



## GRADO EN INGENIERÍA DE ROBÓTICA SOFTWARE

Escuela de Ingeniería de Fuenlabrada

Curso académico 2023-2024

### Trabajo Fin de Grado

Programación de flujo de datos en  
multirobótica con ROS2 y Zenoh-Flow

**Tutor:** Julio Vega Pérez

**Autor:** Unai Sanz Conejo



Este trabajo se distribuye bajo los términos de la licencia internacional CC BY-NC-SA International License (Creative Commons AttributionNonCommercial-ShareAlike 4.0). Usted es libre de *(a) compartir*: copiar y redistribuir el material en cualquier medio o formato; y *(b) adaptar*: remezclar, transformar y crear a partir del material. El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia:

- *Atribución.* Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatante.
- *No comercial.* Usted no puede hacer uso del material con propósitos comerciales.
- *Compartir igual.* Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

# Agradecimientos

---

Quisiera expresar mi más sincero agradecimiento a todas las personas que contribuyeron a la realización de este trabajo. En primer lugar, agradezco al equipo de ZettaScale por darme la oportunidad de realizar mis prácticas con ellos y de aprender incontables aspectos acerca de las telecomunicaciones y por su persistente ayuda. También agradezco a mi tutor de TFG por su orientación constante y su gran paciencia a lo largo de este proceso.

Además estoy profundamente agradecido a mis compañeros de clase por sus ideas y debates constructivos, que han enriquecido enormemente mi investigación.

No puedo dejar de agradecer a mis padres, hermana, amigos cercanos y pareja por sus ideas y debates constructivos, consejos y apoyo moral e incondicional.

Por último, pero no menos importante, quiero expresar mi gratitud a todas las fuentes y recursos que consulté durante la elaboración de este trabajo, así como a cualquier institución o persona que haya contribuido de alguna manera, aunque indirecta, a este proyecto.

Sin el apoyo de todas estas personas y entidades, este trabajo no habría sido posible. Gracias de todo corazón.

*A mi abuelo;  
que estaría sumamente orgulloso de mi trabajo.*

Madrid, xx de xxxxxxx de 2024

*Unai Sanz*

# Resumen

---

Escribe aquí el resumen del trabajo. Un primer párrafo para dar contexto sobre la temática que rodea al trabajo.

Un segundo párrafo concretando el contexto del problema abordado.

En el tercer párrafo, comenta cómo has resuelto la problemática descrita en el anterior párrafo.

Por último, en este cuarto párrafo, describe cómo han ido los experimentos.

# Acrónimos

---

**DDS** *Data Distribution Service*

**JAXA** *Japan Aerospace Exploration Agency*

**NASA** *National Aeronautics and Space Administration*

**RMW** *ROS Middleware Interface*

**ROS2** *Robotic Operating System (ROS) 2*

**STEM** *Ciencia, Tecnología, Ingeniería y Matemáticas*

**UPC** *Universidad Politécnica de Cataluña*

**URJC** *Universidad Rey Juan Carlos*

# Índice general

---

<b>1. Introducción</b>	<b>1</b>
1.1. Problemas de ROS en relación con el aprendizaje . . . . .	8
1.2. Segunda sección . . . . .	8
1.2.1. Números . . . . .	9
1.2.2. Listas . . . . .	9
<b>2. Objetivos</b>	<b>11</b>
2.1. Descripción del problema . . . . .	11
2.2. Requisitos . . . . .	11
2.3. Metodología . . . . .	11
2.4. Plan de trabajo . . . . .	11
<b>3. Plataforma de desarrollo</b>	<b>12</b>
<b>4. Diseño</b>	<b>13</b>
4.1. Snippets . . . . .	13
4.2. Verbatim . . . . .	13
4.3. Ecuaciones . . . . .	14
4.4. Tablas o cuadros . . . . .	14
<b>5. Conclusiones</b>	<b>16</b>
5.1. Conclusiones . . . . .	16
5.2. Corrector ortográfico . . . . .	17
<b>Bibliografía</b>	<b>18</b>

# Índice de figuras

---

1.1.	Rover Perseverance y helicóptero Ingenuity de la NASA en Marte. . . . .	2
1.2.	Robots Spot y Atlas de Boston Dynamics. . . . .	3
1.3.	Sora-Q, robot <i>low-cost</i> enviado a la Luna, por Takara Tomy y la JAXA. .	4
1.4.	Arduino. . . . .	5
1.5.	Código de Arduino en Scratch. . . . .	5
1.6.	Arquitectura de ROS y ROS2. . . . .	6
1.7.	Múltiples robots durante una operación de búsqueda y rescate. . . . .	7
1.8.	Robots educativos, modelos Turtlebot 2 (arriba) y Turtlebot 4 (abajo). .	7
1.9.	Flujo de datos de un robot para la detección y seguimiento de códigos QR. . . . .	8

# Listado de códigos

---

4.1. Función para buscar elementos 3D en la imagen . . . . .	13
4.2. Cómo usar un Slider . . . . .	14

# Listado de ecuaciones

---

4.1. Ejemplo de ecuación con fracciones . . . . .	14
4.2. Ejemplo de ecuación con array y letras y símbolos especiales . . . . .	14

# Índice de cuadros

---

4.1. Parámetros intrínsecos de la cámara . . . . .	15
--	----

---

# Capítulo 1

## Introducción

---

*La creatividad es la inteligencia divirtiéndose*

Albert Einstein

La robótica es un campo multidisciplinario que se concentra en el diseño, construcción, programación y operación de robots. Estos dispositivos electromecánicos, con frecuencia modelados antropomórficamente o zoomórficamente, están destinados a realizar tareas de manera autónoma o semiautónoma en una variedad de entornos. La robótica ha experimentado un rápido crecimiento y expansión desde sus inicios en la década de 1950, abarcando una amplia gama de aplicaciones en la industria, la medicina, la exploración espacial y el entretenimiento, entre otras. La robótica sigue evolucionando y desempeñando un papel cada vez más importante en nuestra sociedad moderna gracias a los avances constantes en tecnologías como la inteligencia artificial, los sensores y los actuadores, y ha conformado un factor crucial en la exploración espacial, que en particular, ha impulsado el desarrollo de tecnologías que luego se han aplicado en la Tierra, mejorando la calidad de vida de las personas. Ejemplos destacados incluyen los sistemas de purificación de agua, los tejidos avanzados como la viscoelástica, los pañales y los dispositivos de imágenes médicas como la resonancia magnética. Estas innovaciones, así como muchas otras, han proporcionado acceso a agua potable en regiones remotas, a colchones y almohadas que promueven un mejor descanso, a mayores facilidades en cuanto al cuidado de los niños y mayores y a diagnósticos médicos precisos sin radiación nociva. Así, la investigación espacial no solo expande nuestro conocimiento del universo, sino que también beneficia directamente a la humanidad en la Tierra. Un ejemplo ilustrativo de la aplicación de la robótica móvil en la exploración espacial se encuentra en los resistentes robots enviados a diferentes planetas del sistema solar en busca de datos científicos, como lo evidencian numerosas misiones científicas, como la ilustrada en la Figura 1.1, donde se muestran dos robots específicamente destinados a Marte por la NASA en 2020.

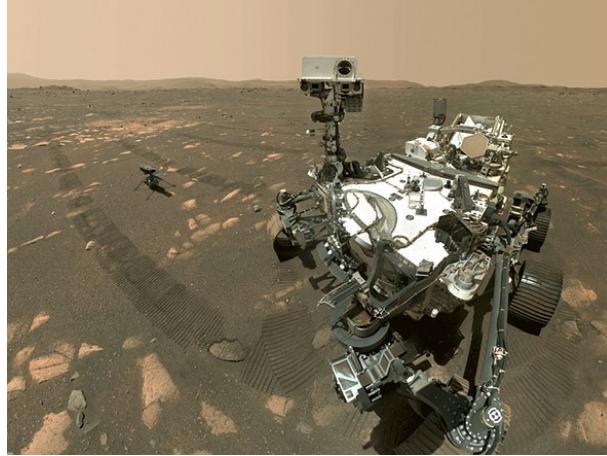


Figura 1.1: Rover Perseverance y helicóptero Ingenuity de la NASA en Marte.

En este contexto introductorio, se exploran conceptos fundamentales de la robótica, así como áreas clave que están redefiniendo los límites de la tecnología, como son la robótica móvil y la robótica de bajo coste. Estas dos áreas no solo son grandes avances tecnológicos, sino que también tienen un gran impacto en cómo usamos la tecnología en nuestra vida diaria.

La robótica móvil ha emergido como un campo multidisciplinario que fusiona la ingeniería, la inteligencia artificial y múltiples ramas de la robótica y la mecatrónica para crear sistemas capaces de moverse y operar en entornos dinámicos, interactuando con los mismos de manera que reciben información mediante sensores y aplican cambios en el medio gracias a sus actuadores, todo ello de manera inteligente debido al software integrado y a su capacidad de cómputo. Desde sus inicios, ha sido impulsada por avances significativos en tecnología, incluyendo sensores y actuadores cada vez más sofisticados, poder de procesamiento mejorado y algoritmos de control más eficientes. Estos avances han permitido la aplicación de robots móviles en una amplia gama de áreas, desde la exploración espacial y submarina hasta la logística industrial y la atención médica, siendo ya parte indispensable de nuestras vidas y mejorando su calidad. En este contexto, la investigación en robótica móvil se centra en desarrollar sistemas autónomos capaces de navegar de manera segura y eficiente en entornos desconocidos, adaptarse a cambios imprevistos y realizar tareas complejas de manera autónoma. Un ejemplo representativo de este tipo de robots se puede ver en la Figura 1.2, donde se pueden observar dos de los robots más desarrollados en el ámbito móvil, demostrando una gran versatilidad en una variedad de entornos para ejecutar una amplia variedad de tareas, desde abrir puertas hasta transportar cargas de peso, realizar trabajos manuales, o

incluso seguir rutinas de deportivas variadas, que en muchos casos iguala o incluso supera la de los humanos, dependiendo de la situación específica.



Figura 1.2: Robots Spot y Atlas de Boston Dynamics.

La robótica de bajo coste se refiere al desarrollo e implementación de sistemas robóticos como los descritos anteriormente utilizando componentes y recursos económicos, con el objetivo de hacer la tecnología robótica más accesible y asequible para una amplia gama de aplicaciones y usuarios. Este enfoque busca reducir los costos asociados con la construcción y operación de robots, empleando materiales económicos, hardware de bajo coste y técnicas de fabricación eficientes. En el contexto de la robótica móvil, los sistemas de bajo coste pueden ofrecer soluciones viables para aplicaciones con presupuestos limitados o despliegues a gran escala, abarcando un papel crucial en áreas como la educación, la investigación académica, la asistencia social y la exploración de entornos remotos o peligrosos. Además de su utilidad práctica, la robótica de bajo coste también promueve la innovación y el desarrollo de nuevas tecnologías al proporcionar una plataforma accesible para la experimentación y la creatividad abierta a una amplia comunidad. Un ejemplo destacado es el robot Sora-Q visible en la Figura 1.3, enviado a la Luna recientemente por la JAXA y desarrollado por una empresa de juguetería japonesa. Tras completar su misión, fue comercializado por 150€, lo que ilustra cómo la tecnología robótica puede volverse accesible para un público más amplio, incluso después de su participación en misiones espaciales. La Educación en robótica suele basarse en este tipo de robótica de bajo coste, ya que las instituciones educativas enfrentan limitaciones presupuestarias que dificultan la adquisición de sistemas más costosos, por su gran número de alumnos, a los que no podrían proveer de sistemas de este calibre de otra manera.



Figura 1.3: Sora-Q, robot *low-cost* enviado a la Luna, por Takara Tomy y la JAXA.

La robótica de bajo coste no solo ha democratizado el acceso a la tecnología robótica, sino que también ha revolucionado la forma en que se enseña la robótica en las escuelas. Este enfoque económico ha permitido a las instituciones educativas superar las limitaciones presupuestarias y proporcionar a un mayor número de estudiantes la oportunidad de involucrarse en actividades prácticas de robótica.

En España, al igual que en el resto de la comunidad europea, la relevancia de la robótica en el ámbito educativo ha ido ganando terreno en los últimos años, con un aumento en la implementación de programas educativos que incluyen actividades prácticas de robótica en escuelas primarias y secundarias, promoviendo la creatividad, el pensamiento crítico y las habilidades tecnológicas entre los estudiantes, además de fomentar el trabajo en equipo y la resolución de problemas complejos, preparándolos para futuras carreras relacionadas con la tecnología. Desde la década de los 90, se han implementado programas piloto y competiciones robóticas, como el programa Robolot (1992), desarrollado por la UPC, las Olimpiadas de Informática (1993), que incorporaron desafíos relacionados con la programación de robots, así como la RoboCupJunior (2000), ofreciendo a los estudiantes la oportunidad de diseñar, construir y programar robots para competir en diferentes categorías. Desde alrededor de 2014, dependiendo de la comunidad autónoma de España, se han introducido programas y asignaturas que incluyen la robótica como parte esencial del plan de estudios. Al ser un área introductoria a la robótica se busca simplificar la programación, utilizando herramientas como placas como Arduino y plataformas como Scratch para simplificar la programación debido a su interfaz de bloques y la asociación de ideas a colores, como se muestra en las Figuras 1.4 y 1.5.

Este enfoque introductorio ha allanado el camino para que los estudiantes se sumerjan en áreas más avanzadas de la robótica, como la robótica móvil, que juega



Figura 1.4: Arduino.



Figura 1.5: Código de Arduino en Scratch.

un papel crucial en el desarrollo de habilidades tecnológicas y en la motivación de los estudiantes, que al poder ver los resultados en movimiento de manera autónoma, les origina una sensación de realización y entusiasmo por aprender.

Las placas utilizadas en el ámbito educativo, mencionadas anteriormente, resultan ideales para estos propósitos debido a su coste y simplicidad, sin embargo, también imponen ciertas limitaciones en las capacidades del robot y en la posibilidad de añadir hardware externo más complejo, y por tanto en la propia originalidad y aprendizaje de los estudiantes, restringiendo así su creatividad, innovación y potencial de creación, una vez han obtenido unos conocimientos básicos. Es entonces cuando se encuentran con el desafío de dar el siguiente paso: la programación de robots utilizando ROS, el *middleware* estándar por excelencia en robótica. Este proceso implica un considerable escalón de aprendizaje, ya que no solo deben dominar un lenguaje de programación más complejo, sino que también deben comprender el entorno que rodea a esta plataforma robótica, que incluye aspectos como las comunicaciones, la arquitectura software, la programación modular y orientada a objetos, algoritmos y estructuras de datos, entre otros, que suelen conllevar decenas de asignaturas individuales en los grados de robótica de cualquier universidad. Dicha complejidad se puede ver ilustrada en la Figura 1.6, donde se muestra un esquema simplificado de la arquitectura de ROS y ROS2 que puede resultar abrumadora para aquellos que se están iniciando en este ámbito. Por este motivo, resulta evidente la necesidad de un paso intermedio que pueda actuar como puente entre estos dos niveles de aprendizaje, el cual podría ser incorporado, por ejemplo, en el programa educativo del bachillerato tecnológico o de centros de educación secundaria. Este nivel intermedio facilitaría la transición entre las habilidades adquiridas en la enseñanza primaria o secundaria y los requisitos más avanzados de la

universidad en el campo de la robótica.

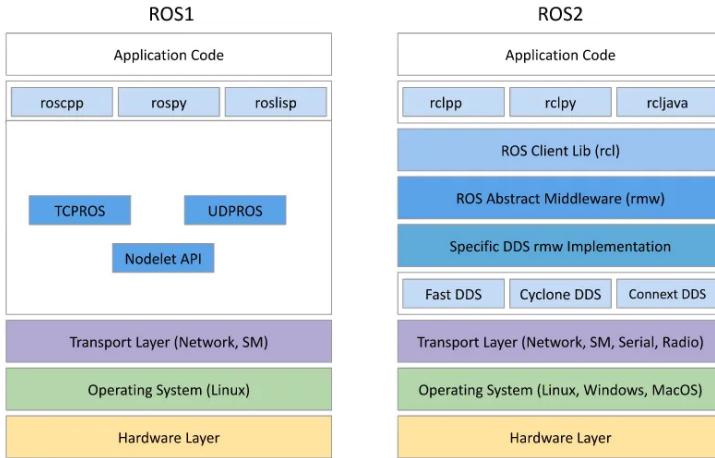


Figura 1.6: Arquitectura de ROS y ROS2.

La robótica móvil sienta las bases a campos más complejos como la multirobótica, que representa un paso adelante en la complejidad y la escala de los sistemas robóticos.

La multirobótica es un campo de investigación que estudia y desarrolla sistemas robóticos compuestos por múltiples robots que trabajan en conjunto para realizar una variedad tareas complejas, que ha sido ampliamente estudiada en artículos como [Verma and Ranga, 2021]. Estos sistemas pueden dividirse las tareas, como la exploración de entornos desconocidos o la búsqueda y rescate en áreas de difícil acceso. La colaboración entre robots implica, entre otros aspectos, establecer comunicaciones para compartir información y entender de un mejor modo el mundo y contexto que les rodea, crear mapas del entorno para localizarse y navegar por el mismo de manera controlada [Trawny et al., 2009, Fox et al., 2000]. Además la necesidad de manipular objetos también es una tarea relevante [Parker, 2003]. En términos de coordinación, los equipos de robots pueden operar eficientemente asignando roles y responsabilidades [Alami et al., 1998], y aunque estos sistemas adoptan una estructura centralizada, donde un robot puede asumir roles específicos, la tendencia actual se inclina hacia sistemas más descentralizados, donde la coordinación y la optimización son fundamentales [Sheng et al., 2006]. Al mejorar la eficiencia y la robustez de los sistemas robóticos, la multirobótica se ha convertido en un campo importante de investigación. Los principios y problemas técnicos en este campo se exploran en diversos contextos, como se ilustra en la Figura 1.7, en la cual están realizando una tarea de búsqueda y rescate, mediante la división de un área, probablemente desconocida.



Figura 1.7: Múltiples robots durante una operación de búsqueda y rescate.

En el ámbito educativo, la multirobótica ofrece una oportunidad única para involucrar a los estudiantes en actividades prácticas y colaborativas. Al trabajar con sistemas multirobot, los estudiantes no solo adquieren conocimientos sobre programación, control y mecánica de robots, sino que también exploran conceptos como son las telecomunicaciones entre robots, la coordinación, la planificación y asignación de tareas con o sin prioridades, la localización y navegación conjunta, así como la seguridad que se requiere para evitar colisiones entre ellos. Además, la multirobótica proporciona un entorno de aprendizaje dinámico y estimulante que despierta aún más la curiosidad y la creatividad de los estudiantes, preparándolos para enfrentar los desafíos del mundo tecnológico en constante evolución. Se puede ver un ejemplo de los robots educativos de nuestro laboratorio de robótica de la URJC en la Figura 1.8.



Figura 1.8: Robots educativos, modelos Turtlebot 2 (arriba) y Turtlebot 4 (abajo).

Las mencionadas telecomunicaciones entre robots son fundamentales en la multirobótica, ya que garantizan una comunicación rápida, óptima, eficiente y ordenada entre los distintos dispositivos, necesaria para un correcto desempeño de la

funcionalidad en cuestión. Sin embargo, este proceso puede enfrentarse a desafíos como la congestión de la red, y la consecuente pérdida de mensajes, que pueden ser críticos para el correcto funcionamiento de los robots. Por este motivo es crucial gestionar cuidadosamente la cantidad de robots y mensajes generados, minimizándolos para optimizar el rendimiento del sistema y evitando de esta manera un cuello de botella.

Mantener un flujo de datos correcto es fundamental para solventar los problemas de telecomunicaciones mencionados en el desarrollo de sistemas de multirobótica. Además, simplifica el desarrollo del software al proporcionar una clara visión de la dirección, el origen y el destino de los datos en cada momento. Esto permite dividir el programa en partes claramente diferenciadas, normalmente llamadas nodos, modularizándolo y dando lugar a la división del problema último en varios problemas más simples y fáciles de atajar. Como resultado, el desarrollo se vuelve un proceso más sostenible y escalable, y por tanto, más fácil de llevar a cabo por los estudiantes. Podemos ver ejemplificado este proceso en la Figura 1.9.

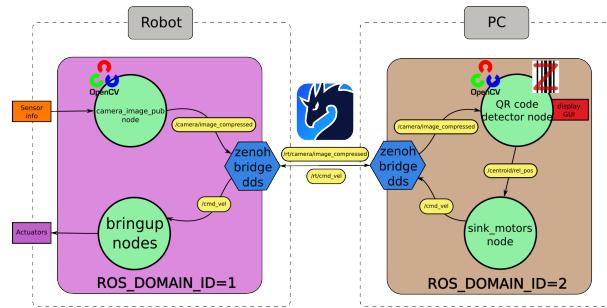


Figura 1.9: Flujo de datos de un robot para la detección y seguimiento de códigos QR.

## 1.1. Problemas de ROS en relación con el aprendizaje

## 1.2. Segunda sección

No olvides incluir imágenes y referenciarlas, como la Figura ??.

Ni tampoco olvides de poner las URLs como notas al pie. Por ejemplo, si hablo de la Robocup<sup>1</sup>.

---

<sup>1</sup><http://www.robocup.org>

### 1.2.1. Números

En lugar de tener secciones interminables, como la Sección 1.1, divídelas en subsecciones.

Para hablar de números, mételos en el entorno *math* de L<sup>A</sup>T<sub>E</sub>X, por ejemplo,  $1,5Kg$ . También puedes usar el símbolo del Euro como aquí:  $1.500\text{€}$ .

### 1.2.2. Listas

Cuando describas una colección, usa `itemize` para ítems o `enumerate` para enumerados. Por ejemplo:

- *Entorno de simulación.* Hemos usado dos entornos de simulación: uno en 3D y otro en 2D.
  - *Entornos reales.* Dentro del campus, hemos realizado experimentos en Biblioteca y en el edificio de Gestión.
1. Primer elemento de la colección.
  2. Segundo elemento de la colección.

**Referencias bibliográficas** Cita, sobre todo en este capítulo, referencias bibliográficas que respalden tu argumento. Para citarlas basta con poner la instrucción `\cite` con el identificador de la cita. Por ejemplo: libros como [Vega et al., 2012], artículos como [Vega and Cañas, 2019], URLs como [Vega, 2019], tesis como [Vega, 2018a], congresos como [Vega, 2018b], u otros trabajos fin de grado como [Vega, 2008].

Las referencias, con todo su contenido, están recogidas en el fichero `bibliografia.bib`. El contenido de estas referencias está en formato BibTex. Este formato se puede obtener en muchas ocasiones directamente, desde plataformas como `Google Scholar` u otros repositorios de recursos científicos.

Existen numerosos estilos para reflejar una referencia bibliográfica. El estilo establecido por defecto en este documento es APA, que es uno de los estilos más comunes, pero lo puedes modificar en el archivo `memoria.tex`; concretamente, cambiando el campo `apalike` a otro en la instrucción `\bibliographystyle{apalike}`.

Y, para terminar este capítulo, resume brevemente qué vas a contar en los siguientes.

---

# **Capítulo 2**

# **Objetivos**

---

*Quizás algún fragmento de libro inspirador...*

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo lo ideal es explicar cuáles han sido los objetivos que te has fijado conseguir con tu trabajo, qué requisitos ha de respetar el resultado final, y cómo lo has llevado a cabo; esto es, cuál ha sido tu plan de trabajo.

## **2.1. Descripción del problema**

Cuenta aquí el objetivo u objetivos generales y, a continuación, concrétalos mediante objetivos específicos.

## **2.2. Requisitos**

Describe los requisitos que ha de cumplir tu trabajo.

## **2.3. Metodología**

Qué paradigma de desarrollo software has seguido para alcanzar tus objetivos.

## **2.4. Plan de trabajo**

Qué agenda has seguido. Si has ido manteniendo reuniones semanales, cumplimentando objetivos parciales, si has ido afinando poco a poco un producto final completo, etc.

---

## **Capítulo 3**

# **Plataforma de desarrollo**

---

*Quizás algún fragmento de libro inspirador...*

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo, explica qué has usado a nivel hardware y software para poder desarrollar tu trabajo: librerías, sistemas operativos, plataformas, entornos de desarrollo, etc.

---

# Capítulo 4

## Diseño

---

*Quizás algún fragmento de libro inspirador...*

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo (y quizás alguno más) es donde, por fin, describes detalladamente qué has hecho y qué experimentos has llevado a cabo para validar tus desarrollos.

### 4.1. Snippets

Puede resultar interesante, para clarificar la descripción, mostrar fragmentos de código (o *snippets*) ilustrativos. En el Código 4.1 vemos un ejemplo escrito en C++.

---

```
void Memory::hypothesizeParallelograms () {
    for(it1 = this->controller->segmentMemory.begin(); it1++) {
        squareFound = false; it2 = it1; it2++;
        while ((it2 != this->controller->segmentMemory.end()) && (!squareFound))
        {
            if (geometry::haveACommonVertex((*it1), (*it2), &square)) {
                dist1 = geometry::distanceBetweenPoints3D ((*it1).start, (*it1).end);
                dist2 = geometry::distanceBetweenPoints3D ((*it2).start, (*it2).end);
            }
        // [...]
    }
}
```

---

Código 4.1: Función para buscar elementos 3D en la imagen

En el Código 4.2 vemos un ejemplo escrito en Python.

### 4.2. Verbatim

Para mencionar identificadores usados en el código —como nombres de funciones o variables— en el texto, usa el entorno literal o verbatim

---

```

def mostrarValores():
    print (w1.get(), w2.get())

master = Tk()
w1 = Scale(master, from_=0, to=42)
w1.pack()
w2 = Scale(master, from_=0, to=200, orient=HORIZONTAL)
w2.pack()
Button(master, text='Show', command=mostrarValores).pack()

mainloop()

```

---

Código 4.2: Cómo usar un Slider

`hypothesizeParallelograms()`. También se puede usar este entorno para varias líneas, como se ve a continuación:

```

void Memory::hypothesizeParallelograms () {
    // add your code here
}

```

### 4.3. Ecuaciones

Si necesitas insertar alguna ecuación, puedes hacerlo. Al igual que las figuras, no te olvides de referenciarlas. A continuación se exponen algunas ecuaciones de ejemplo: Ecuación 4.1 y Ecuación 4.2.

$$H = 1 - \frac{\sum_{i=0}^N \left( \frac{d_{js} + d_{je}}{2} \right)}{M} \quad (4.1)$$

Ecuación 4.1: Ejemplo de ecuación con fracciones

$$v(\text{entrada}) = \begin{cases} 0 & \text{if } \epsilon_t < 0,1 \\ K_p \cdot (T_t - T) & \text{if } 0,1 \leq \epsilon_t < M_t \\ K_p \cdot M_t & \text{if } M_t < \epsilon_t \end{cases} \quad (4.2)$$

Ecuación 4.2: Ejemplo de ecuación con array y letras y símbolos especiales

### 4.4. Tablas o cuadros

Si necesitas insertar una tabla, hazlo dignamente usando las propias tablas de L<sup>A</sup>T<sub>E</sub>X, no usando pantallazos e insertándolas como figuras... En el Cuadro 4.1 vemos

un ejemplo.

Parámetros	Valores
Tipo de sensor	Sony IMX219PQ[7] CMOS 8-Mpx
Tamaño del sensor	3.674 x 2.760 mm (1/4" format)
Número de pixels	3280 x 2464 (active pixels)
Tamaño de pixel	1.12 x 1.12 um
Lente	f=3.04 mm, f/2.0
Ángulo de visión	62.2 x 48.8 degrees
Lente SLR equivalente	29 mm

Cuadro 4.1: Parámetros intrínsecos de la cámara

---

# **Capítulo 5**

# **Conclusiones**

---

*Quizás algún fragmento de libro inspirador...*

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo, que básicamente será una recapitulación de los problemas que has abordado, las soluciones que has prouesto, así como los experimentos llevados a cabo para validarlos. Y con esto, cierras la memoria.

## **5.1. Conclusiones**

Enumera los objetivos y cómo los has cumplido.

Enumera también los requisitos implícitos en la consecución de esos objetivos, y cómo se han satisfecho.

No olvides dedicar un par de párrafos para hacer un balance global de qué has conseguido, y por qué es un avance respecto a lo que tenías inicialmente. Haz mención expresa de alguna limitación o peculiaridad de tu sistema y por qué es así. Y también, qué has aprendido desarrollando este trabajo.

Por último, añade otro par de párrafos de líneas futuras; esto es, cómo se puede continuar tu trabajo para abarcar una solución más amplia, o qué otras ramas de la investigación podrían seguirse partiendo de este trabajo, o cómo se podría mejorar para conseguir una aplicación real de este desarrollo (si es que no se ha llegado a conseguir).

## 5.2. Corrector ortográfico

Una vez tengas todo, no olvides pasar el corrector ortográfico de L<sup>A</sup>T<sub>E</sub>Xa todos tus ficheros *.tex*. En Windows, el propio editor TeXworks incluye el corrector. En Linux, usa aspell ejecutando el siguiente comando en tu terminal:

```
aspell --lang=es --mode=tex check capitulo1.tex
```

# Bibliografía

---

- [Alami et al., 1998] Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1998). Multi-robot cooperation in the martha project. *IEEE Robotics Automation Magazine*, 5(1):36–47.
- [Arai and Parker, 2003] Arai, T. and Parker, L. (2003). Editorial: Advances in multi-robot systems.
- [Chaimowicz et al., 2001] Chaimowicz, L., Sugar, T., Kumar, V., and Campos, M. (2001). An architecture for tightly coupled multi-robot cooperation. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 3, pages 2992–2997 vol.3.
- [Fox et al., 2000] Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344.
- [Parker, 2003] Parker, L. E. (2003). Current research in multirobot systems. *Artificial Life and Robotics*, 7(1):1–5.
- [Sheng et al., 2006] Sheng, W., Yang, Q., Tan, J., and Xi, N. (2006). Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54(12):945–955.
- [Trawny et al., 2009] Trawny, N., Roumeliotis, S. I., and Giannakis, G. B. (2009). Cooperative multi-robot localization under communication constraints. In *2009 IEEE International Conference on Robotics and Automation*, pages 4394–4400.
- [Vega, 2008] Vega, J. (2008). Navegación y autolocalización de un robot guía de visitantes. Master thesis on computer science, Rey Juan Carlos University.
- [Vega, 2015] Vega, J. (2015). De la tiza al robot. Technical report.

- [Vega, 2018a] Vega, J. (2018a). *Educational framework using robots with vision for constructivist teaching Robotics to pre-university students.* Doctoral thesis on computer science and artificial intelligence, University of Alicante.
- [Vega, 2018b] Vega, J. (2018b). JdeRobot-Kids framework for teaching robotics and vision algorithms. In *II jornada de investigación doctoral.* University of Alicante.
- [Vega, 2019] Vega, J. (2019). El profesor Julio Vega, finalista del concurso 'Ciencia en Acción 2019'. URJC, on-line newspaper interview.
- [Vega and Cañas, 2019] Vega, J. and Cañas, J. (2019). PyBoKids: An innovative python-based educational framework using real and simulated Arduino robots. *Electronics*, 8:899–915.
- [Vega et al., 2012] Vega, J., Perdices, E., and Cañas, J. (2012). *Attentive visual memory for robot localization*, pages 408–438. IGI Global, USA. Text not available. This book is protected by copyright.
- [Verma and Ranga, 2021] Verma, J. K. and Ranga, V. (2021). Multi-robot coordination analysis, taxonomy, challenges and future scope. *Journal of Intelligent & Robotic Systems*, 102(1):10.