



Desarrollo de plataforma móvil para SLAM monocular

Autor:

Gonzalo Gabriel Fernandez

Director:

Nombre del Director (pertenencia)

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	6
3. Propósito del proyecto	7
4. Alcance del proyecto	7
5. Supuestos del proyecto.	7
6. Requerimientos	7
7. Historias de usuarios (<i>Product backlog</i>).	8
8. Entregables principales del proyecto	8
9. Desglose del trabajo en tareas	9
10. Diagrama de Activity On Node.	9
11. Diagrama de Gantt	10
12. Presupuesto detallado del proyecto	13
13. Gestión de riesgos	13
14. Gestión de la calidad	14
15. Procesos de cierre	15

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

Acta de constitución del proyecto

Córdoba, 25 de abril de 2023

Por medio de la presente se acuerda con el Ing. Gonzalo Gabriel Fernandez que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará “Desarrollo de plataforma móvil para SLAM monocular”, consistirá esencialmente en **la implementación de un prototipo de un sistema de control de temperatura de un calefón**, y tendrá un presupuesto preliminar estimado de **600** h de trabajo, con fecha de inicio 25 de abril de 2023 y fecha de presentación pública 15 de mayo de 2024.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Leandro Borgnino
Fundación Fulgor

Nombre del Director
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 i ejecutar sus propias acciones de manera autónoma, sin la necesidad de una intervención humana constante. Esta capacidad 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 técnicas de bajo costo es importante porque puede ayudar a reducir los costos de implementación y hacer que la tecnología sea más 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. 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 robot, posibilitando calcular su posición y orientación en relación a una posición inicial. Es particularmente útil en entornos donde el movimiento del robot es predominantemente lateral, ya que se basa en la medición del movimiento de los objetos en la imagen en relación con un marco de referencia. Además, la odometría visual se puede 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.

SLAM (Simultaneous Localization and Mapping) es una técnica de visión por computadora que permite a un robot móvil construir un mapa del entorno en el que se encuentra, mientras simultáneamente estima su propia posición y orientación en ese entorno. En el caso de SLAM monocular, se utiliza una única cámara para capturar imágenes del entorno, y en tiempo real se utilizan diferentes algoritmos para procesar las imágenes y estimar la posición y orientación del robot.

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. Se han realizado experimentos, implementando y comparando 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, y dada su naturaleza modular, permite que la migración consista en el reemplazo de un módulo encapsulado que contenga el algoritmo, como se observa en la Figura 1.

ROS 2 es un framework de software libre y de código abierto diseñado para permitir el desarrollo de aplicaciones robóticas distribuidas. Proporciona un conjunto de herramientas para la creación, gestión, depuración y análisis de sistemas robóticos. Es la segunda versión de ROS (Robot Operating System) y ha sido desarrollado por la comunidad con el objetivo de mejorar la escalabilidad, la fiabilidad y la seguridad de ROS, entre otras características. Una de las

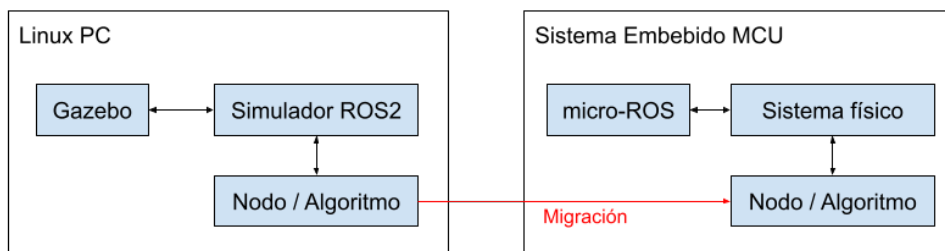


Figura 1. Migración de algoritmos en entorno de simulación a sistema físico

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. ROS 2 también se enfoca en proporcionar una abstracción de hardware más clara y permitir el uso de diferentes sistemas operativos y arquitecturas de hardware.

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 limitados recursos. 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 utilizan tanto sistemas embebidos como sistemas de propósito general, y que todos ellos pueden comunicarse a través de la misma plataforma.

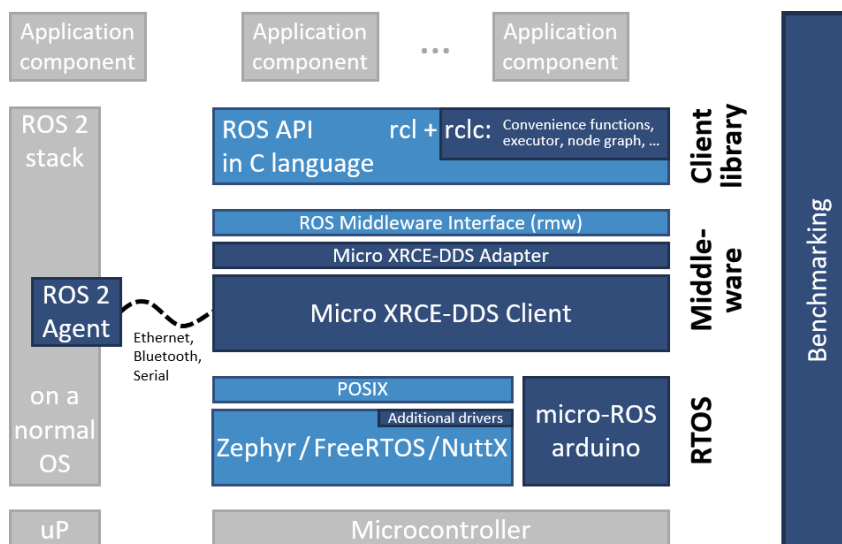


Figura 2. Arquitectura de micro-ROS

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Leandro Borgnino	Fundación Fulgor	Investigador
Responsable	Gonzalo Gabriel Fernandez	FIUBA	Alumno
Orientador	Nombre del Director	pertenencia	Director Trabajo final
Equipo	Evangelina Castellano Gonzalo Gabriel Fernandez	-	Colaboradores
Usuario final	Grupos de investigación	Fundación Fulgor	-

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 embebidos, 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 del mismo. 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 todos los elementos de hardware vinculados al sistema se encuentran adquiridos por Fundación Fulgor y su interconexión es trivial dado el conocimiento de los integrantes del equipo de trabajo.

6. Requerimientos

Los requerimientos deben numerarse y de ser posible estar agruparlos por afinidad, por ejemplo:

1. Requerimientos funcionales
 - 1.1. El sistema debe...
 - 1.2. Tal componente debe...
 - 1.3. El usuario debe poder...
2. Requerimientos de documentación
 - 2.1. Requerimiento 1
 - 2.2. Requerimiento 2 (prioridad menor)
3. Requerimiento de testing...
4. Requerimientos de la interfaz...
5. Requerimientos interoperabilidad...
6. etc...

Leyendo los requerimientos se debe poder interpretar cómo será el proyecto y su funcionalidad.

Indicar claramente cuál es la prioridad entre los distintos requerimientos y si hay requerimientos opcionales.

No olvidarse de que los requerimientos incluyen a las regulaciones y normas vigentes!!!

Y al escribirlos seguir las siguientes reglas:

- Ser breve y conciso (nadie lee cosas largas).
- Ser específico: no dejar lugar a confusiones.
- Expresar los requerimientos en términos que sean cuantificables y medibles.

7. Historias de usuarios (*Product backlog*)

Descripción: En esta sección se deben incluir las historias de usuarios y su ponderación (*history points*). Recordar que las historias de usuarios son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema. La ponderación es un número entero que representa el tamaño de la historia comparada con otras historias de similar tipo.

El formato propuesto es: como [rol] quiero [tal cosa] para [tal otra cosa].”

Se debe indicar explícitamente el criterio para calcular los *story points* de cada historia

8. Entregables principales del proyecto

Los entregables del proyecto son (ejemplo):

- Manual de uso
- Diagrama de circuitos esquemáticos
- Código fuente del firmware
- Diagrama de instalación
- Informe final
- etc...

9. Desglose del trabajo en tareas

El WBS debe tener relación directa o indirecta con los requerimientos. Son todas las actividades que se harán en el proyecto para dar cumplimiento a los requerimientos. Se recomienda mostrar el WBS mediante una lista indexada:

1. Grupo de tareas 1
 - 1.1. Tarea 1 (tantas h)
 - 1.2. Tarea 2 (tantas hs)
 - 1.3. Tarea 3 (tantas h)
2. Grupo de tareas 2
 - 2.1. Tarea 1 (tantas h)
 - 2.2. Tarea 2 (tantas h)
 - 2.3. Tarea 3 (tantas h)
3. Grupo de tareas 3
 - 3.1. Tarea 1 (tantas h)
 - 3.2. Tarea 2 (tantas h)
 - 3.3. Tarea 3 (tantas h)
 - 3.4. Tarea 4 (tantas h)
 - 3.5. Tarea 5 (tantas h)

Cantidad total de horas: (tantas h)

Se recomienda que no haya ninguna tarea que lleve más de 40 h.

10. Diagrama de Activity On Node

Armaz el AoN a partir del WBS definido en la etapa anterior.

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semicríticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color, como se muestra en el siguiente ejemplo:



Figura 3. Diagrama de *Activity on Node*.

11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de Gantt, entre los cuales destacamos:

- Planner
- GanttProject
- Trello + *plugins*. En el siguiente link hay un tutorial oficial:
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately, herramienta online colaborativa.
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- Se puede hacer en latex con el paquete *pgfgantt*
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la “tabla” del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 4, se muestra un ejemplo de diagrama de Gantt realizado con el paquete de *pgfgantt*. En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.



Figura 4. Diagrama de Gantt de ejemplo



Figura 5. Ejemplo de diagrama de Gantt rotado

12. Presupuesto detallado del proyecto

Si el proyecto es complejo entonces separarlo en partes:

- Un total global, indicando el subtotal acumulado por cada una de las áreas.
- El desglose detallado del subtotal de cada una de las áreas.

IMPORTANTE: No olvidarse de considerar los **COSTOS INDIRECTOS**.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
TOTAL			

13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S):
- Ocurrencia (O):

Riesgo 3:

- Severidad (S):

■ Ocurrencia (O):

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN=S \times O$)

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado: Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

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

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación). Nueva asignación de S y O, con su respectiva justificación: - Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S). - Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

14. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento.
 - Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar
 - Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc. Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno. En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
- Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, y los problemas que surgieron y cómo se solucionaron: - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores: - Indicar esto y quién financiará los gastos correspondientes.