas diarias.
- Usuario final: no esta representado por una persona particular. Una vez finalizado el proyecto el prototipo quedará disponible en la Fundación Fulgor para su uso.

3. Propósito del proyecto

El propósito de este proyecto es desarrollar el prototipo de una plataforma de hardware con las capacidades para poder implementar los algoritmos investigados de SLAM monocular y observar resultados.

La técnica de SLAM monocular es de interés por su aplicación en la navegación autónoma de vehículos, su bajo costo económico y simpleza del hardware involucrado en comparación a otras técnicas utilizadas en el área.

4. Alcance del proyecto

El proyecto incluye el diseño del sistema embebido, el desarrollo de los drivers de los sensores asociados a las técnicas de SLAM monocular, la incorporación de la interfaz de micro-ROS en el proyecto, la aplicación de ROS 2 que permitirá interactuar con el sistema a alto nivel y la investigación e implementación de un algoritmo de SLAM monocular adecuado.

El proyecto no incluye el desarrollo de algoritmos de navegación autónoma asociados a la planificación y ejecución de trayectorias del vehículo, ni el modelado matemático de este. Tampoco incluye optimización de los algoritmos planteados con el uso de hardware dedicado como una FPGA, por ser una actividad que se realizará en una segunda etapa del proyecto.

Por lo tanto, el producto final esperado es un prototipo de un sistema embebido con los correspondientes sensores y herramientas necesarias funcionando en conjunto con técnicas de

SLAM monocular, que un vehículo móvil pueda incorporar para dotarlo con capacidades de navegación autónoma.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- La adquisición de materiales no tendrá demoras mayores a una semana, asumiendo que el mercado local posee los materiales requeridos para la fabricación del prototipo.
- No habrá cambios en el equipo de trabajo en el periodo de tiempo que comprende la planificación del proyecto.
- El responsable del proyecto puede dedicar una cantidad mínima de 25 horas semanales a las actividades del plan.
- El responsable del proyecto estará ausente durante dos semanas de Julio del año 2023, con fecha a definir.
- No habrá cambios en los objetivos principales del proyecto ni nuevos requerimientos funcionales de parte del cliente.

6. Requerimientos

1. Requerimientos funcionales:

- 1.1. El driver de la Unidad de Medición Inercial (IMU por sus siglas en inglés) debe ofrecer la información del giróscopo y acelerómetro de forma completa a la aplicación.
- 1.2. El sistema debe utilizar algoritmos y técnicas de fusión sensorial para optimizar el uso de la información obtenida por la IMU.
- 1.3. El driver de la IMU debe brindar todas las posibilidades de configuración necesarias para poder adaptar la escala y frecuencia de muestreo de los sensores a los requisitos de la aplicación.
- 1.4. El driver de la cámara debe proveer la información correcta a la aplicación, dadas las capacidades del dispositivo.
- 1.5. El driver de la IMU y de la cámara deben implementarse de forma modular para que puedan utilizarse independientemente del proyecto.
- 1.6. El sistema embebido debe utilizar micro-ROS para interactuar e integrarse fácilmente con el entorno de desarrollo, monitoreo y evaluación de ROS 2.
- 1.7. El sistema embebido debe hacer un uso adecuado de los mecanismos de comunicación de micro-ROS con ROS 2 para explotar todas las ventajas que ofrece el framework.
- 1.8. El sistema debe estar diseñado de forma modular para favorecer su escalabilidad.
- 1.9. El sistema debe ejecutar un algoritmo investigado previamente de SLAM monocular.

2. Requerimientos de documentación:

- 2.1. Todas aquellas funcionalidades cubiertas en el driver de la IMU deben estar completamente documentadas para que el módulo de software pueda utilizarse independientemente del proyecto.

- 2.2. El proceso de incorporación y uso de las librerías de micro-ROS debe estar correctamente documentado para facilitar el desarrollo continuo del sistema.
- 2.3. La investigación del estado del arte en el tópico de algoritmos de SLAM monocular debe estar debidamente documentada.
- 2.4. El algoritmo seleccionado para implementar en el sistema debe estar documentado en forma detallada.
3. Requerimientos de testing:
 - 3.1. Se deben aplicar herramientas y metodologías de testing en los drivers de los módulos para asegurar su funcionamiento y calidad.
4. Requerimientos de la interfaz:
 - 4.1. Las herramientas desarrolladas en el entorno de ROS 2 deben también cumplir el propósito de interfaz gráfica y permitir la visualización de la información obtenida a través de los sensores.
 - 4.2. El sistema desarrollado con ROS 2 debe ofrecer herramientas que permitan operar y reconfigurar el sistema.
 - 4.3. El sistema desarrollado con ROS 2 debe ofrecer herramientas que permitan evaluar el desempeño del algoritmo implementado, como puede ser el ajuste de parámetros del sistema *on-the-fly*, la visualización del mapa del entorno y la ubicación espacial de la plataforma en él y otras herramientas que se consideren necesarias.

7. Historias de usuarios (*Product backlog*)

Se identifican los siguientes roles/usuarios:

1. Investigador en el área de Ciencias de la Computación, en su rol de estudiante de máster o PhD desea realizar avances en los algoritmos de navegación autónoma. No necesariamente tiene conocimientos de electrónica y sistemas embebidos, su foco es el desarrollo de alto nivel.
2. Investigador en el área de la electrónica, sistemas embebidos y/o firmware, en su rol de estudiante de máster o PhD desea realizar avances en el desarrollo de dispositivos de navegación autónoma mediante la optimización o mejora del hardware y el firmware asociado.
3. Usuario perteneciente a la comunidad de código abierto/libre, en su rol desea implementar el proyecto con sus propios materiales, aplicarlo en un caso de uso específico y colaborar con posibles mejoras. Puede o no poseer conocimiento técnico.

La ponderación de cada historia de usuario se obtiene de una evaluación sobre tres ejes: tiempo, dificultad e incertidumbre. Sobre cada eje se asigna un valor en una escala entre 0 y 5, luego este valor se penaliza según la relevancia asignada a cada aspecto: tiempo 0.4, dificultad 0.2 e incertidumbre 0.4. Para obtener el puntaje final de la historia se divide el resultado parcial por 5 y se multiplica por 100.

Historia 1 de usuario 1: como investigador en el área de Ciencias de la Computación quiero poder utilizar los sensores de SLAM monocular como si fueran una caja negra, quiero que sea sencillo aprender a utilizarlo e incorporarlo en mi flujo de trabajo para bajar de forma sencilla al dispositivo los algoritmos implementados a alto nivel en la simulación.

- Tiempo: 4.
- Dificultad: 3.
- Incertidumbre: 2.
- Puntaje: 60

Historia 2 de usuario 1: como investigador en el área de Ciencias de la Computación quiero poder visualizar la información obtenida por los sensores y utilizarla como entrada a simulaciones virtuales.

- Tiempo: 3.
- Dificultad: 2.
- Incertidumbre: 1.
- Puntaje: 40

Historia 1 de usuario 2: como investigador en el área de la electrónica quiero poder entender rápido cómo está conformado el sistema y que sea lo suficientemente modular como para modificar, mejorar y/o intercambiar algún componente funcional y que el sistema siga operando sin dificultades.

- Tiempo: 4.
- Dificultad: 3.
- Incertidumbre: 2.
- Puntaje: 60

Historia 2 de usuario 2: como investigador en el área de la electrónica quiero poder reutilizar los componentes que conforman el sistema de forma independiente en un proyecto diferente.

- Tiempo: 3.
- Dificultad: 3.
- Incertidumbre: 2.
- Puntaje: 52

Historia 1 de usuario 3: como parte de la comunidad de código abierto/libre quiero que los materiales utilizados en el prototipo sean accesibles y que el proyecto tenga la documentación completa de las distintas etapas para poner en marcha el sistema.

- Tiempo: 5.
- Dificultad: 1.
- Incertidumbre: 3.
- Puntaje: 68

Historia 2 de usuario 3: como parte de la comunidad de código abierto/libre quiero que el proyecto posea un mecanismo para poder contribuir en cualquiera de sus aspectos dado mi conocimiento específico.

- Tiempo: 4.
- Dificultad: 2.
- Incertidumbre: 3.
- Puntaje: 64

8. Entregables principales del proyecto

Los entregables del proyecto son:

- Plataforma de hardware que conforma el prototipo funcional
- Documentación del hardware e interconexión de los módulos componentes
- Código fuente del driver de la IMU
- Documentación del driver de la IMU
- Código fuente del driver de la cámara
- Documentación del driver de la cámara
- Código fuente del firmware con el algoritmo de SLAM monocular y micro-ROS
- Código fuente del agente de ROS 2 utilizado
- Documentación y manual para configuración y uso del entorno de ROS 2
- Documentación del estado del arte de SLAM y del algoritmo seleccionado
- Manual de uso del hardware
- Informe final

9. Desglose del trabajo en tareas

1. Planificación del proyecto (70 h)
 - 1.1. Identificación y contacto con los interesados (7 h)
 - 1.2. Definición del propósito y alcance del proyecto (10 h)
 - 1.3. Definición de los requerimientos y entregables (20 h)
 - 1.4. Desglose del trabajo en tareas y obtención del diagrama de Gantt (20 h)
 - 1.5. Análisis de riesgos, gestión de calidad y definición de procesos de cierre (13 h)
2. Setup del entorno de trabajo para desarrollo del software necesario (5 h)
3. Implementación de micro-ROS en el sistema embebido (18 h)
 - 3.1. Introducción a micro-ROS (4 h)
 - 3.2. Incorporación de micro-ROS en el microcontrolador (8 h)
 - 3.3. Setup de agente de ROS 2 y comunicación con sistema embebido (2 h)
 - 3.4. Documentación del proceso (4 h)
4. Desarrollo de driver para la IMU (40 h)
 - 4.1. Selección y adquisición de la IMU a utilizar (2 h)
 - 4.2. Implementación de driver en lenguaje C para el sistema embebido (30 h)
 - 4.3. Documentación del driver (8 h)
5. Integración de la IMU en el sistema (60 h)
 - 5.1. Integración de driver de la IMU con el sistema con micro-ROS (4 h)
 - 5.2. Repaso (e investigación) de técnicas de fusión sensorial (8 h)
 - 5.3. Implementación de algoritmos de fusión sensorial (30 h)
 - 5.4. Visualización de la información obtenida con la IMU mediante las herramientas de ROS 2 (10 h)
 - 5.5. Evaluación parcial y documentación de resultados (milestone) (8 h)
6. Desarrollo de driver para la cámara (53 h)
 - 6.1. Selección y adquisición de cámara a utilizar (3 h)
 - 6.2. Implementación de driver en lenguaje C para el sistema embebido (40 h)
 - 6.3. Documentación del driver (10 h)
7. Integración de la cámara en el sistema (26 h)
 - 7.1. Integración del driver de la cámara con el sistema con micro-ROS (6 h)
 - 7.2. Visualización de la información obtenida con la cámara mediante las herramientas de ROS 2 (12 h)
 - 7.3. Evaluación parcial y documentación de resultados (milestone) (8 h)
8. Implementación de algoritmo de SLAM monocular (305 h)
 - 8.1. Introducción a la navegación inercial y SLAM monocular, investigación de estado del arte (15 h)

- 8.2. Investigación de algoritmos de SLAM monocular y selección de uno de ellos (30 h)
- 8.3. Implementación del algoritmo seleccionado en un entorno de simulación (40 h)
- 8.4. Optimización del algoritmo implementado en el entorno de simulación (40 h)
- 8.5. Evaluación integral de los componentes del sistema (20 h)
- 8.6. Implementación del algoritmo seleccionado sobre la plataforma desarrollada (40 h)
- 8.7. Optimización del algoritmo implementado sobre la plataforma desarrollada (40 h)
- 8.8. Evaluación final de resultados (milestone) (40 h)
- 8.9. Documentación técnica del algoritmo seleccionado y resumen del estado del arte (40 h)
- 9. Proceso de cierre (91 h)
 - 9.1. Análisis de cumplimiento de objetivos y requerimientos con el director (2 h)
 - 9.2. Análisis de cumplimiento del plan original (diagrama de Gantt) (2 h)
 - 9.3. Análisis de cumplimiento de objetivos y requerimientos con el cliente (2 h)
 - 9.4. Elaboración de memoria técnica (40 h)
 - 9.5. Corrección de memoria técnica (20 h)
 - 9.6. Preparación de defensa pública (20 h)
 - 9.7. Defensa pública (2 h)
 - 9.8. Reunión de cierre con el director (1 h)
 - 9.9. Reunión de cierre con el cliente y el equipo de trabajo (2 h)

Cantidad total de horas: 668 h

10. Diagrama de Activity On Node

Las tareas expuestas en el diagrama de *Activity On Node*, figura 3, se encuentran explicitadas en la siguiente tabla, con su código, duración y tareas predecesoras.

Código	Predecesora	Descripción	Duración
1		Planificación del proyecto	70 h
1.1		Identificación y contacto con los interesados	7 h
1.2	1.1	Definición del propósito y alcance del proyecto	10 h
1.3	1.2	Definición de los requerimientos y entregables	20 h
1.4	1.3	Desglose del trabajo en tareas	20 h
1.5	1.4	Análisis de riesgos y gestión de calidad	13 h
2	1	Setup del entorno de trabajo	5 h
3		Implementación de micro-ROS	18 h
3.1	1	Introducción a micro-ROS	4 h
3.2	2, 3.1	Incorporación de micro-ROS en el sistema	8 h
3.3	3.2	Setup del agente de ROS 2	2 h
3.4	3.3	Documentación del proceso	4 h
4		Desarrollo de driver para la IMU	40 h
4.1	1	Selección y adquisición de la IMU a utilizar	2 h
4.2	2, 4.1	Implementación de driver para el sistema embebido	30 h
4.3	4.2	Documentación del driver	8 h
5		Integración de la IMU en el sistema	60 h
5.1	3.3, 4.2	Integración del driver de la IMU con micro-ROS	4 h
5.2	1	Repaso de técnicas de fusión sensorial	8 h
5.3	4.2, 5.2	Implementación de algoritmos de fusión sensorial	30 h
5.4	5.1	Visualización de la información mediante ROS 2	10 h
5.5	5.3, 5.4	Evaluación parcial de resultados y documentación	8 h
6		Desarrollo de driver para la cámara	53 h
6.1	1	Selección y adquisición de la cámara a utilizar	3 h
6.2	2, 6.1	Implementación de driver para el sistema embebido	40 h
6.3	6.2	Documentación del driver	10 h
7		Integración de la cámara en el sistema	26 h
7.1	3.3, 6.2	Integración del driver de la cámara con micro-ROS	6 h
7.2	7.1	Visualización de la información mediante ROS 2	12 h
7.3	7.2	Evaluación parcial de resultados y documentación	8 h
8		Implementación de algoritmo de SLAM monocular	305 h
8.1	1	Investigación del estado del arte de SLAM	15 h
8.2	8.1	Investigación y selección de algoritmos de SLAM	30 h
8.3	2, 8.2	Implementación del algoritmo en simulación	40 h
8.4	8.3	Optimización del algoritmo en simulación	40 h
8.5	5.5, 7.3	Evaluación integral de los componentes del sistema	20 h
8.6	8.4, 8.5	Implementación del algoritmo en el sistema embebido	40 h
8.7	8.6	Optimización del algoritmo en el sistema embebido	40 h
8.8	8.7	Evaluación final de resultados	40 h
8.9	8.8	Documentación técnica del algoritmo	50 h
9		Proceso de cierre	91 h
9.1	8.9	Análisis de cumplimiento de objetivos con director	2 h
9.2	9.1	Análisis de cumplimiento del plan original	2 h
9.3	9.2	Análisis de cumplimiento de objetivos con cliente	2 h
9.4	9.3	Elaboración de memoria técnica	40 h
9.5	9.4	Corrección de memoria técnica	20 h
9.6	9.5	Preparación de defensa pública	20 h
9.7	9.6	Defensa pública	2 h
9.8	9.7	Reunión de cierre con director	1 h
9.9	9.8	Reunión de cierre con cliente y equipo de trabajo	2 h

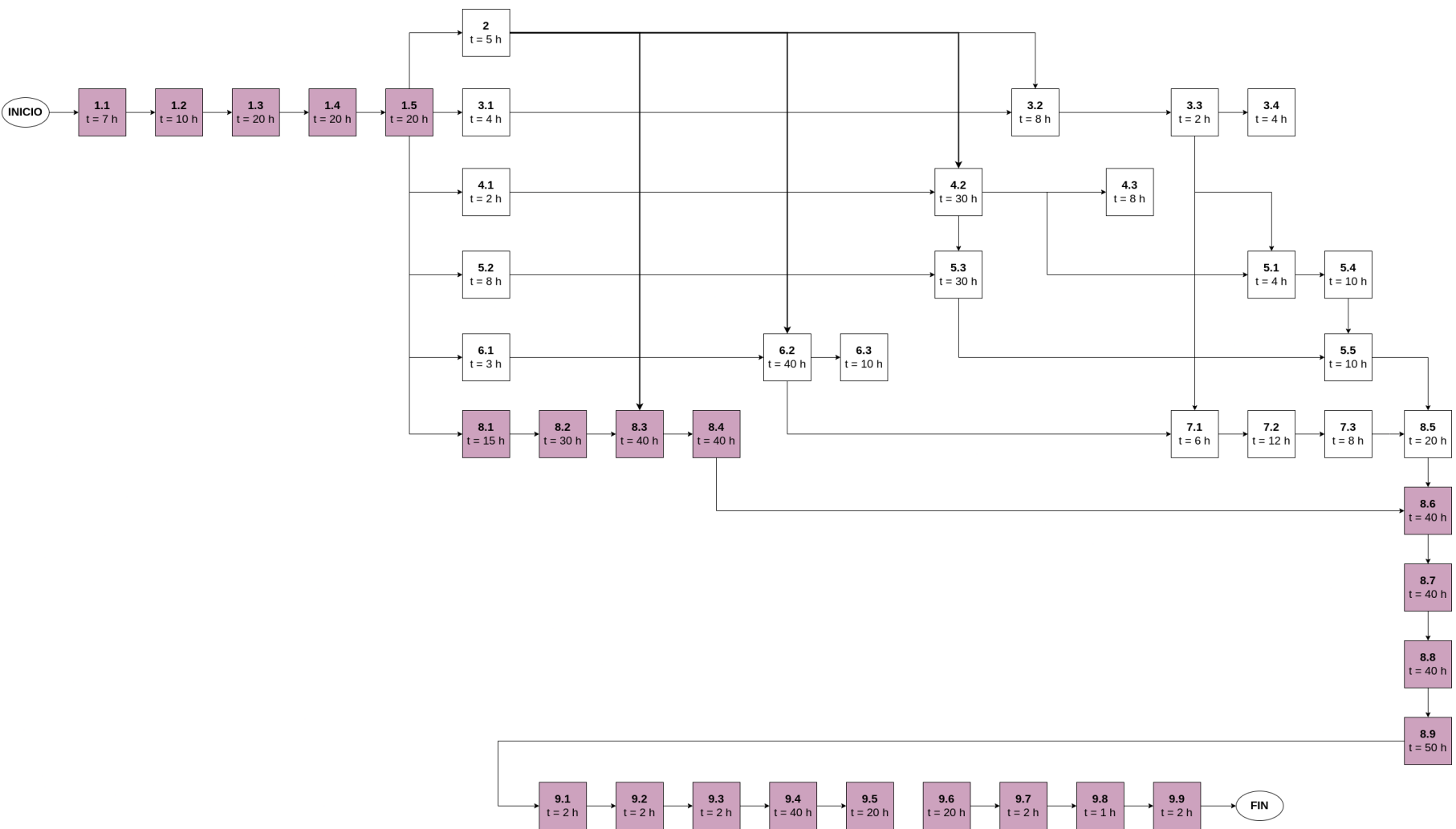
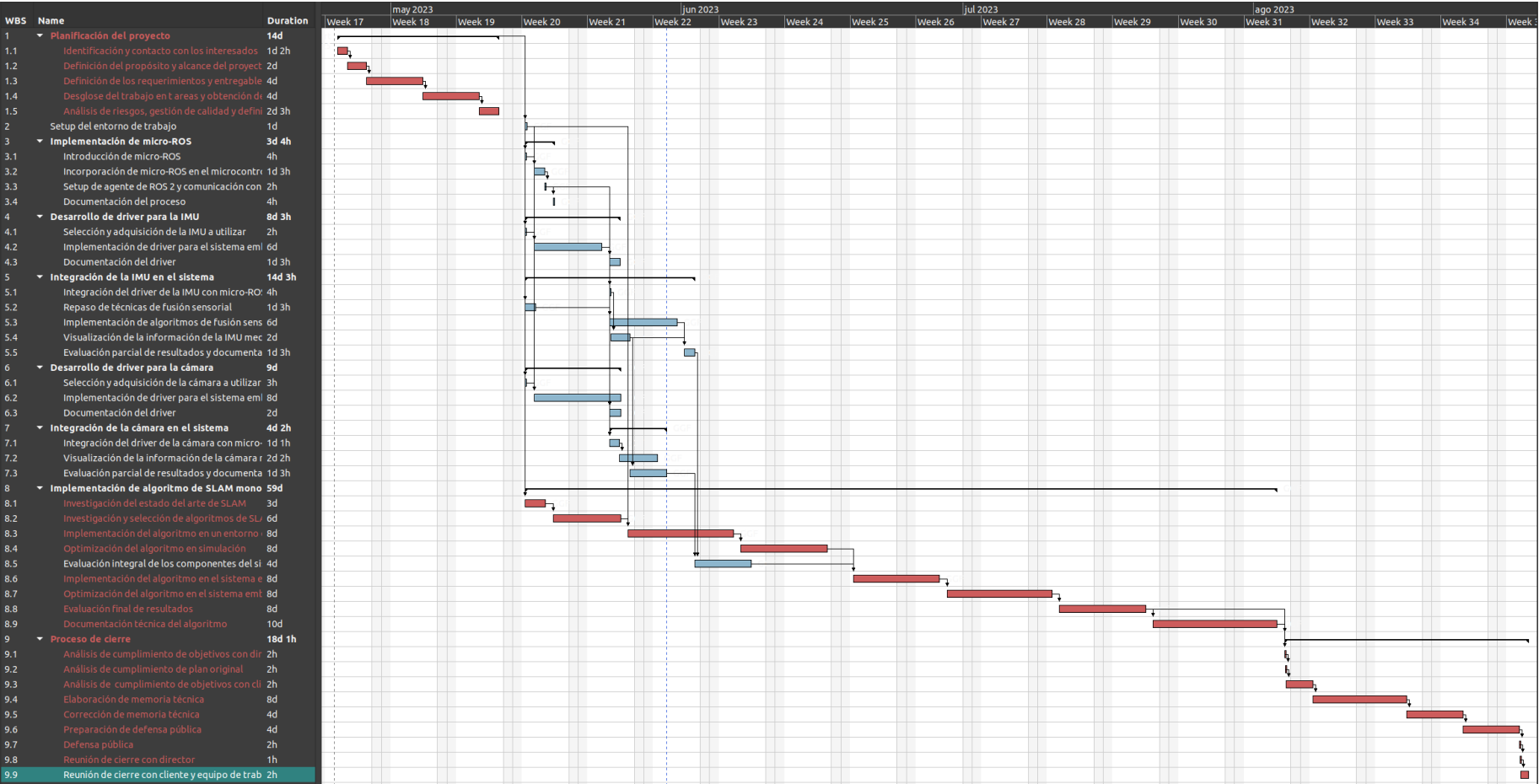


Figura 3. Diagrama de *Activity on Node*.

11. Diagrama de Gantt



12. Presupuesto detallado del proyecto

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Placa de desarrollo STM32 Nucleo	1	30000 ARS	30000 ARS
Cámara	1	5000 ARS	5000 ARS
Unidad de medición inercial	1	5000 ARS	5000 ARS
Componentes de electrónica varios	1	3000 ARS	3000 ARS
Hora de ingeniería	668	2500 ARS	1670000 ARS
SUBTOTAL			1713000 ARS
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Transporte	10	100 ARS	1000 ARS
Energía eléctrica	1	2000 ARS	2000 ARS
5 % de costos directos	1	85650 ARS	85650 ARS
SUBTOTAL			88650 ARS
TOTAL			1801650 ARS

13. Gestión de riesgos

a) Identificación de los riesgos y estimación de sus consecuencias:

Riesgo 1: la IMU o la cámara seleccionadas y adquiridas para el prototipo no tienen buen desempeño en la aplicación del proyecto.

- Severidad (S): 6, porque impacta en la calidad del resultado final del prototipo, pero no impide la obtención de una primera versión de este.
- Probabilidad de ocurrencia (O): 5, porque es un riesgo contemplado al tener un presupuesto acotado para la adquisición de materiales. Dadas las limitaciones del primer prototipo, se desarrollará una segunda versión con mejoras en estos aspectos, contemplando la adquisición de sensores en el extranjero, de ser necesario.

Riesgo 2: hay cambios o imprevistos (ej., licencias) en el equipo de trabajo.

- Severidad (S): 8, porque el grupo de trabajo es reducido y por lo tanto poco flexible a limitaciones adicionales en los recursos humanos disponibles.
- Probabilidad de ocurrencia (O): 6, porque Evangelina Castellano, colaboradora en el proyecto, puede inclinarse por otra temática para su proyecto final de grado una vez finalizada su PPS.

Riesgo 3: el responsable del proyecto puede dedicar menos de 25 horas semanales a las actividades del plan.

- Severidad (S): 9, porque hay un único responsable y el equipo de trabajo es reducido.
- Probabilidad de ocurrencia (O): 3, porque solo ocurriría ante alguna situación personal o laboral particular.

Riesgo 4: hay cambios en los requerimientos funcionales del proyecto.

- Severidad (S): 7, porque dependerá mucho de las características del requerimiento a añadir y de la instancia en la que se encuentra el proyecto.
- Probabilidad de ocurrencia (O): 6, porque pueden presentarse nuevas ideas que valga la pena explorar, tanto de parte del responsable como del cliente.

Riesgo 5: las limitaciones en el hardware impiden un buen desempeño del algoritmo investigado.

- Severidad (S): 8, porque es un riesgo que, de presentarse, será en una instancia avanzada del proyecto.
- Probabilidad de ocurrencia (O): 6, porque es conocida la cantidad de recursos demandada por los algoritmos de SLAM.

b) Tabla de gestión de riesgos:

Criterio adoptado: se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean igual o mayores a 40.

Riesgo	S	O	RPN	S*	O*	RPN*
1. Los sensores seleccionados no tienen buen desempeño.	6	5	30			
2. Cambios imprevistos en el equipo de trabajo	8	6	48	6	6	36
3. Disponibilidad del responsable menor a la estimada.	9	3	27			
4. Cambios en los requerimientos del proyecto.	7	6	42	4	4	16
5. Limitación en el hardware impide buen desempeño	8	6	48	7	4	28

c) Plan de mitigación de los riesgos que exceden el RPN máximo establecido:

Riesgo 2: se mantendrán reuniones mensuales con Evangelina Castellano y el Ing. Leandro Borgnino para estar al tanto de su interés en el tema, el posible surgimiento de nuevas líneas de investigación más prioritarias o cualquier otra variable que pueda afectar al equipo de trabajo.

- Severidad (S): 6, porque todavía existe la posibilidad de que haya cambios en el equipo de trabajo. Lo único que cambia es que se puede anticipar el cambio y adaptarse mediante la búsqueda de otro posible colaborador o realizando los ajustes necesarios en el plan de trabajo.
- Probabilidad de ocurrencia (O): 6, ya que el plan de mitigación no afecta esta variable.

Riesgo 4: se mantendrán reuniones mensuales con el Ing. Leandro Borgnino para intercambiar nuevas ideas y ponderar si merecen ser consideradas.

- Severidad (S): 4, porque el intercambio mensual permite llevar un registro de las ideas que surjan y descartar con anticipación aquellas que no valgan la pena. Las ideas que puedan tener más valor para el proyecto se considerarán con tiempo y se podrá ajustar el plan de trabajo si es necesario.

- Probabilidad de ocurrencia (O): 4, porque las reuniones mensuales actuarán de filtro de todos los posibles cambios en los requerimientos.

Riesgo 5: durante la investigación de los algoritmos SLAM se llevará a cabo un análisis de los recursos necesarios para que el desempeño sea el adecuado, es decir, lo mínimo para considerar que el sistema funciona. Si se anticipa que el hardware tiene características inferiores a las necesarias para el algoritmo, se contemplará ajustar el plan con soluciones como la selección de un algoritmo diferente a implementar, la optimización del algoritmo seleccionado o el uso de una FPGA (actividades no contempladas en el plan de trabajo).

- Severidad (S): 7, porque el impacto en el proyecto sigue siendo considerable, pero de ocurrir, se contará con la información suficiente como para tomar la decisión correcta.
- Probabilidad de ocurrencia (O): 4, porque el análisis en forma concurrente con la investigación puede permitir tener mejor información para tomar decisiones orientadas a las limitaciones de hardware.

14. Gestión de la calidad

- Req #1.1: el driver de la IMU debe ofrecer la información del giróscopo y acelerómetro de forma completa a la aplicación.
 - Verificación mediante el análisis de la hoja de datos de la IMU y mediante ensayos donde se observe la correspondencia entre el movimiento del dispositivo y la visualización de los datos obtenidos para cada eje de los sensores que conforman la IMU.
 - Validación mediante ensayo del tipo caja negra, donde se observe la correspondencia entre el movimiento del dispositivo y la visualización con la interfaz gráfica del entorno ROS 2 de los datos obtenidos con el algoritmo de fusión sensorial.
- Req #1.2: el sistema debe utilizar algoritmos y técnicas de fusión sensorial para optimizar el uso de la información obtenida por la IMU.
 - Verificación mediante ensayos que evalúen el comportamiento del algoritmo: análisis de estabilidad, seguimiento de variables internas, correspondencia experimental entre la salida del algoritmo con el movimiento real del dispositivo y cualquier otra evaluación pertinente.
 - La validación con el cliente es equivalente a la introducida en el requerimiento #1, es decir, mediante la visualización de los datos obtenidos a través del algoritmo de fusión sensorial.
- Req #1.3: el driver de la IMU debe brindar todas las posibilidades de configuración necesarias para poder adaptar la escala y frecuencia de muestreo de los sensores a los requisitos de la aplicación.
 - Verificación mediante el análisis de la hoja de datos de la IMU y la demostración de la correcta configuración del dispositivo.
 - Validación mediante la inspección de la documentación generada para el driver de la IMU y la demostración de la correcta configuración del dispositivo a través de la interfaz gráfica del entorno ROS 2.

- Req #1.4: el driver de la cámara debe proveer la información correcta a la aplicación, dadas las capacidades del dispositivo.
 - Verificación mediante el análisis de la hoja de datos de la cámara y mediante la demostración de las imágenes obtenidas a través del driver desarrollado.
 - Validación mediante la visualización con la interfaz gráfica del entorno ROS 2 de las imágenes obtenidas a través del driver desarrollado.
- Req #1.5: el driver de la IMU y de la cámara deben implementarse de forma modular para que puedan utilizarse independientemente del proyecto.
 - Verificación mediante la inspección del *port* de cada uno de los drivers y la evaluación de su funcionamiento de forma independiente (requerimientos #1 y #4).
 - Validación mediante la inspección de la documentación generada sobre cómo desarrollar el *port* de los drivers para otros microcontroladores.
- Req #1.6: el sistema embebido debe utilizar micro-ROS para interactuar e integrarse fácilmente con el entorno de desarrollo, monitoreo y evaluación de ROS 2.
 - Verificación mediante la demostración del funcionamiento del agente de ROS 2 en comunicación con el sistema embebido, intercambiando datos en un canal genérico.
 - La validación es de forma indirecta con la validación de los requerimientos #1 y #4, dado que la visualización de los datos obtenidos a través de los sensores es con herramientas de ROS2.
- Req #1.7: el sistema embebido debe hacer un uso adecuado de los mecanismos de comunicación de micro-ROS con ROS 2 para explotar todas las ventajas que ofrece el framework.
 - Verificación y validación a través del análisis del grafo de comunicación resultante en ROS 2.
- Req #1.8: el sistema debe estar diseñado de forma modular para favorecer su escalabilidad.
 - Verificación mediante la demostración del funcionamiento del sistema anulando (o desactivando) los distintos módulos de forma secuencial.
 - Validación mediante la demostración del funcionamiento de entornos de trabajo predefinidos en ROS 2 que permitan operar con módulos individuales del sistema.
- Req #1.9: el sistema debe ejecutar un algoritmo investigado previamente de SLAM monocular.
 - Verificación mediante comparación entre los resultados obtenidos con el mismo algoritmo en un entorno de simulación y los obtenidos con el prototipo real.
 - Validación mediante demostración del algoritmo implementado en un entorno de simulación y demostración del prototipo en funcionamiento en un ambiente controlado.
- Req #4.1: las herramientas desarrolladas en el entorno de ROS 2 deben también cumplir el propósito de interfaz gráfica y permitir la visualización de la información obtenida a través de los sensores.
 - Verificación y validación indirecta mediante los requerimientos #1 y #4.

- Req #4.2: el sistema desarrollado con ROS 2 debe ofrecer herramientas que permitan operar y reconfigurar el sistema.
 - Verificación y validación indirecta mediante los requerimientos #3 y #4.
- Req #4.3: el sistema desarrollado con ROS 2 debe ofrecer herramientas que permitan evaluar el desempeño del algoritmo implementado, como puede ser el ajuste de parámetros del sistema *on-the-fly*, la visualización del mapa del entorno y la ubicación espacial de la plataforma en él y otras herramientas que se consideren necesarias.
 - Verificación y validación mediante la demostración del funcionamiento del sistema, cubriendo una lista predefinida detallada de los requisitos que debe cumplir el sistema desarrollado con ROS 2.

15. Procesos de cierre

Una vez finalizado el proyecto se realizarán las siguientes actividades:

- Análisis del grado de cumplimiento de los objetivos y requerimientos junto con el Dr. Ing. Fabio Ardiani.
- Análisis de cumplimiento de la versión original del plan de trabajo, contrastando el registro de actividades realizadas con el diagrama de Gantt original.
- Análisis del grado de cumplimiento de los objetivos y requerimientos junto con el cliente, el Ing. Leandro Borgnino.
- Elaboración de un documento con todos los problemas que surgieron durante el desarrollo del proyecto y las soluciones encontradas.
- Elaboración de un documento con las actividades que no se alcanzaron a realizar en el tiempo estimado y todas aquellas actividades realizadas que no se contemplaron en el plan original.
- Elaboración de un documento con el trabajo a futuro, que sirva como base de un plan de proyecto que continúe la misma línea de investigación y desarrollo.
- Defensa pública del proyecto por parte del responsable.
- Reunión privada con el equipo de trabajo para compartir *feedback* sobre la experiencia obtenida en el lapso del proyecto.
- Reunión virtual, organizada por el responsable, con el Dr. Ing. Fabio Ardiani como acto de agradecimiento.
- Reunión con el Ing. Leandro Borgnino y el equipo de trabajo, organizada por el responsable, para dar un cierre al proyecto y evaluar los próximos proyectos y actividades a realizar en conjunto.