

Trabajo Fin de Grado  
Ingeniería en Electrónica, Robótica y  
Mecatrónica

Análisis de los sensores de un robot móvil con  
configuración diferencial

Autor: Guillermo Cartes Domínguez

Tutor: Federico Cuesta Rojo

Dpto. de Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2019





Trabajo Fin de Grado  
Ingeniería en Electrónica, Robótica y Mecatrónica

# **Análisis de los sensores de un robot móvil con configuración diferencial**

Autor:

Guillermo Cartes Domínguez

Tutor:

Federico Cuesta Rojo

Profesor titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019



Trabajo Fin de Grado: Análisis de los sensores de un robot móvil con configuración diferencial

Autor: Guillermo Cartes Domínguez

Tutor: Federico Cuesta Rojo

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

*A mi familia*

*A mis compañeros*

*A mis profesores*



# Resumen

---

La odometría es un factor imprescindible para la autonomía de un robot móvil. Para ello, es necesario el uso de sensores que proporcionen una información precisa y fiable que haga posible la localización del robot. En este documento se lleva a cabo un análisis de los sensores de Andruino, en especial de los sensores proporcionados por el teléfono inteligente con el que cuenta. Se estudia la calidad de la medida con plataformas que ayudan a realizar al robot movimientos tipo, así como la fabricación de dichas plataformas. Se analiza también el ruido añadido por los motores del robot.



# Abstract

---

Odometry is an essential factor for mobile robot's autonomy. To have odometry, it's necessary the use of sensors which provide a precise and reliable information that made possible robot's location. In this document, a sensory analysis of Andruino is carried out, specially from Andruino smartphone sensors. Platforms which help the robot to make known movements are used to study the quality of measure. The making of these platforms is also shown. The noise which motors cause is also studied.



<b>Resumen</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Índice</b>	<b>xiii</b>
<b>Índice de Tablas</b>	<b>xv</b>
<b>Índice de Figuras</b>	<b>xvi</b>
<b>Índice de Gráficas</b>	<b>xvii</b>
<b>Índice de Ilustraciones</b>	<b>xviii</b>
<b>1 Introducción</b>	<b>1</b>
1.1 <i>Descripción del problema</i>	1
1.2 <i>Objetivos</i>	2
1.3 <i>Plataformas de desarrollo</i>	2
1.3.1 ROS	2
1.3.2 Python	4
1.3.3 Arduino	4
1.3.4 Android	5
1.4 <i>Teléfonos inteligentes</i>	5
1.4.1 Smartphone 1	5
1.4.2 Smartphone 2	5
<b>2 Estado del Arte</b>	<b>7</b>
<b>3 Plataformas de Estudio</b>	<b>11</b>
3.1 <i>Plataforma para trayectoria en línea recta</i>	11
3.1.1 Plataforma sin motores	11
3.1.2 Plataforma con motores	13
3.2 <i>Plataforma para rotación</i>	15
3.2.1 Plataforma sin motores	15
3.2.2 Plataforma con motores	17
<b>4 Análisis en reposo</b>	<b>19</b>
4.1 <i>Smartphone 1</i>	19
4.1.1 Giroscopio sin calibrar y orientación con magnetómetro	19
4.1.2 Giroscopio calibrado y orientación sin magnetómetro	20
4.1.3 Ruido de los motores	21
4.2 <i>Smartphone 2</i>	23
4.2.1 Giroscopio sin calibrar y orientación con magnetómetro	23
4.2.2 Giroscopio calibrado y orientación sin magnetómetro	24
4.2.3 Ruido de los motores	25
4.3 <i>Conclusiones</i>	28
<b>5 Análisis en línea recta</b>	<b>31</b>
5.1 <i>Smartphone 1</i>	31

5.1.1	Orientación con magnetómetro	31
5.1.2	Orientación sin magnetómetro	32
5.2	<i>Smartphone 2</i>	33
5.2.1	Orientación con magnetómetro	33
5.2.2	Orientación sin magnetómetro	34
5.3	<i>Conclusiones</i>	35
<b>6</b>	<b>Análisis en rotación</b>	<b>37</b>
6.1	<i>Smartphone 1</i>	37
6.1.1	Orientación con magnetómetro	37
6.1.2	Orientación sin magnetómetro	38
6.2	<i>Smartphone 2</i>	39
6.2.1	Orientación con magnetómetro	39
6.2.2	Orientación sin magnetómetro	40
6.3	<i>Conclusiones</i>	41
<b>7</b>	<b>Conclusiones</b>	<b>43</b>
7.1	<i>Lista de conclusiones</i>	43
7.2	<i>Línea de trabajo futuro</i>	44
	<b>Referencias</b>	<b>45</b>
	<b>Índice de Conceptos</b>	<b>46</b>
	<b>Glosario</b>	<b>47</b>

# ÍNDICE DE TABLAS

---

Tabla 1. Comparativa de dispositivos	5
Tabla 2. Azimut con y sin magnetómetro. Test en reposo.	28
Tabla 3. Giroscopio con y sin compensación de deriva. Test en reposo.	29
Tabla 4. Ruido de los motores.	29
Tabla 5. Azimut con y sin magnetómetro. Test en línea recta.	35
Tabla 6. Azimut con y sin magnetómetro. Test de rotación.	42

# ÍNDICE DE FIGURAS

---

Figura 1. Error azimut.	2
Figura 2. Estructura de ROS. Recuperado de: <a href="https://answers.ros.org/question/12882/why-ros-topic-should-get-xmlrpc-addressport-but-ros-service-doesnt/">https://answers.ros.org/question/12882/why-ros-topic-should-get-xmlrpc-addressport-but-ros-service-doesnt/</a>	3

# ÍNDICE DE GRÁFICAS

---

Gráfica 1. Reposo con smartphone 1. Giroscopio sin calibrar y orientación con magnetómetro.	20
Gráfica 2. Reposo con smartphone 1. Giroscopio calibrado y orientación sin magnetómetro.	21
Gráfica 3. Reposo con smartphone 1. Funcionamiento de los motores. Azimut y velocidad angular.	22
Gráfica 4. Reposo con smartphone 1. Funcionamiento de los motores. IMU.	22
Gráfica 5. Reposo con smartphone 1. Funcionamiento de los motores. IMU. Detalle de ruido en el eje x.	23
Gráfica 6. Reposo con smartphone 2. Giroscopio sin calibrar. Azimut y velocidad angular.	24
Gráfica 7. Reposo con smartphone 2. Giroscopio calibrado. Azimut y velocidad angular.	25
Gráfica 8. Reposo con smartphone 2. Orientación con magnetómetro. Azimut y velocidad angular.	26
Gráfica 9. Reposo con smartphone 2. Orientación con magnetómetro. IMU.	26
Gráfica 10. Reposo con smartphone 2. Azimut y velocidad angular.	27
Gráfica 11. Reposo con smartphone 2. IMU.	28
Gráfica 12. Test en línea recta con smartphone 1. Orientación con magnetómetro.	32
Gráfica 13. Test en línea recta con smartphone 1. Orientación sin magnetómetro.	33
Gráfica 14. Test en línea recta con smartphone 2. Orientación con magnetómetro.	34
Gráfica 15. Test en línea recta con smartphone 2. Orientación sin magnetómetro.	35
Gráfica 16. Test de rotación con smartphone 1. Orientación con magnetómetro.	38
Gráfica 17. Test de rotación con smartphone 1. Orientación sin magnetómetro.	39
Gráfica 18. Test de rotación con smartphone 2. Orientación con magnetómetro.	40
Gráfica 19. Test de rotación con smartphone 2. Orientación sin magnetómetro.	41

# ÍNDICE DE ILUSTRACIONES

---

Ilustración 1. Comando rostopic list.	4
Ilustración 2. Smartphone 1 a la izquierda y smartphone 2 a la derecha.	6
Ilustración 3. Robot de Amazon Robotics. Recuperado de: <a href="https://www.amazon.es/b?ie=UTF8&amp;node=13495453031">https://www.amazon.es/b?ie=UTF8&amp;node=13495453031</a>	7
Ilustración 4. Testa Model S. Recuperado de: <a href="https://www.lespanol.com/omicron/20180611/tesla-recibiran-funciones-completamente-autonomas-proxima-actualizacion/314219842_0.html">https://www.lespanol.com/omicron/20180611/tesla-recibiran-funciones-completamente-autonomas-proxima-actualizacion/314219842_0.html</a>	8
Ilustración 5. Roomba serie i. Recuperado de: <a href="https://www.irobot.es/roomba/serie-i">https://www.irobot.es/roomba/serie-i</a>	8
Ilustración 6. Robot de Hubot. Recuperado de: <a href="https://hubot.cl/product-category/kits/kit-robotica-programables/">https://hubot.cl/product-category/kits/kit-robotica-programables/</a>	9
Ilustración 7. Kit usado en Andruino. Recuperado de: <a href="https://www.micro-log.com/384-general">https://www.micro-log.com/384-general</a>	9
Ilustración 8. Diseño de plataforma para test en línea recta.	12
Ilustración 9. Plataforma para línea recta.	12
Ilustración 10. Plataforma para línea recta junto con robot.	13
Ilustración 11. Diseño del soporte para plataforma con motores.	13
Ilustración 12. Soporte para plataforma con motores.	14
Ilustración 13. Test en línea recta completo.	14
Ilustración 14. Diseño de plataforma para test de rotación.	15
Ilustración 15. Plataforma para test de rotación.	16
Ilustración 16. Tornillo en eje de rotación.	16
Ilustración 17. Tornillo y varilla para limitar el ángulo de giro.	17

# 1 INTRODUCCIÓN

---

*“Entender el problema es el principio de la solución.”*

*- Guillermo Cartes -*

**A**ndruino es un proyecto realizado por Francisco López, cuyo nombre está compuesto por las palabras “Android” y “Arduino”, dos partes esenciales como se verá más adelante. Este proyecto utiliza un robot con configuración diferencial dotado de multitud de sensores, integrado en ROS<sup>1</sup>, una plataforma de código abierto que proporciona un gran abanico de posibilidades en la robótica móvil.

## 1.1 Descripción del problema

Entre los sensores de Andruino, se encuentran tres sensores de ultrasonidos HC-SR04, tres sensores fotorresistivos o LDR<sup>2</sup>, y un teléfono inteligente, que proporciona datos como el azimut, la velocidad angular, y la aceleración en los tres ejes gracias a una Unidad de Medida Inercial (IMU).

El valor del azimut es de especial importancia en odometría, ya que errores pequeños podrían acumularse y provocar grandes errores en el transcurso de la trayectoria. Es lo que muestra la Figura 1, un error en la medida del azimut inicial provoca una gran diferencia entre la trayectoria real descrita por el robot, y la obtenida mediante odometría. Por ello, se considera necesario caracterizar las medidas proporcionadas por el smartphone.

---

<sup>1</sup> Robot Operating System

<sup>2</sup> Light-Dependent Resistor

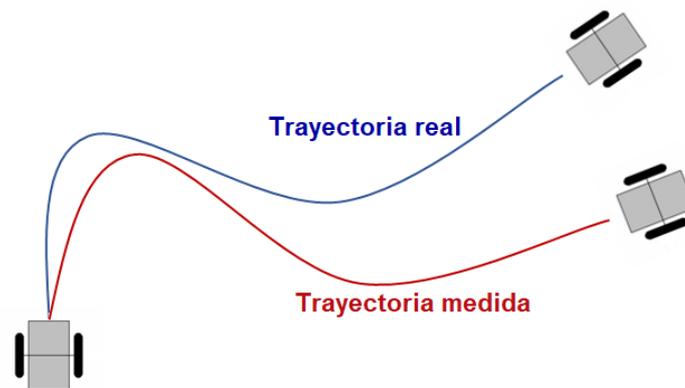


Figura 1. Error azimut.

## 1.2 Objetivos

En este documento se pretende analizar las medidas proporcionadas por el dispositivo android, especialmente el azimut ya que, como se ha comentado antes, es una medida crítica para la odometría.

Se estudiará la dependencia de las medidas de los sensores con el tipo de dispositivo. Evidentemente, dicho dispositivo debe ser android, pero hay un amplio catálogo de dispositivos con dicho sistema operativo con diferentes prestaciones, según sus sensores, nivel API, procesador o memoria RAM.

Por otro lado, es necesario conocer la repetibilidad de la medida y su comportamiento ante diferentes escenarios, obteniendo así un análisis que permita conocer mejor la medida que proporciona el dispositivo.

## 1.3 Plataformas de desarrollo

### 1.3.1 ROS

Robotic Operating System (ROS) es un framework para la programación de robots, gestionado por Open Robotics. ROS provee librerías y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots. Está bajo licencia de código libre, por lo que cualquier persona tiene acceso a dicho sistema. [1]

ROS se compone de: [2]

- Nodos: un nodo es un ejecutable que usa ROS para comunicarse con otros nodos. Para ello, necesita la librería de un cliente de ROS que le permite escribir en diferentes lenguajes de programación para comunicarse. En este documento, se usará *rospy*, que permite programar en python.
- Mensajes: estructura de datos de ROS que se usa al suscribirse o publicar en un topic. Es decir, son datos que se mandan a través de los topics y sirven de comunicación entre nodos. Están definidos como archivos .msg.
- Topics: canales para compartir datos entre diferentes nodos. Para ello, un nodo debe suscribirse a un topic para recibir un mensaje, o publicar un mensaje para enviarlo a través del topic.
- Master: nombre del servidor de ROS, es decir, el servidor que ayuda a los nodos a ubicarse entre sí, manejando las direcciones de los nodos. Cuando se establece la comunicación, esta se produce

entre pares: publishers/subscribers. El ROS Master debe ser el primero en ejecutarse, para lo que se usa el comando *roscore*, que carga el ROS Master junto con otros componentes esenciales.

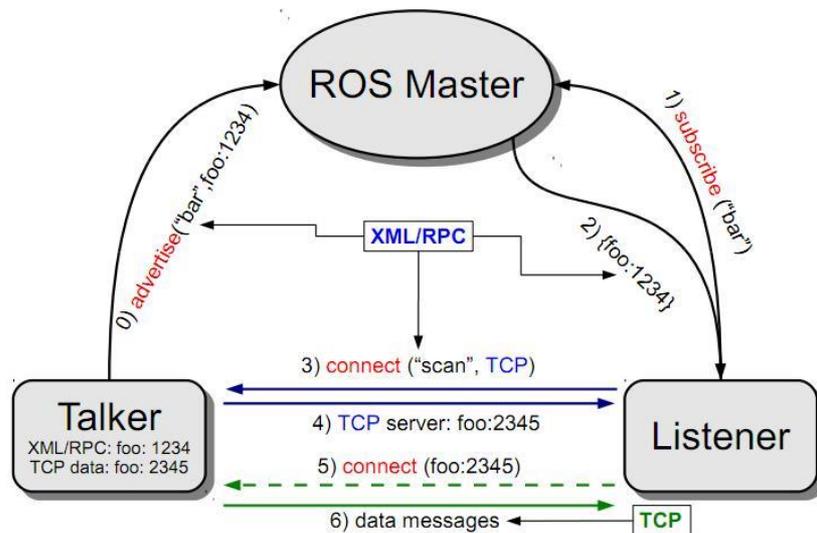


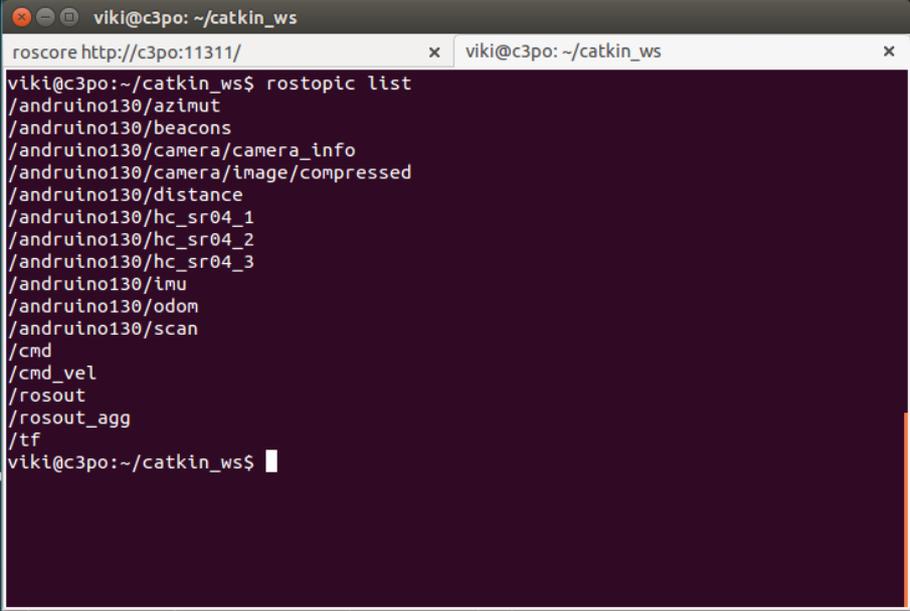
Figura 2. Estructura de ROS. Recuperado de: <https://answers.ros.org/question/12882/why-ros-topic-should-get-xmlrpc-addressport-but-ros-service-doesnt/>

En cuanto a Andruino, cuenta con una serie de topics cuya lista puede verse usando el comando *rostopic list* (Ilustración 1). Entre los topics, se pueden diferenciar los que reciben mensajes del robot, y los que reciben mensajes del usuario. Es decir, el robot mandará información a través de una serie de topics, y recibirá órdenes a través de otros. En el nombre de cada topic se encuentra la palabra andruino seguida de un número de 3 cifras, que corresponden a las 3 últimas cifras de la dirección IP del dispositivo. La lista se muestra a continuación, y las cifras correspondientes a la dirección IP se mostrarán como XXX, ya que estas dependerán del dispositivo que esté conectado:

- andruinoXXX/azimut: devuelve un mensaje del tipo `std_msgs/String` dando los valores del azimut y la velocidad angular estimados.
- andruinoXXX/camera/camera\_info: devuelve un mensaje de tipo `sensor_msgs/CameraInfo` con información de la imagen como sus dimensiones en píxeles.
- andruinoXXX/camera/image/compressed: devuelve un mensaje del tipo `sensor_msgs/CompressedImage`.
- andruinoXXX/distance: devuelve un mensaje del tipo `std_msgs/String` con los valores de los sensores de ultrasonidos.
- andruinoXXX/hc\_sr04\_Y: donde Y = 1, 2, 3, devuelve un mensaje de tipo `sensor_msgs/Range` con la distancia medida por el sensor de ultrasonidos correspondiente
- andruinoXXX/imu: devuelve un mensaje del tipo `sensor_msgs/Imu` con los valores de la Unidad de Medida Inercial.
- andruinoXXX/odom: devuelve un mensaje de tipo `nav_msgs/Odometry` dando las coordenadas x,y y la orientación estimados.
- andruinoXXX/scan: devuelve un mensaje del tipo `sensor_msgs/LaserScan` con la unión de los tres sensores de ultrasonidos.
- Cmd: recibe un mensaje de tipo `std_msgs/String` para configurar o controlar el robot. Es un comando general que afectará a todas las unidades de Andruino conectadas a un mismo maestro.

- `Cmd_vel`: recibe un mensaje de tipo `geometry_msgs/Twist` con el movimiento que debe tomar el robot.
- `tf`: devuelve la transformación entre los diferentes sistemas de referencia de un robot, conocida como  $TF^2$ , donde Andruino R2 sigue la recomendación 105<sup>3</sup> en la denominación de los sistemas de coordenadas.

Toda la información completa en [3].



```
viki@c3po: ~/catkin_ws
roscore http://c3po:11311/
viki@c3po:~/catkin_ws$ rostopic list
/andruino130/azimut
/andruino130/beacons
/andruino130/camera/camera_info
/andruino130/camera/image/compressed
/andruino130/distance
/andruino130/hc_sr04_1
/andruino130/hc_sr04_2
/andruino130/hc_sr04_3
/andruino130/imu
/andruino130/odom
/andruino130/scan
/cmd
/cmd_vel
/rosout
/rosout_agg
/tf
viki@c3po:~/catkin_ws$
```

Ilustración 1. Comando `rostopic list`.

### 1.3.2 Python

Python es un lenguaje de programación de código abierto, que es dinámico, interpretado y multiplataforma. Es administrado por la Python Software Foundation, y fue creado a finales de los ochenta por Guido van Rossum. [4]

Gracias a la librería `rospy` es posible trabajar con Python en ROS. Es un lenguaje sencillo que tiene una gran comunidad y multitud de librerías que permiten programar con facilidad. En el presente documento, se ha hecho uso de la documentación suministrada por la Python Software Foundation [5] para el desarrollo de los distintos nodos que han sido necesarios.

### 1.3.3 Arduino

Arduino es una plataforma de creación de electrónica de código abierto, basada en hardware y software libre, flexible y fácil de utilizar para creadores y desarrolladores [6].

El microprocesador utilizado en Andruino es una placa Arduino UNO. Esta placa se encarga de la lectura de los 3 ultrasonidos HC-SR04, las 3 fotorresistencias LDR y del funcionamiento de los motores. Para el transporte de datos, establece una conexión mediante un cable USB con Android, que luego se encargará de enviar los datos al ordenador junto con los demás datos que recoja.

### 1.3.4 Android

Android es un sistema operativo gestionado por Google que está presente en la gran mayoría de los dispositivos móviles, además de relojes, tablets, televisores o coches. Su presencia en la mayoría de los terminales dota a Andruino de una gran versatilidad para contar con un dispositivo móvil.

Andruino cuenta con una aplicación creada por Francisco López. Dicha aplicación se encarga de la comunicación con ROS y con Arduino. Para la comunicación con ROS, la aplicación publica los topics en la dirección IP que se le introduce por pantalla. Para la comunicación con Arduino, hay una conexión por cable USB. Además, la aplicación recoge los datos de los sensores del móvil como son el giróscopo, el acelerómetro y la cámara.

## 1.4 Teléfonos inteligentes

Todos los experimentos del estudio se llevarán a cabo con dos smartphones diferentes, con el propósito de analizar la dependencia de la medida con el tipo de dispositivo. Ambos terminales son de propiedad del autor, uno de uso habitual (Smartphone 2), y otro un teléfono ya en desuso, olvidado por casa (Smartphone 1).

### 1.4.1 Smartphone 1

Se trata de un BQ Aquaris M5, un dispositivo de febrero de 2015 que cuenta con un procesador Qualcomm Snapdragon 615, 2GB de RAM y la versión 7.1.2 de Android, conocida como Android Nougat. Dada sus especificaciones, se considera un terminal de gama media.

### 1.4.2 Smartphone 2

Se trata de un Xiaomi Pocophone F1, un dispositivo de septiembre de 2018, que cuenta con un procesador Qualcomm Snapdragon 845, 6GB de RAM y la versión 9 de Android, conocida como Android Pie. Sus altas prestaciones hacen considerarlo un terminal de gama alta. Además, aún no ha sufrido la pronta obsolescencia que achaca a los smartphones.

	Smartphone 1	Smartphone 2
<b>Versión de Android</b>	7.1.2	9
<b>Nivel API</b>	25	28
<b>Procesador</b>	Snapdragon 615 (1.46GHz)	Snapdragon 845 (2.8GHz)
<b>Memoria RAM</b>	2GB	6GB
<b>Giroscopio</b>	BMI160	BMI160
<b>Acelerómetro</b>	BMI160	BMI160
<b>Magnetómetro</b>	Akm09911	Ak0991x

Tabla 1. Comparativa de dispositivos

Se puede observar que ambos dispositivos cuentan con el mismo acelerómetro y giroscopio (BMI 160). Si cambia el magnetómetro, que es un Akm09911 para el BQ y un Ak0991x para el Xiaomi. Por tanto, las principales diferencias entre ambos residen en el procesador, que otorga al segundo una velocidad de cómputo mayor, la versión del software, y el sensor del magnetómetro.



Ilustración 2. Smartphone 1 a la izquierda y smartphone 2 a la derecha.

## 2 ESTADO DEL ARTE

---

*“La educación es el pasaporte hacia el futuro, el mañana pertenece a aquellos que se preparan para él en el día de hoy.”*

*- Malcom X -*

La robótica es una disciplina en pleno desarrollo cuyo origen no es muy lejano. Aunque algunos inventos que pueden ser considerados como los primeros robots datan de hace más de un milenio, la palabra «robot» aparece por primera vez en 1920 cuando el checo Karel Capek escribe *Rossum's Universal Robots*, una obra teatral que trata sobre la creación de hombres artificiales que aligeran la carga de trabajo del resto de las personas [7]. De hecho, la palabra «robot» tiene su origen en la palabra checa «robota», que significa servidumbre. Fue entonces cuando aparece la palabra robot y empieza a usarse para hablar de dicha disciplina, mucho más amplia de los robots humanoides de los que trata la obra.

Dentro de la robótica se pueden diferenciar varias disciplinas, entre las que destacan la robótica de manipulación y la robótica móvil. Es la segunda en la que se encuadra el presente documento. Desde Shakey, el primer robot móvil, desarrollado entre 1966 y 1972, dicho sector ha sufrido un gran desarrollo. Actualmente los robots móviles se encuentran en multitud de ámbitos, algunos de los cuales se desarrollarán a continuación.

En la industria, destaca Amazon Robotics. En los almacenes de dicha compañía, los robots se deslizan por debajo de las torres de estanterías en las que se almacenan los productos y la levantan y desplazan por todo el centro logístico. Estos robots pesan 150kg, soportan una carga de más de 1300kg y trabajan a una velocidad de hasta 1,7 metros por segundo. Entre las ventajas de este entorno de trabajo, destaca el aumento de la capacidad de almacenamiento en hasta un 50%, y la velocidad de almacenamiento. [8]



Ilustración 3. Robot de Amazon Robotics. Recuperado de: <https://www.amazon.es/b?ie=UTF8&node=13495453031>

En el sector de la automoción se está produciendo un auge en los últimos años. Gran parte de culpa tiene la empresa Tesla, la cual se encuentra a la cabeza en cuanto a coches autónomos comerciales se refiere. Sus coches constan de multitud de sensores y un software denominado Autopilot, que le permiten circular de forma autónoma. No obstante, por razones legislativas y de seguridad aún requiere la atención humana, de manera que, si el vehículo detecta que el piloto no presta atención durante unos segundos, la conducción autónoma se desactiva. Estos coches constan de un nivel 3 de automatización [9]. Según el estándar SAE J3016, realizado por la Sociedad de Ingenieros Automotrices (SAE<sup>3</sup> por sus siglas en inglés), existen 6 niveles de conducción autónoma que van desde ninguna automatización en el nivel 0, hasta automatización total de la conducción en el nivel 5 [10]. Actualmente, sólo el prototipo de coche autónomo de Google podría considerarse como perteneciente al nivel 5 de autonomía.



Ilustración 4. Testa Model S. Recuperado de: [https://www.elespanol.com/omicron/20180611/tesla-recibiran-funciones-completamente-autonomas-proxima-actualizacion/314219842\\_0.html](https://www.elespanol.com/omicron/20180611/tesla-recibiran-funciones-completamente-autonomas-proxima-actualizacion/314219842_0.html)

En la domótica también se encuentran los robots móviles. Destaca iRobot, con sus robots aspiradoras Roomba. Estos dispositivos son aspiradoras móviles con inteligencia artificial capaces de generar un mapa del entorno gracias a un sensor óptico, de forma que el robot puede saber dónde está, dónde ha estado, y qué le queda por limpiar, pudiendo programar así rutas para su objetivo. Además, varios dispositivos pueden comunicarse entre sí y realizar tareas de forma cooperativa. [11]



Ilustración 5. Roomba serie i. Recuperado de: <https://www.irobot.es/roomba/serie-i>

<sup>3</sup> SAE: Society of Automotive Engineers

Por otro lado, dentro de la robótica educativa, se encuentran multitud de robots diseñados para su uso en investigación y docencia. Se tratan de robots programables, fáciles de montar, económicos y muy flexibles. En el mercado se encuentran tanto robots móviles como robots manipuladores, aunque al ser equipos económicos y, por tanto, de menores prestaciones, el alcance funcional de los robots manipuladores de este estilo es muy limitado. Según su diseño, cuentan con diferentes actuadores y sensores, además de permitir una amplia integración de componentes. Respecto a las configuraciones de estos robots, en su mayoría son robots diferenciales, dada su facilidad de implementación y diseño.

Hay un amplio catálogo en el mercado. Por un lado, hay kits muy completos como es el caso de los robots de Hubot, que cuentan con numerosos sensores, como ultrasonidos o sensores IR; y actuadores, donde además de las ruedas motrices, tienen servomotores para mover sensores o crear pinzas para manipulación. También disponen de código libre para que, una vez montado el robot, realice aquellas tareas para las que está diseñado como seguimiento de líneas o evitación de obstáculos [12].

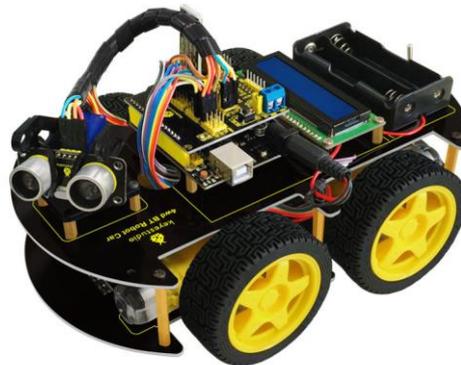


Ilustración 6. Robot de Hubot. Recuperado de: <https://hubot.cl/product-category/kits/kit-robotica-programables/>

Por otro lado, hay kits más simples que sólo cuentan con el chasis, los motores y las ruedas, pero permiten así una mayor expansión y alcance del proyecto. En Andruino, se usa un kit cuyo chasis tiene multitud de ranuras para incorporar sensores, placas o cables. Esta estructura aporta una gran flexibilidad de diseño, lo que supone un elemento clave para Andruino. Además, facilita la adición de nuevos elementos estructurales, una opción que será importante en el estudio del comportamiento de los sensores en el presente documento.



Ilustración 7. Kit usado en Andruino. Recuperado de: <https://www.micro-log.com/384-general>



# 3 PLATAFORMAS DE ESTUDIO

---

*“Si le das pescado a un hombre hambriento, le nutres una jornada. Si le enseñas a pescar, le nutrirás toda la vida.”*

*- Lao-tsé -*

Se pretende estudiar el comportamiento de las medidas de los sensores ante ciertos movimientos conocidos. Para realizar dichos movimientos, es necesario el uso de una plataforma para que los movimientos realizados estén acotados y no varíen de una prueba a otra. Se tendrán en cuenta dos escenarios diferentes: una trayectoria en línea recta y un giro sobre el eje del robot de un ángulo determinado. Además, cada test se diseñará en un primer momento sin considerar el funcionamiento de los motores, y luego con éstos en marcha. La fabricación de cada una de las plataformas se detallará a continuación:

## 3.1 Plataforma para trayectoria en línea recta

### 3.1.1 Plataforma sin motores

Dado que el objetivo es que el robot siga una línea recta, la plataforma deberá diseñarse atendiendo a dos factores: transportar al robot y realizar el movimiento rectilíneo. Es evidente que hay múltiples formas de realizar una plataforma que cumpla con lo acometido, pero la elegida atenderá a la mayor simplicidad y robustez posible. La plataforma podrá ser guiada por estructuras comunes con partes rectas como puede ser el caso de una pared o una mesa. Por tanto, el diseño será el siguiente.

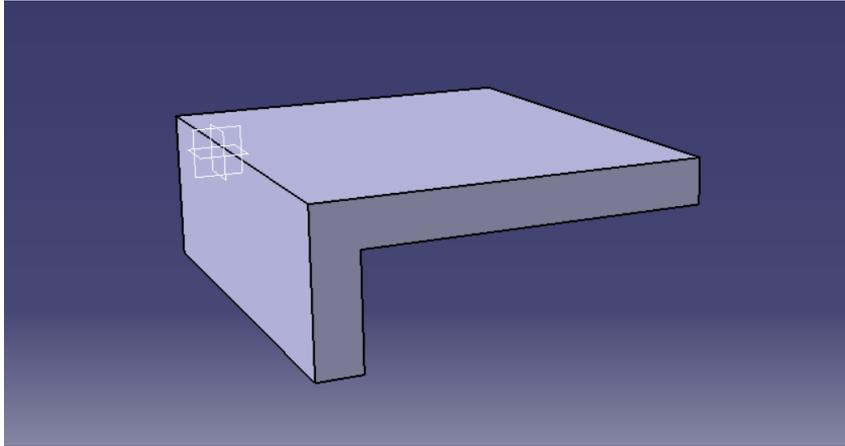


Ilustración 8. Diseño de plataforma para test en línea recta.

El material de construcción adecuado podría ser plástico o madera, materiales ligeros para realizar el estudio con facilidad y resistentes. La madera es más sencilla de trabajar, y las herramientas necesarias son muy comunes y fáciles de encontrar. Pero la sencillez del diseño hace posible encontrar alguna pieza que se pueda reciclar y ahorrar los costes y el tiempo de fabricación. En este caso, se encontró una pieza de metacrilato que cumplía perfectamente con los objetivos. Por tanto, la plataforma es la siguiente:



Ilustración 9. Plataforma para línea recta.

Que junto con el robot y la superficie guía (la mesa) sería:

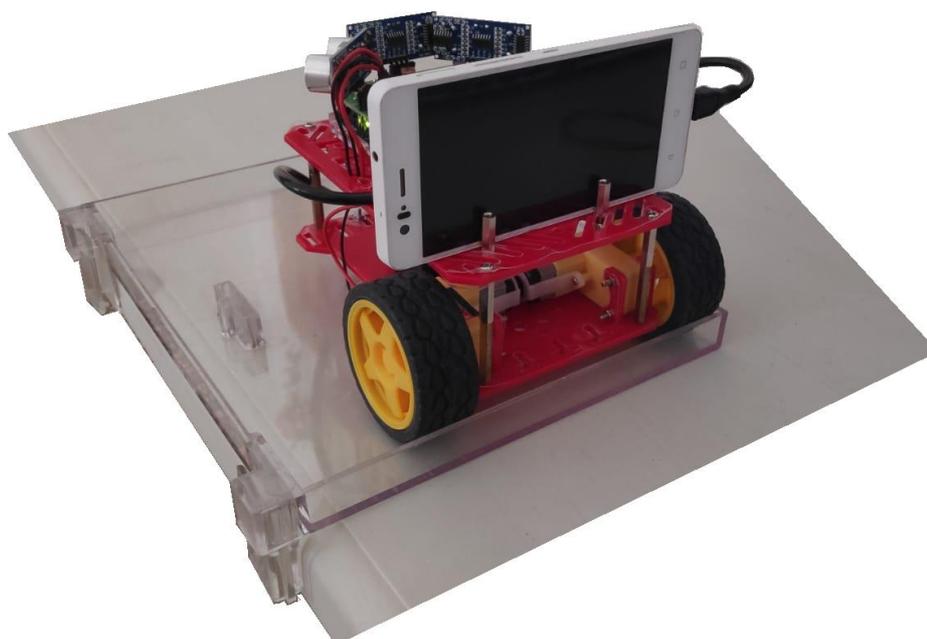


Ilustración 10. Plataforma para línea recta junto con robot.

### 3.1.2 Plataforma con motores

Para que los motores del robot entren en funcionamiento y se estudie el comportamiento de los sensores en condiciones reales de funcionamiento, será necesario elevar el robot en la misma plataforma hasta que las ruedas puedan girar sin contacto con el suelo. Si las ruedas tocasen el suelo, el robot saldría de la plataforma de estudio y dejaría de seguir la línea recta a la que se quiere someter a los sensores. Por tanto, se diseña una plataforma que se añada a la ya existente y cumpla con lo citado anteriormente. Así, el robot se someterá exactamente al mismo test, con la única diferencia de la puesta en marcha de los motores, pudiendo evaluar el efecto de éstos en la medida de los sensores.

El diseño se realiza de forma que el nuevo soporte se ajuste a la plataforma anterior, sujetando el robot para que no se desplace y permitiendo el libre movimiento de las ruedas. Se obtiene el siguiente modelo:

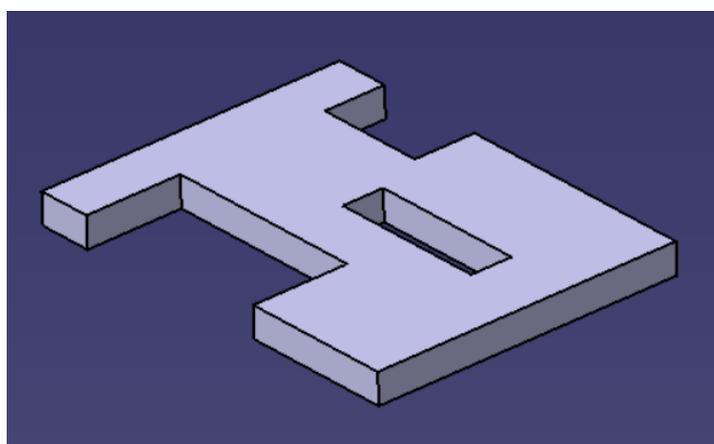


Ilustración 11. Diseño del soporte para plataforma con motores.

Se puede observar que el diseño de la plataforma deja espacio libre para las dos ruedas motrices y para la rueda loca, por lo que el robot se sostiene mediante el chasis. Uno de los espacios libres para las ruedas se aumenta para que encaje bien en la plataforma de metacrilato anterior.

Para su fabricación se utiliza un tablero de aglomerado de 2 cm de espesor, un material fácil de modelar y muy resistente. Además del diseño anterior, se le añade un tornillo que encaja con el chasis y proporciona mayor sujeción al robot para la realización del test. El resultado es el siguiente:

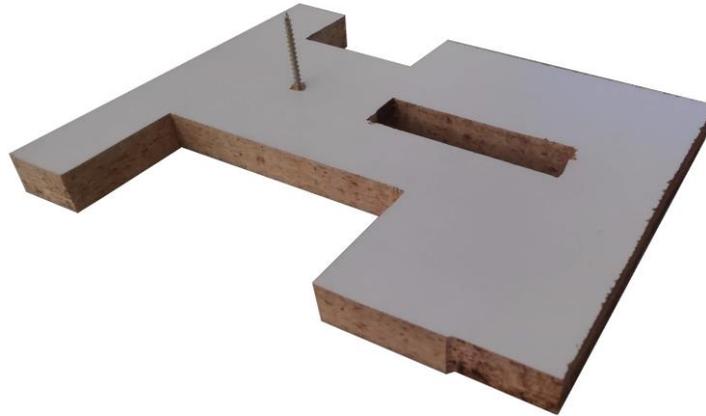


Ilustración 12. Soporte para plataforma con motores.

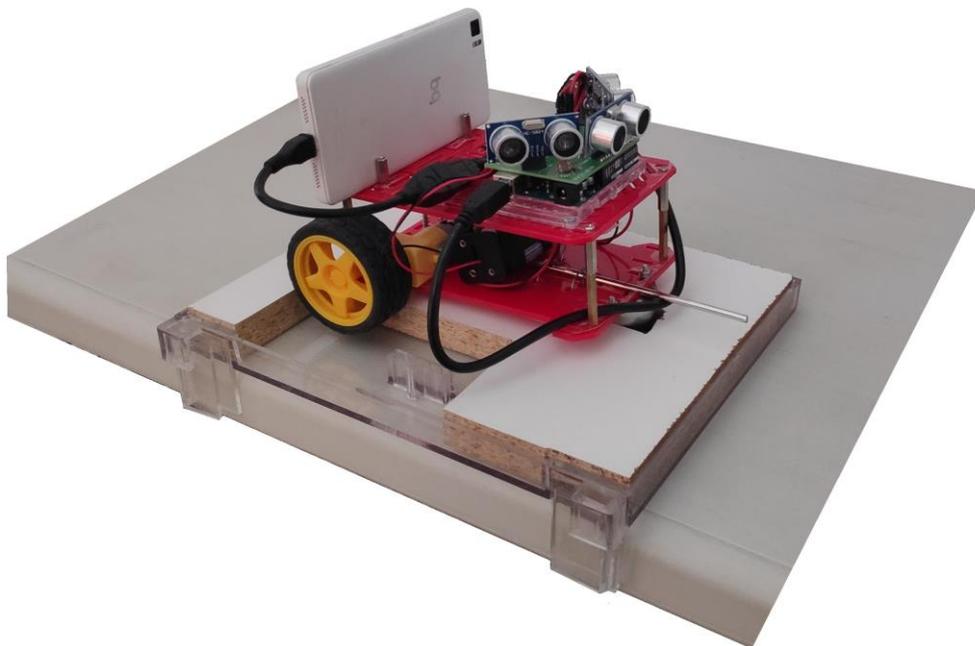


Ilustración 13. Test en línea recta completo.

## 3.2 Plataforma para rotación

### 3.2.1 Plataforma sin motores

En este caso, el robot debe realizar un giro determinado sobre su centro de rotación, que sería el centro del eje de rotación de sus ruedas. Para ello, la plataforma deberá realizar el giro en torno a dicho centro y solo permitirlo en un ángulo que se fija a  $90^\circ$ . Hay dos opciones: realizar una plataforma activa, es decir, que realice ella el giro y soporte el robot durante el mismo; o una *plataforma pasiva*, es decir, que permita al robot el movimiento deseado y restrinja el resto. El modelo anterior para el movimiento en línea recta sería una plataforma activa, ya que es la plataforma la que sostiene y guía al robot.

En este caso es interesante utilizar el chasis del robot, que consta de multitud de ranuras. Una de ellas se sitúa en el centro de rotación, por lo que es útil para realizar el movimiento de rotación. Por otro lado, es necesario limitar el giro, para lo que se usará una varilla situada en el eje longitudinal del robot. De esta forma, podría utilizarse una base plana a la que se añadirán 3 tornillos: uno para hacer de centro de rotación y otros dos para limitar el movimiento junto con la varilla. El diseño es el siguiente:

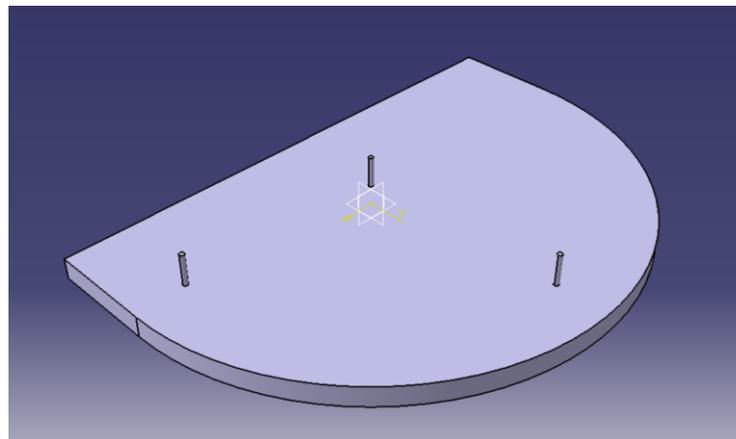


Ilustración 14. Diseño de plataforma para test de rotación.

A diferencia de la primera plataforma, este diseño no es fácil de encontrar, por lo que sí es necesaria la etapa de fabricación. Para la base se usará una tabla de madera de 2 cm de espesor y se le enroscarán los 3 tornillos, de 4 mm de diámetro. Hay que tener en cuenta tanto el diámetro de los tornillos como el de la varilla (todos de 4 mm) para que el robot realice el giro deseado de  $90^\circ$ . Cabe destacar que podría cambiarse el valor del ángulo simplemente cambiando uno de los tornillos que limitan la varilla.



Ilustración 15. Plataforma para test de rotación.

A pesar de haber tenido en cuenta los diámetros de los objetos, es necesario comprobar el giro que realizará el robot, ya que es muy posible que se haya tenido algún error de fabricación. Apenas se desplace el tornillo algún milímetro al ser enroscado, el ángulo real variará. De hecho, se comprueba que el giro real es de  $90.9^\circ$ , y no de  $90^\circ$  como se estimó. Este error no tiene importancia una vez conocido, ya que el objetivo del test es realizar un giro conocido. A continuación, se mostrará con más detalle la plataforma:

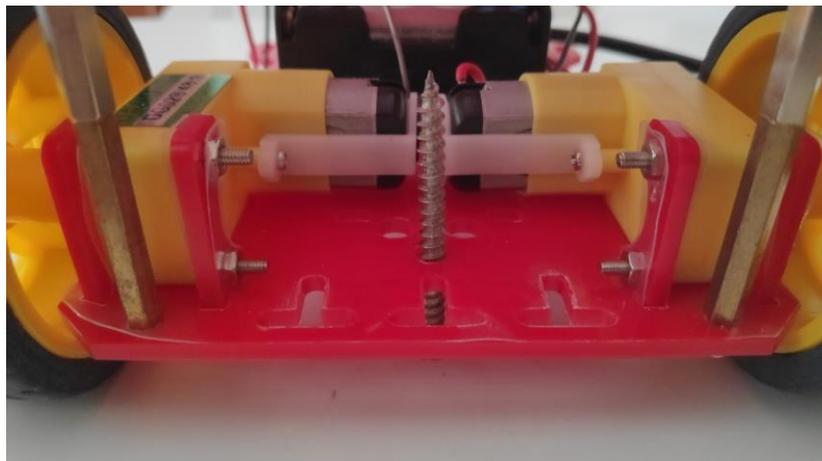


Ilustración 16. Tornillo en eje de rotación.



Ilustración 17. Tornillo y varilla para limitar el ángulo de giro.

### 3.2.2 Plataforma con motores

En este caso, la inclusión de los motores en el test es más sencilla que en el test de línea recta. A diferencia del primero, en este test se pueden usar los motores como fuerza motriz, ya que el movimiento está acotado y los motores no podrían provocar otro movimiento que no sea el previsto. Por tanto, la plataforma no cambia, y la única diferencia será a nivel de software, en el que además de recibir los datos de los sensores, se le ordenará al robot que cambie su velocidad angular cuando corresponda.



# 4 ANÁLISIS EN REPOSO

---

*“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”*

*- Albert Einstein -*

**E**n primer lugar, se estudiará el comportamiento de los sensores mientras el robot se encuentra en reposo. Se analizará las diferencias entre los sensores existentes para calcular el azimut y la velocidad angular. Además, se verá cómo afecta la puesta en marcha de los motores a la medida de los sensores.

El test en reposo permite evaluar fácilmente la deriva de los sensores, ya que idealmente deberían proporcionar una medida estable.

Para analizar la influencia de los motores en los sensores, se someterá el robot a la siguiente prueba. El robot permanecerá en reposo a lo largo de la prueba, por lo que será necesario dejar las ruedas suspendidas, ya que estas se moverán durante la prueba. Los motores comenzarán apagados. Pasados dos segundos se encenderán y, pasados cinco, volverán a apagarse. De esta forma, se obtendrá el ruido introducido por los motores.

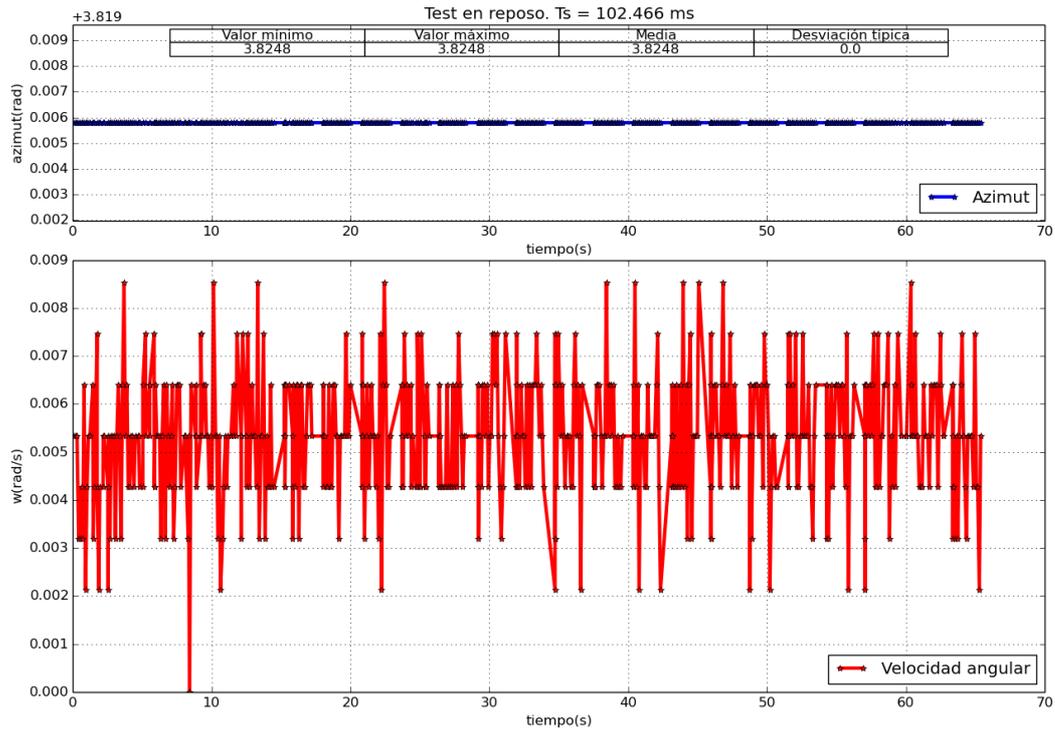
Se estudiará con los dos dispositivos, smartphone 1 y smartphone 2, procedimiento que se repetirá en todos los análisis posteriores.

## 4.1 Smartphone 1

### 4.1.1 Giroscopio sin calibrar y orientación con magnetómetro

En primer lugar, se obtiene el azimut usando el magnetómetro y la velocidad angular con el giroscopio sin calibrar. Dichos sensores están definidos como `TYPE_ROTATION_VECTOR` y `TYPE_GYROSCOPE_UNCALIBRATED`, respectivamente. El primero proporciona el azimut usando el acelerómetro, el giroscopio y el magnetómetro, y el segundo estima la velocidad angular sin la compensación de la deriva [13].

En la Gráfica 1, se observa que la velocidad angular no es nula a pesar de estar parado el robot. De hecho, siempre es positiva, con una media de 0.00532 rad/s. Si la orientación del dispositivo se obtuviese integrando esta velocidad, tendría una deriva con pendiente positiva, que se haría cada vez más notable con el paso del tiempo. En cambio, se aprecia que el azimut, obtenido con `TYPE_ROTATION_VECTOR`, tiene un valor constante, con una desviación típica nula. Durante la prueba se ha sometido al dispositivo a diferentes campos magnéticos sin obtener ninguna variación en el azimut.

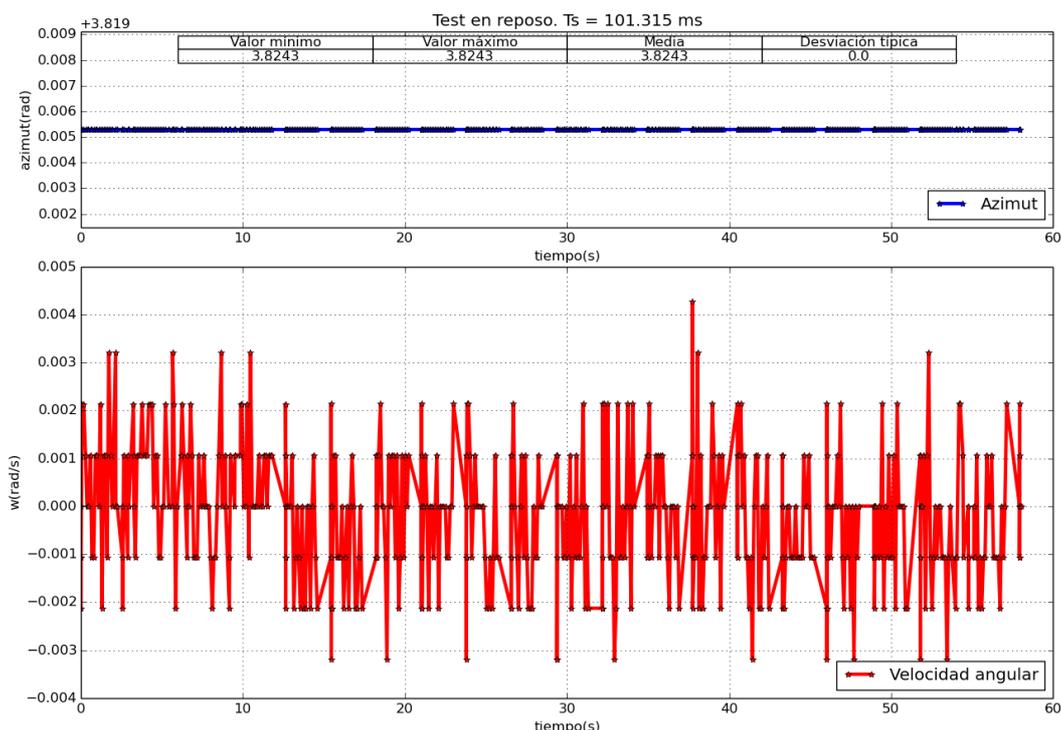


Gráfica 1. Reposo con smartphone 1. Giroscopio sin calibrar y orientación con magnetómetro.

#### 4.1.2 Giroscopio calibrado y orientación sin magnetómetro

En este caso, se obtiene el azimut sin usar el magnetómetro y la velocidad angular con el giroscopio calibrado, definidos como `TYPE_GAME_ROTATION_VECTOR` y `TYPE_GYROSCOPE`, respectivamente. El primero proporciona el azimut usando el acelerómetro y el giroscopio, y el segundo usa el giroscopio con una compensación de deriva [13].

Se puede observar en la Gráfica 2 que en este caso la velocidad angular tiene una media de  $-0.00018$  rad/s, por lo que la deriva disminuye en un orden de magnitud. Aunque la deriva haya disminuido considerablemente, aún sigue existiendo, mientras que el azimut obtenido con `TYPE_ROTATION_VECTOR`, al igual que en la prueba anterior, se mantiene constante con desviación típica nula. Por tanto, se aprecia una clara mejora en la deriva del giroscopio gracias a la compensación, mientras que no se aprecian diferencias entre el uso o no del magnetómetro para obtener el azimut.



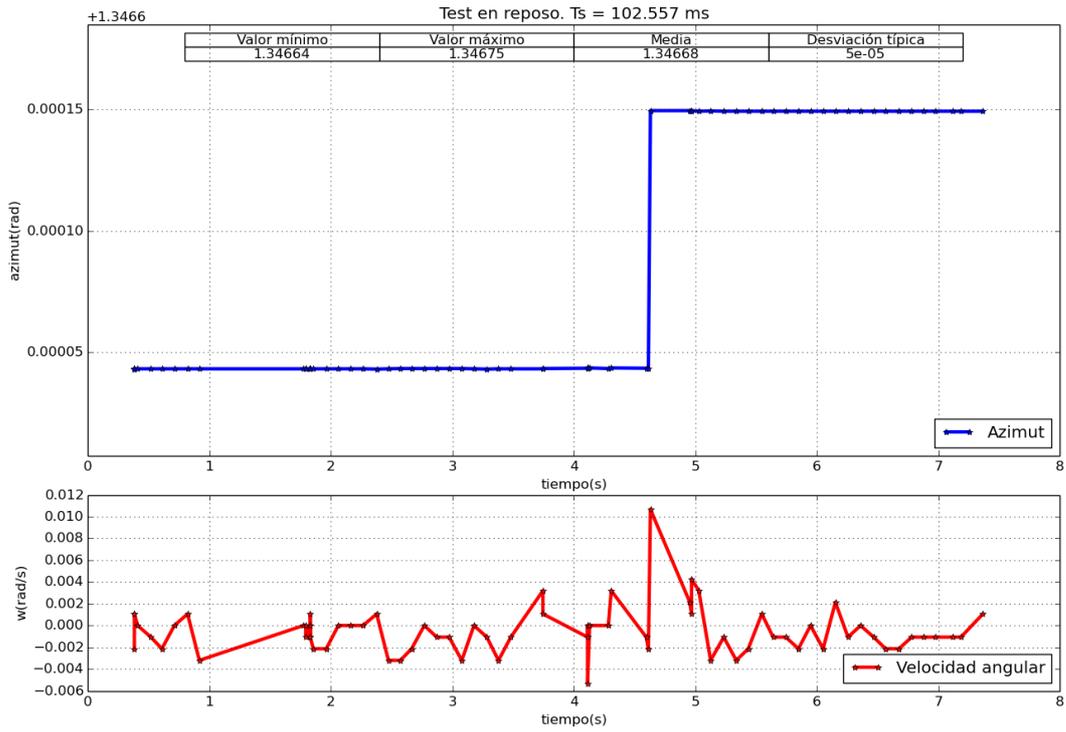
Gráfica 2. Reposo con smartphone 1. Giroscopio calibrado y orientación sin magnetómetro.

#### 4.1.3 Ruido de los motores

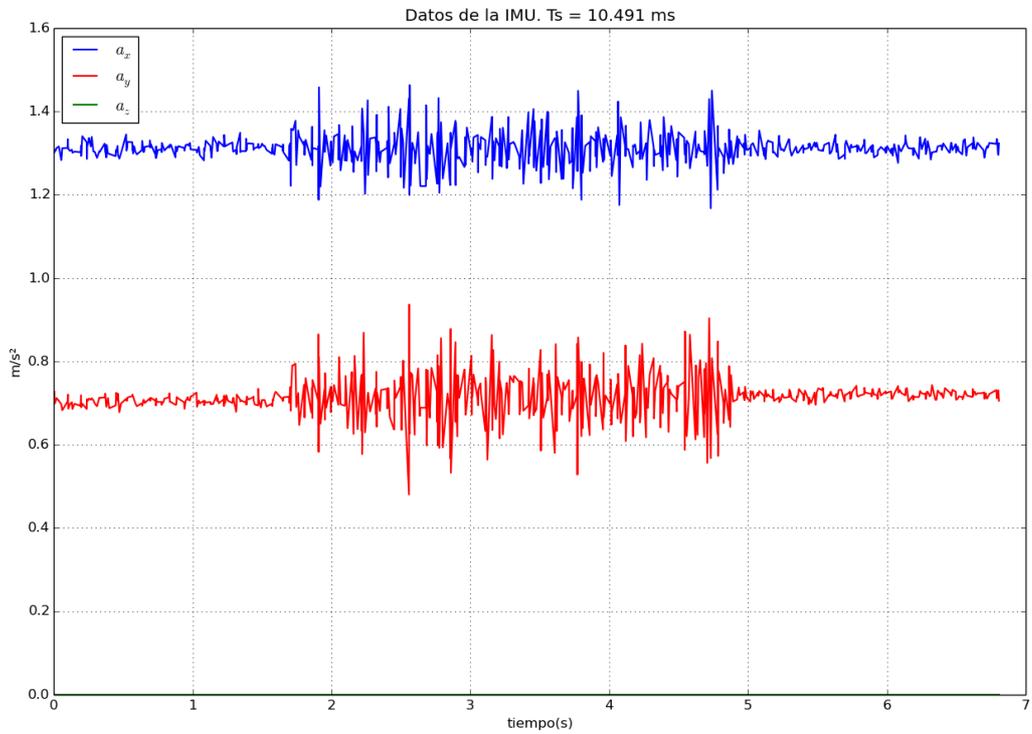
En este apartado se estudiará el comportamiento de los sensores con el funcionamiento de los motores. En la Gráfica 3 se puede ver el azimut y la velocidad angular obtenidos con el uso del magnetómetro y con la compensación de deriva, respectivamente. Las medidas cambian levemente con la puesta en marcha de los motores. A los 4.7 segundos, el azimut sufre una subida, pero esta apenas es de una diezmilésima, cambio que puede ser debido a un pequeño movimiento del mismo smartphone.

Con el giróscopo sin compensación, la velocidad angular tendría una deriva mayor, por lo que se ha optado por obtenerla directamente con la compensación. En cambio, sí es interesante ver las posibles diferencias entre usar el magnetómetro o no para la estimación del azimut, ya que los motores provocan un campo eléctrico que podría afectar a la medida. En cambio, se comprueba que no afecta, ya que se obtienen los mismos resultados independientemente del uso o no del magnetómetro. Esto ratifica lo ya visto en el apartado 4.1.1, donde la introducción de campos magnéticos en el entorno del dispositivo no afectaba a la medida del azimut.

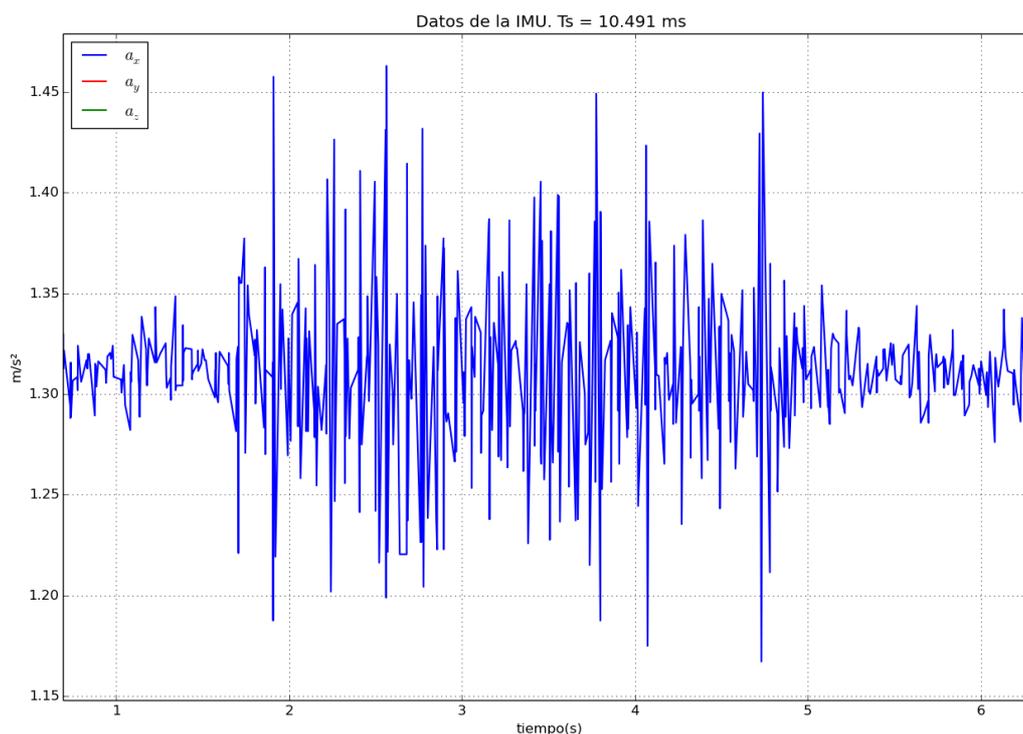
En la Gráfica 4 se observa los datos de la Unidad de Medida Inercial. Puede observarse que entre los segundos 2 y 5 existe un ruido superior al resto de la prueba. Esto coincide con el tiempo en el que los motores están en funcionamiento. Por tanto, es dicho ruido el provocado por la puesta en marcha de los motores. Se puede ver el detalle de la amplitud del ruido en la Gráfica 5, donde se aprecia la subida de la amplitud pico a pico del ruido, desde los  $0.05 \text{ m/s}^2$  hasta los  $0.25 \text{ m/s}^2$ , es decir, un valor 5 veces superior.



Gráfica 3. Reposo con smartphone 1. Funcionamiento de los motores. Azimut y velocidad angular.



Gráfica 4. Reposo con smartphone 1. Funcionamiento de los motores. IMU.



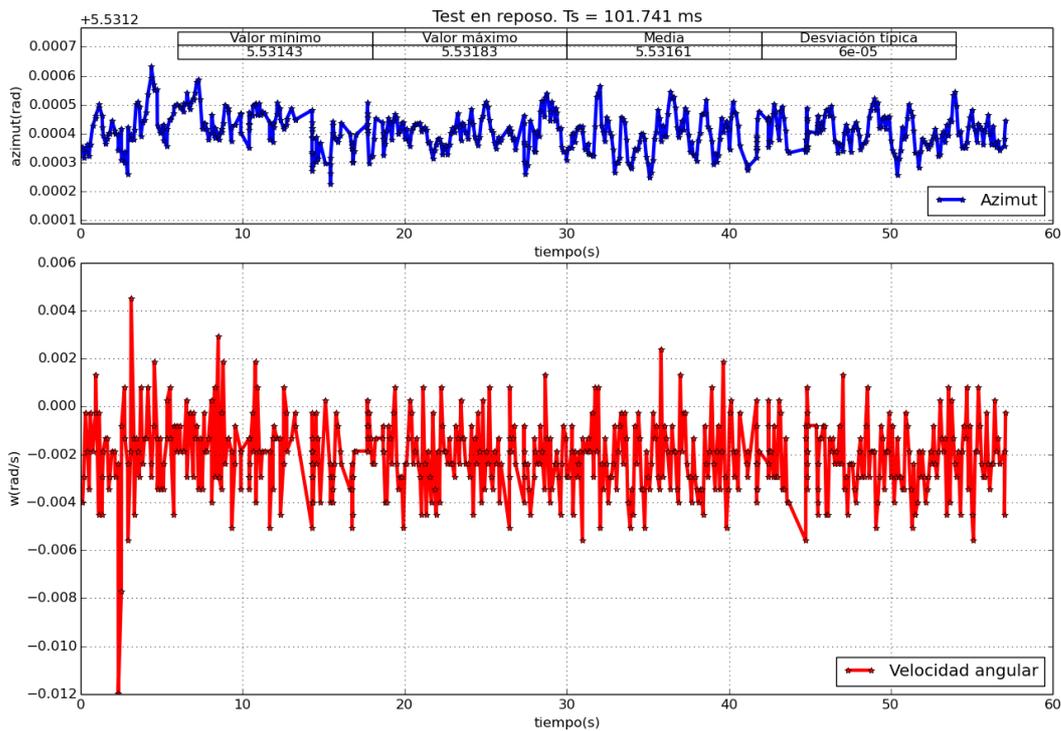
Gráfica 5. Reposo con smartphone 1. Funcionamiento de los motores. IMU. Detalle de ruido en el eje x.

## 4.2 Smartphone 2

### 4.2.1 Giroscopio sin calibrar y orientación con magnetómetro

De la misma forma que se realizó para el smartphone 1, se obtendrán los resultados para este smartphone con el azimut usando el magnetómetro, y la velocidad angular con el giroscopio sin la compensación de deriva. En la Gráfica 6 se puede observar que la velocidad angular tiene una media no nula, pero en este caso la media es negativa. Esto significa que, en caso de obtener la orientación mediante la integración de la velocidad angular, ésta tendría una deriva con pendiente negativa, a diferencia de la pendiente positiva que obtenía el smartphone 1. Esta deriva depende del sensor, por lo que a priori se sabía que no iban a coincidir.

Por otro lado, se observa que el azimut en este caso presenta una desviación típica de  $6 \cdot 10^{-5}$ , por lo que deja de ser constante. No obstante, la desviación típica tiene un valor muy bajo.

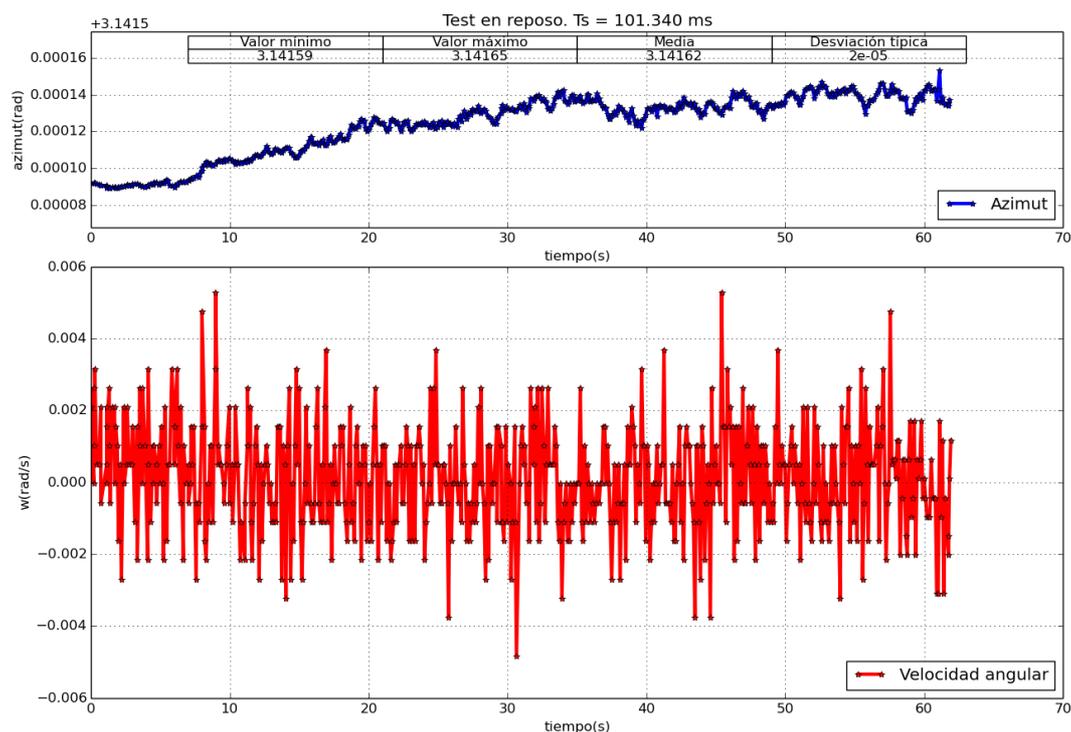


Gráfica 6. Reposo con smartphone 2. Giroscopio sin calibrar. Azimut y velocidad angular.

#### 4.2.2 Giroscopio calibrado y orientación sin magnetómetro

En este caso, se obtendrá el azimut sin usar el magnetómetro y la velocidad angular con la compensación de deriva del giroscopio. En la Gráfica 7 se observa que en este caso la media de la velocidad angular ha disminuido hasta 0.00014 rad/s. Al igual que en el caso del smartphone 1, el giróscopo calibrado disminuye la deriva en un orden de magnitud.

Como ya ocurría en el apartado anterior, el azimut no presenta una desviación típica nula, siendo ésta de  $2 \cdot 10^{-5}$ . Además, en este caso, presenta una deriva con pendiente ascendente, causada por la media positiva que presentaba la velocidad angular. Por tanto, aunque la desviación típica haya disminuido, ya que la señal es menos ruidosa sin usar el magnetómetro, ésta presenta una deriva que daría lugar a errores a largo plazo.

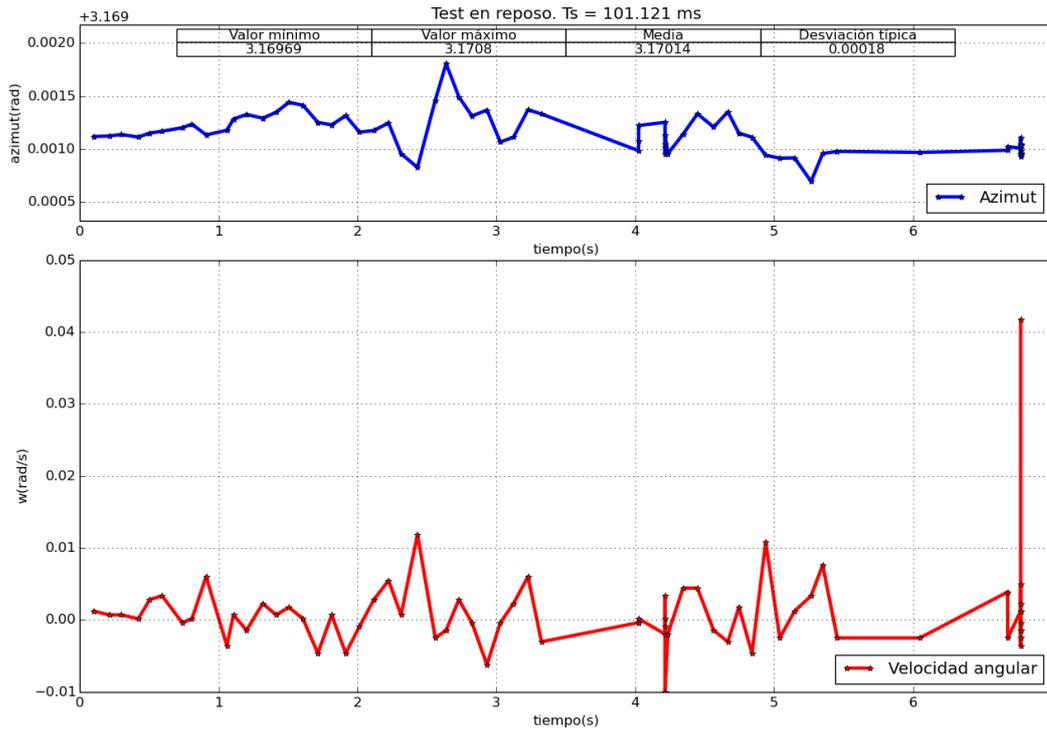


Gráfica 7. Reposo con smartphone 2. Giroscopio calibrado. Azimut y velocidad angular.

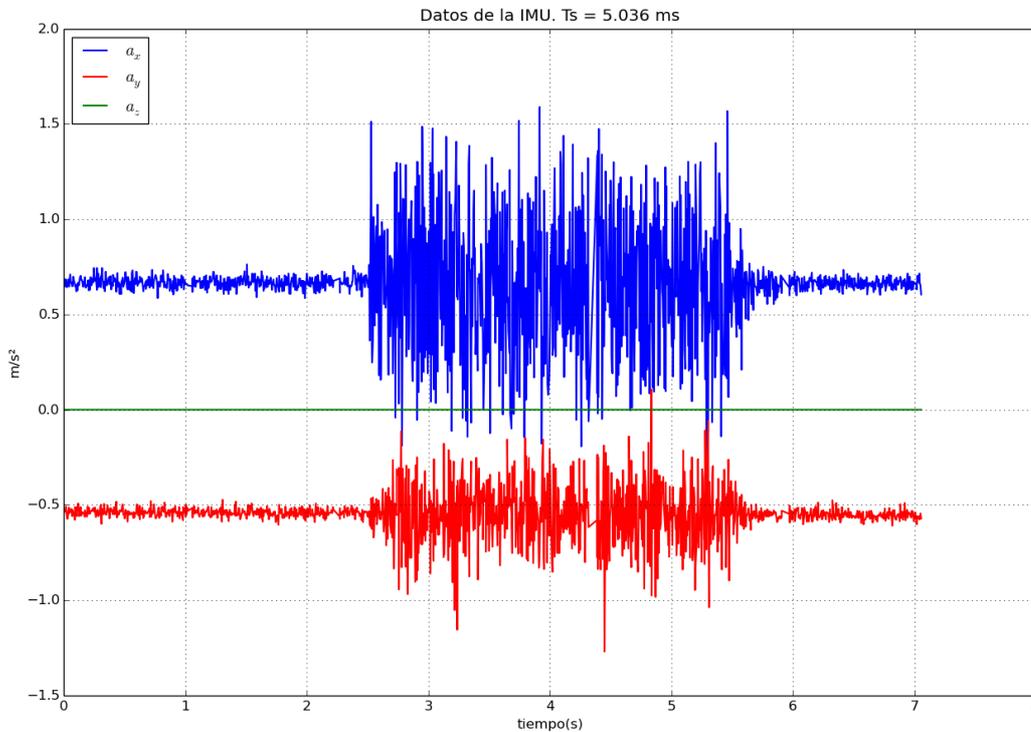
#### 4.2.3 Ruido de los motores

La Gráfica 8 representa los datos obtenidos del azimut sin magnetómetro y la velocidad angular con el giroscopio calibrado. Los motores se ponen en funcionamiento entre los segundos 2 y 5, pero no se aprecia ningún cambio. De hecho, en el apartado 4.2.1 no se obtenía ningún cambio con la introducción de campos magnéticos en el entorno del dispositivo.

En la Gráfica 9 se representa los datos de la IMU. De nuevo, en el periodo que están los motores en funcionamiento, el ruido aumenta. En la aceleración sobre el eje x se observa un aumento de la amplitud pico a pico del ruido, desde los  $0.2 \text{ m/s}^2$  hasta los  $1 \text{ m/s}^2$ . Han aumentado las amplitudes, pero la relación entre el ruido con los motores en marcha y en reposo sigue siendo la misma, 5 veces superior con los motores en marcha. Sobre el eje y se obtiene de nuevo la misma relación.



Gráfica 8. Reposo con smartphone 2. Orientación con magnetómetro. Azimut y velocidad angular.

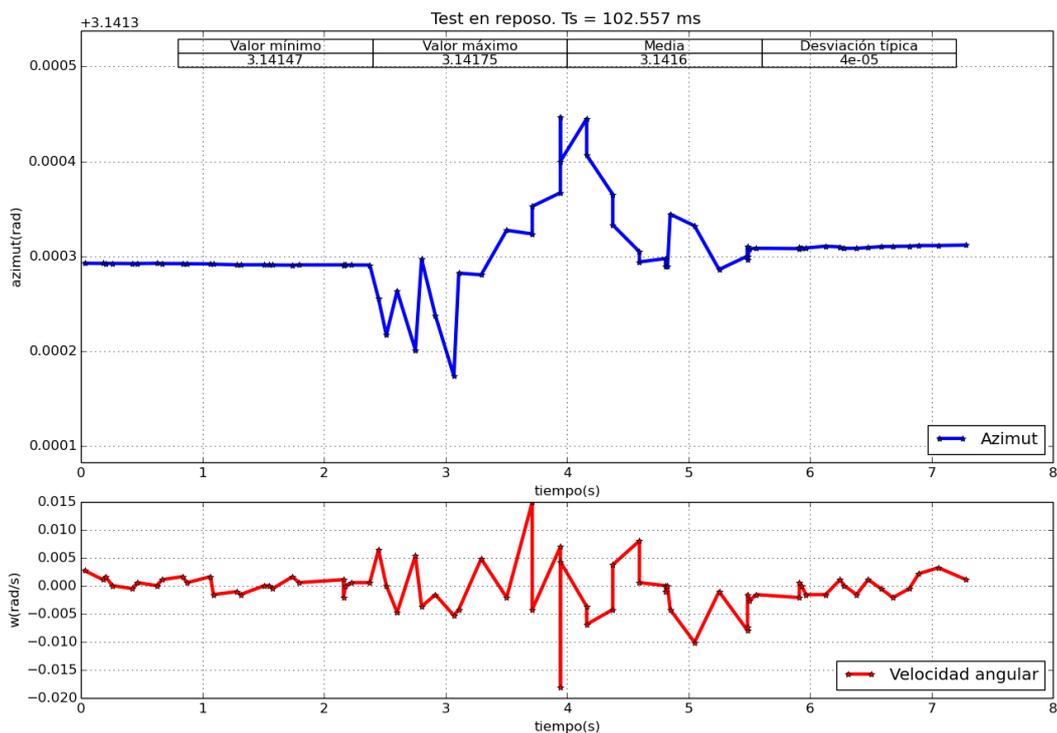


Gráfica 9. Reposo con smartphone 2. Orientación con magnetómetro. IMU.

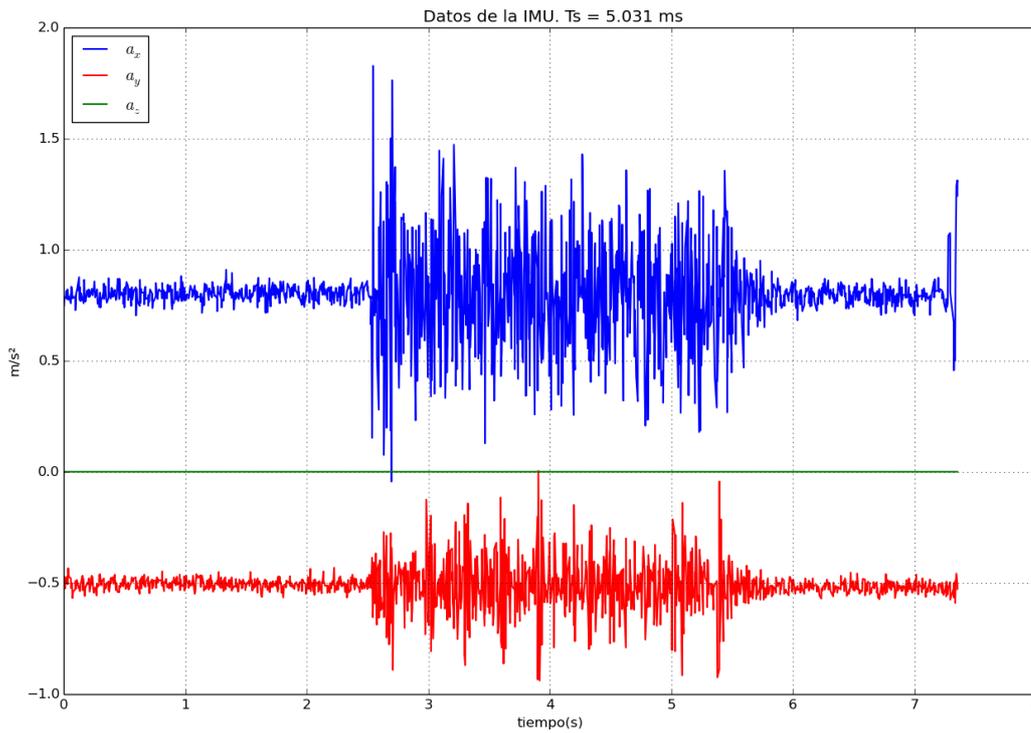
La Gráfica 10 representa los datos obtenidos del azimut sin magnetómetro y la velocidad angular con el giroscopio calibrado. Se observa que el azimut sufre un cambio de comportamiento entre los segundos 2 y 5. Esto significa que, en este caso, el azimut sí se ve afectado por el funcionamiento de los motores, aunque la repercusión es mínima. De hecho, la desviación típica se mantiene por debajo de la diezmilésima. También se observa una pequeña deriva, fenómeno que ya se obtuvo usando el mismo sensor en el apartado 4.2.2.

En la Gráfica 11 se representa los datos de la IMU. En este caso no se aprecia diferencias con la prueba anterior, ya que el sensor usado es el mismo.

Por otro lado, puede observarse que la frecuencia de la medida de la IMU ha aumentado respecto a la obtenida en el smartphone 1. De hecho, el periodo de muestreo en el caso del smartphone 1 es de unos 10 ms, mientras que para el smartphone 2 es de unos 5 ms.



Gráfica 10. Reposo con smartphone 2. Azimut y velocidad angular.



Gráfica 11. Reposo con smartphone 2. IMU.

### 4.3 Conclusiones

En resumen, los resultados obtenidos en los apartados 4.1 y 4.2 son los siguientes:

	TYPE_ROTATION_VECTOR		TYPE_GAME_ROTATION_VECTOR	
	Smartphone 1	Smartphone 2	Smartphone 1	Smartphone 2
<b>Desviación típica</b>	0.0	$6 \cdot 10^{-5}$	0.0	$2 \cdot 10^{-5}$
<b>Deriva</b>	NO	NO	NO	SÍ

Tabla 2. Azimut con y sin magnetómetro. Test en reposo.

El smartphone 1 proporciona una medida mucho más estable del azimut. En cambio, el smartphone 2 es más sensible, presentando una pequeña variación con desviaciones típicas del orden de  $10^{-5}$ , además de una leve deriva en el caso de no usar el magnetómetro.

	TYPE_GYROSCOPE_UNCALIBRATED		TYPE_GYROSCOPE	
	Smartphone 1	Smartphone 2	Smartphone 1	Smartphone 2
<b>Velocidad angular media (rad/s)</b>	0.00532	-0.00209	-0.00018	0.00014

Tabla 3. Giroscopio con y sin compensación de deriva. Test en reposo.

En cuanto al giroscopio, no se aprecian diferencias entre dispositivos. Sí entre sensores, ya que el sensor con la compensación de deriva disminuye a esta en un orden de magnitud.

	TYPE_ROTATION_VECTOR		TYPE_GAME_ROTATION_VECTOR	
	Smartphone 1	Smartphone 2	Smartphone 1	Smartphone 2
<b>Desviación típica</b>	$5 \cdot 10^{-5}$	$1.8 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
<b>Afectan los motores</b>	NO	NO	NO	SÍ

Tabla 4. Ruido de los motores.

En el smartphone 1, los motores afectan levemente al azimut debido a un posible movimiento de este con el funcionamiento de los motores, pero la medida se mantiene estable una vez sufrido un pequeño escalón del orden de la diezmilésima. No se aprecian cambios con el uso o no del magnetómetro. Respecto al acelerómetro, éste obtiene un ruido 5 veces superior.

En el smartphone 2, sí se aprecian diferencias, especialmente con la estimación de la orientación sin el magnetómetro, ya que sufre una mayor desviación durante el tiempo que los motores están en funcionamiento respecto al tiempo que permanecen en reposo. Este cambio no se aprecia con el uso del magnetómetro. Esto es debido a que el ruido que introducen los motores en el azimut sin magnetómetro es de menor magnitud que el ruido que presenta el azimut con el uso del magnetómetro.

Por último, el smartphone 2 proporciona una mayor tasa de transferencia para la IMU.



# 5 ANÁLISIS EN LÍNEA RECTA

---

*“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.”*

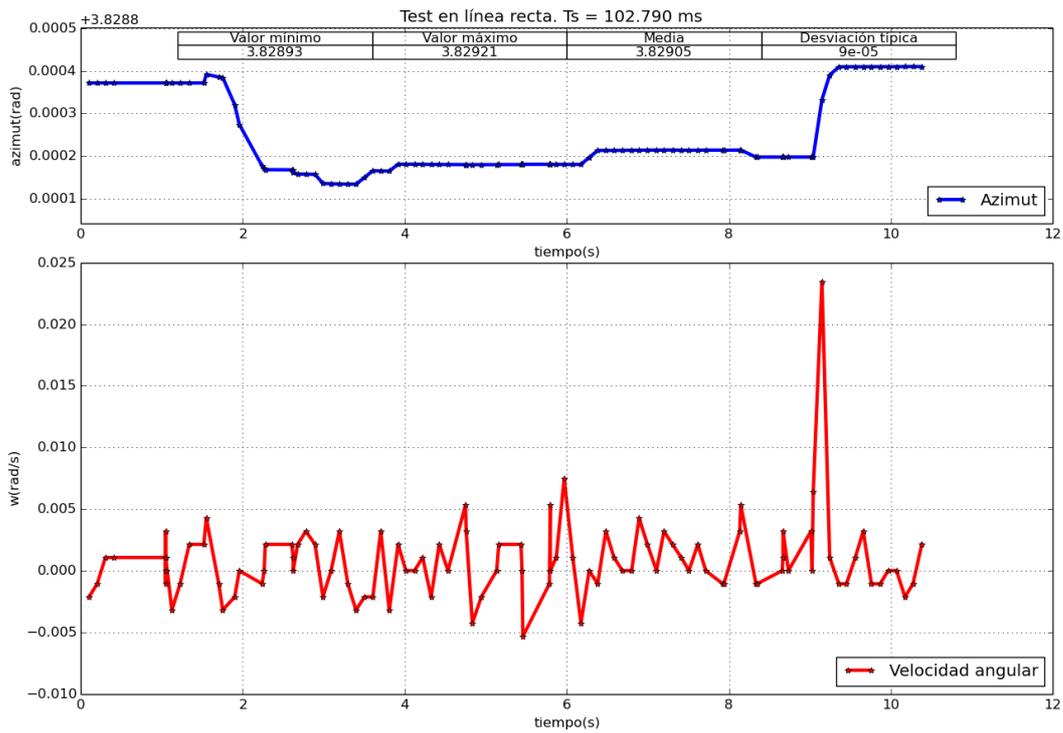
*- Albert Einstein -*

En este apartado, se someterá a Andruino a la prueba de trayectoria en línea recta, haciendo uso de la plataforma presentada en el apartado 3.1. Se realizará tanto con el smartphone 1 (BQ Aquaris M5), como con el smartphone 2 (Xiaomi Pocophone F1), pudiéndose apreciar las diferencias de usar un dispositivo u otro. Además, se verán las diferencias entre usar el magnetómetro o no para la estimación de la orientación. La velocidad angular se obtiene con el giroscopio con la compensación de deriva, ya que las diferencias ya han sido expuestas en el apartado anterior. Los resultados obtenidos se detallan a continuación:

## 5.1 Smartphone 1

### 5.1.1 Orientación con magnetómetro

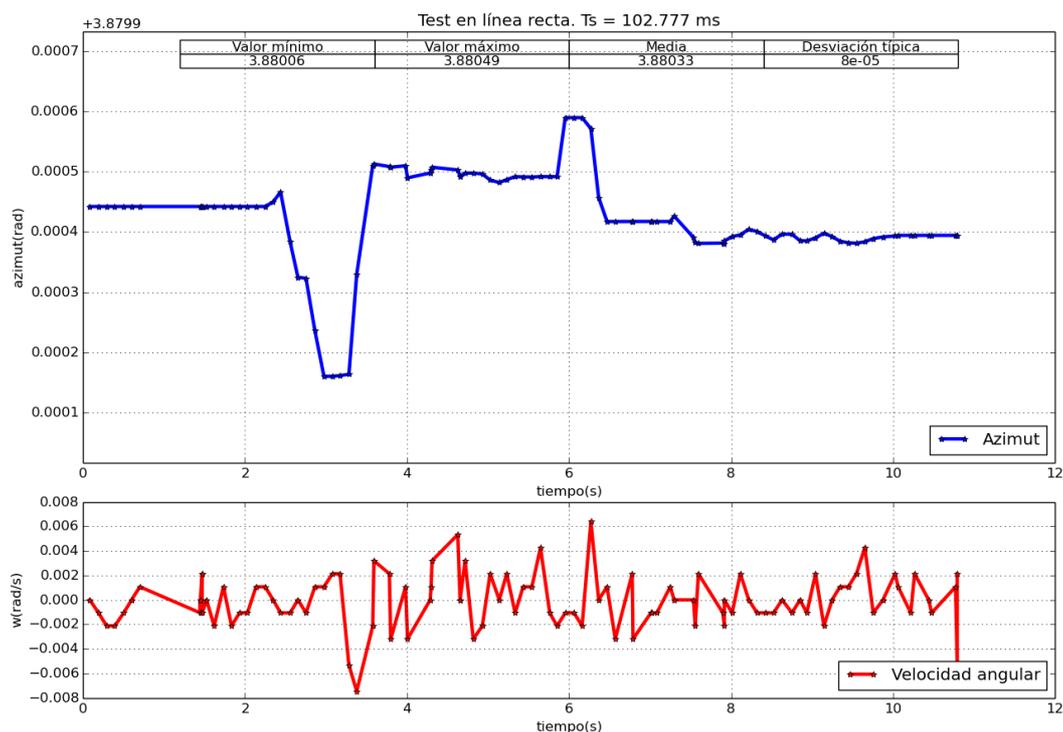
En primer lugar, se realiza el test en línea recta con el sensor TYPE\_ROTATION\_VECTOR, que proporciona el azimut con el uso del acelerómetro, giroscopio y magnetómetro. En la Gráfica 12, se ilustra el azimut y la velocidad angular obtenidos durante la prueba. Se observa un azimut sin deriva, con una desviación típica de  $9 \cdot 10^{-5}$ , y una velocidad angular con compensación de deriva, ya que como se ha comentado antes, se ha usado el giroscopio calibrado.



Gráfica 12. Test en línea recta con smartphone 1. Orientación con magnetómetro.

### 5.1.2 Orientación sin magnetómetro

En este caso, se realiza el test en línea recta con el sensor TYPE\_GAME\_ROTATION\_VECTOR. La Gráfica 13 representa el azimut y la velocidad angular obtenidos en dicha prueba. El azimut tiene una desviación típica de  $8 \cdot 10^{-5}$ , valor prácticamente idéntico al del apartado anterior. Tampoco se obtiene deriva. Por tanto, no se aprecia ninguna mejora en el uso de un sensor u otro, conclusiones que ratifican lo obtenido en el análisis en reposo.

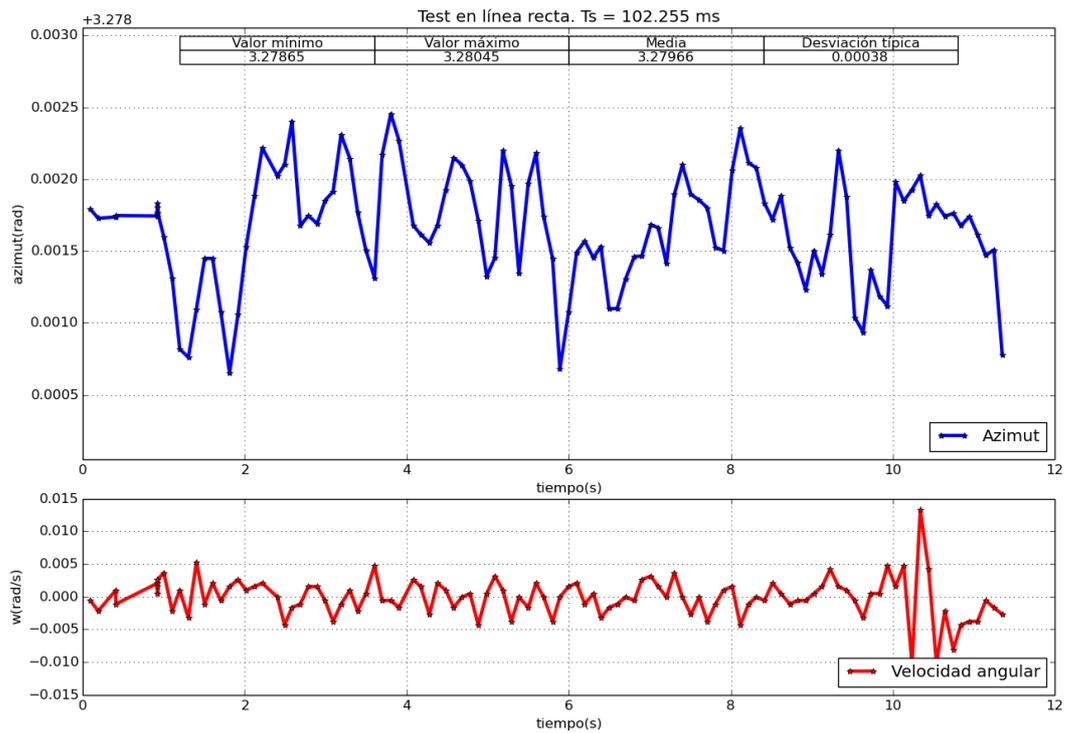


Gráfica 13. Test en línea recta con smartphone 1. Orientación sin magnetómetro.

## 5.2 Smartphone 2

### 5.2.1 Orientación con magnetómetro

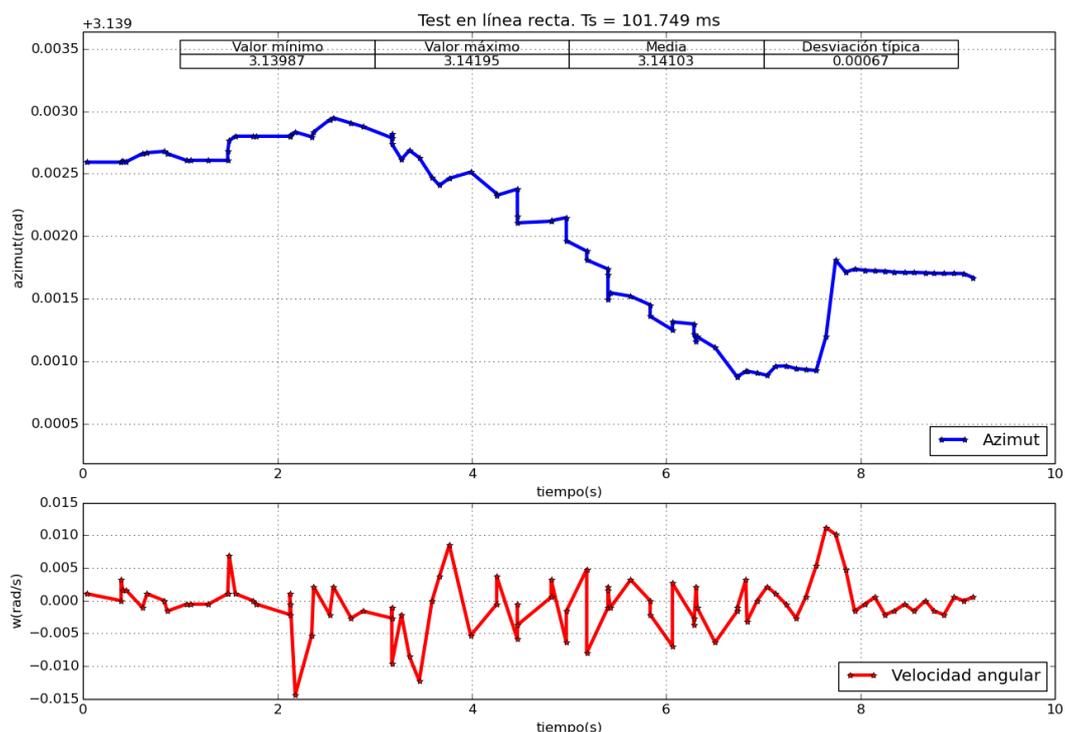
Los resultados del test en línea recta con el sensor TYPE\_ROTATION\_VECTOR para el smartphone 2 se muestra en la Gráfica 14. Se obtiene un azimut con una desviación típica de  $3.8 \cdot 10^{-4}$ , que no presenta deriva. Respecto al análisis en reposo, la desviación obtenida es 6 veces superior, dado que ahora el smartphone está en movimiento, aunque siga con la orientación constante. No obstante, el valor de la desviación típica es muy bajo, por lo que el azimut obtenido sigue siendo bastante bueno.



Gráfica 14. Test en línea recta con smartphone 2. Orientación con magnetómetro.

## 5.2.2 Orientación sin magnetómetro

Haciendo uso del sensor `TYPE_GAME_ROTATION_VECTOR`, se obtiene los siguientes resultados. En la Gráfica 15 se observa que se obtiene un azimut con una deriva descendente y con una desviación típica de  $6.7 \cdot 10^{-4}$ . Contrasta con el azimut obtenido en el apartado anterior, que no obtiene deriva y cuenta con una desviación típica menor. Confirman los resultados obtenidos en el apartado 4.3, donde se obtenía deriva sin el uso del magnetómetro.



Gráfica 15. Test en línea recta con smartphone 2. Orientación sin magnetómetro.

### 5.3 Conclusiones

Los resultados obtenidos son similares a los del capítulo 4, donde se analizó el comportamiento de los sensores con el robot en reposo. Para el test en línea recta, todas las pruebas han obtenido una desviación típica mayor ya que, aunque el robot apunte hacia el mismo sitio, el movimiento hace variar más la medida.

Respecto a la deriva del azimut, sólo se aprecia en el smartphone 2 sin el uso del magnetómetro, que es lo mismo que se observaba en el test con el robot en reposo.

	TYPE_ROTATION_VECTOR		TYPE_GAME_ROTATION_VECTOR	
	Smartphone 1	Smartphone 2	Smartphone 1	Smartphone 2
<b>Desviación típica</b>	$9 \cdot 10^{-5}$	$3.8 \cdot 10^{-4}$	$8 \cdot 10^{-5}$	$6.7 \cdot 10^{-4}$
<b>Deriva</b>	NO	NO	NO	SÍ

Tabla 5. Azimut con y sin magnetómetro. Test en línea recta.



# 6 ANÁLISIS EN ROTACIÓN

---

*“La educación ayuda a la persona a aprender a ser lo que es capaz de ser.”*

*- Hesíodo-*

**E**n este caso se someterá a Andruino a la prueba de rotación. Para ello, se hará uso de la plataforma del apartado 3.2. De forma análoga al capítulo anterior, se realizarán todas las pruebas con ambos dispositivos, obteniendo la orientación con y sin el uso del magnetómetro.

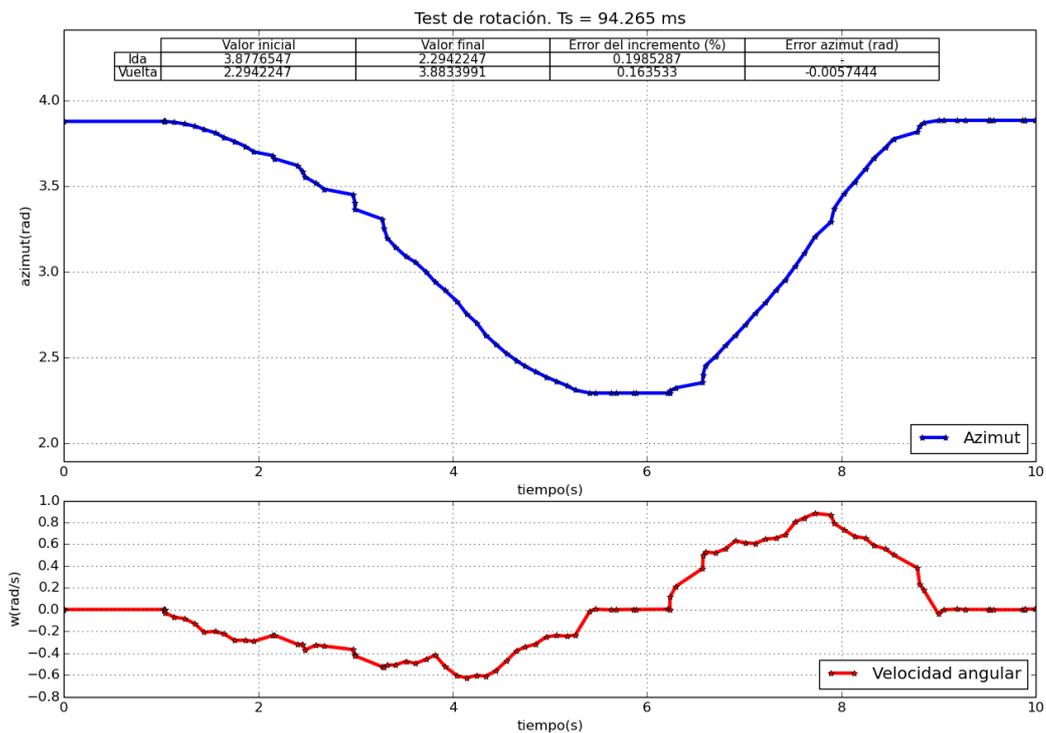
Los aspectos más importantes de esta prueba son el incremento del azimut medido y la diferencia entre las medidas del azimut antes y después de realizar el movimiento. El incremento ideal es de  $90.9^\circ$ . Para cada prueba se calculará el error en el incremento y el error entre los valores iniciales y finales del movimiento.

Se podrá realizar el movimiento de dos formas: rotando primero en sentido horario y regresando en sentido antihorario o viceversa. Se realizará de ambas formas.

## 6.1 Smartphone 1

### 6.1.1 Orientación con magnetómetro

En primer lugar, se realiza la prueba con el sensor TYPE\_ROTATION\_VECTOR. En la Gráfica 16 se observa un error del 0.19% en el incremento del azimut en la ida y del 0.16% en la vuelta. Al finalizar la prueba, se obtiene un azimut 0.0057 radianes mayor. Esta diferencia en la milésima del azimut es mayor que las variaciones en la diezmilésima obtenidas en el test de rotación, por lo que dicho error proviene de la realización del movimiento.

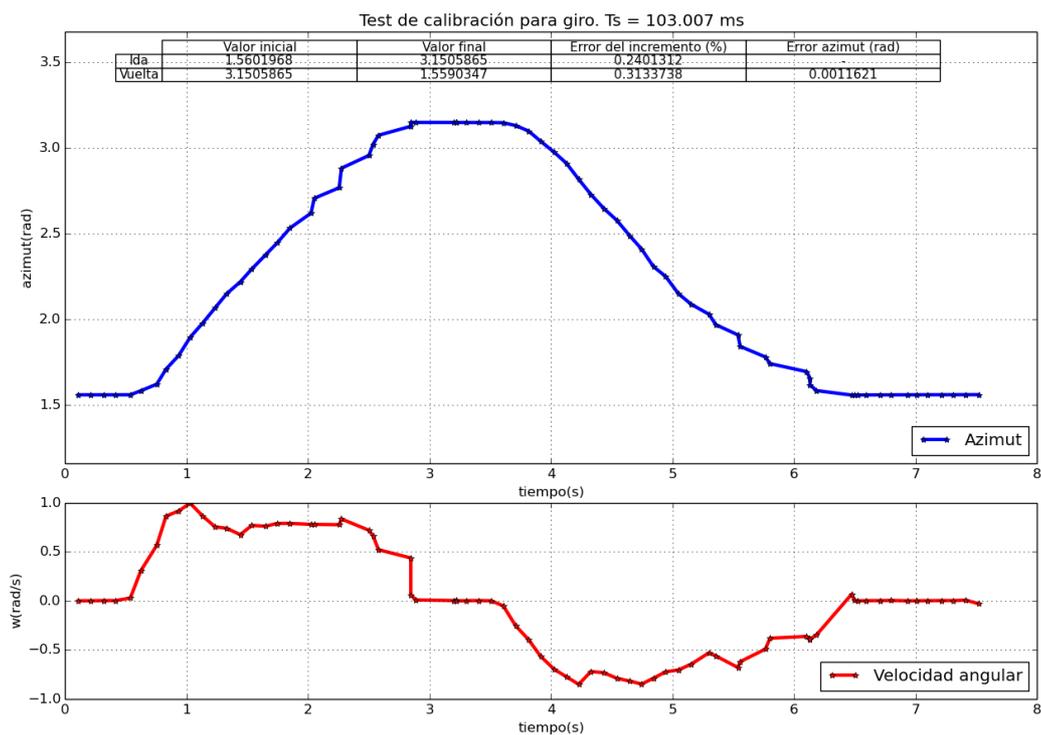


Gráfica 16. Test de rotación con smartphone 1. Orientación con magnetómetro.

### 6.1.2 Orientación sin magnetómetro

En la Gráfica 17, el valor inicial del azimut en la ida es de 1.56 radianes, y de 3.15 radianes de valor final. Esto proporciona un error del 0.24% en la medida del incremento que debería proporcionar. En la vuelta, este error es del 0.31%. Ambos son errores muy pequeños, por debajo del 1%.

En este caso, el error entre el azimut medido antes y después de realizar el movimiento es de 0.00116 radianes. Disminuye respecto a la prueba anterior, pero no significativamente. De hecho, los errores incrementales han subido en esta prueba, por lo que no se aprecia una mejora significativa.

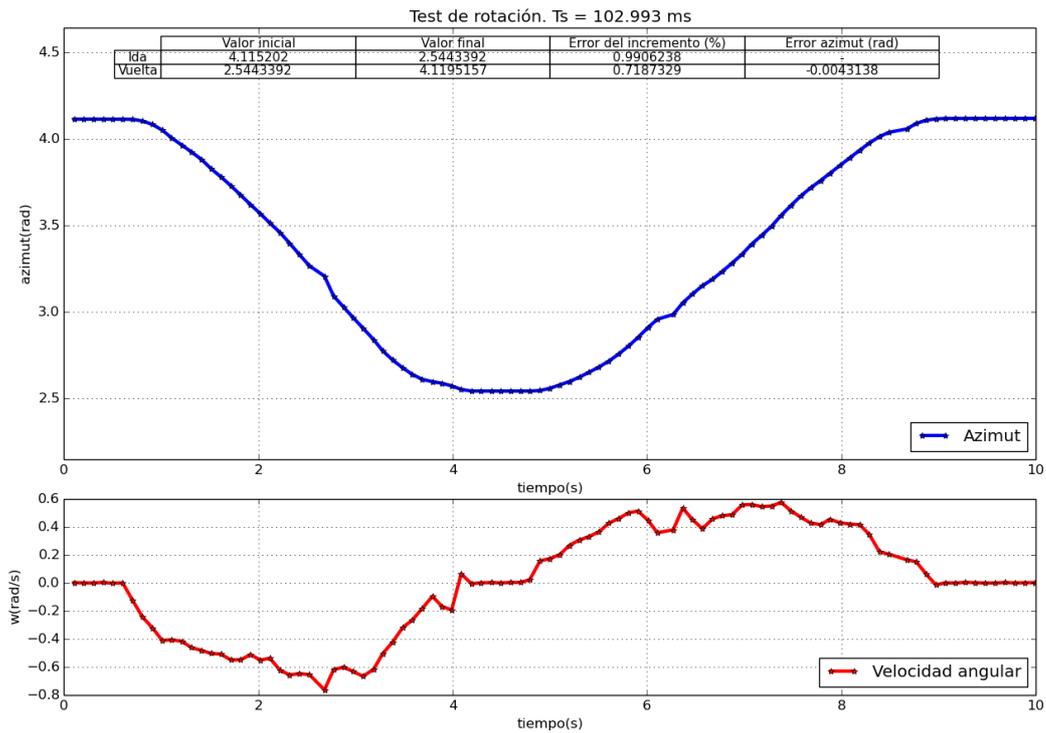


Gráfica 17. Test de rotación con smartphone 1. Orientación sin magnetómetro.

## 6.2 Smartphone 2

### 6.2.1 Orientación con magnetómetro

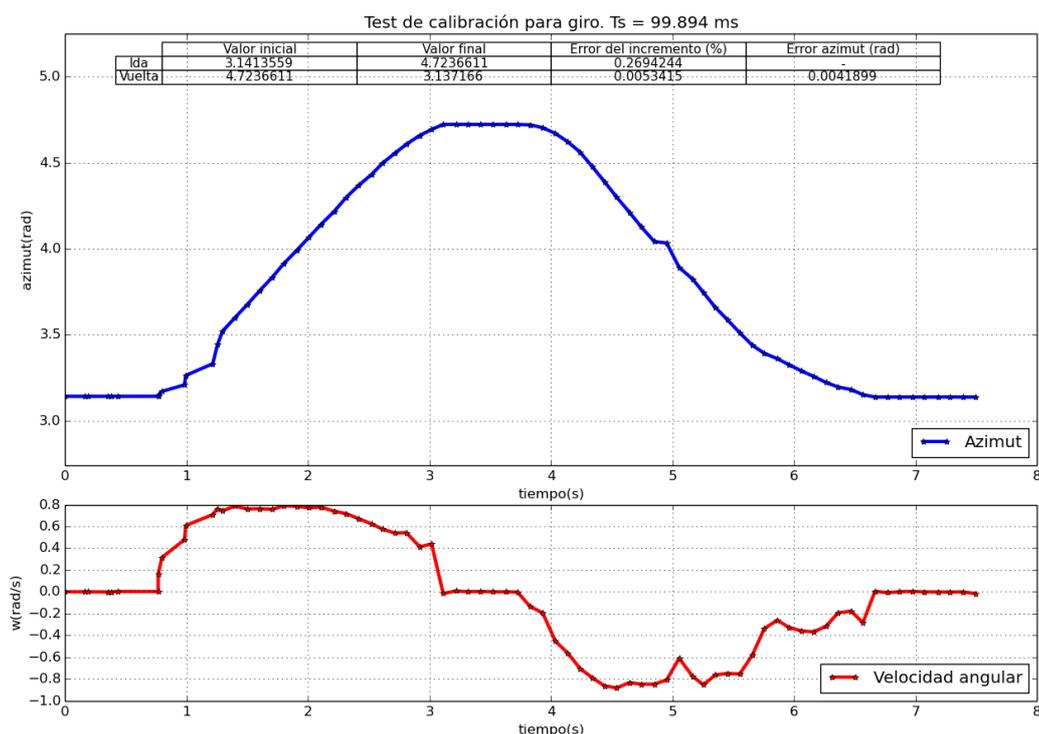
En la Gráfica 18, se representan los resultados obtenidos al realizar el test de rotación con el smartphone 2 y el sensor `TYPE_ROTATION_VECTOR`, que proporciona el azimut con el magnetómetro. Los errores incrementales aumentan respecto al smartphone 1, aunque siguen situándose por debajo del 1%. El error del azimut inicial y final se mantiene en el mismo orden de magnitud, disminuyendo en este caso 0.0043 radianes.



Gráfica 18. Test de rotación con smartphone 2. Orientación con magnetómetro.

### 6.2.2 Orientación sin magnetómetro

En la Gráfica 19 se obtiene un error del 0.269% para el incremento del azimut para la ida, y un 0.005% para la vuelta. Ambos errores han bajado respecto al apartado anterior, donde la orientación se obtuvo con magnetómetro. La mejora es más significativa en el movimiento de vuelta. Respecto al error del azimut inicial y final, en este caso aumenta 0.0042 radianes, misma magnitud que en el caso anterior, pero de sentido opuesto. Por tanto, no significa ninguna mejora.



Gráfica 19. Test de rotación con smartphone 2. Orientación sin magnetómetro.

### 6.3 Conclusiones

Puede verse un resumen de los resultados en la Tabla 6. Los errores en la medida del azimut antes y después de realizar el movimiento son del mismo orden de magnitud en las diferentes pruebas. Se observa que el azimut aumenta con el sensor con magnetómetro y disminuye con el sensor que no lo usa. Es decir, el error obtenido es negativo con `TYPE_ROTATION_VECTOR` y positivo con `TYPE_GAME_ROTATION_VECTOR`. Esto no sigue ningún patrón ya que, repitiendo la misma prueba, el error cambia de signo, manteniendo los mismos órdenes de magnitud de los errores analizados.

Respecto a los errores incrementales, todos están por debajo del 1%. Se observa que en el caso del smartphone 2, estos errores bajan con el sensor que no usa el magnetómetro, aunque la mejora no sea muy grande. En cambio, en el smartphone 1, no se aprecian diferencias significativas entre sensores.

	TYPE_ROTATION_VECTOR		TYPE_GAME_ROTATION_VECTOR	
	Smartphone 1	Smartphone 2	Smartphone 1	Smartphone 2
<b>Error azimut (rad)</b>	-0.0057	-0.0043	0.0012	0.0042
<b>Error incremental en la ida (%)</b>	0.1985	0.9906	0.2401	0.2694
<b>Error incremental en la vuelta (%)</b>	0.1635	0.7187	0.3134	0.0053

Tabla 6. Azimut con y sin magnetómetro. Test de rotación.

# 7 CONCLUSIONES

---

*“Dime y lo olvido, enséñame y lo recuerdo,  
involúcrame y lo aprendo.”*

*- Benjamín Franklin-*

A lo largo del documento se ha ido desarrollando las conclusiones obtenidas en cada apartado. En este último, se hará un resumen de los mismos. A su vez, se propone una línea de trabajo a seguir como continuación del presente documento.

## 7.1 Lista de conclusiones

- Es importante tener una buena base de pruebas. Gracias a las plataformas diseñadas y fabricadas se ha podido estudiar el comportamiento de los sensores de forma precisa. Si estas plataformas no cumplen con lo requerido, el estudio no será válido, así como las conclusiones obtenidas.
- El ruido que añaden los motores no influye significativamente en la medida de los sensores. Sólo se aprecia levemente en el smartphone 2 con el sensor TYPE\_GAME\_ROTATION\_VECTOR con el robot en reposo, donde el azimut tiene un ruido de una amplitud de una diezmilésima.
- El giroscopio calibrado disminuye la deriva, pero esta no desaparece por completo.
- En línea recta, el azimut presenta una medida muy estable, con desviaciones típicas por debajo de la milésima. La medida es más estable en el smartphone 1 que en el smartphone 2, independientemente del sensor utilizado.
- En rotación, el azimut no presenta apenas diferencias entre dispositivos ni entre sensores. Simplemente se aprecia una leve mejora en el incremento del azimut a favor del smartphone 1 con el uso del sensor con magnetómetro.
- El dispositivo android utilizado influye en la medida. El tiempo de muestreo de la IMU baja a la mitad si se usa el dispositivo con mejor procesador. Por otro lado, el azimut es ligeramente mejor con el smartphone con la API inferior. La importancia de estas pequeñas mejoras vendrá dada por el tipo de seguimiento que se quiera hacer de los sensores, o el tipo de control usado.

## 7.2 Línea de trabajo futuro

Como se comentó al inicio del documento, el propósito de éste era analizar los sensores que intervienen en el proceso de odometría, imprescindible para la autonomía del robot. Por tanto, la principal línea de trabajo será conseguir dicha autonomía.

Para ello, es importantísimo un buen control que gobierne la localización del robot. Hay múltiples formas de implementarlo, desde controles simples con arquitectura reactiva, hasta complejas arquitecturas deliberativas que necesitan una planificación y seguimiento de la trayectoria.

Otro aspecto importante sería el programa implementado en Arduino. Se podría estudiar una forma más sencilla computacionalmente de recoger los datos de los sensores que permita una mayor tasa de muestreo, así como un mayor ahorro energético.

En definitiva, Andruino cuenta con un gran campo de desarrollo que permitirá ampliar el alcance del proyecto.

# REFERENCIAS

---

- [1] Open Source Robotics Foundation, «ROS,» 18 2 2018. [En línea]. Available: <http://wiki.ros.org/es>. [Último acceso: 2019].
- [2] Open Source Robotics Foundation, «ROS Tutorials,» [En línea]. Available: <http://wiki.ros.org/es/ROS/Tutorials>. [Último acceso: 2019].
- [3] F. López, «Resumen Andruino».
- [4] Wikipedia, «Python,» [En línea]. Available: <https://es.wikipedia.org/wiki/Python>. [Último acceso: 2019].
- [5] Python Software Foundation, «Python Documentation,» 2001-2019. [En línea]. Available: <https://docs.python.org>. [Último acceso: 2019].
- [6] Xataka, «Qué es Arduino,» [En línea]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. [Último acceso: 2019].
- [7] Wikipedia, «RUR,» [En línea]. Available: [https://es.wikipedia.org/wiki/R.U.R.\\_\(Robots\\_Universales\\_Rossum\)](https://es.wikipedia.org/wiki/R.U.R._(Robots_Universales_Rossum)). [Último acceso: 2019].
- [8] Amazon, «Amazon Robotics,» [En línea]. Available: <https://www.amazon.es/b?ie=UTF8&node=13495453031>. [Último acceso: 2019].
- [9] Xataka, «Niveles de conducción autónoma,» [En línea]. Available: <https://www.xataka.com/automovil/de-0-a-5-cuales-son-los-diferentes-niveles-de-conduccion-autonoma>. [Último acceso: 2019].
- [10] SAE, «Estándar SAE J3016,» [En línea]. Available: <https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic>. [Último acceso: 2019].
- [11] iRobot, «iRobot,» [En línea]. Available: <https://www.irobot.es/>. [Último acceso: 2019].
- [12] Hubot, «Hubot. Robótica educativa,» [En línea]. Available: <https://hubot.cl/>. [Último acceso: 2019].
- [13] Google Developers, «Motion Sensors,» [En línea]. Available: [https://developer.android.com/guide/topics/sensors/sensors\\_motion](https://developer.android.com/guide/topics/sensors/sensors_motion). [Último acceso: 2019].
- [14] Google, «Developer Android,» 2019. [En línea]. Available: <https://developer.android.com/guide/topics/sensors>. [Último acceso: 2019].
- [15] Matplotlib Development Team, «Matplotlib,» 2018. [En línea]. Available: <https://matplotlib.org/>. [Último acceso: 2019].

# ÍNDICE DE CONCEPTOS

---

- **azimut:** ángulo de orientación sobre la superficie de una esfera. 1
- **domótica:** conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda. 8
- **framework:** conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular. 2
- **metacrilato:** sólido transparente, rígido y resistente a los agentes atmosféricos, y uno de los materiales plásticos más utilizados. 12
- **odometría:** procedimiento para determinar la localización y orientación de un vehículo. 1
- **sensores fotorresistivos:** componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. 1

# GLOSARIO

---

API: Application Programming Interface	5
IP: Internet Protocol	3
LDR: Light Depending Resistor	1
RAM: Random Access Memory	2
ROS: Robot Operating System	1
SAE: Society of Automotive Engineers	8
USB: Universal Serial Bus	4