

	<b>UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA</b>			
	Documento	Código	Fecha	Revisión
	<b>FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO</b>	<b>F-AC-DBL-007</b>	<b>08-07-2021</b>	<b>B</b>
Dependencia	Aprobado		Pág.	
<b>DIVISIÓN DE BIBLIOTECA</b>	<b>SUBDIRECTOR ACADEMICO</b>		<b>1(128)</b>	

## RESUMEN – TRABAJO DE GRADO

<b>AUTORES</b>	Yair Eduardo Santiago Peña John Jhader Vega Quintero		
<b>FACULTAD</b>	Ingenierías		
<b>PLAN DE ESTUDIOS</b>	Ingeniería De Sistemas		
<b>DIRECTOR</b>	Alveiro Alonso Rosado Gómez		
<b>TÍTULO DE LA TESIS</b>	Diseño y construcción de un kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de arduino para la materia de robótica de la Universidad Francisco de Paula Santander Ocaña.		
<b>TITULO EN INGLES</b>	Design and construction of a learning kit for the assembly and programming of an autonomous mobile robot based on the arduino platform for robotics at the Francisco de Paula Santander Ocaña University.		
<b>RESUMEN</b> (70 palabras)			
Este proyecto investigativo abarca el proceso de diseño y construcción de un kit de aprendizaje que sirva como herramienta en el desarrollo de robots móviles en la materia de robótica. Parte de un análisis en los contenidos programáticos de la materia de robótica y de las dificultades que presentan los estudiantes en los procesos de aprendizaje entorno a esta área. Como resultado de esta investigación se propone un kit de aprendizaje como instrumento para incentivar el desarrollo del pensamiento lógico y la asimilación de conceptos prácticos entorno a la robótica.			
<b>RESUMEN EN INGLES</b>			
This research project covers the design and construction process of a learning kit that serves as a tool in the development of mobile robots in the field of robotics. Part of an analysis of the programmatic contents of the robotics subject and the difficulties that students present in the learning processes around this area. As a result of this research, a learning kit is proposed as an instrument to encourage the development of logical thinking and the assimilation of practical concepts around robotics.			
<b>PALABRAS CLAVES</b>	Arduino, Robótica, Automata, Electrónica, Kit robótico		
<b>PALABRAS CLAVES EN INGLES</b>	Arduino, Robotics, Automaton, Electronics, Robotic Kit		
<b>CARACTERÍSTICAS</b>			
PÁGINAS: 129	PLANOS:	ILUSTRACIONES: 81	CD-ROM: 1



DISEÑO Y CONSTRUCCIÓN DE UN KIT DE APRENDIZAJE PARA EL ENSAMBLAJE Y  
LA PROGRAMACIÓN DE UN ROBOT MÓVIL AUTÓNOMO BASADO EN LA  
PLATAFORMA DE ARDUINO PARA LA MATERIA DE ROBÓTICA DE LA  
UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

AUTORES:

YAIR EDUARDO SANTIAGO PEÑA

JOHN JHADER VEGA QUINTERO

Trabajo de grado para Optar el título de Ingeniero de Sistemas

Director

Msc. ALVEIRO ALONSO ROSADO GÓMEZ

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

FACULTAD DE INGENIERÍAS

PLAN DE ESTUDIOS INGENIERIA DE SISTEMAS

Ocaña, Colombia

Agosto, 2021

## Índice

<b>Capítulo 1. Diseño y construcción de un kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino para la materia de Robótica de la Universidad Francisco de Paula Santander Ocaña .....</b>	<b>1</b>
1.1 Planteamiento del problema .....	1
1.2 Formulación del problema .....	3
1.3 Objetivos .....	3
1.4 Justificación .....	4
1.5 Delimitaciones .....	6
<b>Capítulo 2. Marco referencial .....</b>	<b>8</b>
2.1 Marco histórico .....	8
2.2 Marco Conceptual .....	11
2.3 Marco Teórico .....	17
2.4 Marco legal .....	19
<b>Capítulo 3. Diseño metodológico .....</b>	<b>26</b>
3.1 Tipo de investigación .....	26
3.2 Población y muestra .....	26
<b>Capítulo 4. Resultados .....</b>	<b>27</b>
4.1 Diseño y construcción de un circuito electrónico que permite leer diferentes señales analógicas y digitales .....	31
4.1.1 Diseño Esquemas electrónicos .....	33
4.1.2 Digitalización de diagramas electrónicos en SolidWorks PCB .....	45
4.1.3 Creación de PCB y posicionamiento de componentes. ....	47
4.1.4 Generación de pistas para PCB .....	49
4.1.5 Impresión de PCB y montaje de componentes. ....	51
4.1.6 Quema de Bootloader. ....	52
4.2 Construcción de estructura mecánica adaptable a la placa electrónica que facilite la movilidad del robot .....	56
4.2.1 Plataforma N1: Velocista .....	58
4.2.2 Plataforma N2: Todo terreno .....	59
4.2.3 Plataforma N3: Modelo Híbrido .....	60

4.3 Diseño de algoritmo en lenguaje Arduino para el procesamiento de señales y control, con el fin de obtener el mejor rendimiento posible del robot .....	61
4.3.1 Ejemplos de código .....	61
4.3.2 Algoritmo PID para un robot velocista .....	68
4.4 Creación del manual que servirá de guía para el ensamblaje, programación y buen manejo del robot. ....	73
4.5 Socialización del diseño y construcción del kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino.....	73
4.5.1 Temática abarcada .....	73
4.5.2 Encuesta .....	74
<b>Capítulo 5. Conclusiones .....</b>	<b>81</b>
<b>Capítulo 6. Recomendaciones .....</b>	<b>82</b>
<b>Referencias.....</b>	<b>83</b>
<b>Apéndices.....</b>	<b>85</b>
Apéndice A. Manual de uso kit robótico .....	86

## Lista de Figuras

<i>Figura 1.</i> Circuito integrado .....	12
<i>Figura 2.</i> Microcontrolador .....	12
<i>Figura 3.</i> Elementos de un microprocesador .....	13
<i>Figura 4.</i> Controlador o regulador PID .....	14
<i>Figura 5.</i> Arduino .....	15
<i>Figura 6.</i> Hardware libre .....	16
<i>Figura 7.</i> Sistemas kit robótico.....	30
<i>Figura 8.</i> UFPSO.INO .....	31
<i>Figura 9.</i> Pasos construcción .....	32
<i>Figura 10.</i> Etapas de la placa electrónica .....	33
<i>Figura 11.</i> Etapa de control .....	34
<i>Figura 12.</i> Microcontrolador Atmega328 AU .....	35
<i>Figura 13.</i> Oscilador.....	37
<i>Figura 14.</i> Condensadores .....	37
<i>Figura 15.</i> Programador.....	38
<i>Figura 16.</i> FT232RL.....	39
<i>Figura 17.</i> Etapa de potencia. ....	41
<i>Figura 18.</i> Driver de motor.....	42
<i>Figura 19.</i> Etapa de alimentación.....	43
<i>Figura 20.</i> Buzzer. ....	44
<i>Figura 21.</i> SolidWorks PCB.....	46
<i>Figura 22.</i> Esquemas electrónicos SolidWorks.....	46
<i>Figura 23.</i> Croquis PCB. ....	47
<i>Figura 24.</i> Posicionamiento de componentes PCB. ....	48
<i>Figura 25.</i> Respaldo de tarjeta.....	49
<i>Figura 26.</i> Generación de pistas PCB.....	50
<i>Figura 27.</i> PCB 3D.....	50
<i>Figura 28.</i> Resultado final de la placa. ....	51
<i>Figura 29.</i> Arduino ISP. ....	53
<i>Figura 30.</i> Conexión Arduino Programado. ....	54
<i>Figura 31.</i> Conexión UFPSO.INO .....	54
<i>Figura 32.</i> Configuración ISP.....	55
<i>Figura 33.</i> Quema de Bootloader. ....	56
<i>Figura 34.</i> Plataforma Seguidor de Línea.....	58
<i>Figura 35.</i> Plataforma Todo Terreno.....	59
<i>Figura 36.</i> Plataforma Híbrida.....	60
<i>Figura 37.</i> Código Led Pin 13. ....	62
<i>Figura 38.</i> Código Led Pin 9 PWM.....	62
<i>Figura 39.</i> Código Buzzer Pin 4.....	63
<i>Figura 40.</i> Código Drivers de Motor.....	64
<i>Figura 41.</i> Código Sensor Ultrasonido.....	65
<i>Figura 42.</i> Código Sensor Qtr8-A. Fuente .....	67
<i>Figura 43.</i> Algoritmo Seguidor de Línea P1 .....	69

<i>Figura 44.</i> Algoritmo Seguidor de Línea P2 .....	70
<i>Figura 45.</i> Algoritmo Seguidor de Línea P3 .....	71
<i>Figura 46.</i> Algoritmo Seguidor de Línea P4 .....	72
<i>Figura 47.</i> Pregunta 1 Encuesta socialización .....	74
<i>Figura 48.</i> Pregunta 2 Encuesta socialización .....	75
<i>Figura 49.</i> Pregunta 3 Encuesta socialización .....	76
<i>Figura 50.</i> Pregunta 4 Encuesta socialización .....	76
<i>Figura 51.</i> Pregunta 5 Encuesta socialización .....	77
<i>Figura 52.</i> Pregunta 6 Encuesta socialización .....	78
<i>Figura 53.</i> Pregunta 7 Encuesta socialización .....	79
<i>Figura 54.</i> Manual de Uso. Diagrama de Componentes .....	90
<i>Figura 55.</i> Manual de Uso. Resumen características .....	91
<i>Figura 56.</i> Manual de Uso. Características destacadas .....	92
<i>Figura 57.</i> Manual de Uso. Diagrama de pines .....	93
<i>Figura 58.</i> Manual de Uso. Cable USB- Mini-USB.....	94
<i>Figura 59.</i> Manual de Uso. Detección de puerto Arduino .....	95
<i>Figura 60.</i> Manual de Uso. Administrador de dispositivos.....	96
<i>Figura 61.</i> Manual de Uso. Actualización de controlador.....	96
<i>Figura 62.</i> Manual de Uso. Selección de controlador .....	97
<i>Figura 63.</i> Manual de Uso. Diagrama conexiones LED. ....	98
<i>Figura 64.</i> Manual de Uso. Esquemático LED.....	98
<i>Figura 65.</i> Manual de Uso. Código Led Arduino.....	99
<i>Figura 66.</i> Manual de Uso. Código Buzzer Arduino.....	100
<i>Figura 67.</i> Manual de Uso. Conexiones drivers de motor.....	101
<i>Figura 68.</i> Manual de Uso. Código drivers de motor.....	102
<i>Figura 69.</i> Manual de Uso. Plataforma todo terreno .....	103
<i>Figura 70.</i> Manual de Uso. Plataforma híbrida .....	104
<i>Figura 71.</i> Manual de Uso. Plataforma seguidora de línea .....	105
<i>Figura 72.</i> Manual de Uso. Conexiones seguidor de línea.....	107
<i>Figura 73.</i> Manual de Uso. Estructura mecánica seguidor de línea .....	108
<i>Figura 74.</i> Manual de Uso. Código seguidor de línea P1.....	109
<i>Figura 75.</i> Manual de Uso. Código seguidor de línea P2.....	110
<i>Figura 76.</i> Manual de Uso. Código seguidor de línea P3.....	111
<i>Figura 77.</i> Manual de Uso. Código seguidor de línea P4.....	112
<i>Figura 78.</i> Manual de Uso. Rueda loca. ....	113
<i>Figura 79.</i> Manual de Uso. PID.....	115
<i>Figura 80.</i> Manual de Uso. Corrección de error PID .....	116
<i>Figura 81.</i> Manual de Uso. Pista .....	120

# **Capítulo 1. Diseño y construcción de un kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino para la materia de Robótica de la Universidad Francisco de Paula Santander Ocaña**

## **1.1 Planteamiento del problema**

La robótica experimenta en la actualidad una actuación en muchos campos como: industrial, médico, educativo, ocio, entre muchos otros. En la robótica educativa intervienen: el diseño, la construcción y la programación de un robot e involucra varias áreas de conocimiento como: electrónica, informática, mecánica, matemáticas y física. (Pinto Salamanca & Pérez Holguín, 2010)

La robótica educativa también está siendo utilizada como herramienta de apoyo en el proceso enseñanza-aprendizaje en todos los niveles educativos, porque facilita y motiva el entendimiento de conceptos complejos y permite el desarrollo de competencias básicas. Un robot puede ser complejo o sencillo, móvil o con articulaciones y puede constar de alguno de los siguientes elementos: estructura mecánica, transmisores, sistemas de accionamiento, sistema sensorial, sistema de control y terminales. (Mayorga, 2015)

De otra parte, con los kits se puede dar un acercamiento al uso de microcontroladores, sensores, motores, temporizadores y a la programación, entre otras muchas cosas. En el diseño de un robot se involucra tiempo e integración de conocimientos, se tiene la ventaja de buscar

componentes específicos que con los avances tecnológicos pueden ser de bajo costo y de un consumo menor de potencia.

En las últimas décadas, se ha incrementado el uso de robots en las aulas y como existe una gran variedad, es factible su adquisición por los diferentes costos y funcionamientos, es así que al integrarlos en clase se promueve el interés y el desarrollo cognitivo de los estudiantes al tiempo que trabajan con ciencia y tecnología. Como existen concursos nacionales e internacionales con robots el interés y la participación de uso se incrementan en la comunidad estudiantil. (Barrera Lombana, 2014)

En este sentido, se diseñó y aplicó una encuesta a los estudiantes de la asignatura de robótica con el objetivo de verificar y conocer las dificultades que presentan los mismos y de esta forma hacer un sondeo para proponer una herramienta de aprendizaje que facilite la adquisición de competencias en el ensamblaje y la programación de robots móviles.

En dichas encuestas los estudiantes afirman que sienten algunas falencias en cuanto a las habilidades en construcción, diseño y programación de robots, puesto que la materia tiene un fuerte enfoque teórico y las horas de practica en los laboratorios a veces no son suficientes para propiciar la adquisición de competencias prácticas que serían de mucha ayuda en el desempeño laboral.

Al preguntar sobre la identificación correcta de los elementos y características de un sistema robótico, los encuestados afirman que en muchos casos no los conocen a detalle ni saben

cuáles son los apropiados para el robot o mecanismo a desarrollar. Por otra parte, no se puede desconocer que poseen competencias y ventajas en el desarrollo de aplicativos de software y hardware orientados a proyectos robóticos, ya que con estos temas desde el primer semestre se han ido familiarizando.

## **1.2 Formulación del problema**

¿Es posible facilitar el aprendizaje en la materia de robótica mediante la construcción de un kit de para la creación de un robot móvil?

## **1.3 Objetivos**

**1.3.1 General.** Diseñar y construir un kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino para la materia de Robótica de la Universidad Francisco de Paula Santander Ocaña.

### **1.3.2 Específicos.**

-Diseñar y construir un circuito electrónico que permita leer las diferentes señales analógicas y digitales.

-Construir la estructura mecánica que se adapte a la placa electrónica y facilite la movilidad del robot.

-Diseñar algoritmo en lenguaje Arduino para el procesamiento de señales y control con el fin de obtener el mejor rendimiento posible del robot.

-Crear el manual que servirá de guía para el ensamblaje, programación y buen manejo del robot.

-Socializar el diseño y construcción del kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino, con los estudiantes de la materia de Robótica de la Universidad Francisco de Paula Santander Ocaña.

#### **1.4 Justificación**

La electrónica pretende ser una herramienta eficaz en la sociedad que cada día requiere de más tecnología para la evolución de la educación, cuyas necesidades hasta el momento han sido abastecidas con aplicaciones desarrolladas por conceptos electrónicos, procesos de control, automatización y robótica, siendo esto lo que permite generar sistemas de control inteligentes que sean aplicados a los procesos de aprendizaje y así lograr proyectar teorías y conocimientos adquiridos anticipándose a las necesidades cada vez mayores de la educación.

De esta manera, se pretende generar y cultivar el interés sobre la educación en áreas de la investigación, robótica y su aplicabilidad, creando soluciones innovadoras a las necesidades de los estudiantes y entidades educativas.

Según un estudio realizado por Ojeda, (2018), el doctor Holman Ariza, profesor y coordinador del Semillero de Robótica de la Universidad El Bosque, dice que es consciente de la necesidad de formar profesionales con estas características en Colombia: “A partir de la cuarta revolución industrial estamos marcando tendencia hacia la robótica y las telecomunicaciones. Hacia allá va el mundo, en ese rumbo se está desarrollando la sociedad. Por lo tanto, nosotros buscamos que los humanos se capaciten para ese escenario”.

De otra parte, con el presente proyecto se pretende articular varios componentes curriculares de acuerdo con el área de la ingeniería, logrando mejorar el aprendizaje en la asignatura de la robótica, desarrollando el diseño digital, robots móviles, control e informática, herramientas integradoras de conocimiento y apoye procesos de investigación. Por lo tanto, se debe decir que la robótica se ha posicionado globalmente como un sinónimo de desarrollo en los países y la innovación en el campo de la tecnología y la ciencia trae grandes beneficios económicos para los países que invierten en estos ámbitos, por esta razón Colombia debe promover la investigación y participación de los estudiantes en proyectos que permitan el desarrollo de tecnologías. Por ende, la realización de este artefacto proporciona condiciones que facilitan el desarrollo del pensamiento tecnológico y sirven como plataforma para el desarrollo de tecnologías. (Zayas Barreras, 2010)

De igual forma, este trabajo de grado puede desarrollarse a futuro como un proyecto de investigación en Colciencias orientados a la automatización en el campo, generando un impacto social en el ámbito educativo, utilizando los sistemas de visión artificial como elemento

integrador de conocimiento del plan de estudios de las carreras de ingeniería: electrónica, sistemas, telecomunicaciones, mecatrónica, entre otros.

Como también se pretende impulsar la robótica en la Universidad Francisco de Paula Santander Ocaña, de manera que la participación de estudiantes de ingeniería aumente su conocimiento generado en el proceso de construcción de robots, será también de utilidad para el desarrollo de futuras aplicaciones de control y procesamiento de señales que busquen dar solución a problemas locales y nacionales. Incentivando el uso tecnologías con enfoque en robótica, programación y mecánica por parte de las instituciones educativas y los semilleros de la universidad esperando a mediano y largo plazo un aumento en el desarrollo de proyectos con énfasis en robótica y automatización.

## **1.5 Delimitaciones**

**1.5.1 Operativa.** Se dispone de los medios físicos, tecnológicos y financieros para el cumplimiento de los diferentes objetivos específicos.

Se trabajará el proyecto de diseño empleando la plataforma de Arduino para el desarrollo del trabajo de grado.

**1.5.2 Conceptual.** Para el diseño del proyecto se tendrán en cuenta los siguientes términos: Microcontroladores, Matlab, IA, sensores, PID, Arduino, SolidWorks, hardware libre, robótica.

**1.5.3 Geográfica.** El proyecto de investigación se llevará a cabo en la Universidad Francisco de Paula Santander Ocaña, que es donde nace, partiendo de las necesidades y problemas que actualmente se presentan con los estudiantes de la asignatura de robótica.

**1.5.4 Temporal.** El proyecto por desarrollar se llevará a cabo en la Universidad Francisco de Paula Santander Ocaña, será de dos meses, como se muestra en el cronograma de actividades.

## Capítulo 2. Marco referencial

### 2.1 Marco histórico

**2.1.1 Antecedentes históricos a nivel internacional.** En la investigación realizada por Ramírez y Reyes (2015), se dice que es de conocimiento general, la tecnología y esta avanza cada día más y más, por lo que los estudiantes de ingenierías deben actualizar sus conocimientos para poder hacer frente a los nuevos equipos e instrumentos que puedan encontrar a lo largo de su carrera profesional.

Por lo tanto, la investigación se basó en el universo de la robótica ya que esta área es multifacética pues se aplican conocimientos mecánicos, eléctricos, electrónicos y programación de controladores, por lo que, si se quiere aprender más y profundizar en dichos temas, la realización de un robot es una oportunidad perfecta para crecer profesionalmente.

Es por esto, que el club de robótica de la Universidad Salesiana intenta incentivar a los estudiantes con actividades extracurriculares en el marco de competiciones de robots para que los alumnos encuentren la motivación necesaria para el desarrollo de la robótica y la electrónica general, por lo que se pretende con este proyecto aportar al mejoramiento de dicha área y ayudar a los estudiantes a reconocer diferentes tecnologías (Ramírez & Reyes, 2015).

Para lo anterior una de las principales características de este proyecto es la utilización de una tarjeta NI sbRIO-9631 de National Instrument la cual estuvo basada en tecnología FPGA,

de las cuales su uso se está viendo más extendido y necesita más comprensión para mayor manejo de estas (Ramírez & Reyes, 2015).

De otra parte, en la primera década del 2000 hasta la actualidad, la robótica, es un área multidisciplinaria, se ha desplegado y evolucionado en universidades de todo el mundo, promoviéndose cada vez más su desarrollo mediante concursos de robótica (varias modalidades), como el concurso CER-2010 que ocurrió en el año 2011 en la Universidad Católica de Santiago de Guayaquil.

A pesar de que la implementación de un laboratorio de robótica está fuera del presupuesto de muchas universidades por los altos costos de robots manipuladores y móviles comerciales, es posible hacer prácticas de robótica con paquetes comerciales de bajo costo y construcciones propias (Sagal & Mora, 2018).

El uso de los robots y de las tareas para las cuales fueron construido, se han ido diversificándose, como por ejemplo se pueden encontrar robots que realizan (ejecutar) tareas muy distintas para las cuales se pensaba que los robots solo eran utilizados para el ensamblaje de automóviles y exploración (navegación) de planetas. Las formas y los tamaños son diversos, se pueden encontrar robots grandes y pequeños, en forma de brazo utilizado para de pintura automotriz y en forma de insectos respectivamente (Sagal & Mora, 2018).

**2.1.2 Antecedentes históricos a nivel nacional.** La robótica educativa según (Acuña, 2017), hace su aparición en los años 60, cuando algunos investigadores como Symourt Papert,

Marvin Minsky y Mitchael Resnik, propusieron la construcción de robots didácticos con el apoyo de la empresa LEGO, para que los niños interactuaran con éstos. Cabe mencionar que esta empresa lidera el mercado de robots educativos, cuya variedad va desde robots para uso de niños hasta adultos.

Luego, en los años 80 se popularizaron estos dispositivos, logrando incluirlos dentro de actividades en las escuelas por medio de proyectos importantes que permitieron que el estudiante obtuviera grandes competencias en áreas como matemática, informática, electrónica y mecánica; a la vez que se divierten y juegan al proponer soluciones a problemas específicos (Acuña, 2017).

Por otro lado, el pensamiento computacional goza de gran importancia en nuestros días ya que permite el desarrollo de competencias y como expresa Wing, es una habilidad fundamental para todo el mundo y no sólo para estudiantes y profesionales del área de la computación, ya que es un enfoque que permite resolver problemas de una forma innovadora. El pensamiento computacional genera en el estudiante la capacidad de realizar varios tipos de análisis como el ascendente-descendente, de ser creativos, de generar un pensamiento divergente-abstracto que permita abordar un problema de forma más sencilla y creativa (García & Deco, 2018).

En cuanto al enfoque pedagógico Herrera (2018), dice que la Robótica Educativa se concibe como la manera en la cual los estudiantes se enfrentan a este mundo de creatividad, de solución de problemas y de diversión, puesto que además de ser en el ámbito educativo una materia enriquecedora y llena de saberes, es una manera divertida de aprender, ya que a los estudiantes no solo les gusta comprender de manera teórica sino más bien yendo a la práctica.

La robótica es trabajada a manera grupal, esta se presta mucho a la exploración y comunicación de diferentes sujetos para lograr un objetivo específico. Con el pasar de los tiempos se fue perdiendo el temor a conocer más sobre los robots, ya que parecía ser un tema difícil de aprender, además de costoso, con respecto a los materiales, pero estas creencias se han ido quebrando, porque la robótica ahora es un tema interesante para aprender y comprender (Herrera, 2018).

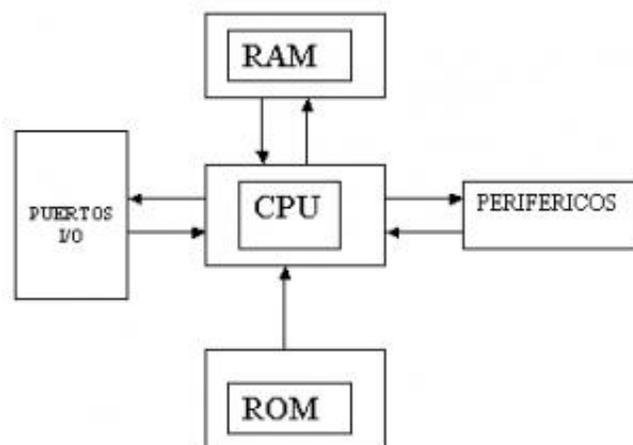
De otra parte, fue utilizada como un auxilio para darle nuevamente vida y generar interés por otras áreas del conocimiento, como lo es la informática que, con ayuda de la robótica y la programación, dio un nuevo sentido a esta, en vista de que los aprendizajes sobre cómo manejar el computador y miles de software era algo cotidiano en las personas, que permanecían a diario varias horas en el mundo del internet, generando habilidades en el manejo de la informática (Herrera, 2018).

A nivel local se debe mencionar que, en cuanto a estudios relacionados con la robótica, en la actualidad no existen, por lo tanto, este sería pionero en la Universidad Francisco de Paula Santander Ocaña.

## **2.2 Marco Conceptual**

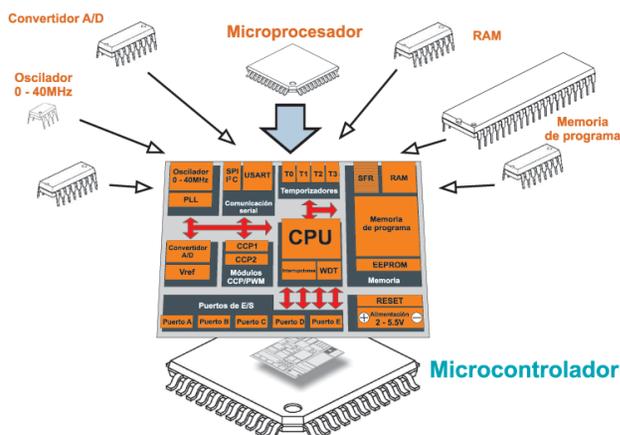
Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y

periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.



**Figura 1.** Circuito integrado. Fuente <https://www.electronicaestudio.com/que-es-un-microcontrolador/>

El procesador o CPU. El centro de procesamiento del computador ya sea de la Gama Intel o AMD, siempre comparando el número de núcleos y la velocidad en GHz. Memoria Ram. Con navegadores como Chrome, siempre necesitarás esos Gbytes altos. Disco Duro. Para almacenar tus series y películas favoritas en 1080p, sin duda se necesitaras 1 terabyte.



**Figura 2.** Microcontrolador. Fuente. <https://devcode.la/articulos/programacion-de-microcontroladores/>

Al igual que un computador, posee dispositivos de entrada (mouse, teclado) y salida (monitor, parlantes) un microcontrolador se basa en este principio contando como entradas a sensores (temperatura, luz, sonido, etc.) y como salidas a los actuadores (leds, motores, relés, etc.).

	Instrucciones (MIPS)	Ciclo de reloj (MHz)	Arquitectura (bits)
Intel Core i7 2600K	117160	3400	64
Raspberry Pi 2	4744	1000	32
ATmega328 (Arduino UNO)	20	16	8
PIC16F877A	5	4	8

**Figura 3.** Elementos de un microprocesador. Fuente. <https://devcode.la/articulos/programacion-de-microcontroladores/>

En la figura anterior los dos primeros elementos de la lista son microprocesadores, y los dos siguientes son microcontroladores. Las diferencias de los millones de instrucciones por segundo que realizan (MIPS) son más que evidentes, los microprocesadores tienen una potencia computacional de más de 200 veces más que los microcontroladores (Devcode, 2018).

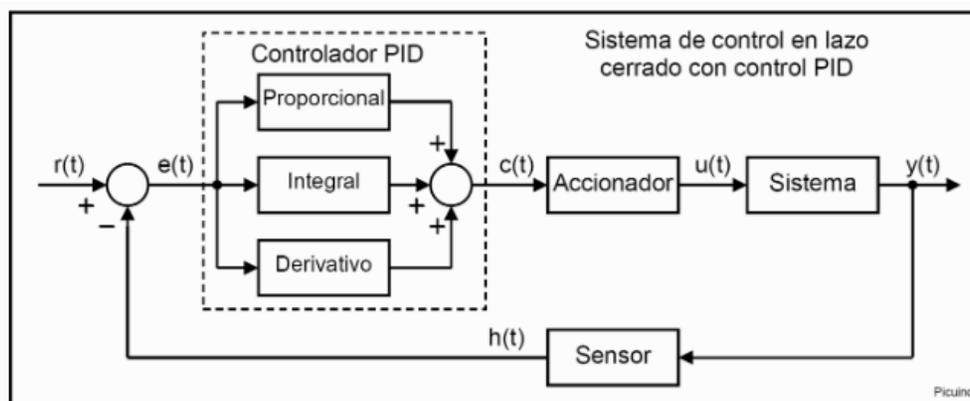
El Matlab es un lenguaje del cálculo técnico, es un entorno de programación para el desarrollo de algoritmos, análisis de datos, visualización y cálculo numérico. Simulink es un entorno gráfico para la simulación y el diseño basado en modelos de sistemas multidominio dinámicos y embebidos (Universidad San Buenaventura, 2019).

Inteligencia Artificial. Es una expresión técnica referida a artefactos empleados para detectar contextos o llevar a cabo acciones en respuesta a contextos detectados. La capacidad de construir dichos artefactos ha aumentado y, con ello, el impacto que tienen en la sociedad. Los

cambios sociales y económicos propiciados por el uso de la IA en particular, pero no exclusivamente en la década transcurrida desde la aparición de los teléfonos inteligentes, han contribuido de manera sustancial a los macrodatos y, por tanto, a la eficacia del aprendizaje de las máquinas (Bryson, 2018).

En cuanto a los sensores este es un concepto genérico que se entiende como las unidades que emite una señal analógica, para múltiples aplicaciones en la industria e investigación una señal binaria. En todos aquellos lugares donde no sea posible detectar magnitudes eléctricas se requiere de los sensores (Iberica.es, 2018).

Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. Estas tres acciones son las que dan nombre al controlador PID (Ogata, 2017).



**Figura 4.** Controlador o regulador PID. Fuente:(Ogata, 2017)

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso (Yúbal, 2018).



*Figura 5.* Arduino. Fuente (Yúbal, 2018)

SolidWorks es un software de diseño CAD 3D (diseño asistido por computadora) para modelar piezas y ensamblajes en 3D y planos en 2D. El software que ofrece un abanico de soluciones para cubrir los aspectos implicados en el proceso de desarrollo del producto. Sus productos ofrecen la posibilidad de crear, diseñar, simular, fabricar, publicar y gestionar los datos del proceso de diseño.

El hardware libre comparte sus modificaciones, y cualquier sugerencia o conocimiento adquirido a través del estudio del código y su funcionamiento. Aunque el concepto de hardware libre es ciertamente menos conocido que su contraparte de software, lo cierto es que se han llevado a cabo proyectos muy interesantes con el paso de los años (Ingenio.com, 2018).



*Figura 6.* Hardware libre. Fuente (Ingenio.com, 2018)

En cuanto a la robótica pedagógica, el cambio de los paradigmas tradicionales en la educación motiva a los docentes a no caer en obsolescencia, para ello es necesario buscar y generar estrategias que les permita innovar su praxis. Una manera de lograrlo es a través del uso de las diferentes herramientas que brindan las nuevas tecnologías de la información (TIC), como recursos didácticos, surgiendo lo que en la actualidad se conoce como Robótica Pedagógica, la cual es pilar de esta Experiencia Pedagógica (Arias, 2016).

Por otra parte, es bien sabido por todos que las nuevas generaciones muestran interés por actividades donde tengan la oportunidad de interactuar con objetos y de elaborar diversos prototipos, de allí la inclinación e interés que manifiestan por todas aquellas temáticas en las que pueden ser sujetos activos y a su vez liderar su conocimiento, cobrando aquí gran importancia la Robótica Pedagógica, al permitir que el educando, construya su propio conocimiento y aprenda creando de manera lúdica (Arias, 2016).

### 2.3 Marco Teórico

La robótica es la ciencia encaminada a diseñar y construir aparatos y sistemas capaces de realizar tareas propias de un ser humano.

En la actualidad la robótica automatizada en la sociedad se interesa en aspectos tanto teóricos como prácticos de la automatización y la robótica, a través de máquinas inteligentes (inteligencia artificial) y de sistemas que permite ofrecer servicios a los seres humano, a la industria e inclusive para exploración espacial (Asimov, 2019).

De otra parte, los robots móviles que operan en grandes ambientes no estructurados deben enfrentarse con significativas incertidumbres en la posición e identificación de objetos (Calderón, Ceres, & Sánchez, 2018).

Los diferentes índices de operación (medidos por la incertidumbre y precisión requeridas) de los robots móviles en relación con los manipuladores industriales se debe a la existencia de un conjunto diferente de prioridades en investigación. Las prioridades para los robots móviles están firmemente orientadas en las áreas de sensado y raciocinio.

El robot móvil autónomo se caracteriza por una conexión inteligente entre las operaciones de percepción y acción, que define su comportamiento y le permite llegar a la consecución de los objetivos programados sobre entornos con cierta incertidumbre. El grado de autonomía depende en gran medida de la facultad del robot para abstraer el entorno y convertir la información

obtenida en órdenes, de tal modo que, aplicadas sobre los actuadores del sistema de locomoción, garantice la realización eficaz de su tarea (Borenstein, 2014).

**Percepción:** El robot móvil debe ser capaz de determinar la relación con su entorno de trabajo, mediante el sistema sensorial a bordo. La capacidad de percepción del robot móvil se traduce en la síntesis de toda la información provista por los sensores, con el objeto de generar mapas globales y locales del entorno de acuerdo con los diversos niveles de control.

**Razonamiento:** El robot móvil debe ser capaz de decidir qué acciones son requeridas en cada momento, según el estado del robot y el de su entorno, para alcanzar su(s) objetivo(s). La capacidad de razonamiento del robot móvil se traduce en la planificación de trayectorias globales seguras y en la habilidad para modificarlas en presencia de obstáculos inesperados (control local de trayectoria) para permitirle, al robot, la consecución de los objetivos encomendados (Borenstein, 2014).

**Generador Global de Trayectorias (GGT):** Es el nivel jerárquico superior. Este nivel es el encargado de decidir, en base a la tarea asignada, las coordenadas del punto destino, de puntos intermedios en la trayectoria y, en caso de que el camino esté obstruido, redefinir la trayectoria elegida. La información que emplea este nivel jerárquico puede ser generada fuera de línea (conocimiento a priori del ambiente de trabajo) o en línea, en base a criterios predefinidos y utilizando la información provista por el sistema sensorial (desconocimiento parcial o total del ambiente de trabajo) a partir de la elaboración de mapas del entorno (SLAM).

Generador Local de Trayectorias (GLT): Es el nivel jerárquico intermedio. Este nivel jerárquico hace las veces de operador (piloto) del robot móvil, evitando los obstáculos del camino, realizando correcciones en la trayectoria y adecuando la velocidad del vehículo de acuerdo a la maniobra que se realiza.

Permite un control dinámico del robot móvil. Mantiene informado al GGT sobre los resultados del objetivo asignado, y en caso de no tener un conocimiento a priori del ambiente de trabajo, genera información para ser almacenada en la memoria del GGT.

Está directamente comunicado con el sistema sensorial, lo que le permite tomar decisiones en línea y además genera los valores de referencia para el Control Local del Sistema de Tracción y Dirección. Se han desarrollado GLT tanto con algoritmos clásicos del tipo Maze-Search como con la utilización de elementos de la Inteligencia Artificial que emulan el comportamiento del operador humano (Calderón, Ceres, & Sánchez, 2018).

## **2.4 Marco legal**

**Propiedad intelectual en Colombia.** La Propiedad Intelectual es la denominación asignada a la protección jurídica del Estado sobre bienes inmateriales específicos; estos últimos hacen referencia a toda creación del talento o del ingenio humano, dentro del ámbito científico, artístico, literario, industrial o comercial. Así, mediante la regulación de diversos instrumentos como la Constitución Política, leyes, decretos y tratados internacionales se procura la defensa del intelecto (Ministerio de Tecnologías de la Información y las Comunicaciones, 2015)

**Constitución Política de Colombia.** La Constitución Política de 1991, en su artículo 61 dispone: “El Estado protegerá la propiedad intelectual por el tiempo y mediante las formalidades que establezca la ley”.

Es debido aclarar que, si bien es una protección jurídica, las demás disposiciones concernientes al tema no realizan una denominación conceptual y literal de la Propiedad Intelectual, sino que hacen referencia a los dos tipos de protección que la componen, a saber: los Derechos de Autor y la Propiedad Industrial.

A continuación, se adelanta la descripción de las disposiciones jurídicas dictadas en la materia, las cuales aplican para todo el territorio nacional, siempre teniendo claro que estas se sustentan en disposiciones internacionales que intentan una protección similar, propendiendo por la cooperación de los diversos países en la determinación de una aplicación uniforme (República de Colombia, Constitución Política de Colombia, 2012)

**Ley 23 de 1982.** Constituida como la ley “sobre Derechos de Autor”, los sujetos protegidos por dicho cuerpo normativo serán los autores de obras literarias, científicas y artísticas, los cuales gozarán de protección para sus obras en la forma prescrita en esta. En adición, comprende a los intérpretes o ejecutantes, a los productores de programas y a los organismos de radiodifusión, en sus derechos conexos a los del autor; los causahabientes, a título singular o universal, de los titulares, anteriormente citados; y a la persona natural o jurídica que, en virtud de contrato obtenga por su cuenta y riesgo, la producción de una obra científica,

literaria o artística realizada por uno o varios autores en las condiciones previstas en el artículo 20 de esta Ley (República de Colombia, Ley 23 de 1982, 2011)

En el artículo 3, los derechos de autor comprenden para sus titulares las facultades exclusivas:

De disponer de su obra a título gratuito u oneroso bajo las condiciones lícitas que su libre criterio les dicte. De aprovecharla, con fines de lucro o sin él, por medio de la imprenta, grabado, copias, molde, fonograma, fotografía, película cinematografía, ideograma, y por la ejecución, recitación, representación, traducción, adaptación, exhibición, transmisión, o cualquier otro medio de reproducción, multiplicación, o difusión conocida o por conocer (República de Colombia, Ley 23 de 1982, 2011)

De ejercer las prerrogativas, aseguradas por esta Ley en defensa de su "derecho moral".

Siguiendo con la protección del autor, según el artículo 21, ésta perdura durante toda su vida, más 80 años después de su muerte. En caso de ser una obra literaria, científica o artística cuyo titular sea una persona jurídica o una entidad oficial o cualquier institución de derecho público, se considerará que el plazo de protección será de 30 años contados a partir de su publicación (artículo 27) (República de Colombia, Ley 23 de 1982, 2011)

Según el artículo 29 (modificado por el artículo 2 de la ley 44 de 1993), la protección consagrada por la presente ley a favor de los artistas, intérpretes y ejecutantes, de los productores de fonogramas y de los organismos de radiodifusión, será de ochenta años a partir de la muerte del respectivo titular, si este fuere persona natural; si el titular fuere persona jurídica, el término

será de treinta años a partir de la fecha en que tuvo lugar la interpretación o ejecución o la primera fijación del fonograma, o la emisión de la radiodifusión (Congreso de Colombia, ley 44 de 1993, 2000)

Por otra parte, de vital importancia resulta la subdivisión que realiza la ley en cuanto al contenido del derecho de autor, pues comprende tanto los derechos morales como los patrimoniales.

En primer lugar, los derechos morales consisten en el reconocimiento de la paternidad del autor sobre la obra realizada y el respeto a la integridad de esta. En el artículo 30, el autor tendrá sobre su obra un derecho perpetuo, inalienable, e irrenunciable para: (Congreso de Colombia, ley 44 de 1993, 2000)

Reivindicar en todo tiempo la paternidad de su obra y, en especial, para que se indique su nombre seudónimo cuando se realice cualquiera de los actos mencionados en el artículo 12 de esta Ley.

A oponerse a toda deformación, mutilación u otra modificación de la obra, cuando tales actos puedan causar o acusen perjuicio a su honor o a su reputación, o la obra se demerite, y a pedir reparación por esto;

A conservar su obra inédita o anónima hasta su fallecimiento, o después de él cuando así lo ordenase por disposición testamentaria;

A modificarla, antes o después de su publicación;

A retirarla de la circulación o suspender cualquier forma de utilización, aunque ella hubiere sido previamente autorizada (Congreso de Colombia, ley 44 de 1993, 2000)

En todos los casos en que los autores o las asociaciones de autores, celebren contratos con los usuarios o con las organizaciones que los representen, respecto al derecho de autor, por concepto de ejecución, representación, exhibición y en general, por uso o explotación de las obras protegidas por la presente Ley, serán las tarifas concertadas en los respectivos contratos, las que tendrán aplicación, siempre que no sean contrarias a los principios consagrados por la misma (artículo 73) (Congreso de Colombia, ley 44 de 1993, 2000)

Así mismo, el artículo 78 es claro en disponer que la interpretación de los negocios jurídicos sobre derechos de autor sea siempre restrictiva. En este sentido, no se admite el reconocimiento de derechos más amplios de los expresamente concedidos por el autor en el instrumento respectivo (Congreso de Colombia, ley 44 de 1993, 2000)

Respecto a este tema, el artículo 20, modificado por el art. 28 de la Ley 1450 de 2015, disponía: “cuando uno o varios autores, mediante contrato de servicios, elaboren una obra según plan señalado por persona natural o jurídica y por cuenta y riesgo de ésta, solo percibirán, en la ejecución de ese plan, los honorarios pactados en el respectivo contrato. Por este solo acto, se entiende que el autor o autores transfieren los derechos sobre la obra, pero conservarán las prerrogativas consagradas en el artículo 30 de la presente Ley, en sus literales a), y b)” (Congreso de Colombia, Ley 1450 de 2015, 2015)

Actualmente, con la modificación mencionada anteriormente, el artículo 20 relativo a propiedad intelectual en cumplimiento de un contrato de prestación de servicios o de un contrato de trabajo, en su nueva redacción dispone: “En las obras creadas para una persona natural o jurídica en cumplimiento de un contrato de prestación de servicios o de un contrato de trabajo, el autor es el titular originario de los derechos patrimoniales y morales; pero se presume, salvo pacto en contrario, que los derechos patrimoniales sobre la obra han sido transferidos al encargado o al empleador, según sea el caso, en la medida necesaria para el ejercicio de sus actividades habituales en la época de creación de la obra (Congreso de Colombia, Ley 1450 de 2015, 2015)

Este tema ya había sido objeto de aclaración por vía de jurisprudencia por parte de la Corte Constitucional, al decidir la asequibilidad del artículo 20 de la Ley 23 de 1982, donde el alto tribunal se pronunció en el sentido en que, para que opere la presunción señalada, se requiere que el contrato conste por escrito. Adicionalmente, si él “en encargado” o empleador quiere disponer de los derechos sobre la obra para otro tipo de actividades, más allá del giro ordinario de sus negocios, deberá obtener una autorización adicional del autor y, si es del caso, pagarle una remuneración adicional por concepto de derechos de autor” (Congreso de Colombia, Ley 1450 de 2015, 2015)

Debe hacerse claridad en que las distintas formas de utilización de una obra son independientes entre sí, pues pueden presentarse diversas formas, tales como: la reproducción, traducción, adaptación, etc. Sobre estas formas, el autor o titular de los derechos patrimoniales puede cederlos, lo cual implica entregarlos a otra persona, o en su defecto autorizar su uso,

donde si bien se da un permiso, no se ceden los derechos, y se consideran aspectos como el tiempo, la cobertura y la retribución económica (Congreso de Colombia, Ley 1450 de 2015, 2015)

**Ley 44 de 1993.** Esta ley, “por la cual se modifica y adiciona la Ley 23 de 1982 y se modifica la Ley 29 de 1944”, realiza cambios importantes en la ley de Derechos de Autor como: El artículo 2, modifica el artículo 29 de la Ley 23 de 1982, ampliando el termino de los derechos conexos consagrados cuando el titular sea persona jurídica, pasando de un término de protección de 30 a 50 años, contados a partir del último día del año en que se tuvo lugar la interpretación o ejecución, la primera publicación del fonograma o, de no ser publicado, de su primera fijación, o la emisión de su radiodifusión. (Congreso de Colombia, Ley 44 de 1993, 2011)

## **Capítulo 3. Diseño metodológico**

### **3.1 Tipo de investigación.**

Para el desarrollo de la investigación, se implementará la metodología propuesta por (Hernandez, Fernandez, & Baptista, 2010), la cual, a través de los procesos planteados con enfoque cualitativos y cuantitativos, siendo una metodología de investigación mixta, permite diseñar y construir un kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino para la materia de Robótica de la Universidad Francisco de Paula Santander Ocaña.

De otra parte, la metodología de investigación con enfoque experimental, ya que el investigador tiene la oportunidad de manipular una o más variables de estudio, para controlar el aumento o disminución de esas variables y su efecto en las conductas observadas. Dicho de otra forma, un experimento consiste en hacer un cambio en el valor de una variable (variable independiente) y observar su efecto en otra variable (variable dependiente). Esto se lleva a cabo en condiciones rigurosamente controladas, con el fin de describir de qué modo o por qué causa se produce una situación o acontecimiento particular. Los métodos experimentales son los adecuados para poner a prueba hipótesis de relaciones causales (Murillo, 2018).

### **3.2 Población y muestra**

La población beneficiada será en su totalidad los estudiantes de la asignatura de robótica en la Universidad Francisco de Paula Santander Ocaña. La muestra se conforma por los estudiantes de ingeniería de sistemas que cursaron la asignatura de robótica.

## Capítulo 4. Resultados

Para poder llevar a cabo el diseño y construcción del kit robótico fue necesario realizar un diagnóstico en relación con los aportes que dicho kit generaría con respecto a la asignatura de robótica de la universidad Francisco de Paula Santander Ocaña. En este sentido, se identificaron las diferentes unidades temáticas que comprenden la asignatura de robótica y posteriormente se aplicó una encuesta a un grupo de estudiantes del programa de ingeniería de sistemas de la universidad, con el fin de establecer el punto de partida en el que se encuentran los estudiantes de la asignatura, en relación a los conocimientos que estos poseen con respecto a la robótica y las prácticas que pudieron llevar a cabo con las herramientas y tiempo disponibles de la asignatura.

### **Contenido programático**

Como punto de partida se realizó una revisión del contenido programático de la asignatura de robótica la cual tiene como objetivo “familiarizar al estudiante con la programación, diseño, y construcción de robots, de tal forma que pueda decidir de manera autónoma su comportamiento, permitiendo incorporarlos en sistemas de producción”.

La revisión de las diferentes unidades temáticas de la asignatura ayudó a identificar los requerimientos que regirían el diseño y construcción de este kit. Entre las unidades temáticas que se consideraron podemos resaltar las siguientes:

- Fundamentos de Robótica
- Control de los robots
- Programación de un robot

## **Encuesta**

Se realizó una encuesta a un grupo de estudiantes que cursaron la asignatura de robótica, en donde se tuvo en cuenta varios factores con el fin de realizar un diagnóstico que permitiera identificar las potencialidades y falencias que presentan los estudiantes entorno al aprendizaje de la asignatura robótica, en este sentido se relacionan dichos factores a continuación y los resultados obtenidos:

### **-Habilidades adquiridas entorno a la robótica**

Se tuvo en cuenta la pregunta ¿considera que, al culminar la asignatura de robótica, ha adquirido las habilidades en construcción, diseño y programación de robots?

En este apartado se pudo evidenciar que un 60% de estudiantes encuestados no cuentan con las suficientes habilidades en construcción, diseño y programación de robots al culminar la asignatura y que solo el 40% considera que cuenta con dichas habilidades, en este sentido más del 50% del grupo de estudiantes encuestados poseen falencias respecto a la construcción, diseño y programación de robots.

### **-Importancia de Arduino y electrónica en las organizaciones**

Se tuvo en cuenta la pregunta ¿considera que los conocimientos que posee en la plataforma Arduino y electrónica en general pueden ser incorporados a las organizaciones?

Un 90% considera que es importante para las organizaciones los conocimientos de Arduino y electrónica general. En este sentido es importante desarrollar de manera adecuada el aprendizaje entorno a dichos factores.

### **-Conocimiento de elementos y características de un robot**

Se tuvo en cuenta la pregunta ¿Usted identifica correctamente los elementos y características de un sistema robótico?

Un 70% considera que no identifica correctamente los elementos de un sistema robótico. En este sentido es importante contar con herramientas de aprendizaje que permitan fácilmente identificar los componentes de un robot y su funcionamiento.

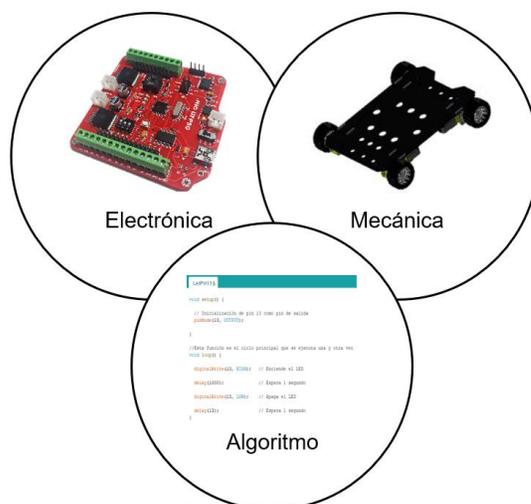
### **-Competencias en software y hardware**

Se tuvo en cuenta la pregunta ¿Usted posee competencias en el desarrollo de aplicativos de software y hardware orientados a proyectos robóticos?

Un 80% considera que no posee competencias en el desarrollo de aplicativos de software y hardware orientado a proyectos. Por consiguiente, es necesario contar con herramientas de aprendizaje que faciliten la adquisición de competencias en torno al desarrollo de hardware y software orientado a la robótica.

Posterior a la revisión del contenido temático de la asignatura y a la identificación de falencias en los estudiantes, se planteó el desarrollo de un kit de aprendizaje para el ensamblaje y programación de un robot móvil autónomo basado en la plataforma de Arduino con el fin de servir como apoyo al docente en los procesos de aprendizaje de manera que los estudiantes puedan adquirir competencias relacionadas a esta área durante las prácticas y laboratorios de la asignatura.

Se identificaron 3 componentes o áreas importantes que conformarían el kit:



**Figura 7.** Sistemas kit robótico. Fuente. Autores del proyecto

En primer lugar, el sistema electrónico representado principalmente por una tarjeta para procesamiento de señales y control de actuadores, esto con el fin de permitir al estudiante identificar fácilmente diferentes componentes electrónicos y las etapas o sistemas comprendidos en la unidad temática “Fundamentos de robótica” tales como: sistema de control, sistema de sensores, elementos terminales, entre otros.

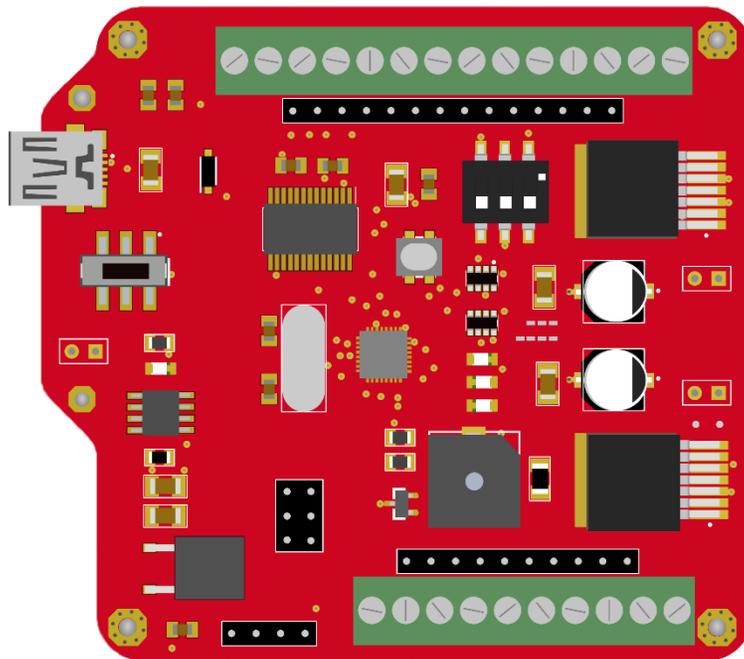
En segundo lugar, una estructura mecánica que posibilitara la construcción de robots móviles que permitieran un entendimiento práctico de la cinemática, control y programación de un robot. Puntos abarcados en las unidades temáticas: cinemática de los robots, control de los robots y programación de un robot.

En tercer y último lugar, se abarcó la estructura algorítmica en la cual se tomó como base la plataforma Arduino con el fin de permitir al estudiante hacer uso de la extensa documentación, foros y tutoriales existentes para este lenguaje. En este apartado se buscó inducir al estudiante en el desarrollo de algoritmos para robot móviles, brindando diferentes ejemplos de funcionamiento

para componentes de la estructura electrónica. Además de ello se implementó un algoritmo de control aplicado a un robot seguidor de línea con el objetivo de soportar el aprendizaje de la unidad temática “programación de un robot”.

#### **4.1 Diseño y construcción de un circuito electrónico que permite leer diferentes señales analógicas y digitales**

Al momento de diseñar la placa y escoger los componentes electrónicos necesarios para su funcionamiento, se buscó mantener un equilibrio entre tamaño, prestaciones y comodidad en lo referente a su uso.



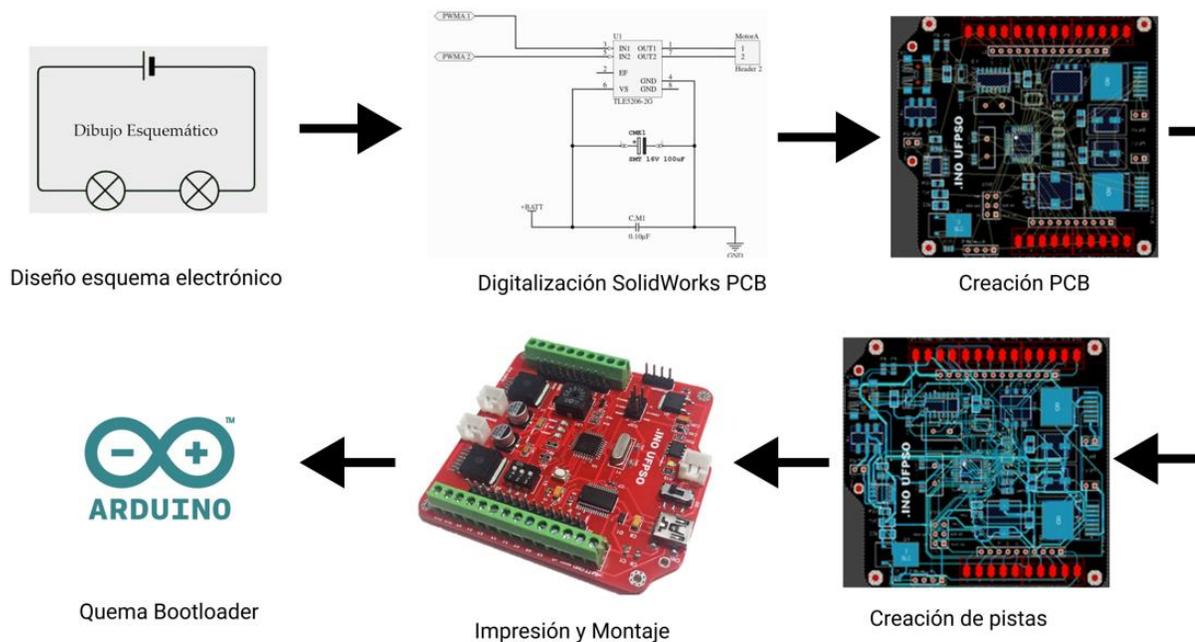
*Figura 8.* UFPSO.INO. Fuente. Autores del proyecto

Se optó en su mayoría por componentes superficiales (SMD) por las siguientes razones:

- Minimizar los costos de fabricación.
- Reducir el peso y las dimensiones de la placa.
- Obtener un diseño mucho más compacto al integrar ambas caras del circuito impreso.
- Conseguir valores mucho más precisos en el caso de componentes pasivos como resistencias y condensadores
- Permitir una mayor automatización en el proceso de fabricación de las placas.

Además de lo anterior se seleccionó cuidadosamente cada uno de los componentes buscando maximizar las prestaciones de la placa electrónica sin que su uso requiera de conocimientos avanzados en el tema.

Para la construcción de la placa se siguió la siguiente ruta:

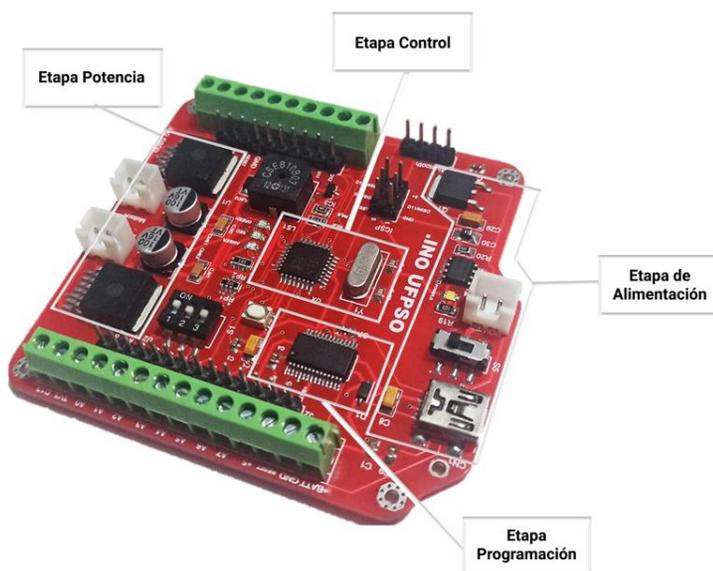


**Figura 9.** Pasos construcción. Fuente. Autores del proyecto

### 4.1.1 Diseño Esquemas electrónicos

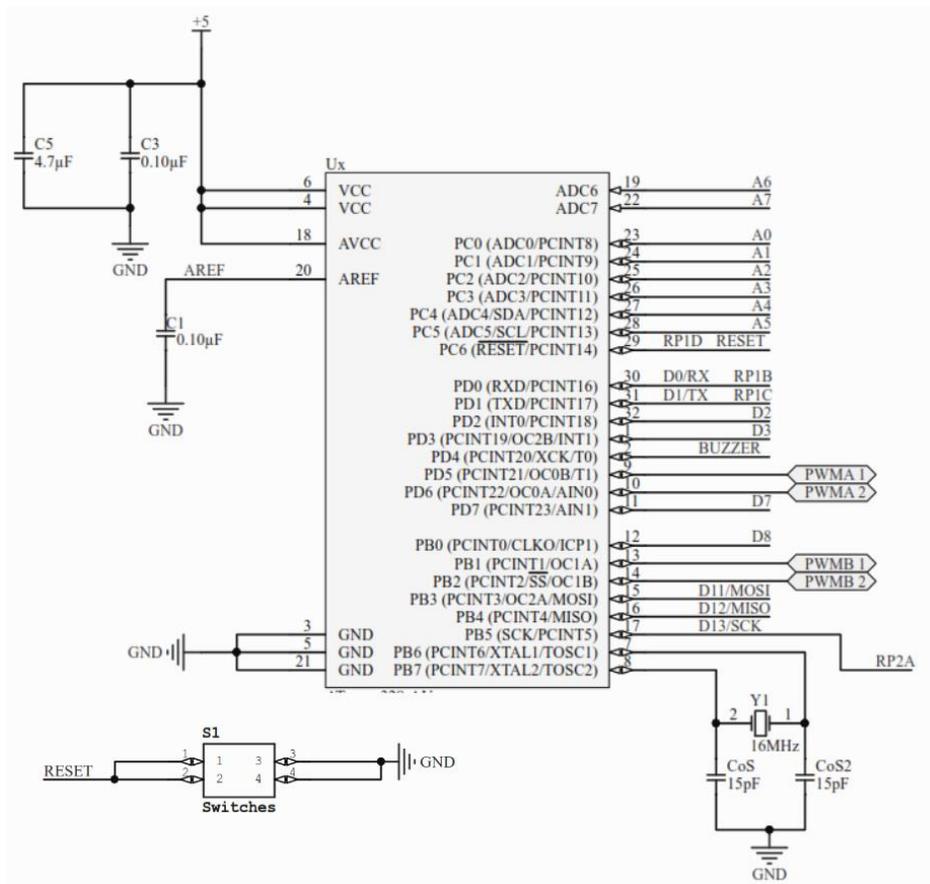
En esta primera fase de creación se realizó un análisis de los diferentes componentes necesarios para el procesamiento de señales y la creación de un sistema robótico. Posteriormente se agruparon en etapas según su funcionamiento principal, para luego diseñar los esquemáticos electrónicos que comprenden los componentes y conexiones necesarias entre ellos.

Según Tapiero Santa, (2019), entre las etapas involucradas en un circuito para el procesamiento de señales y control de un robot se encuentran: la etapa de control encargada del procesamiento de información y la toma de decisiones, la etapa de alimentación encargada de energizar el sistema, la etapa sensorial correspondiente a la percepción del entorno, la etapa de potencia encargada de suministrar energía a los motores, y por último una etapa de programación necesaria para la carga de código al microcontrolador.



**Figura 10.** Etapas de la placa electrónica. Fuente. Autores del proyecto

### 4.1.1.1 Etapa de control.

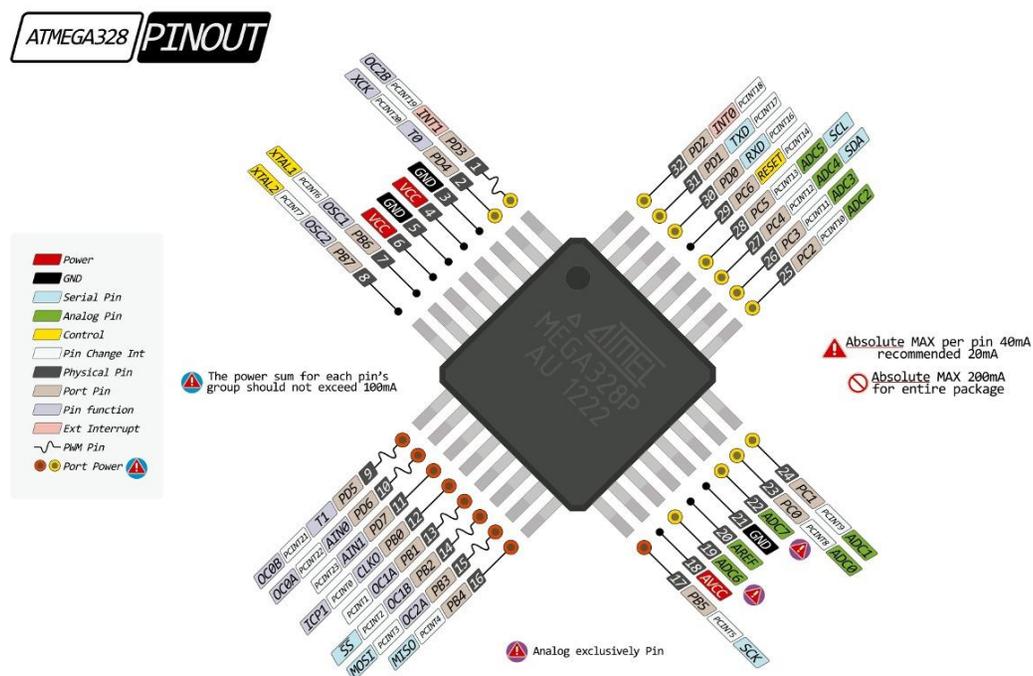


**Figura 11.** Etapa de control. Fuente. Autores del proyecto

Esta etapa abarca los elementos necesarios para que la placa tenga la capacidad de recibir, procesar y emitir señales.

En este apartado el principal elemento fue el microcontrolador, el cual necesita de algunos elementos adicionales para funcionar correctamente.

- Microcontrolador Atmega 328p AU



**Figura 12.** Microcontrolador Atmega328 AU. Fuente. <https://naylampmechatronics.com/atmel/262-atmega328p-smd.html>

El microcontrolador escogido para el diseño de esta placa fue el Atmega328p AU, caracterizado por ser pequeño, compacto y además poseer pines que sobresalen en los laterales del micro, lo que facilita en gran medida su correcto ensamblaje en la placa.

Este micro en particular cuenta con las siguientes especificaciones técnicas lo que lo convierte en una opción idónea para las prestaciones finales de la placa.

## Especificaciones técnicas

- Chip: ATMEGA328P-AU
- Formato SMD: TQFP
- Pines: 32
- Memoria FLASH: 32KB
- Memoria RAM: 2KB
- Máxima frecuencia de funcionamiento: 20Mhz
- CPU: 8-bit AVR
- Pines de entrada/salida: 23
- Entradas Analógicas (ADC): 8

Este componente es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

- **Oscilador**

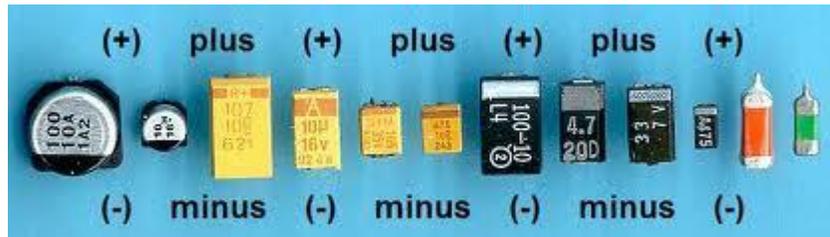
Para controlar la velocidad de trabajo del microcontrolador se utiliza un oscilador de frecuencia de 16Mhz (Y1), adicionalmente lo acompañan dos condensadores (CoS y CoS2) de 15pf conectados a GND, es recomendable que el valor de estos condensadores esté entre 15pf y 22pf.



**Figura 13.** Oscilador. Fuente: <https://sandorobotics.com/producto/hr0525-16/>

- **Condensadores**

Los capacitores C1, C3 Y C5, se utilizan para reducir el ruido, estabilizar el voltaje y prevenir picos, los de menor valor ayudan a filtrar las frecuencias altas, y los de mayor valor, las frecuencias bajas



**Figura 14.** Condensadores. Fuente: <https://www.arrow.com/es-mx/research-and-events/articles/the-ups-and-downs-of-tantalum-capacitors>

Un capacitor o también conocido como condensador es un dispositivo capaz de almacenar energía a través de campos eléctricos (uno positivo y uno negativo). Este se clasifica dentro de los componentes pasivos ya que no tiene la capacidad de amplificar o cortar el flujo eléctrico.

- **Pulsador s1**

Este pulsador nos permite reiniciar el microcontrolador de forma manual, al presionarlo cierra el circuito entre el pin RESET y GND y reinicia la placa.

Cabe aclarar que realizar un reset, no se afecta la programación de la placa, el código que se le haya cargado anteriormente permanecerá, simplemente se generará un reinicio de la placa, teniendo que volver a ejecutar el código desde su primera línea.

#### 4.1.1.2 Etapa de Programación.

Para la programación del microcontrolador se utilizó el chip ft232rl que permite realizar la conversión entre la interfaz serie del MCU a una interfaz USB con la cual se podrá comunicar con el pc. La interfaz USB en el ordenador, permite tener un puerto serie virtual que será usado para establecer la conexión.

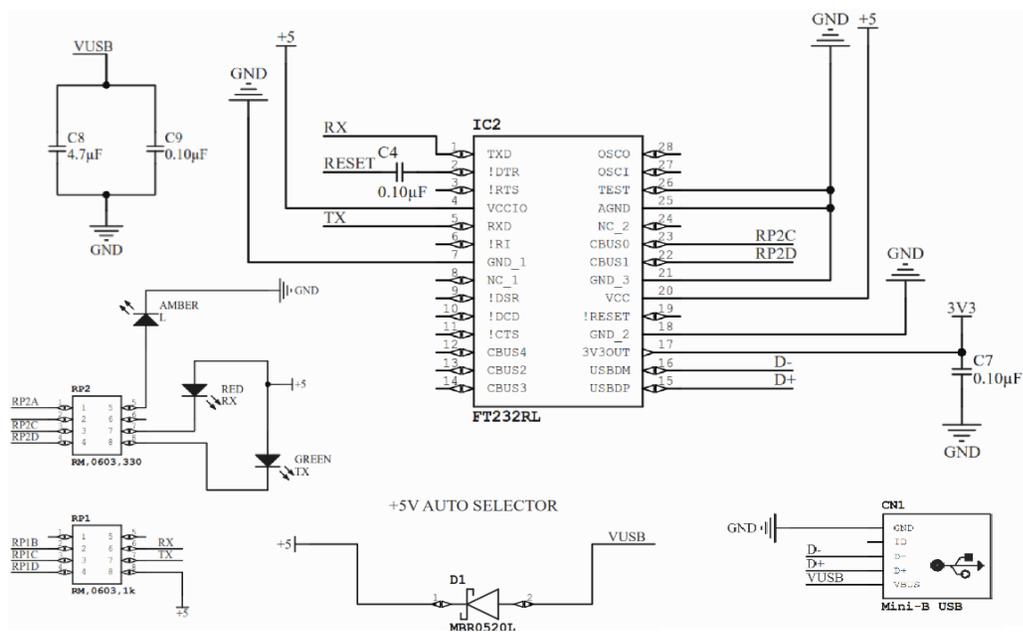


Figura 15. Programador. Fuente. Autores del proyecto.



*Figura 16.* FT232RL. Fuente. <https://ftdichip.com/products/ft232rl/>

- **Capacitores**

Los capacitores C8, C9 y C7 actúan como capacitores de desacoplo, ayudando a limpiar la señal de corriente continua al permitir que el ruido desemboque en GND.

El capacitor C4 ubicado entre el pin DTR y el pin RESET del MCU, permite que el código cargue correctamente al alternar los estados del pin reset entre bajo y alto por la acción de la resistencia pull-up (RP1-4) conectada entre reset y 5v, la cual se encargará de cargar lentamente el condensador y de esta forma cambiar el estado del pin de bajo a alto.

Otra forma de cargar el código sin necesidad de conectar el pin reset al DTR por medio del capacitor, sería hacer el reset de forma manual a través del pulsador (s1) en el momento preciso en el que se está cargando el código.

- **Resistencias**

En esta área del circuito se cuenta con dos resistencias en matriz cuya función será regular el paso de la corriente a niveles seguros para los componentes conectados a ellas, tales como leds y pines del microcontrolador.

La ventaja de utilizar resistencias en matriz es el ahorro de espacio, puesto que en una pequeña matriz se encuentran 4 resistencias que es posible utilizar por separado.

**RP2:** Se encuentra conectada a ella el led del pin 13 del MCU, el cual será muy utilizado para realizar pruebas de código sin necesidad de conectar un led externo a la placa.

También se conectó a esta resistencia dos leds que estarán accionados por los pines 22 y 23 del chip FTDI, los cuales mostrarán el comportamiento de los pines RX y TX, pudiendo verificar en tiempo real que el código se está transmitiendo correctamente al microcontrolador.

**RP1:** Dos de ellas se utilizaron como protección para los pines RX y TX del micro, y la tercera como una resistencia Pull-up necesaria para la acción del reset.

- **Conector Mini USB**

Este conector se utilizó para comunicar la placa y el computador a través de un cable de datos. A la salida del conector se encuentra 4 pines: dos encargados de la alimentación de la placa y dos para la transmisión de datos, los cuales van a los pines 15 y 16 del chip FTDI que realiza la respectiva conversión entre las interfaces.

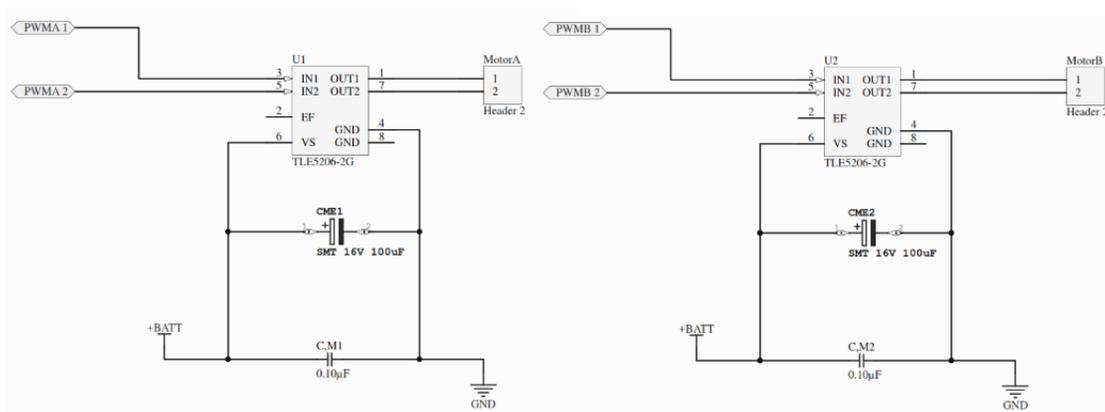
La razón por la cual se utilizó un conector Mini USB en lugar de uno Micro USB fue por su robustez y tolerancia a movimientos incorrectos del usuario al momento de la conexión del cable, además de esto su ensamble a la placa supone menos dificultades.

- **Diodo rectificador Schottky**

Este diodo de protección tiene diferentes funcionalidades, una de ellas es permitir únicamente el paso de la corriente desde el conector hacia el circuito y no en sentido contrario, esto protege el puerto del computador en caso de que se haya conectado una batería adicional al circuito.

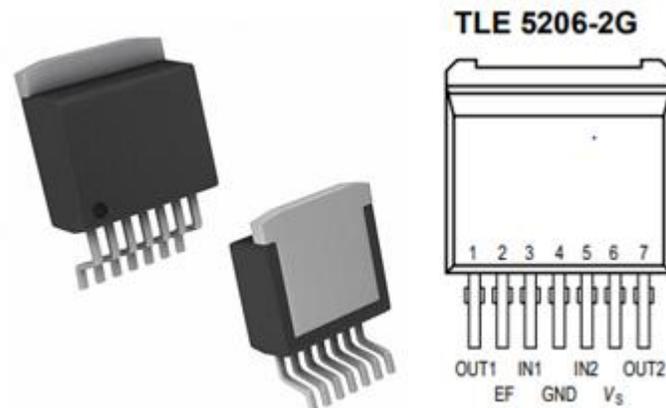
Otra función es la de rectificar la corriente en caso de ruidos, además de ser altamente eficiente por su baja caída de tensión, la cual no supera los 0,45v.

#### 4.1.1.3 Etapa de potencia.



**Figura 17.** Etapa de potencia. Fuente. Autores del proyecto.

- **Driver de Motor TLE5206-2G**



**Figura 18.** Driver de motor. Fuente. <http://www.datasheetdir.com/TLE5206-2G+DC-Motor-Drivers-Controllers>

Se utilizó este controlador debido a sus excelentes prestaciones cuyo uso puede abarcar un sin fin de aplicaciones en robótica. La robustez de este controlador permite al usuario conectar motores de diferentes potencias abarcando un espectro bastante amplio de estos y potenciando en gran medida la utilidad de la placa en proyectos con requerimientos exigentes.

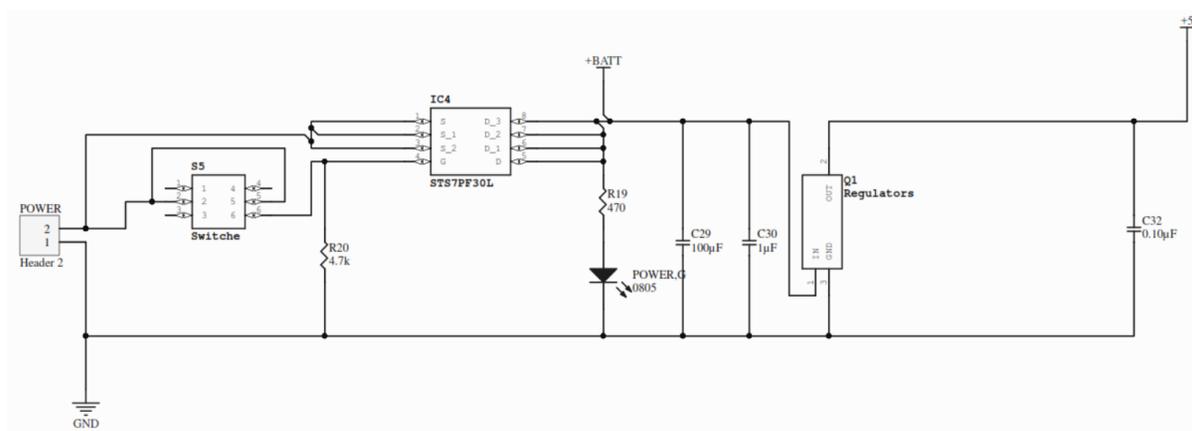
Algunas características que caben resaltar son:

- Permite cambiar el giro del motor
- Su salida de corriente puede alcanzar los 5A de corriente continua
- Permite un voltaje de entrada de hasta 40V
- Protección contra cortocircuito en su salida

- **Capacitores**

Los capacitores CM1, CME1, CM2 y CME2 actúan como filtro permitiendo que el voltaje de entrada sea más estable

#### 4.1.1.4 Etapa de alimentación.



*Figura 19.* Etapa de alimentación. Fuente. Autores del proyecto.

- **Switch**

En este apartado se cuenta en primer lugar con un Switch cuya función es permitir el paso de la corriente, en otras palabras, un interruptor de encendido y apagado.

- **Regulador 78M05**

Se incorporó también un regulador con una salida de 5v para alimentar el microcontrolador, el programador y algunos componentes que requieren este voltaje.

Este regulador cuenta con las siguientes características:

- Voltaje de entrada: De 8V a 40V
- Voltaje de Salida: De 4.75V a 5.2V
- Corriente Máxima de Salida: 500mA

Se puede concluir que la placa soportará una entrada de Voltaje en los pines POWER de hasta 40V, voltaje que podrá ser utilizado directamente para alimentar los motores a través de los drivers TLE5206-2G y que serán regulados a 5v por el 78M05 para alimentar el MCU.

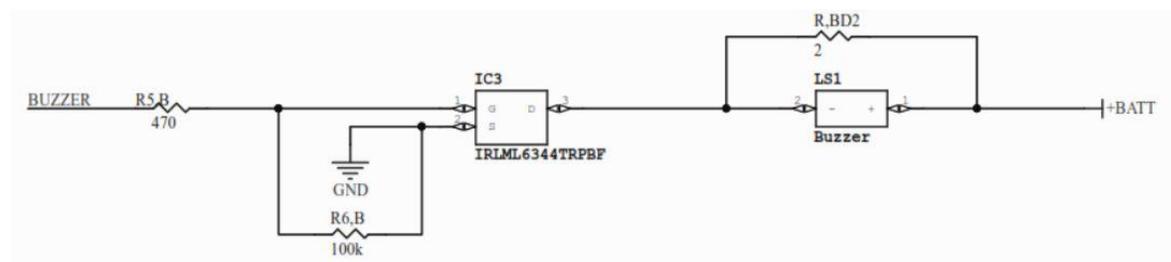
- **Capacitores**

Se encuentran 3 condensadores de desacoplo C29, C30 y C32, encargados de filtrar señales de ruido y estabilizar el voltaje.

- **Led**

Se añadió un Led emisor de luz que se encenderá cuando se conecte una batería a la placa, acompañado a este hay una resistencia de protección que evitará que el led se quemé.

### 1.1.1.5 Buzzer



**Figura 20.** Buzzer. Fuente. Autores del proyecto.

El buzzer piezoeléctrico podrá ser utilizado al poner en alto el pin digital D4, este al igual que el led en el pin 13 se incorporan a la placa con el fin de que el estudiante no necesite de hardware externo sino lo único que necesita es un indicador auditivo o visual en su proyecto.

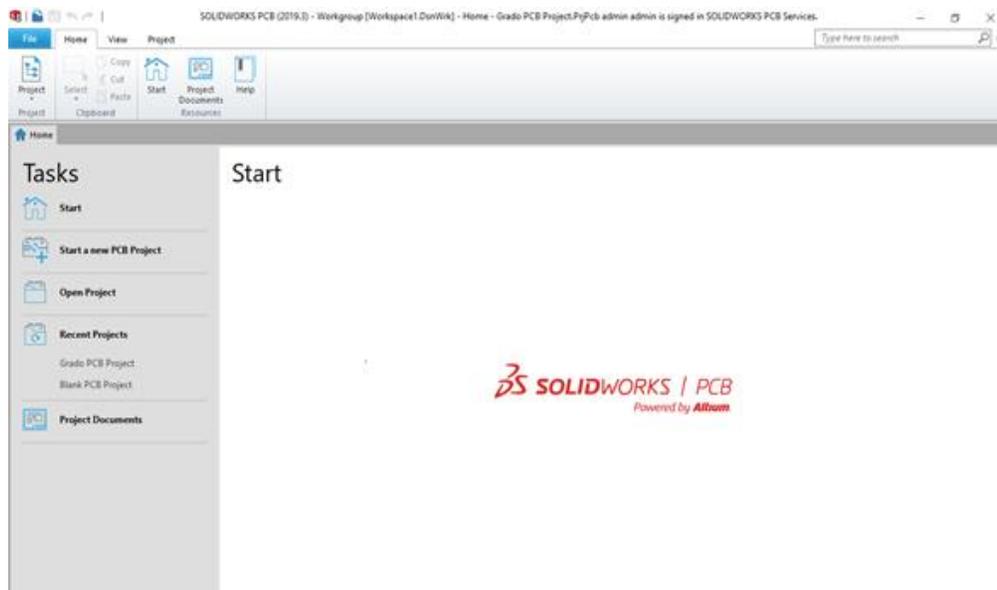
Para el funcionamiento del buzzer se utiliza en primer lugar un transistor MOSFET a modo de interruptor que habilita la alimentación de este cuando ponemos en alto el pin D4, el buzzer es alimentado directamente con el voltaje de la batería principal.

R5B y R6B son opcionales en este caso, puede conectar el transistor directamente, pero proporcionan cierta medida de protección contra conexiones de falla abiertas y en cortocircuito.

RBD2 actúa como un diodo de protección, permitiendo que la corriente circule por el diodo cuando se desactiva el buzzer y de esta forma se eviten picos de voltaje que pueden dañar otros componentes del circuito.

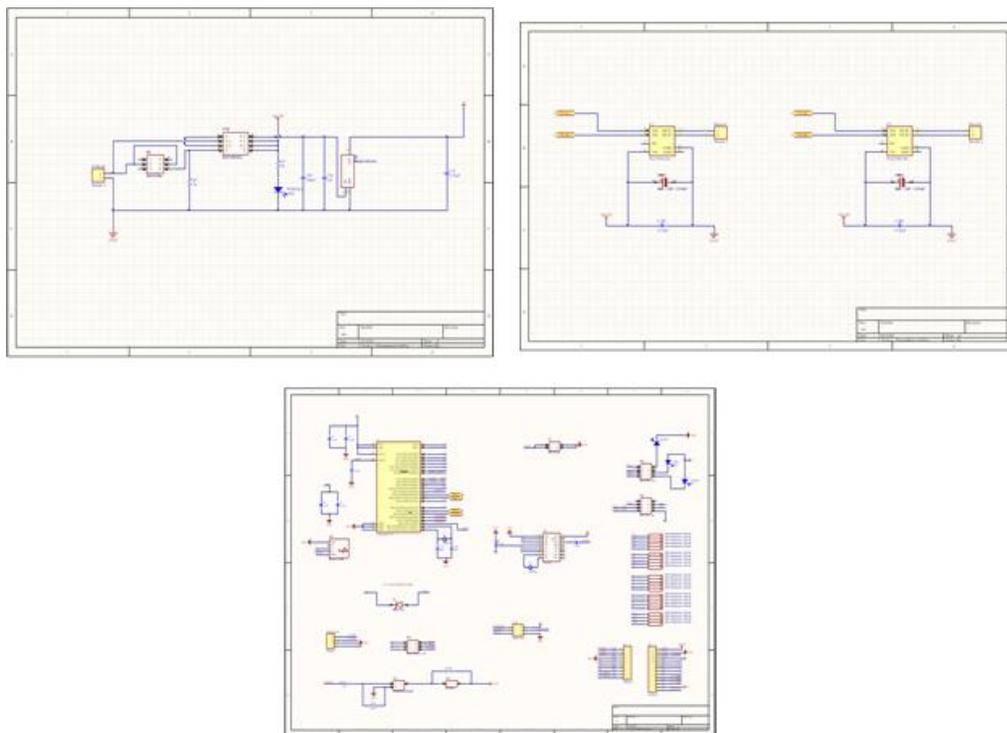
#### **4.1.2 Digitalización de diagramas electrónicos en SolidWorks PCB.**

Para el diseño de la placa se utilizó el software SolidWorks PCB por sus amplias fortalezas en el diseño electrónico de PCB (Printed Circuit Board) y su interfaz robusta y fácil de usar.



**Figura 21.** SolidWorks PCB. Fuente. Autores del proyecto.

Luego de maquetado los diferentes diagramas electrónicos mostrados en puntos anteriores, se procedió a pasar ese mismo diseño a la interfaz de SolidWorks.



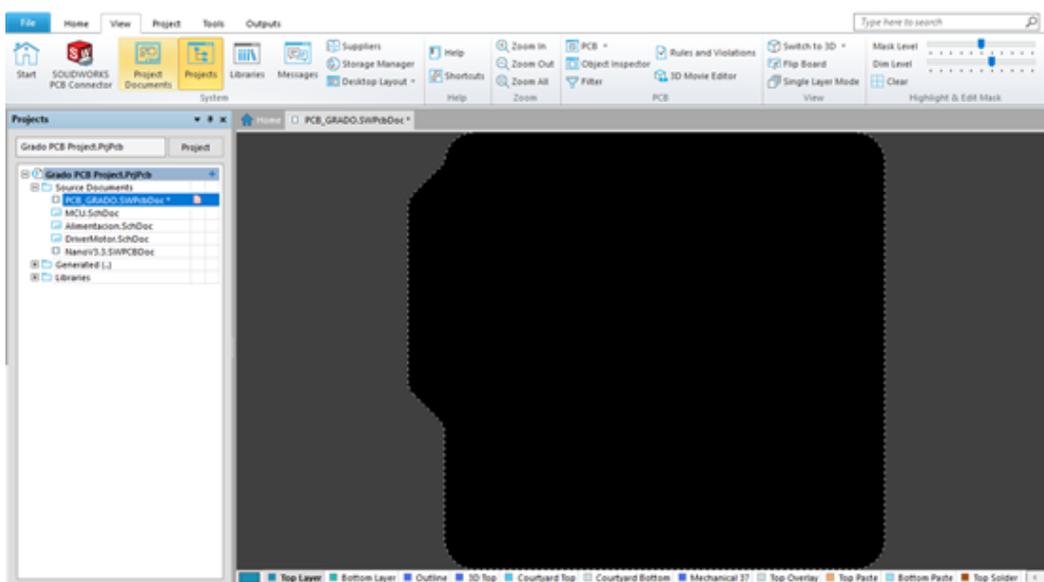
**Figura 22.** Esquemas electrónicos SolidWorks. Fuente. Autores del proyecto.

Cada conjunto de pines de componentes que están conectados entre sí forma lo que se conoce como una red. A cada red se le asigna automáticamente un nombre generado por el sistema, que se basa en uno de los pines del componente de dicha red.

Para facilitar la identificación de redes importantes en el diseño, se puede agregar etiquetas de red para asignar nombres.

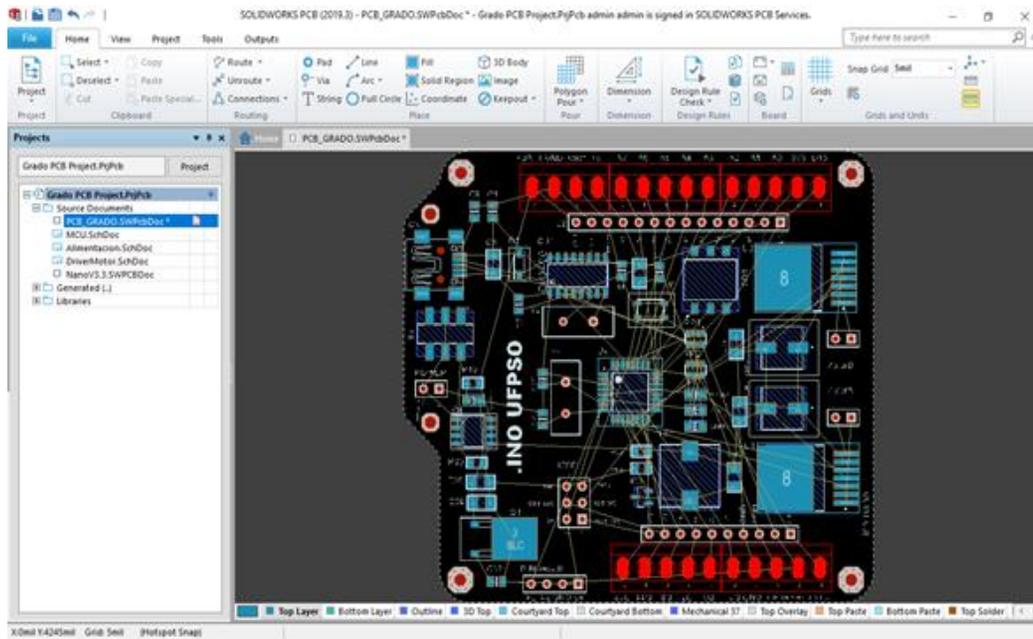
#### 4.1.3 Creación de PCB y posicionamiento de componentes.

Antes de transferir el diseño del Editor de esquemas al Editor de PCB, se creó el PCB con la forma y dimensiones deseadas.



*Figura 23.* Croquis PCB. Fuente. Autores del proyecto.

Seguido de esto se procedió a ubicar los componentes estratégicamente con el fin de aprovechar el espacio de la mejor manera mientras se priorizaba la ubicación de pines y conectores para que favoreciera su usabilidad y conexiones futuras.

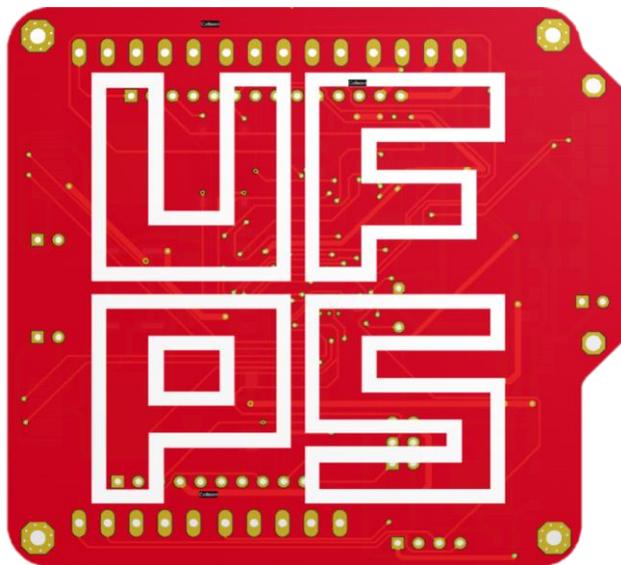


**Figura 24.** Posicionamiento de componentes PCB. Fuente. Autores del proyecto.

Esta placa posee un tamaño de 7cm x 7.5cm y un grosor de 3mm.

En el momento de su elaboración, se optó por un diseño compacto, resistente y liviano que cupiese perfectamente en la palma de la mano, buscando la comodidad al momento de su uso, también se tuvo en cuenta el acoplamiento con las diferentes partes mecánicas y módulos electrónicos adicionales, sobre todo para la ubicación de los componentes electrónicos y pines de salida.

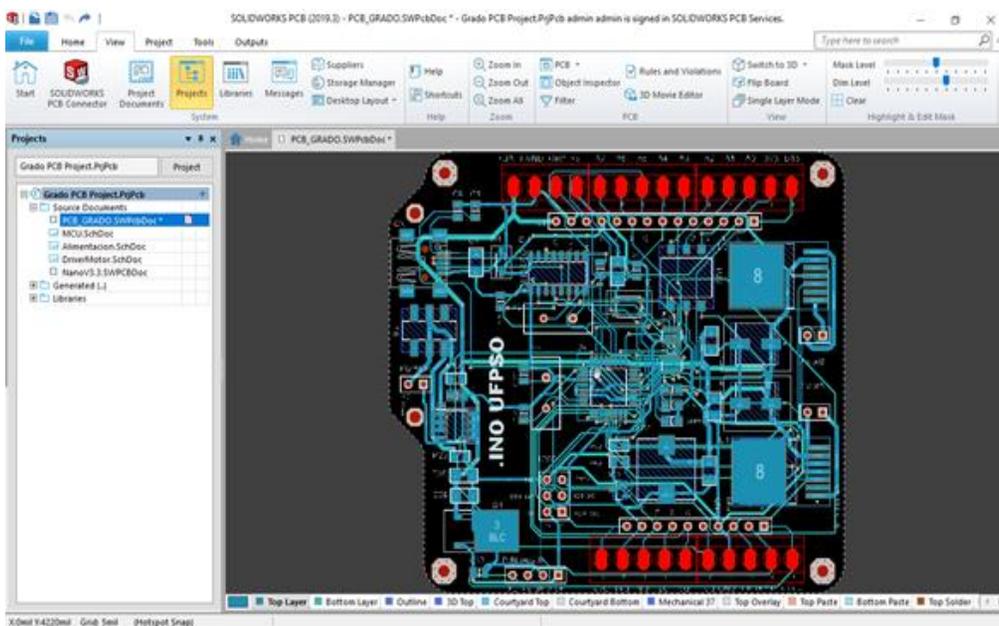
En cuanto al color, se eligieron los colores rojos y blancos los cuales son los colores distintivos de la universidad y le dan un matiz institucional reforzado con las iniciales UFPS marcadas al respaldo de la placa.



*Figura 25.* Respaldo de tarjeta. Fuente. Autores del proyecto.

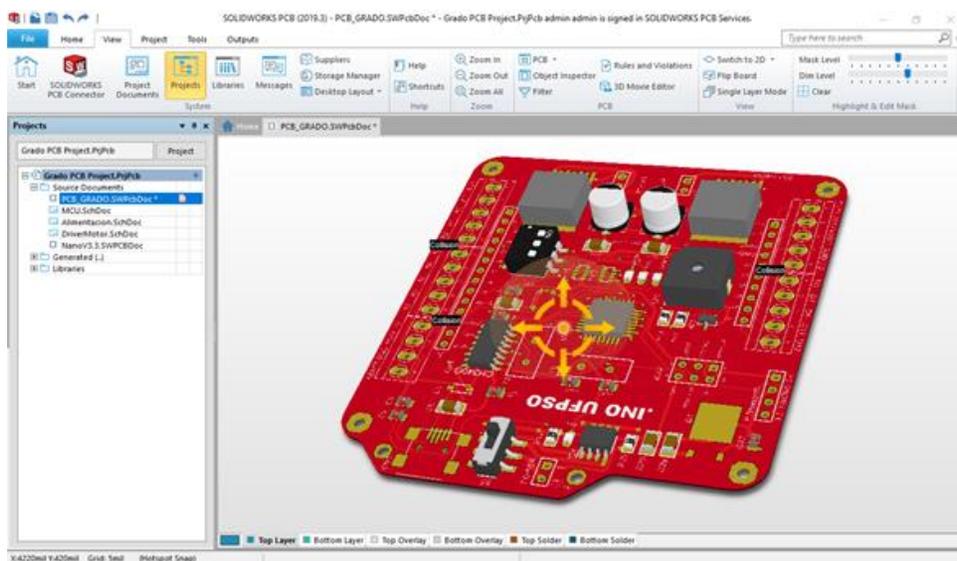
#### **4.1.4 Generación de pistas para PCB.**

El enrutamiento es el proceso de colocar pistas y vías en la placa para conectar los pines del componente. El editor de PCB facilita este trabajo al proporcionar herramientas sofisticadas de enrutamiento interactivo, así como el enrutador automático topológico, que enruta de manera óptima todo o parte de la placa.



*Figura 26.* Generación de pistas PCB. Fuente. Autores del proyecto.

Finalmente, el diseño de la placa está completado y es posible visualizar el resultado en 3D así como exportar los diferentes archivos necesarios para su impresión.



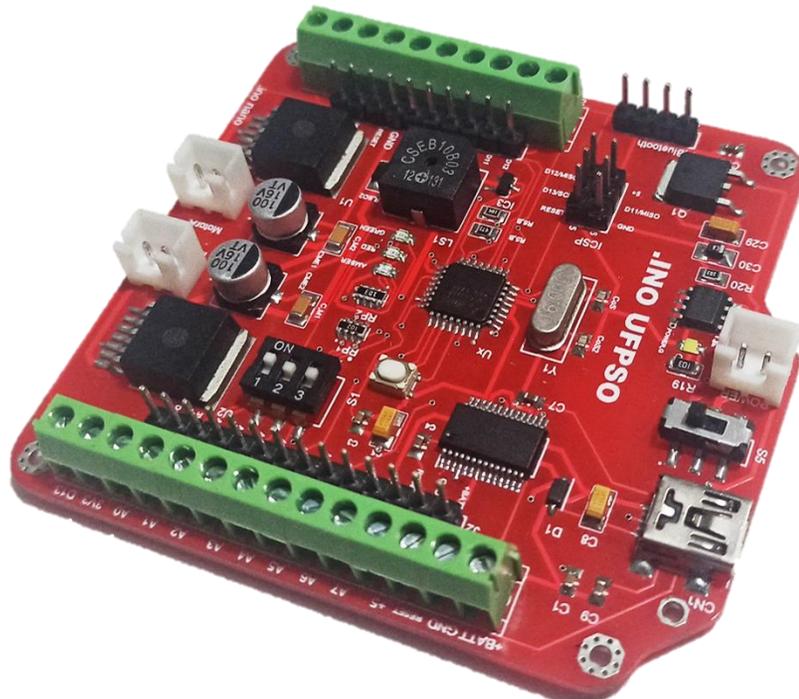
*Figura 27.* PCB 3D. Fuente. Autores del proyecto.

#### 4.1.5 Impresión de PCB y montaje de componentes.

La impresión de la placa se realizó en una empresa extranjera a fin de disminuir costos y obtener como resultado una placa con acabados profesionales y materiales de excelente calidad.

La placa fue diseñada con la idea de que fuese liviana y lo más compacta posible, es por esto por lo que todos los componentes utilizados fueron de montaje superficial (SMD).

El montaje de todos los componentes se realizó de forma manual haciendo uso de herramientas tales como caudín, estaño, pinzas, lupa, y otros.



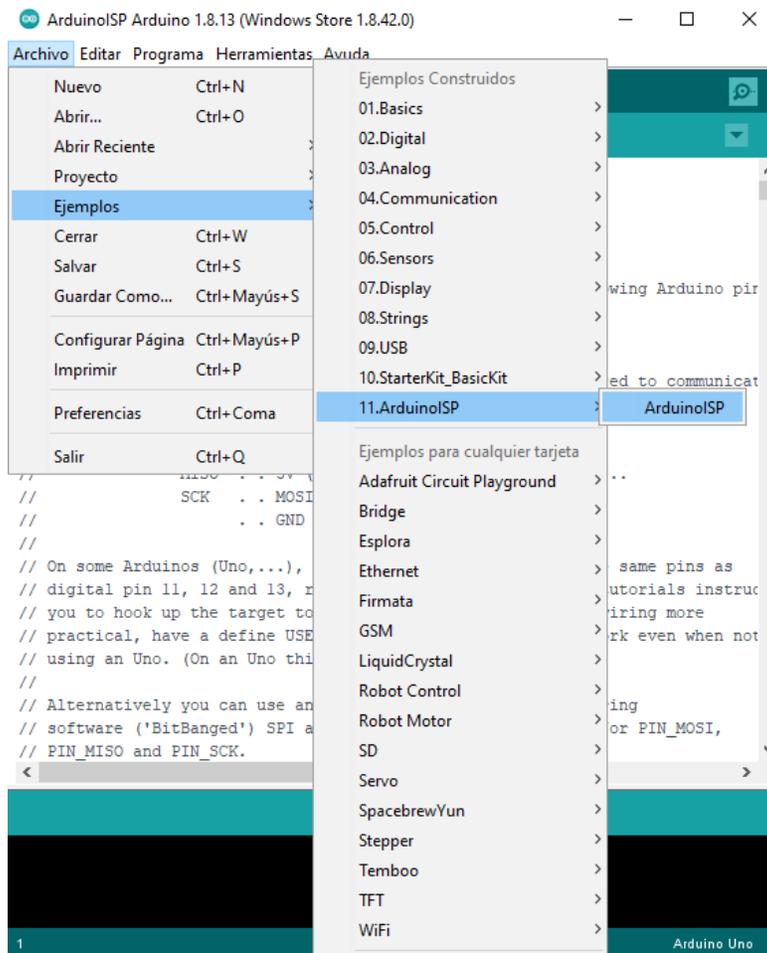
**Figura 28.** Resultado final de la placa. Fuente. Autores del proyecto.

#### **4.1.6 Quema de Bootloader.**

Antes de poder hacer uso de la placa es necesario abarcar un componente intangible pero indispensable para el funcionamiento de la placa, ese es conocido como Bootloader.

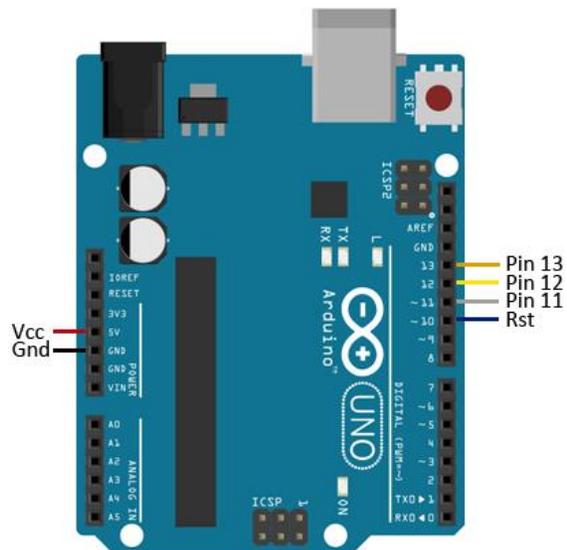
El Bootloader es un firmware que permite la rápida descarga de programas al microcontrolador a través de una conexión USB. Este se ejecuta cuando el microcontrolador se enciende o se pulsa el botón reset y durante un corto espacio de tiempo espera que le llegue por el puerto serie un nuevo sketch desde el IDE de Arduino (este distingue un sketch de otra cosa porque tiene un formato definido). Si llega un sketch, este es guardado en la memoria flash y ejecutado, en caso contrario ejecuta el sketch anteriormente cargado.

La manera más sencilla de realizar el proceso de grabado del Bootloader es utilizando una tarjeta Arduino como programador. Para ello, se carga el Sketch “Arduino as ISP” de la librería de ejemplos de Arduino.

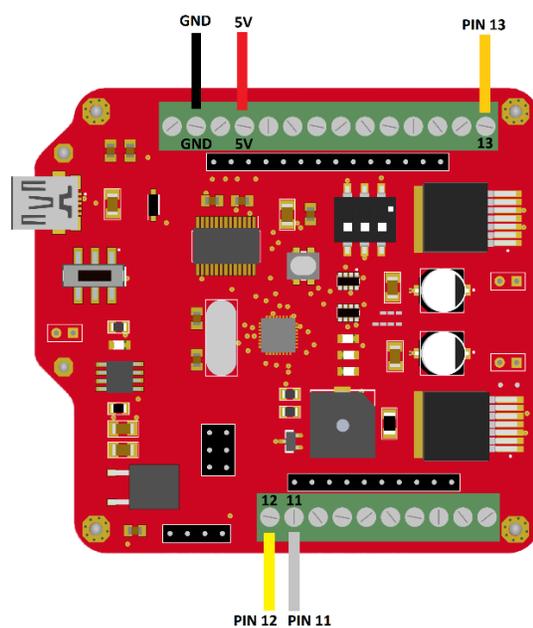


**Figura 29.** Arduino ISP. Fuente. Autores del proyecto.

El siguiente paso es conectar el Arduino programador al chip, para esto la placa diseñada cuenta con un slot con los pines requeridos en la conexión.



**Figura 30.** Conexión Arduino Programado. Fuente. Autores del proyecto.



**Figura 31.** Conexión UFPSO.INO. Fuente. Autores del proyecto.

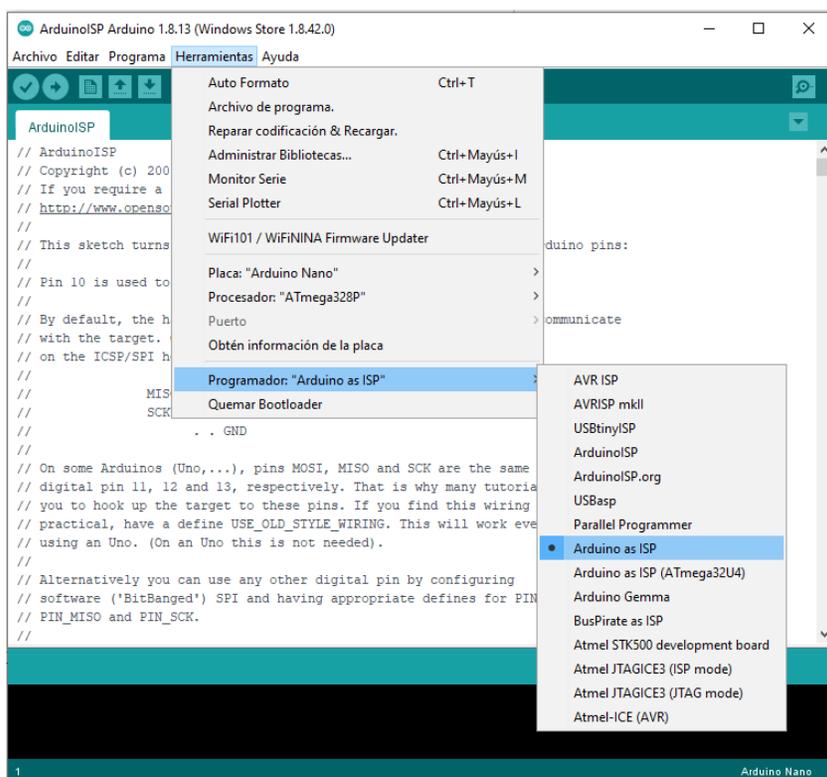
Luego de realizar la conexión se configura el IDE de Arduino para la tarjeta programadora teniendo en cuenta las siguientes pautas:

Programador: “Arduino as ISP”

Placa: “Arduino Nano”

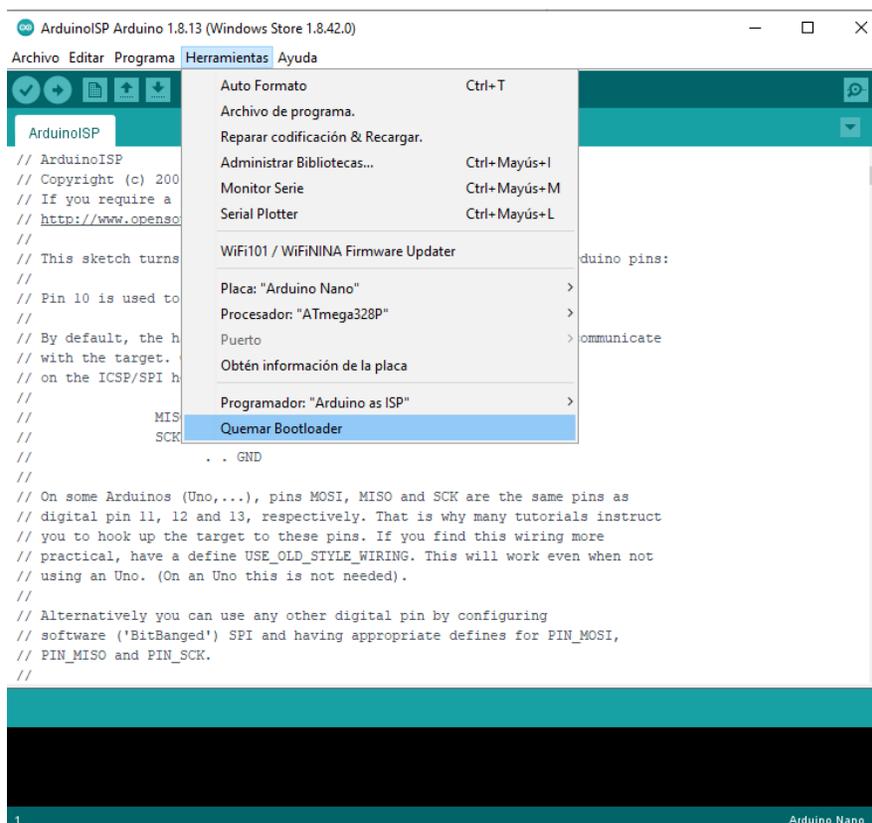
Procesador: “ATmega328”

El Bootloader que se escogió para la placa resultante fue el del Arduino Nano puesto que tiene el mismo microcontrolador y cumple con todas las características esperadas.



**Figura 32.** Configuración ISP. Fuente. Autores del proyecto.

Por último, se selecciona la opción “Quemar Bootloader” la cual iniciara el proceso de grabado del Bootloader a la tarjeta destino



*Figura 33.* Quema de Bootloader. Fuente. Autores del proyecto.

## 4.2 Construcción de estructura mecánica adaptable a la placa electrónica que facilite la movilidad del robot

Según Tapiero Santa (2019), la construcción de una estructura mecánica comprende la definición de su forma y la manera como los diferentes componentes electrónicos se acoplan a ella.

Para la estructura mecánica se diseñaron 3 piezas principales correspondientes a los diferentes tipos de robots que se podrían ensamblar.

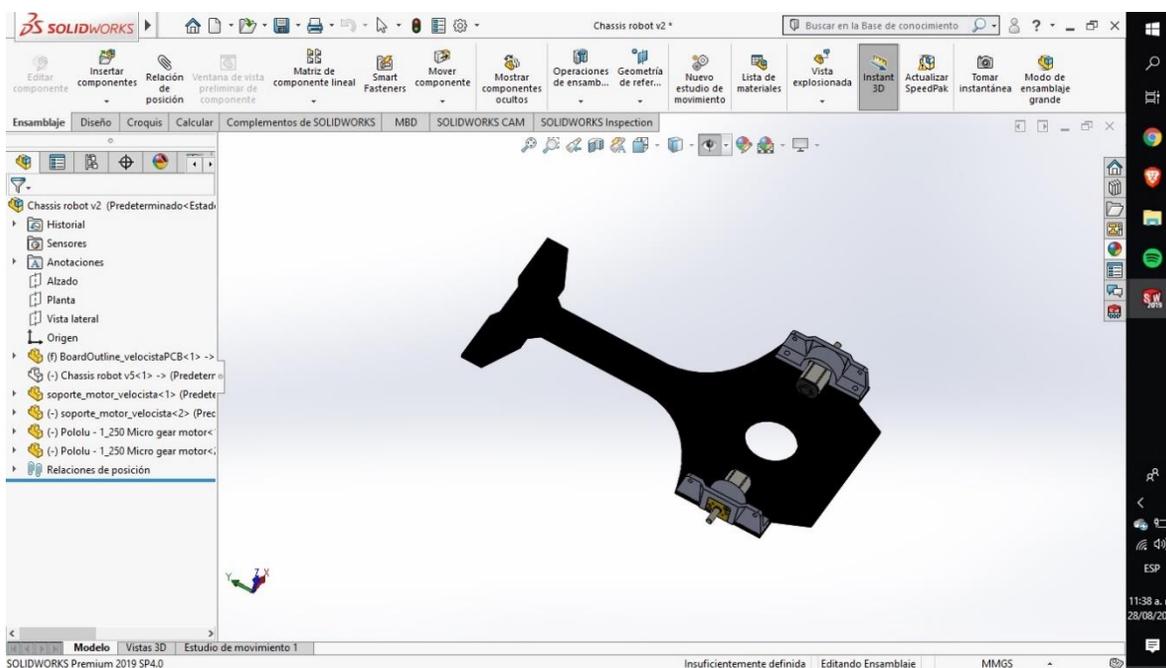
En el diseño de cada una de las plataformas se tuvo en cuenta en una primera instancia el acople con la Board o placa de circuitos, seguidamente se tuvo en cuenta los posibles módulos o complementos que se podrían ajustar a la estructura básica del robot, por lo que se crearon una serie de ranuras y orificios multipropósito cuyo objetivo es facilitar el acople con estos distintos elementos.

El material utilizado para la construcción de estos diseños fue acrílico reforzado de 3mm, el cual fue escogido por su resistencia a roturas y flexibilidad.

Además de los diferentes chasises se incluyeron ciertos componentes de terceros con la finalidad de posibilitar al estudiante construir algunos prototipos básicos y así comprender el funcionamiento general de un robot y de esta manera adquirir los conocimientos básicos que le permitan incorporar nuevos elementos y programar una lógica propia de funcionamiento. Entre los elementos añadidos podemos destacar:

- Soportes para motores
- Ruedas todo terreno
- Ruedas para terreno liso
- Tornillos y tuercas

### 4.2.1 Plataforma N1: Velocista



**Figura 34.** Plataforma Seguidor de Línea. Fuente. Autores del proyecto.

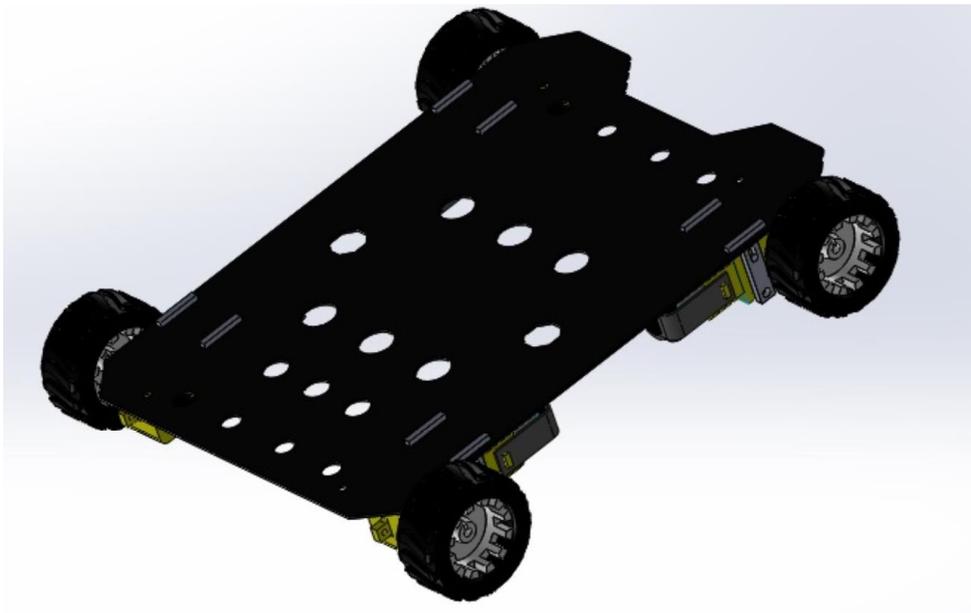
Esta plataforma se creó con el fin de ensamblar un robot velocista seguidor de línea, robot cuyo objetivo es recorrer un circuito de línea negra sobre una superficie blanca.

En el diseño de esta estructura se tuvo en cuenta algunos principios para mejorar el funcionamiento de este robot entre los cuales se pueden destacar:

- Los sensores de línea se sitúan al frente del robot para que este pueda captar con anticipación las señales de luz correspondiente a la dirección de la línea negra. Esto mejora significativamente la velocidad de respuesta del robot y evita que este se salga de la línea cuando su velocidad es alta.

- Los elementos pesados como son la batería y la placa base se sitúan en una zona central para mejorar la estabilidad del robot y evitar que pierda el control en las curvas.
- La distancia de los sensores y de la estructura en general se sitúan lo más cercanos al suelo para mejorar aún más la estabilidad del robot.

#### 4.2.2 Plataforma N2: Todo terreno



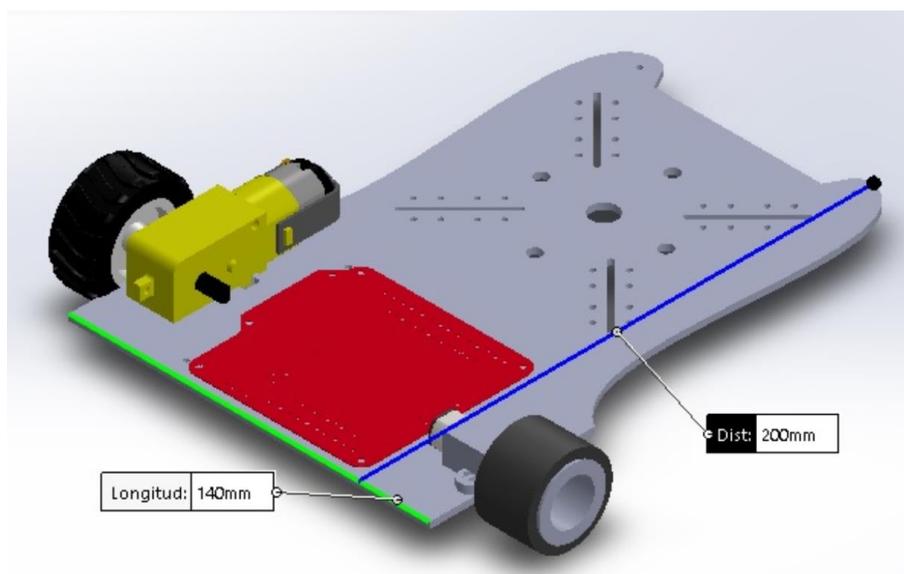
*Figura 35.* Plataforma Todo Terreno. Fuente. Autores del proyecto.

Esta plataforma tiene un diseño multipropósito con diferentes tipos de perforaciones que facilitan el acople de múltiples componentes que se quieran conectar al prototipo. Esta plataforma facilita en gran medida el manejo de cables al tener diferentes agujeros por donde pueden pasar.

Cuenta con ranuras especiales para el acople de motorreductores de plástico y llantas todo terreno de 65mm, pensado para armar ejemplares más resistentes y que puedan transitar por superficies más ásperas.

Adicional a esta también se diseñó una variante de este modelo con igual forma y tamaño, pero con modificación en los tipos de ranura y en la ubicación de estos mismos, esto con el objetivo de brindar aún más posibilidades al estudiante a la hora de crear su propio robot.

#### 4.2.3 Plataforma N3: Modelo Híbrido



**Figura 36.** Plataforma Híbrida. Fuente. Autores del proyecto.

Esta plataforma es una mezcla entre las plataformas 1 y 2. Permite el acople de dos motores que pueden ser micromotores o motorreductores de plástico. Cuenta con ranuras de diferentes formas y tamaños que le permiten acoplar diferentes elementos externos.

### **4.3 Diseño de algoritmo en lenguaje Arduino para el procesamiento de señales y control, con el fin de obtener el mejor rendimiento posible del robot**

La placa UFPSO.INO cuenta con un gestor de arranque de un Arduino Pro-mini, lo cual facilita su programación teniendo la posibilidad de crear y cargar todo tipo de algoritmos escritos en este lenguaje tan difundido y con una comunidad tan amplia.

El proceso de carga de un algoritmo es extremadamente sencillo y similar al usado con una placa Arduino cualquiera, no obstante, este proceso se encuentra detallado en el manual de uso.

Para el desarrollo de este objetivo se tomó la decisión de en primera instancia generar una serie de algoritmos básicos para conocer el funcionamiento de la placa con los diferentes componentes integrados y algunos módulos y sensores externos tales como sensores de línea y de proximidad. Posteriormente se diseñó un algoritmo para el procesamiento de señales y control más complejo que abarcara todos los conceptos analizados en los ejemplos y profundizara en su uso con el fin de obtener un alto rendimiento de un robot seguidor de línea.

#### **4.3.1 Ejemplos de código**

En este apartado se presentan diferentes ejemplos de código para el funcionamiento individual de los componentes integrados y módulos externos.

### 4.3.1.1 Led Pin 13 Salida Digital

```
LedPin13$
void setup() {
    // Inicialización de pin 13 como pin de salida
    pinMode(13, OUTPUT);
}

//Esta función es el ciclo principal que se ejecuta una y otra vez
void loop() {

    digitalWrite(13, HIGH); // Enciende el LED

    delay(1000);           // Espera 1 segundo

    digitalWrite(13, LOW); // Apaga el LED

    delay(13);             // Espera 1 segundo
}
```

**Figura 37.** Código Led Pin 13. Fuente. Autores del proyecto.

### 4.3.1.2 Led Pin 13 Salida Analógica

```
LedPin9_PWM$
int ledPin = 9; // LED Pin 9 controlado por PWM

void setup(){
}

void loop() {

    for (int i = 0; i <= 255; i++){
        analogWrite(ledPin, i); // Establece el brillo a la variable creciente i
        delay(100);             // Pausa de 100 ms
    }

    for (int i=255; i>=0; i--) {
        analogWrite(ledPin, i); // Establece el brillo a la variable decreciente i
        delay(100);             // Pausa de 100 ms
    }
}
```

**Figura 38.** Código Led Pin 9 PWM. Fuente. Autores del proyecto.

En este ejemplo se emplea la modulación por ancho de pulso (PWM) para controlar el brillo de un led. Esta señal puede tomar valores entre 0 y 255 siendo 255 el punto de máximo voltaje y 0 el del mínimo.

#### 4.3.1.3 Buzzer Integrado

En este código se hace uso del Buzzer integrado en la placa cuyo pin asociado es el número 4. Su funcionamiento es muy sencillo y práctico y brinda la oportunidad de ser usado en cualquier proyecto como señal de alerta, como puede ser el caso de un robot anti-obstáculos.

```
BuzzerPin4 §  
// Asignación de pin 4 a la variable buzzer  
int buzzer = 4;  
  
void setup() {  
  
    // Inicialización de pin 4 como pin de salida  
    pinMode(buzzer, OUTPUT);  
  
}  
  
//Esta función es el ciclo principal que se ejecuta una y otra vez  
void loop() {  
  
    digitalWrite(buzzer, HIGH); // Enciende el buzzer  
  
    delay(1000);                // Espera 1 segundo  
  
    digitalWrite(buzzer, LOW);  // Apaga el buzzer  
  
    delay(1000);                // Espera 1 segundo  
}
```

**Figura 39.** Código Buzzer Pin 4. Fuente. Autores del proyecto.

### 4.3.1.4 Driver Integrado

```

Motores §

#define MotorA1 5 // Pin 5 PWM Motor N1
#define MotorA2 6 // Pin 6 PWM Motor N1
#define MotorB1 9 // Pin 9 PWM Motor N2
#define MotorB2 10 // Pin 10 PWM Motor N2

void setup()
{
  Serial.begin(9600); // Configuración de monitor serial para visualizar velocidad
  pinMode(MotorA1, OUTPUT); //Configuración de pines asociados a cada motor
  pinMode(MotorA2, OUTPUT);
  pinMode(MotorB1, OUTPUT);
  pinMode(MotorB2, OUTPUT);

  digitalWrite(MotorA1, LOW); //Estado inicial de los motores en OFF
  digitalWrite(MotorA2, LOW);
  digitalWrite(MotorB1, LOW);
  digitalWrite(MotorB2, LOW);
}

void MotorA_Forward(int i) { //Función Adelante Motor N1
  Serial.println(i);
  analogWrite(MotorA1, i);
  digitalWrite(MotorA2, LOW);
}

void MotorB_Forward(int i) { //Función Adelante Motor N2
  analogWrite(MotorB1, i);
  digitalWrite(MotorB2, LOW);
}

void MotorA_Backward(int i) { //Función de retroceso Motor N1
  digitalWrite(MotorA1, LOW);
  analogWrite(MotorA2, i);
}

void MotorB_Backward(int i) { //Función de retroceso Motor N2
  digitalWrite(MotorB1, LOW);
  analogWrite(MotorB2, i);
}

void loop()
{
  for (int i = 0; i <= 125; i++){ // Variación de Velocidad Motor N1 de 0 a 125
    MotorA_Forward(i);
    delay(100);
  }

  for (int i=125; i>=0; i--) { // Variación de Velocidad Motor N2 de 0 a 125
    MotorA_Forward(i);
    delay(100);
  }
}

```

**Figura 40.** Código Drivers de Motor. Fuente. Autores del proyecto.

Esta tarjeta cuenta con un par de controladores para motor que pueden ser utilizados de manera muy sencilla. Los pines asociados al Driver N1 son el 5 y 6, y para el Driver N2, los pines 9 y 10 respectivamente. Estos 4 pines cuentan con salida PWM (Modulación por ancho de pulso), la cual permite variar la velocidad del motor en un rango de 0 a 255, contrario a lo que sucede con pines netamente digitales que solo permiten estados binarios (HIGH-LOW).

#### 4.3.1.5 Sensor Ultrasonido HC-SR04

```

SensosUltrasonido $
const int Trigger = 2; //Pin digital 2 para el Trigger del sensor
const int Echo = 3; //Pin digital 3 para el Echo del sensor

void setup() {
  Serial.begin(9600); //Se inicializa la comunicación serial
  pinMode(Trigger, OUTPUT); //Pin Triguer configurado como salida
  pinMode(Echo, INPUT); //Pin Echo configurado como entrada
  digitalWrite(Trigger, LOW); //Se inicializa el pin con un estado bajo
}

void loop()
{

  long t; //Tiempo que tarda en llegar el eco
  long d; //Distancia en centímetros

  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10); //Se envía un pulso de 10us
  digitalWrite(Trigger, LOW);

  t = pulseIn(Echo, HIGH); //Se obtiene el ancho del pulso
  d = t/59; //Se escala el tiempo a una distancia en cm

  Serial.print("Distancia: ");
  Serial.print(d); //Se imprime la distancia en el monitor serial
  Serial.print("cm");
  Serial.println();
  delay(100); //Se realiza una pausa de 100ms
}

```

**Figura 41.** Código Sensor Ultrasonido. Fuente. Autores del proyecto.

Esta placa brinda la posibilidad de conectar un sinnúmero de sensores y módulos externos como puede ser el caso del sensor ultrasonido de distancia HC-SR04. A continuación, se muestra un código de ejemplo que permite medir la distancia desde el sensor a cualquier superficie e imprimirla en el monitor serial.

#### ***4.3.1.5 Sensor Qtr8-A***

El sensor Qtr8-A es ampliamente utilizado en la elaboración de robots seguidores de línea por su forma compacta que integra más de 6 sensores individuales, además de ello cuenta con una librería oficial que facilita su uso y acoplamiento a un robot autónomo.

Este ejemplo está diseñado para usarse con seis sensores QTR analógicos. Estos sensores de reflectancia deben conectarse a los pines analógicos A0 a A5. El pin de control del emisor de los sensores LEDON se puede conectar opcionalmente al pin digital 2, o se puede dejar desconectado y eliminar la llamada a `setEmitterPin ()`.

La fase de configuración de este ejemplo calibra los sensores durante diez segundos y enciende el LED de Arduino (generalmente en el pin 13) mientras se realiza la calibración. Durante esta fase, es necesario exponer cada sensor de reflectancia a las lecturas más claras y oscuras que haya. Por ejemplo, si se está haciendo un seguidor de línea, se debe deslizar los sensores a lo largo de la línea durante la fase de calibración para que cada sensor pueda obtener una lectura de qué tan oscura es la línea y qué tan claro es el suelo. Una calibración incorrecta resultará en lecturas deficientes.

## QTRExample 5

```

#include <QTRSensors.h>
QTRSensors qtr;
const uint8_t SensorCount = 6;
uint16_t sensorValues[SensorCount];

void setup()
{
  // Configuración de los sensores
  qtr.setTypeAnalog();
  qtr.setSensorPins((const uint8_t[]){A0, A1, A2, A3, A4, A5}, SensorCount);
  qtr.setEmitterPin(2);

  delay(500);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH); // Enciende el led integrado para indicar el modo calibración

  //La función calibrate() toma 4 muestras por sensor con una duración de 24ms
  //Esta función es llamada 400 veces para obtener una calibración precisa
  for (uint16_t i = 0; i < 400; i++)
  {
    qtr.calibrate();
  }
  digitalWrite(LED_BUILTIN, LOW); // Apaga el led integrado para indicar el fin de la calibración

  // Imprime los valores mínimos de calibración cuando los emisores estaban encendidos
  Serial.begin(9600);
  for (uint8_t i = 0; i < SensorCount; i++)
  {
    Serial.print(qtr.calibrationOn.minimum[i]);
    Serial.print(' ');
  }
  Serial.println();

  //Imprime los valores máximos de calibración cuando los emisores estaban encendidos
  for (uint8_t i = 0; i < SensorCount; i++)
  {
    Serial.print(qtr.calibrationOn.maximum[i]);
    Serial.print(' ');
  }
  Serial.println();
  Serial.println();
  delay(1000);
}

void loop()
{
  //Lee los valores del sensor calibrado y obtiene una medida de la posición de la línea
  uint16_t position = qtr.readLineBlack(sensorValues);
  //Imprime los valores del sensor como números del 0 al 1000, donde 0 significa reflectancia máxima
  //y 1000 significa reflectancia mínima, seguido de la posición de la línea

  for (uint8_t i = 0; i < SensorCount; i++)
  {
    Serial.print(sensorValues[i]);
    Serial.print('\t');
  }
  Serial.println(position);
  delay(250);
}

```

*Figura 42.* Código Sensor Qtr8-A. Fuente. Autores del proyecto.

El bucle principal del ejemplo lee los valores del sensor calibrado y los usa para estimar la posición de una línea. Puede probar esto pegando un trozo de cinta aislante negra de 3/4 " en un trozo de papel blanco y deslizando el sensor a través de él. Imprime los valores del sensor en el monitor en serie como números del 0 (reflectancia máxima) a 1000 (mínimo reflectancia) seguido de la ubicación estimada de la línea como un número de 0 a 5000, 1000 significa que la línea está directamente debajo del sensor 1, 2000 significa directamente debajo del sensor 2, etc. 0 significa que la línea está directamente debajo del sensor 0 o se vio por última vez antes de perderse por el sensor 0, 5000 significa que la línea está directamente debajo del sensor 5 o fue vista por última vez por el sensor 5 antes de perderse.

#### **4.3.2 Algoritmo PID para un robot velocista.**

Se diseñó este algoritmo con el fin de aprovechar las características de la placa y de esta forma crear un robot autónomo en el que se puedan evidenciar sus diferentes componentes y la manera como interactúan entre sí.

La funcionalidad del autómatas representado en este código es recorrer lo más rápido posible un circuito definido por una línea negra sobre una superficie blanca. Esta tarea supone diferentes retos técnicos que abarcan desde la estructura mecánica hasta la eficiencia del algoritmo de control (PID).

Para su construcción se hizo uso de diferentes sensores que permiten ubicar la línea negra en tiempo real, y posteriormente procesar esa información entrante con un algoritmo de control

PID que calcula la desviación del robot y genera una corrección. Estos datos de salida son traducidos a una señal PWM enviada a los motores, que finalmente son los encargados de guiar al autómatas por el circuito. (ver **Apéndice A**)

```

RobotVelocista
#include <QTRSensors.h>

//Configuración pines driver de motor
#define MOTORLEFT_DIR_A 5
#define MOTORLEFT_DIR_B 6
#define MOTORRIGHT_DIR_A 9
#define MOTORRIGHT_DIR_B 10

#define NUM_SENSORS          6 // Número de sensores usados
#define NUM_SAMPLES_PER_SENSOR 4 // Muestras por sensor
#define EMITTER_PIN          2 // Emisor de Qtr8-A es controlado por el pin 2

#define LEDPIN      13 // Led integrado conectado a pin 13
#define BUTTONPIN   3  // Pulsador conectado al pin 3

#define esperarBoton() while(!digitalRead(BUTTONPIN)); while(digitalRead(BUTTONPIN))

// Configuración inicial de barra de sensores Qtr8-A
QTRSensorsAnalog qtra((unsigned char[]) {A0, A1, A2, A3, A4, A5, A6, A7},
    NUM_SENSORS, NUM_SAMPLES_PER_SENSOR, EMITTER_PIN);
unsigned int sensorValues[NUM_SENSORS];

//Función para girar hacia la izquierda
void setMotorLeft(int value)
{
    if ( value >= 0 )
    {
        analogWrite(MOTORRIGHT_DIR_A,value);
        digitalWrite(MOTORRIGHT_DIR_B,LOW);
    }
    else
    {
        digitalWrite(MOTORRIGHT_DIR_A,LOW);
        analogWrite(MOTORRIGHT_DIR_B,value);
        value *= -1;
    }
}
}

```

**Figura 43.** Algoritmo Seguidor de Línea P1. Fuente. Autores del proyecto.

```

// Función para girar hacia la derecha
void setMotorRight(int value)
{
  if ( value >= 0 )
  {
    analogWrite(MOTORLEFT_DIR_A,value);
    digitalWrite(MOTORLEFT_DIR_B,LOW);
  }
  else
  {
    digitalWrite(MOTORLEFT_DIR_A,LOW);
    analogWrite(MOTORLEFT_DIR_B,value);
    value *= -1;
  }
}

// Función de arranque para motores
void setMotors(int left, int right)
{
  setMotorLeft(left);
  setMotorRight(right);
}

// Función de frenado de motores
void setBrake(boolean left, boolean right, int value)
{
  if ( left )
  {
    analogWrite(MOTORRIGHT_DIR_A,value);
    analogWrite(MOTORRIGHT_DIR_B,value);
  }

  if ( right )
  {
    analogWrite(MOTORLEFT_DIR_A,value);
    analogWrite(MOTORLEFT_DIR_B,value);
  }
}

```

**Figura 44.** Algoritmo Seguidor de Línea P2. Fuente. Autores del proyecto.

```

void setup()
{
  //Configuración de pines de salida y entrada
  pinMode(LEDPIN          ,OUTPUT);
  pinMode(MOTORRIGHT_DIR_A ,OUTPUT);
  pinMode(MOTORRIGHT_DIR_B ,OUTPUT);
  pinMode(MOTORLEFT_DIR_A  ,OUTPUT);
  pinMode(MOTORLEFT_DIR_B  ,OUTPUT);
  pinMode(BUTTONPIN       ,INPUT);
  digitalWrite(BUTTONPIN,HIGH);

  //Espera a que el pulsador sea accionado para empezar la calibración
  while ( digitalRead(BUTTONPIN) );

  //Calibración QTR8-A
  for ( int i=0; i<70; i++)
  {
    digitalWrite(LEDPIN, HIGH); delay(20);
    qtra.calibrate();
    digitalWrite(LEDPIN, LOW);  delay(20);
  }

  digitalWrite(LEDPIN, LOW);

  //Espera a que el pulsador sea nuevamente accionado para iniciar el robot
  while ( digitalRead(BUTTONPIN) );

  delay(500); //Tiempo de espera antes de arrancar

  setMotors(90, 90);

  delay(300);
}
//Inicialización de constantes de algoritmo PID
unsigned int position = 0;
int derivative = 0;
int proportional = 0;
int power_difference = 0;

int max = 90;

int last_proportional;

float KP = 0.15;
float KD = 2.2;

#define RANGEBRAKE 3500

```

*Figura 45.* Algoritmo Seguidor de Línea P3. Fuente. Autores del proyecto.

```

//Bucle principal
void loop()
{
  //Posición de la línea
  position = qtra.readLine(sensorValues);

  //Cálculo de posición relativa
  proportional = ((int)position) - 3500;

  //Activación de función de frenado si la línea se encuentra a los extremos
  if ( proportional <= -RANGEBRAKE )
  {
    setMotorRight(0);
    setBrake(true, false, 255);
    delay(1);
  }
  else if ( proportional >= RANGEBRAKE )
  {
    setMotorLeft(0);
    setBrake(false, true, 255);
    delay(1);
  }

  //Aplicación de algoritmo PID para encontrar el error con respecto
  //al centro de la línea
  derivative = proportional - last_proportional;

  last_proportional = proportional;

  int power_difference = ( proportional * KP ) + ( derivative * KD );

  if ( power_difference > max ) power_difference = max;
  else if ( power_difference < -max ) power_difference = -max;

  ( power_difference < 0 ) ?
    setMotors(max+power_difference, max) : setMotors(max, max-power_difference);
}

```

*Figura 46.* Algoritmo Seguidor de Línea P4. Fuente. Autores del proyecto.

#### **4.4 Creación del manual que servirá de guía para el ensamblaje, programación y buen manejo del robot.**

Se diseñó un manual que contiene los detalles de funcionamiento de la placa, así como instrucciones de uso, ejemplos y recomendaciones. Además de ello se muestra el proceso de ensamblaje y programación de un autómatas que permite entender y asimilar diferentes conceptos importantes relacionados al campo de la robótica. (ver **Apéndice A**)

#### **4.5 Socialización del diseño y construcción del kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino.**

Se realizó la socialización a través de la plataforma meet con un grupo de estudiantes del programa de Ingeniería de Sistemas con el fin dar a conocer el proceso de creación y funcionamiento del kit y medir la percepción de los estudiantes frente al proyecto y cómo este puede ayudar en el proceso de aprendizaje en el campo de la robótica.

##### **4.5.1 Temática abarcada**

Para el desarrollo de la reunión se abordaron diferentes temas con respecto al proyecto. En primer lugar, se hizo una pequeña introducción para contextualizar a los estudiantes y mostrarles las diferentes partes de un sistema robótico y cómo interactúan entre sí. Luego de esto se realizó un recorrido detallado de las diferentes etapas de creación de la placa, abarcando desde el análisis electrónico hasta el montaje final de componentes. Posteriormente se mostraron los

diferentes componentes de la placa y como estos generaban un impacto positivo en el proceso de creación de sistemas robóticos.

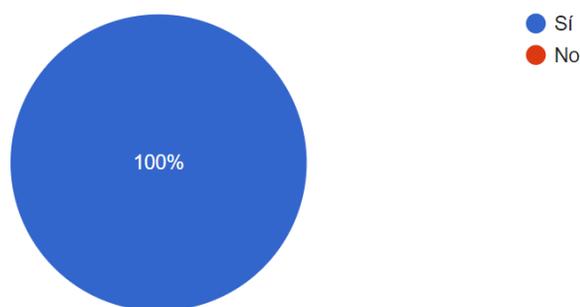
Además del análisis del sistema electrónico del kit, también se habló acerca de la estructura mecánica y los diferentes puntos que se tuvieron en cuenta en su creación.

Finalmente se mostró el proceso de ensamblaje de un robot seguidor de línea abarcando su estructura electrónica, mecánica y los diferentes algoritmos de control utilizados.

#### 4.5.2 Encuesta

Se realizó una encuesta como herramienta para observar y medir la percepción de los estudiantes con respecto al impacto de este kit en los procesos de enseñanza y aprendizaje.

1. ¿Considera que este kit robótico podrá incentivar a los estudiantes a adquirir habilidades en relación con la construcción, diseño y programación de robots?

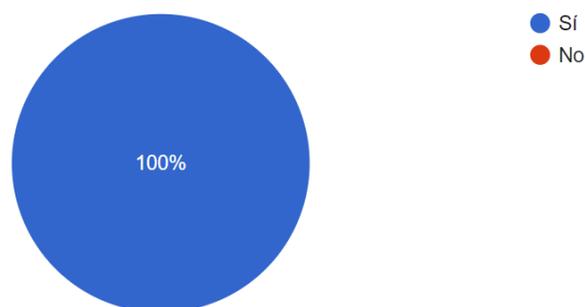


**Figura 47.** Pregunta 1 Encuesta socialización. Fuente. Autores del proyecto.

El 100% de los estudiantes encuestados considera que este kit robótico incentivará a los estudiantes en la adquisición de competencias en lo relacionado a la construcción, diseño y programación de robots.

Esto se plantea como una meta altamente alcanzable al brindar al estudiante herramientas, componentes y una guía detallada que posibilita la adquisición de dichas competencias.

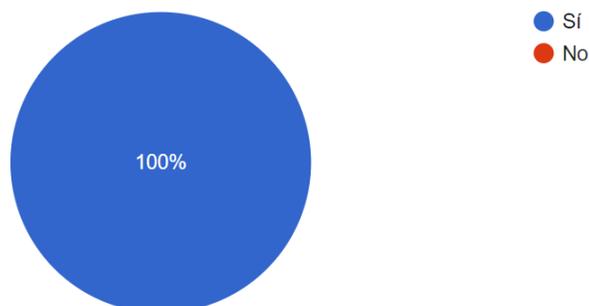
2. ¿Considera que los conocimientos relacionados con robótica, electrónica y algoritmos de control pueden ser útiles para las organizaciones?



**Figura 48.** Pregunta 2 Encuesta socialización. Fuente. Autores del proyecto.

La percepción de los estudiantes con respecto a la utilidad que tienen los conocimientos relacionados al campo de la robótica, electrónica y algoritmos de control en entornos organizacionales, es importante para promover el interés en el uso de este kit.

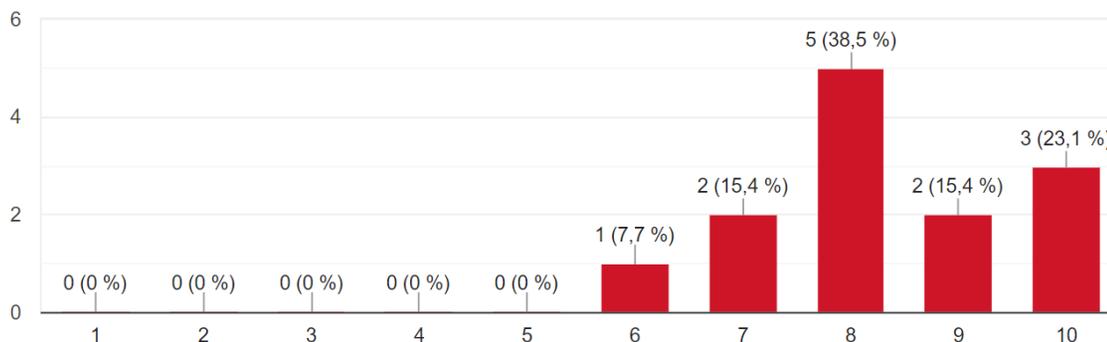
3. ¿Cree usted que este kit ayudará a los estudiantes a identificar correctamente los elementos y características de un sistema robótico?



**Figura 49.** Pregunta 3 Encuesta socialización. Fuente. Autores del proyecto.

Los estudiantes consideran que este proyecto los ayudará a identificar correctamente los diferentes elementos que componen un sistema robótico. Este hecho se soporta mediante la presencia de un manual de uso adjunto al kit.

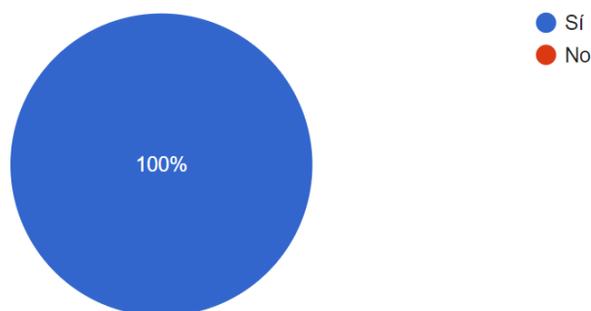
4. En una escala de 1 a 10. ¿Qué tanto cree que este proyecto promoverá el desarrollo de competencias en desarrollo de aplicativos de software y hardware orientado a proyectos robóticos?



**Figura 50.** Pregunta 4 Encuesta socialización. Fuente. Autores del proyecto.

El desarrollo de competencias en relación con la creación de aplicativos de software y hardware orientado a proyectos robóticos se espera que sea impulsado a través de este kit. Para ello se brinda al estudiante un camino o ruta de aprendizaje donde se recoge la experiencia vivida por parte de los autores de este proyecto, con el ánimo de sentar las bases para proyectos más complejos nacidos de la innovación e interés del estudiante.

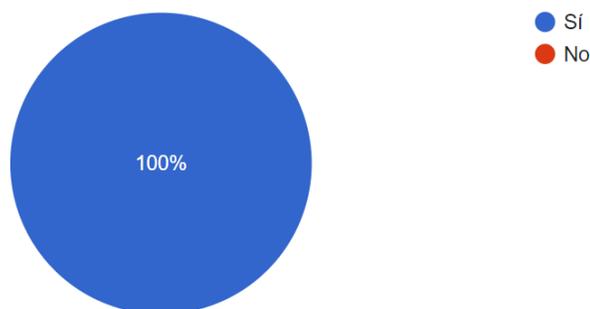
5. ¿Piensa usted que este tipo de proyectos podrá ser útil en el desarrollo de prácticas en la asignatura de robótica y afines?



**Figura 51.** Pregunta 5 Encuesta socialización. Fuente. Autores del proyecto.

La percepción de los estudiantes concuerda con lo esperado por el objetivo futuro de este proyecto, el cual es servir como una herramienta para el desarrollo de prácticas en la asignatura de robótica, permitiendo entender de forma práctica el funcionamiento de un sistema robótico y ofrecer mecanismos e instrumentos que motiven la profundización en este campo.

6. Está de acuerdo con la idea: “la robótica educativa como instrumento para incentivar el desarrollo del pensamiento lógico y la asimilación de conceptos prácticos relacionados a la tecnología”?

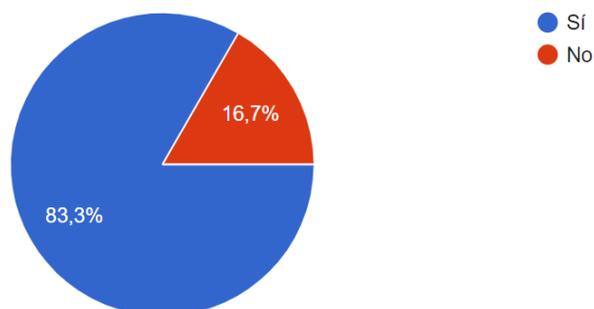


**Figura 52.** Pregunta 6 Encuesta socialización. Fuente. Autores del proyecto.

La idea de ver la robótica como un instrumento para incentivar el desarrollo del pensamiento lógico y la asimilación de conceptos prácticos, es cada vez más aceptada por los estudiantes y docentes, como se muestra en los resultados de la encuesta.

En este sentido, un ambiente de aprendizaje con robótica educativa es una experiencia que contribuye al desarrollo de nuevas habilidades, nuevos conceptos, fortalece el pensamiento sistémico, lógico, estructurado y formal del estudiante, al tiempo que desarrolla su capacidad de resolver problemas concretos, dando así una respuesta eficiente a los entornos cambiantes del mundo actual. (Odorico, 2004)

Considera que este kit, es un instrumento para comprender como funciona el Internet de las cosas



**Figura 53.** Pregunta 7 Encuesta socialización. Fuente. Autores del proyecto.

Los resultados muestran un grado de aceptación alto por parte de los estudiantes para con el proyecto y una sintonía con la idea de que este kit impactará de forma positiva los procesos de enseñanza y aprendizaje de la robótica.

Los estudiantes de la materia de robótica, semilleros investigativos y afines pueden encontrar en este kit de aprendizaje una herramienta que facilite el proceso de creación de proyectos electrónicos y sistemas robóticos al mismo tiempo que asimilan conceptos relacionados a esta área. Esto es posible gracias al enfoque práctico del kit, donde se proporcionan de antemano diferentes componentes preensamblados a la tarjeta principal, que facilita y acelera el desarrollo y entendimiento de sistemas complejos.

En la fase de ensamblaje guiado propuesto por el kit se evidencian las diferentes etapas implicadas en la elaboración de un robot, así como el funcionamiento específico y general de

todos los componentes. Todo esto brindando un panorama amplio al estudiante e incitándolo a profundizar en las diferentes temáticas y continuar el desarrollo de sistemas autónomos más complejos.

Según un estudio realizado por Ojeda, (2018), el doctor Holman Ariza, profesor y coordinador del Semillero de Robótica de la Universidad El Bosque, menciona lo consciente que es de la necesidad de formar profesionales con estas competencias en Colombia: “A partir de la cuarta revolución industrial estamos marcando tendencia hacia la robótica y las telecomunicaciones. Hacia allá va el mundo, en ese rumbo se está desarrollando la sociedad. Por lo tanto, nosotros buscamos que los humanos se capaciten para ese escenario”.

En este orden de ideas, con este kit se pretende articular varios componentes curriculares de acuerdo con el área de la ingeniería, logrando mejorar el aprendizaje en la asignatura de robótica y electrónica y brindando herramientas integradoras de conocimiento que apoyen procesos de investigación.

## Capítulo 5. Conclusiones

En la construcción de la tarjeta para procesamiento de señales se evidenciaron diferentes características que aportan usabilidad y que son de vital importancia para que, en el momento de su uso, la persona se sienta cómoda utilizándola e incremente la interacción con esta. Algunas características que podemos mencionar son: Diferentes pads de conexión, tamaño compacto, elementos integrados, entre otras.

Diferentes tipos de robot móviles requieren de diferentes estructuras mecánicas para obtener el máximo rendimiento en su funcionamiento, por lo cual se diseñaron diferentes plataformas que se acoplan a diferentes terrenos y tareas.

Si bien el hardware es una parte esencial de la robótica, también lo es el software. Los diferentes algoritmos vistos en este trabajo investigativo inducen al estudiante en buenas prácticas de ordenamiento de código, así como el uso de algoritmos de control complementarios que permiten maximizar la eficiencia y reducir errores ocasionados por variables externas.

Finalmente, los resultados obtenidos con la investigación fueron los deseados puesto que se logró desarrollar un kit de aprendizaje para el ensamblaje y la programación de un robot móvil autónomo basado en la plataforma de Arduino que podrá ser usado para respaldar conceptos prácticos impartidos en la materia de robótica y afines.

## Capítulo 6. Recomendaciones

El sistema operativo Windows brinda mayor compatibilidad con la tarjeta de procesamiento, por lo que se recomienda el uso de este.

Los componentes electrónicos son susceptibles a cortocircuitos y daños si son expuestos a líquidos o sustancias similares. Se recomienda mantenerlos en un lugar seco y con temperaturas menores a 40°.

Antes de hacer uso de los diferentes componentes del kit se recomienda leer cuidadosamente el manual de uso para garantizar su correcto funcionamiento.

Mantenga el kit alejado de niños menores 5 años.

Si presenta problemas en la ejecución de los códigos de ejemplo, asegúrese de tener instalada la última versión del IDE, el controlador indicado en el manual, y las librerías propias de los diferentes módulos externos.

## Referencias

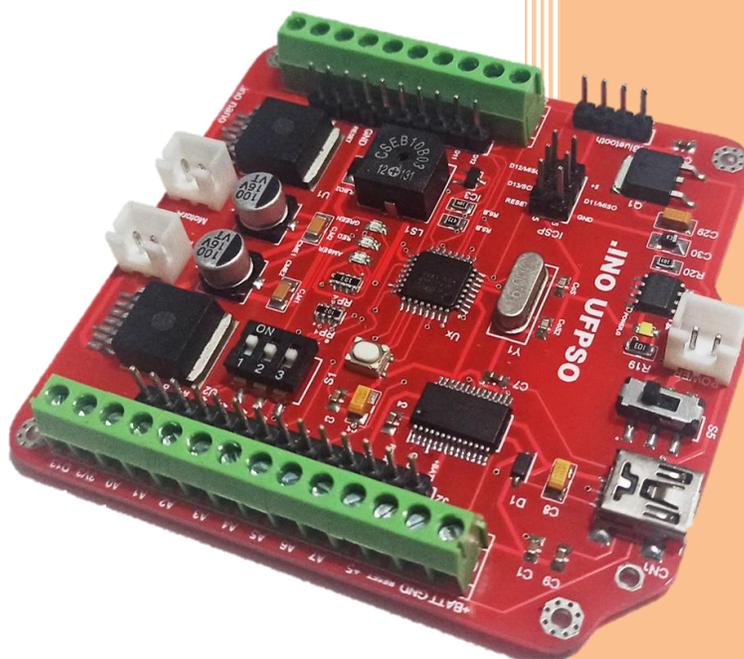
- Acuña, A. (2017). *Robótica y aprendizaje por Diseño en Educación*. Obtenido de <http://www.educoas.org/portal/bdigital/lae-ducacion/home.html>.
- Arias, R. (2016). *La Robótica Pedagógica como Herramienta para la Construcción de Aprendizajes Significativos en el Aula*. Popayan: Coloquio internacional de educación.
- Asimov, I. (2019). [http://www.asimovonline.com/asimov\\_home\\_page.html](http://www.asimovonline.com/asimov_home_page.html). Obtenido de Escritor famoso de ciencia ficción y de variedad de libros de ciencia.
- Barrera Lombana, N. (2014). Uso de la robótica educativa como estrategia didáctica en el aula. *Tunja: Universidad Pedagógica y Tecnológica de Colombia*.
- Borenstein, J. (2014). *Sensors and Methods for Mobile Robot Positioning*. Michigan: Universidad de Michigan.
- Bryson, J. (2018). *La última década y el futuro del impacto de la IA en la sociedad*. Obtenido de <https://www.bbvaopenmind.com/articulos/la-ultima-decada-y-el-futuro-del-impacto-de-la-ia-en-la-sociedad/>.
- Calderón, L., Ceres, R., & Sánchez, J. (2018). *Sensores de distancia en robótica*. España: Revista Robótica. No.12.
- Congreso de Colombia. (2000). *ley 44 de 1993*. Bogotá.
- Congreso de Colombia. (2011). *Ley 44 de 1993*. Bogotá.
- Congreso de Colombia. (2015). *Ley 1450 de 2015*. Bogotá.
- Devcode. (2018). <https://devcode.la/articulos/programacion-de-microcontroladores/>. Obtenido de Arduino: Programación de Microcontroladores.
- García, M., & Deco, C. (2018). *Estrategias Basadas en Robótica para Apoyar el Pensamiento Computacional*. Popayan: Universidad del Cauca.
- Herrera, C. (2018). *Robótica educativa*. Bogotá: Universidad Minuto de Dios.
- Iberica.es. (2018). <https://www.pce-iberica.es/instrumentos-de-medida/sistemas/sensores.htm>. Obtenido de Sensores.
- Ingenio.com. (10 de Enero de 2018). *¿Qué es el Hardware Libre?* Obtenido de <https://www.softeingenio.com/tecnologia/hardware/51-hardware-industria/95-hardwarelibre.html>.
- Mayorga, M. E. (2015). Programación de mini robots para el desarrollo de aprendizaje significativo. *Pistas Educativas, No. 112, Noviembre 2015. México, Instituto Tecnológico de Celaya.* .

- Ministerio de Tecnologías de la Información y las Comunicaciones. (13 de Febrero de 2015). *Marco Jurídico, Subsistema de innovación para el uso y apropiación de TIC en el Gobierno*. Obtenido de [http://vivedigital.gov.co/idi/wp-content/uploads/2012/07/Marco\\_Juridico\\_V\\_2\\_0\\_0.pdf](http://vivedigital.gov.co/idi/wp-content/uploads/2012/07/Marco_Juridico_V_2_0_0.pdf): [http://vivedigital.gov.co/idi/wp-content/uploads/2012/07/Marco\\_Juridico\\_V\\_2\\_0\\_0.pdf](http://vivedigital.gov.co/idi/wp-content/uploads/2012/07/Marco_Juridico_V_2_0_0.pdf)
- Odorico, A. (2004). Marco teórico para una robótica pedagógica. *Revista de Informática Educativa y Medios Audiovisuales*, 34-46.
- Ogata, K. (2017). *Ingeniería de Control Moderna*. Canada: Tercera edición. Editorial Prentice Hall.
- Pinto Salamanca, M. L., & Pérez Holguín, W. J. (2010). Uso de la robótica educativa como herramienta en los procesos de enseñanza. *Bogotá: Universidad pedagógica y tecnológica de Colombia*.
- Ramírez, R., & Reyes, R. (2015). *Diseño e implementación de un robot autónomo móvil usando tecnología FPGA*. Ecuador: Universidad Salesiana.
- República de Colombia. (2011). *Ley 23 de 1982*. Bogotá.
- República de Colombia. (2012). *Constitución Política de Colombia*. Bogotá: Cupido.
- Sagal, K., & Mora, F. (2018). *Estudio de la aplicación didáctica del Lego Mindstorms NXT para la enseñanza de la robótica en la carrera de ingeniería en telecomunicaciones y electrónica automatizado*. Guayaquil: Universidad Católica de Santiago.
- Tapiero Santa, Y. D. (2019). *DISEÑO E IMPLEMENTACIÓN DE UN ROBOT SEGUIDOR DE LÍNEA DE COMPETENCIA PARA LA CATEGORÍA VELOCISTA*. Obtenido de Repositorio Universidad Ibagué: <https://repositorio.unibague.edu.co/bitstream/20.500.12313/1297/1/Trabajo%20de%20grado.pdf>
- Universidad San Buenaventura. (2019). *MATLAB para la Universidad*. Obtenido de <https://www.usbbog.edu.co/matlab/>.
- Yúbal, Y. (3 de Agosto de 2018). *Qué es Arduino, cómo funciona y qué puedes hacer con uno*. Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>.
- Zayas Barreras, I. (2010). El desarrollo tecnológico y la innovación como ente principal de competitividad en las empresas del sector agropecuario en el municipio de angostura, sinaloa. *Revista Mexicana de Agronegocios*, vol. 42, pp. 867-877, 2018.

# Apéndices

2021

# Manual de uso-Kit Robótico



Universidad Francisco  
de Paula Santander

## Contenido

<b>Introducción .....</b>	<b>88</b>
<b>Componentes del Kit.....</b>	<b>89</b>
<b>1. Board.....</b>	<b>90</b>
1.1 Diagrama de componentes.....	90
1.3 Resumen de Características.....	91
1.3 Características destacadas.....	92
1.4 Diagrama de Pines.....	93
1.5 Instalación.....	94
1.5.1 Instalación de Arduino.....	94
1.5.1 Conexión USB e instalación de controlador.....	94
1.6 Ejemplos de Uso .....	97
1.6.1 Led Pin 13 .....	97
1.6.2 Buzzer Integrado.....	100
1.6.3 Driver de Motor Integrado.....	101
<b>2. Estructura mecánica.....</b>	<b>103</b>
2.1 Plataforma todo terreno.....	103
2.2 Plataforma Híbrida.....	104
2.3 Plataforma seguidor de línea.....	105
<b>3. Ensamblaje de Robot Seguidor de Línea.....</b>	<b>106</b>
3.1 Elementos a utilizar.....	106
3.2 Conexiones.....	107
3.3 Estructura mecánica .....	108
3.4 Programación .....	109
3.5 Anotaciones Importantes .....	113
3.4.1 Rueda Loca .....	113
3.4.2 Tasa de descarga de batería.....	113
3.4.3 Qtr8-A .....	114
3.4.3 PID.....	115
3.4.4 Frenado Micromotores.....	119
3.6 Cómo hacer tu propia pista .....	119
3.7 Funcionamiento y recomendaciones.....	120

## Introducción

Este kit cuenta con todo lo necesario para que construyas un robot desde cero. Incluye una placa central para programar la funcionalidad de tu robot, diferentes tipos de chasises y módulos y componentes que puedes integrar en proyectos de tu gusto.

La placa UFPSO.INO es un controlador de robot todo en uno compatible con Arduino. Está diseñado especialmente para proyectos de robótica u otras soluciones tecnológicas.

Incluye potentes controladores de motor, reguladores, y componentes integrados. Todo está en una placa para muchos proyectos de robots de diferentes tipos.

Además de las herramientas tangibles, también cuenta con algunos materiales de apoyo, como códigos de inicio, librerías y un paso a paso de cómo hacer tu primer robot seguidor de línea.

## Componentes del Kit

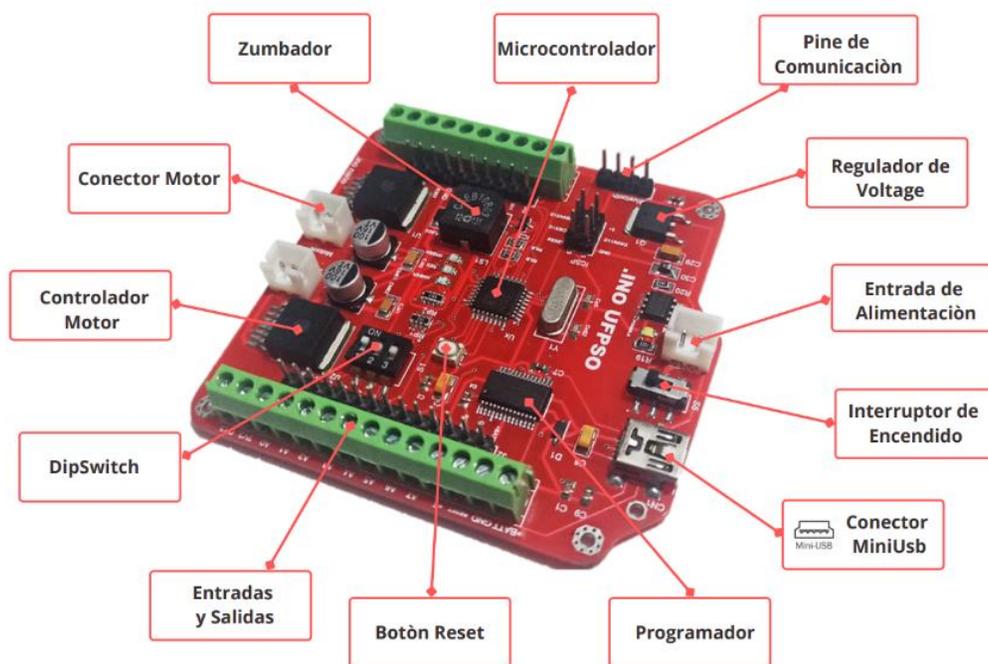
- Tarjeta controladora UFPSO.INO
- Set de Chasis x 4
- Micromotores x 2
- Motorreductores de plástico x2
- Ruedas todo terreno x 2
- Ruedas velocista x 2
- Ball caster de plástico x1
- Soporte para Micromotores x2
- Soporte para Motorreductores x 2
- Barra de Sensores QTR8-A
- Sensor ultrasonido HC-SR04
- Batería Lipo 7.4V 1500 mAh
- Cable Jumper macho-macho x 20
- Cables Jumper hembra-hembra x 20
- Tornillos 1.5cm 2mm x 10
- Conector JST hembra x 4
- Cinta doble faz 1.5m x 1
- Cargador batería Lipo

## 1. Board

A primera vista, si planeas hacer proyectos de robótica, hay muchas placas controladoras en el mercado, pero suelen no ser lo suficientemente prácticas o fáciles de usar. La placa UFPSO.INO posee ciertas características que te facilitan el proceso de ensamblaje de un robot autónomo.

Una de sus principales fortalezas es la presencia de componentes preensamblados que puedes usar directamente en un proyecto real.

### 1.1 Diagrama de componentes



*Figura 54.* Manual de Uso. Diagrama de Componentes. Fuente. Autores del proyecto.

## 1.1 Resumen de Características

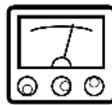
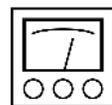
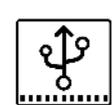
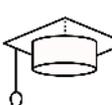
	<p><b>Microcontrolador</b> Atmega 328P-AU con Bootloader de Arduino Leonardo subido. Fácil desarrollo.</p>
	<p><b>Voltage Regulador</b> Regulador de Voltaje 78M05 para voltaje de entrada de 7-40V. Corriente máxima de salida 500mA 5V.</p>
	<p><b>Driver de Motor</b> Driver de motor TLE5206 para controlar hasta 2 motores con corriente de salida máxima de 5A. Voltaje de entrada de hasta 40V.</p>
	<p><b>Leds Indicadores</b> Leds indicadores para programador serial que permiten visualizar el estado de carga del código a la tarjeta. Led libre conectado al pin 13.</p>
	<p><b>Interruptor de encendido</b> Interruptor de encendido y apagado para detener fácilmente la alimentación de la tarjeta via batería.</p>
	<p><b>Dipswitch</b> 3 posiciones, 8 combinaciones digitales posibles.</p>
	<p><b>Capacitores</b> Capacitores cerámicos SMD distribuidos a lo largo de la placa.</p>
	<p><b>Protección de placa</b> Protección de polaridad de batería invertida, cortocircuito, sobrecalentamiento driver de motor.</p>
	<p><b>Conexión Mini USB</b> La tarjeta puede ser fácilmente programada utilizando el puerto Mini USB.</p>
	<p><b>Pads de Conexión</b> Los motores y las baterías pueden ser fácilmente conectados a través de los pads de conexión sin necesidad de soldarlos</p>
	<p><b>I/O Terminales</b> 17 pines de entrada y salida (I/O). 9 pines digitales; 6 pines que pueden ser usados como digitales y analógicos y 2 exclusivamente analógicos.</p>
	<p><b>Conectores Pines</b> Salida doble de pines para mayor flexibilidad al realizar las conexiones. Terminales de bloque con tornillo y pines macho 2.54mm.</p>
	<p><b>Buzzer</b> Buzzer piezoeléctrico integrado en la placa, conectado al pin digital D4 a través del cual puede ser utilizado.</p>

**Peso:** 25 gramos

**Dimensiones:** 70mm x 75mm x 11mm

*Figura 55.* Manual de Uso. Resumen características. Fuente. Autores del proyecto.

### 1.3 Características destacadas

 <b>Dimensiones</b>	70mm x 75mm x 11mm
 <b>Voltaje de Entrada</b>	7v - 40v
 <b>Corriente Driver Motor</b>	5 Amperios X2
 <b>Corriente Regulador Voltaje</b>	500 mA
 <b>Conectividad USB</b>	Si (Mini USB)
 <b>Curva de Aprendizaje</b>	Fácil
 <b>Anti Cortocircuito-Ruido</b>	Si

*Figura 56.* Manual de Uso. Características destacadas. Fuente. Autores del proyecto.

- Tamaño reducido
- Alto amperaje de salida para motores
- Entrada de voltaje protegida para conexión inversa
- Fácil manejo
- Voltaje de entrada de hasta 40V
- Conectividad USB

## 1.2 Diagrama de Pines

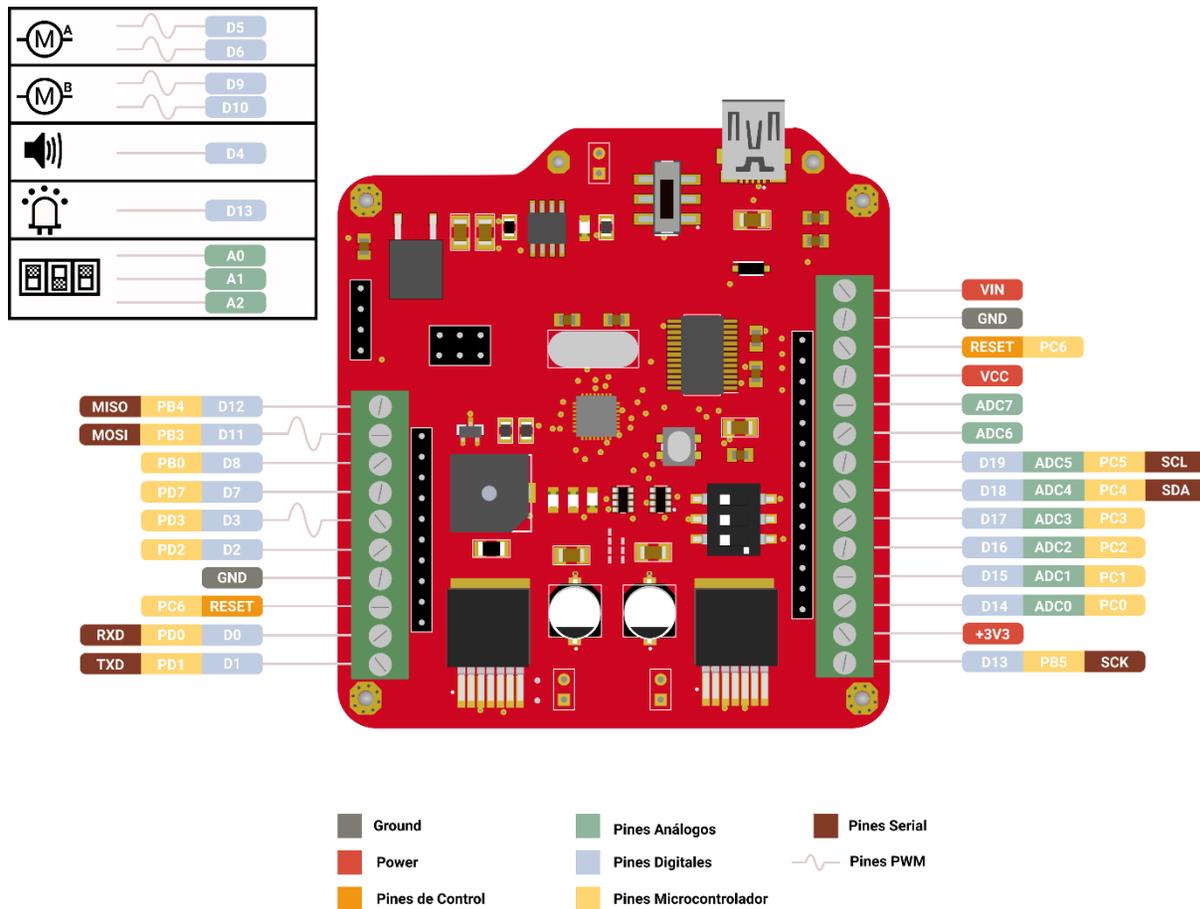


Figura 57. Manual de Uso. Diagrama de pines. Fuente. Autores del proyecto.

- 15 pines digitales
- 8 pines analógicos
- 2 pines PWM
- 3 pines dip Switch

## 1.5 Instalación

### 1.5.1 Instalación de Arduino

Para el uso y programación de la placa es necesario descargar e instalar en tu computador el IDE de Arduino. Este lo puedes descargar gratuitamente en la tienda de Windows o en la página oficial de Arduino <https://www.arduino.cc/en/software>.

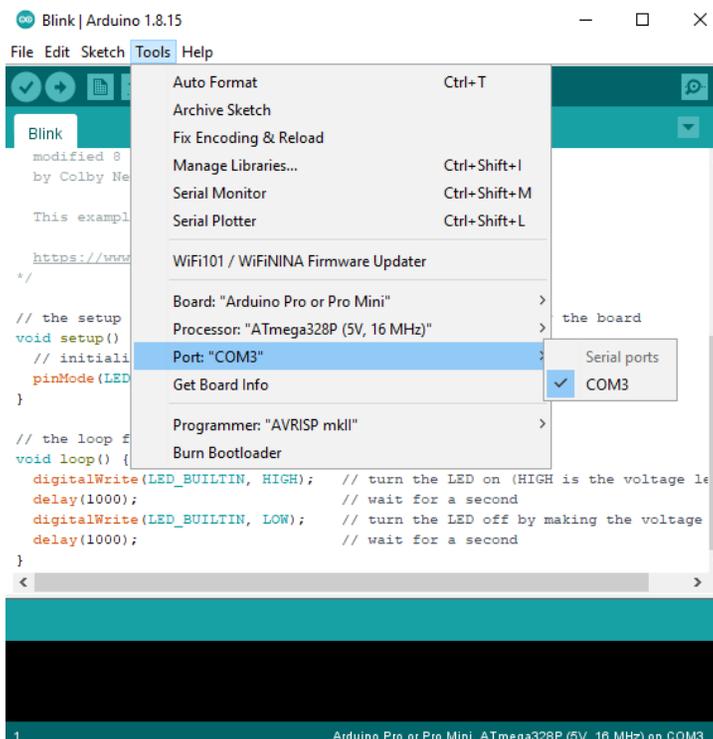
### 1.5.1 Conexión USB e instalación de controlador

Antes de poder hacer uso de la placa es necesario que la conectes a tu pc a través de un cable Mini-USB como el que se muestra en la figura.



**Figura 58.** Manual de Uso. Cable USB- Mini-USB. Fuente. <https://es.rs-online.com/web/p/cables-usb/1862805/>.

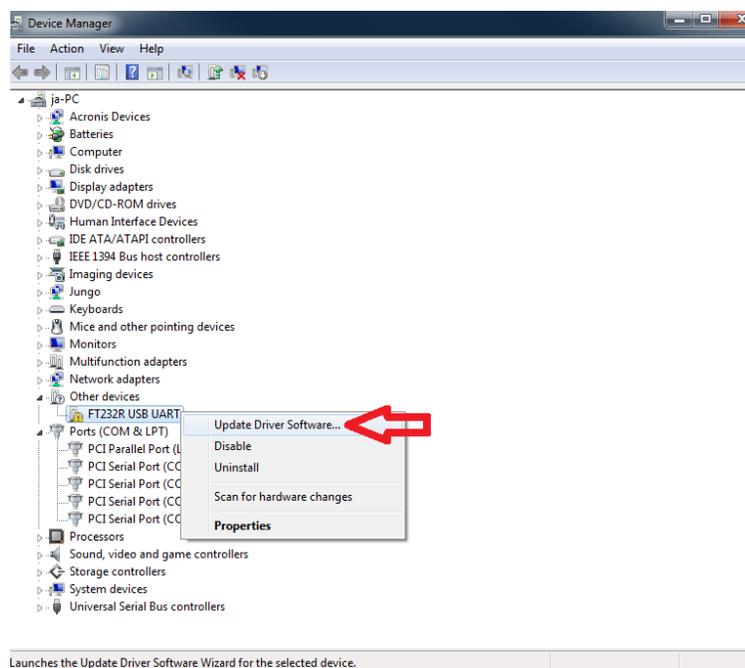
En algunos casos tu sistema operativo instalará por defecto el controlador de la placa. Puedes verificar fácilmente si la instalación fue exitosa abriendo el IDE de Arduino y observando si hay un puerto en uso. Además de ello en la parte inferior de la ventana aparecerá un mensaje referente a la placa detectada.



**Figura 59.** Manual de Uso. Detección de puerto Arduino. Fuente. Autores del proyecto.

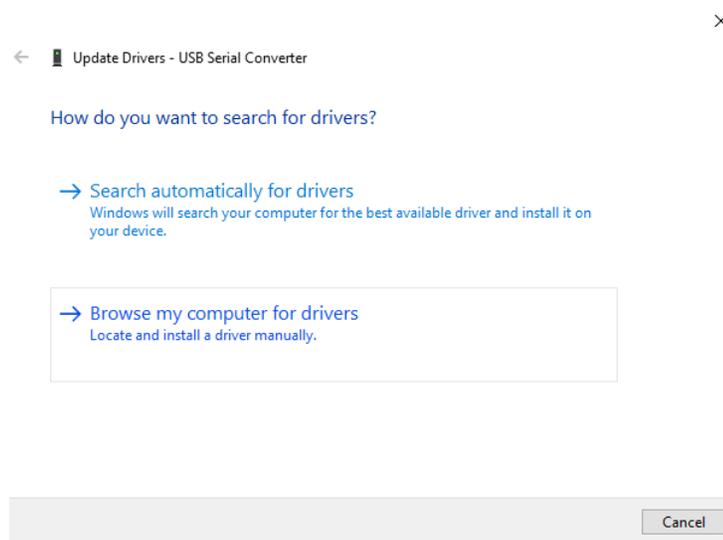
En caso de que tu placa no haya sido reconocida, debes instalar el controlador de forma manual como se muestra a continuación:

1. Descarga el controlador FTDI de la página oficial según tu sistema operativo  
<https://ftdichip.com/drivers/vcp-drivers/>
2. Abre el administrador de dispositivos y localiza el dispositivo FT232 USB UART



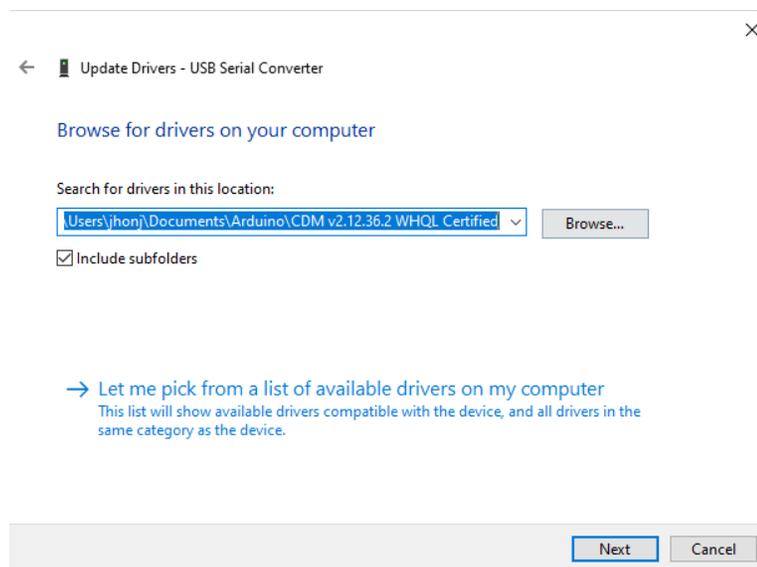
**Figura 60.** Manual de Uso. Administrador de dispositivos. Fuente. Autores del proyecto.

3. Seleccionar la opción de actualizar el controlador buscando el controlador en tu PC.



**Figura 61.** Manual de Uso. Actualización de controlador. Fuente. Autores del proyecto.

4. Selecciona la ubicación donde descargaste y descomprimiste anteriormente el controlador y marca la opción “Incluir subcarpetas”.



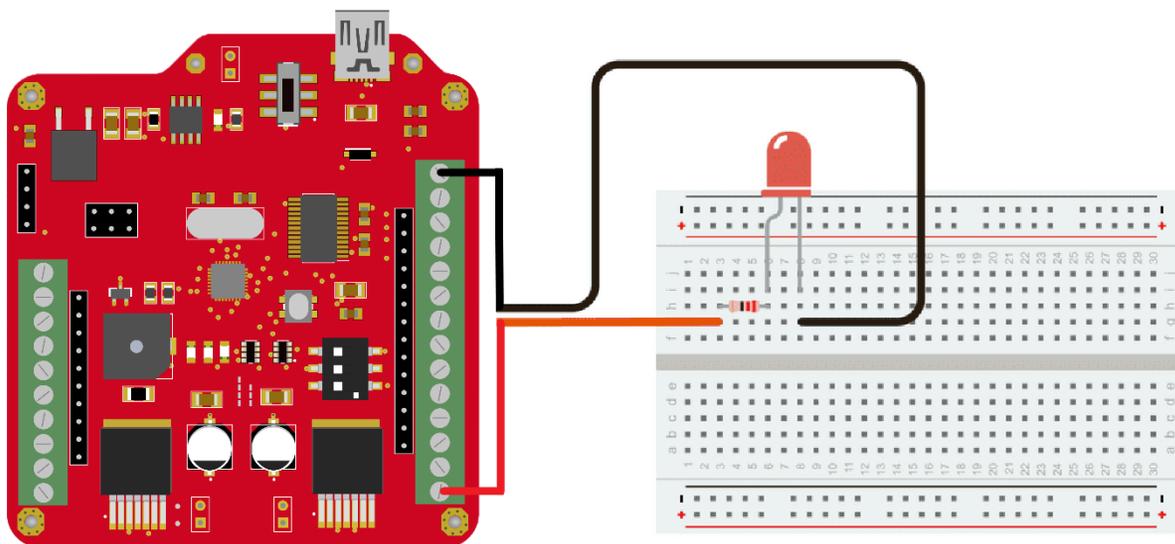
*Figura 62.* Manual de Uso. Selección de controlador. Fuente. Autores del proyecto.

5. La instalación del controlador debería realizarse correctamente y ahora tu tarjeta será reconocida por el IDE de Arduino. En caso de que aún no la detecte, puedes encontrar más ayuda en la siguiente dirección <https://www.usb-drivers.org/ft232r-usb-uart-driver.html>.

## 1.6 Ejemplos de Uso

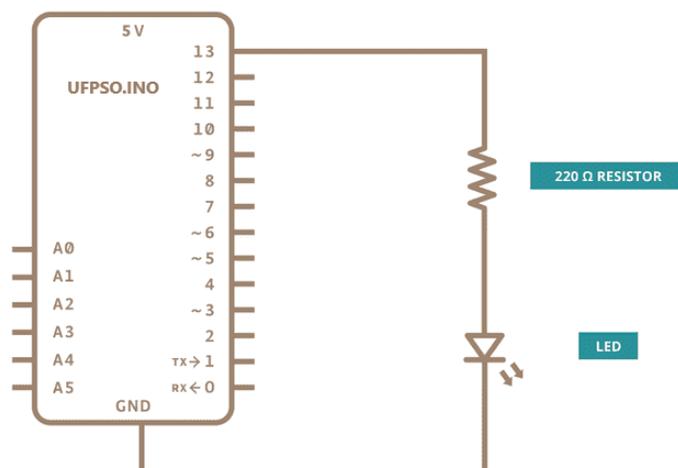
### 1.6.1 Led Pin 13

Para este primer ejemplo podemos usar el led integrado a la tarjeta o podemos conectar un led externo como se muestra en la figura. Si usas un led externo, asegúrate de conectar una resistencia de 220 Ohms para su protección.



*Figura 63.* Manual de Uso. Diagrama conexiones LED. Fuente. Autores del proyecto.

## Esquemático



*Figura 64.* Manual de Uso. Esquemático LED. Fuente. Autores del proyecto.

## Código

Luego de realizar las conexiones, inicia tu IDE de Arduino y escribe el código mostrado en la imagen.

Es importante entender a través de este sencillo ejemplo la estructura básica de un código en Arduino. Esencialmente tenemos dos funciones principales llamadas *void setup()* y *void loop()*. En donde *setup()* es la parte encargada de recoger la configuración y *loop()* es la que contiene el programa que se ejecutará cíclicamente.

La función de configuración debe contener la declaración de las variables, la inicialización de los pines que van a ser utilizados y la configuración de la comunicación serial, entre otros.

La función bucle contiene el código que se ejecutará continuamente y es la parte que realiza la mayor parte del trabajo.

```
LedPin13 §  
  
void setup() {  
  
    // Inicialización de pin 13 como pin de salida  
    pinMode(13, OUTPUT);  
  
}  
  
//Esta función es el ciclo principal que se ejecuta una y otra vez  
void loop() {  
  
    digitalWrite(13, HIGH); // Enciende el LED  
  
    delay(1000);           // Espera 1 segundo  
  
    digitalWrite(13, LOW); // Apaga el LED  
  
    delay(13);            // Espera 1 segundo  
  
}
```

**Figura 65.** Manual de Uso. Código Led Arduino. Fuente. Autores del proyecto.

*pinMode()*: Se encarga de inicializar el pin y establecerlo como una salida o entrada.

*digitalWrite()*: Envía una señal al pin, que puede ser baja (0 Voltios) o alta (5v).

*delay()*: Esta función detiene la ejecución del programa a la cantidad de ms indicada.

### 1.6.2 Buzzer Integrado

Para este ejemplo utilizaremos el buzzer integrado de la placa por lo que no necesitaremos ningún elemento externo

#### Código

```
BuzzerPin4 $
// Asignación de pin 4 a la variable buzzer
int buzzer = 4;

void setup() {

    // Inicialización de pin 4 como pin de salida
    pinMode(buzzer, OUTPUT);

}

//Esta función es el ciclo principal que se ejecuta una y otra vez
void loop() {

    digitalWrite(buzzer, HIGH); // Enciende el buzzer

    delay(1000);                // Espera 1 segundo

    digitalWrite(buzzer, LOW);  // Apaga el buzzer

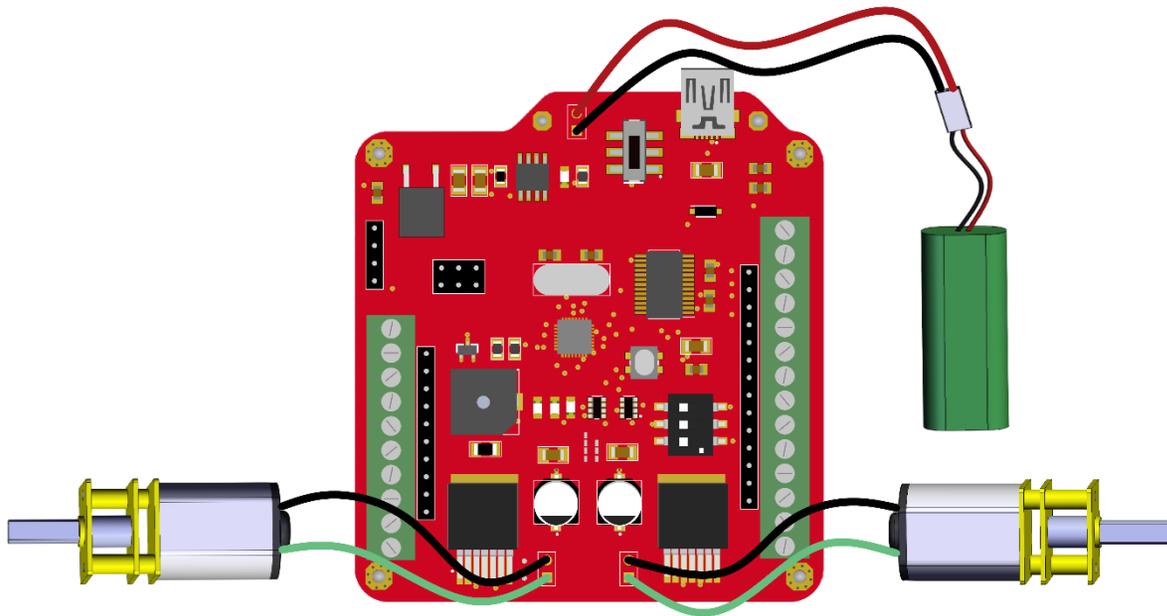
    delay(1000);                // Espera 1 segundo

}
```

**Figura 66.** Manual de Uso. Código Buzzer Arduino. Fuente. Autores del proyecto.

El buzzer se encuentra conectado internamente al pin 4 de la placa, siempre que necesitemos hacer uso de él, podemos configurarlo llamando a este pin.

### 1.6.3 Driver de Motor Integrado



*Figura 67.*Manual de Uso. Conexiones drivers de motor. Fuente. Autores del proyecto.

Esta tarjeta cuenta con dos controladores de motor altamente eficientes que puedes usar para todo tipo de robots. Además de ello cuenta con conectores especiales que permiten fácilmente enchufar y desenchufar las terminales de los motores.

Los motores utilizados para este tipo de proyectos robóticos deben tener una “caja reductora” acoplada puesto que sin ella la fuerza de torque del motor sería muy poca para mover cualquier estructura. Igualmente es necesario utilizar una batería externa con una potencia suficiente para alimentar todo el circuito.

## Código

```

Motores §

#define MotorA1 5 // Pin 5 PWM Motor N1
#define MotorA2 6 // Pin 6 PWM Motor N1
#define MotorB1 9 // Pin 9 PWM Motor N2
#define MotorB2 10 // Pin 10 PWM Motor N2

void setup()
{
  Serial.begin(9600); // Configuración de monitor serial para visualizar velocidad
  pinMode(MotorA1, OUTPUT); //Configuración de pines asociados a cada motor
  pinMode(MotorA2, OUTPUT);
  pinMode(MotorB1, OUTPUT);
  pinMode(MotorB2, OUTPUT);

  digitalWrite(MotorA1, LOW); //Estado inicial de los motores en OFF
  digitalWrite(MotorA2, LOW);
  digitalWrite(MotorB1, LOW);
  digitalWrite(MotorB2, LOW);
}

void MotorA_Forward(int i) { //Función Adelante Motor N1
  Serial.println(i);
  analogWrite(MotorA1, i);
  digitalWrite(MotorA2, LOW);
}

void MotorB_Forward(int i) { //Función Adelante Motor N2
  analogWrite(MotorB1, i);
  digitalWrite(MotorB2, LOW);
}

void MotorA_Backward(int i) { //Función de retroceso Motor N1
  digitalWrite(MotorA1, LOW);
  analogWrite(MotorA2, i);
}

void MotorB_Backward(int i) { //Función de retroceso Motor N2
  digitalWrite(MotorB1, LOW);
  analogWrite(MotorB2, i);
}

void loop()
{
  for (int i = 0; i <= 125; i++){ // Variación de Velocidad Motor N1 de 0 a 125
    MotorA_Forward(i);
    delay(100);
  }

  for (int i=125; i>=0; i--) { // Variación de Velocidad Motor N2 de 0 a 125
    MotorA_Forward(i);
    delay(100);
  }
}

```

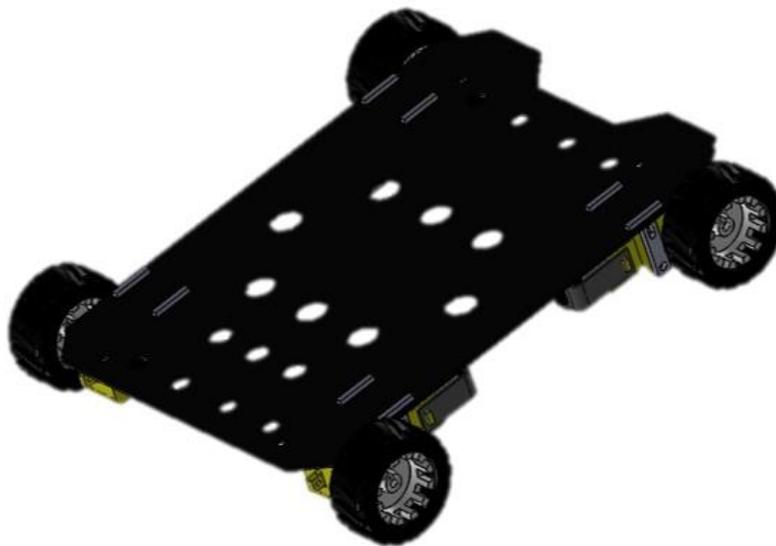
**Figura 68.** Manual de Uso. Código drivers de motor. Fuente. Autores del proyecto.

***analogWrite()***: Esta instrucción permite hacer uso de la tecnología PWM, a través de la cual simulamos una señal analógica que luego es enviada al pin indicado. El rango de valores que podemos utilizar en esta instrucción va desde 0 a 255 y simularán una tensión de entre 0V y 5V.

El lenguaje Arduino está basado en C/C++ y es posible utilizar diferentes estructuras lógicas y funciones propias del lenguaje. Puedes acceder a la guía de referencia dirigiéndote a <https://www.arduino.cc/reference/en/>

## 2. Estructura mecánica

### 2.1 Plataforma todo terreno



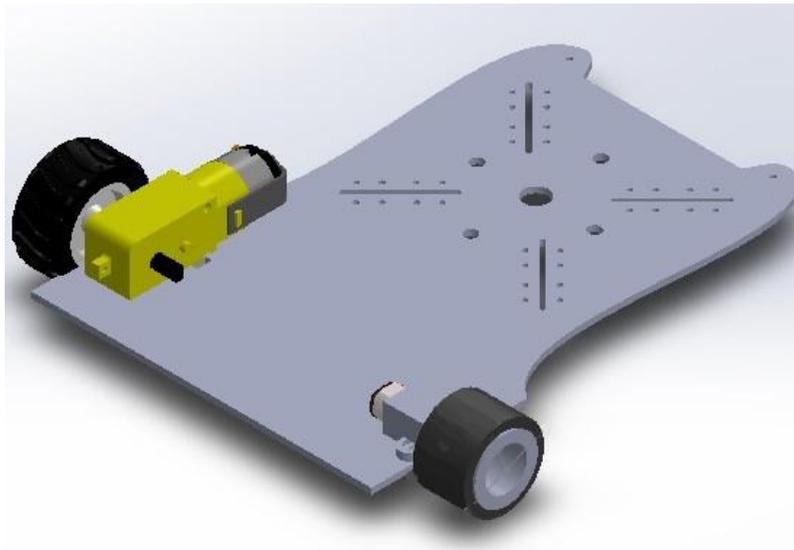
**Figura 69.** Manual de Uso. Plataforma todo terreno. Fuente. Autores del proyecto.

Esta plataforma tiene un diseño multipropósito con diferentes tipos de perforaciones que facilitan el acople de múltiples componentes que se quieran conectar al prototipo. Su estructura facilita en gran medida el manejo de cables al tener diferentes agujeros por donde pueden pasar.

Cuenta con ranuras especiales para el acople de motorreductores de plástico y llantas todo terreno de 65mm, pensado para armar ejemplares más resistentes y que puedan transitar por superficies más ásperas.

Adicional a esta también se diseñó una variante de este modelo con igual forma y tamaño, pero con modificación en los tipos de ranura y en la ubicación de estos mismos, esto con el objetivo de brindarte más opciones a la hora de crear tu propio robot.

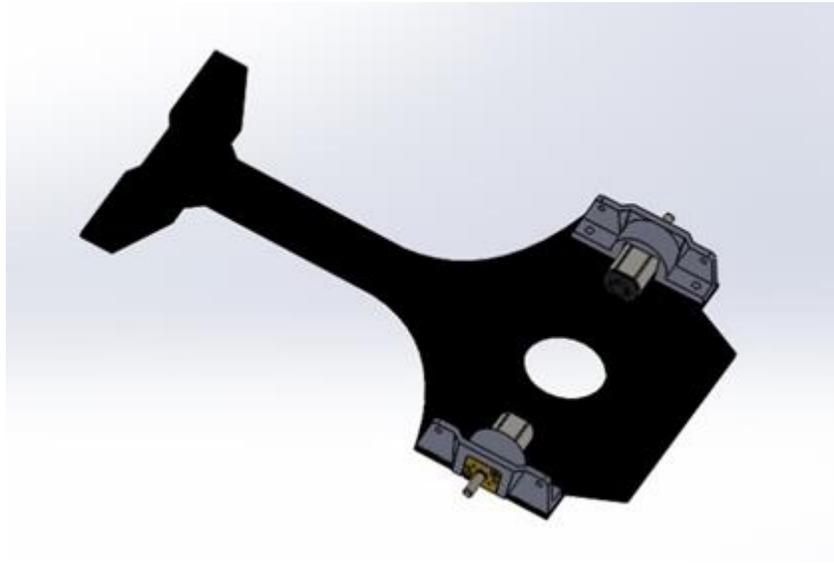
## 2.2 Plataforma Híbrida



*Figura 70.* Manual de Uso. Plataforma híbrida. Fuente. Autores del proyecto.

Esta plataforma es una mezcla entre las plataformas todo terreno y velocista. Permite el acople de dos motores que pueden ser micromotores o motorreductores de plástico. Cuenta con ranuras de diferentes formas y tamaños que le permiten acoplar diferentes elementos externos.

### 2.3 Plataforma seguidor de línea



*Figura 71.* Manual de Uso. Plataforma seguidora de línea. Fuente. Autores del proyecto.

Esta plataforma se creó con el fin de ensamblar un robot velocista seguidor de línea, robot cuyo objetivo es recorrer un circuito de línea negra sobre una superficie blanca.

En el diseño de esta estructura se tiene en cuenta algunos principios para mejorar el funcionamiento de este robot, entre los cuales se pueden destacar:

- Los sensores de línea se sitúan al frente del robot para que este pueda captar con anticipación las señales de luz correspondiente a la dirección de la línea negra. Esto mejora significativamente la velocidad de respuesta del robot y evita que este se salga de la línea cuando su velocidad es alta.
- Los elementos pesados como son la batería y la placa base se sitúan en una zona central para mejorar la estabilidad del robot y evitar que pierda el control en las curvas.

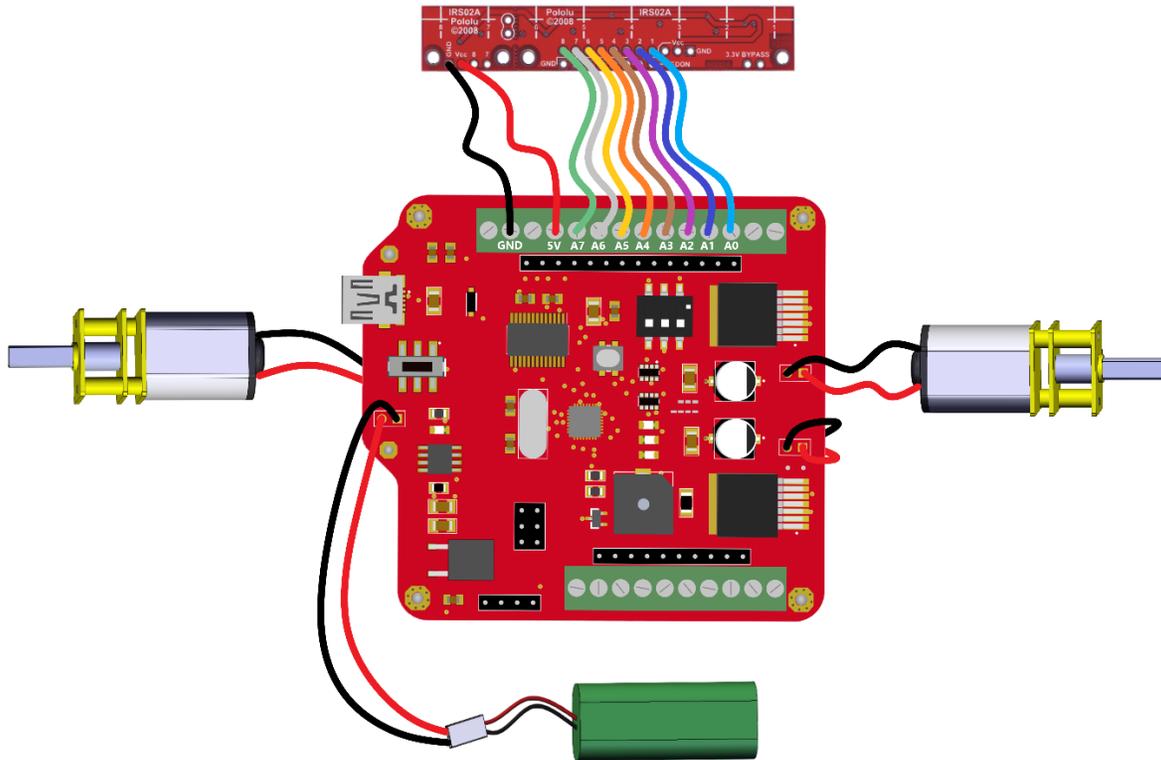
- La distancia de los sensores y de la estructura en general se sitúan lo más cercanos al suelo para mejorar aún más la estabilidad del robot.

### **3. Ensamblaje de Robot Seguidor de Línea**

#### **3.1 Elementos a utilizar**

1. Tarjeta controladora UFPSO.INO
2. Chasis velocista
3. Micromotores x 2
4. Ruedas velocista x 2
5. Ball caster de plástico x 1
6. Soporte para Micromotores x2
7. Barra de Sensores QTR8-A
8. Batería Lipo 7.4V 2000 mAh
9. Cables Jumper hembra-hembra x 10
10. Tornillos 1.5cm 2mm x 8
11. Cinta doble faz 1.5m x 1

### 3.2 Conexiones



*Figura 72.* Manual de Uso. Conexiones seguidor de línea. Fuente. Autores del proyecto.

Para armar este robot seguidor de línea las conexiones son muy sencillas puesto que la mayoría de los elementos necesarios para su funcionamiento ya se encuentran integrados a la placa. Tendremos 4 elementos principales:

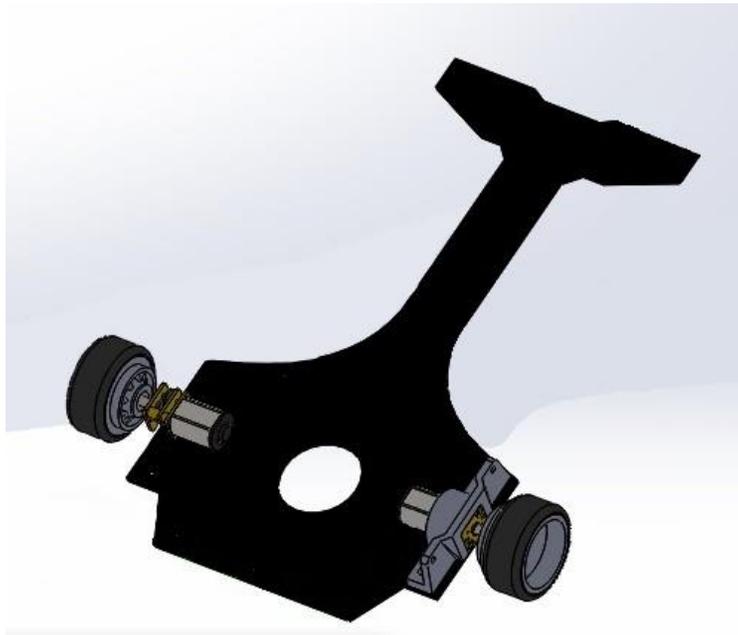
- **Tarjeta UFPSO.INO:** encargada de procesar todas las señales, aplicar el algoritmo de control y enviar las señales de salida a los motores.
- **Barra de sensores Qtr8-A:** Tarjeta equipada de 8 sensores detectores de línea que serán los encargados de capturar las señales referentes a la ubicación de la línea.
- **Micromotores:** Motores con caja reductora integrada caracterizados por su pequeño tamaño y alto rendimiento, muy útiles cuando el peso es una limitante.

- **Batería Lipo de alto rendimiento:** Batería de 7.4V con alta tasa de descarga que puede suministrar la potencia necesaria a un robot exigente.

### 3.3 Estructura mecánica

Para armar la estructura mecánica del robot puedes disponer del chasis velocista, a continuación, deberás acoplar los motores al chasis utilizando los soportes para micromotores y cuatro pares de tornillos de 2mm.

En cuanto a las ruedas, utilizaremos las ruedas para pista de goma, que tienen como característica un agarre bastante alto que impide deslizamientos del robot cuando la velocidad aumenta y hay curvas cerradas en la pista.



*Figura 73.* Manual de Uso. Estructura mecánica seguidor de línea. Fuente. Autores del proyecto.

### 3.4 Programación

```

RobotVelocista
#include <QTRSensors.h>

//Configuración pines driver de motor
#define MOTORLEFT_DIR_A 5
#define MOTORLEFT_DIR_B 6
#define MOTORRIGHT_DIR_A 9
#define MOTORRIGHT_DIR_B 10

#define NUM_SENSORS          6 // Número de sensores usados
#define NUM_SAMPLES_PER_SENSOR 4 // Muestras por sensor
#define EMITTER_PIN          2 // Emisor de Qtr8-A es controlado por el pin 2

#define LEDPIN      13 // Led integrado conectado a pin 13
#define BUTTONPIN   3 // Pulsador conectado al pin 3

#define esperarBoton() while(!digitalRead(BUTTONPIN)); while(digitalRead(BUTTONPIN))

// Configuración inicial de barra de sensores Qtr8-A
QTRSensorsAnalog qtra((unsigned char[]) {A0, A1, A2, A3, A4, A5, A6, A7},
    NUM_SENSORS, NUM_SAMPLES_PER_SENSOR, EMITTER_PIN);
unsigned int sensorValues[NUM_SENSORS];

//Función para girar hacia la izquierda
void setMotorLeft(int value)
{
    if ( value >= 0 )
    {
        analogWrite(MOTORRIGHT_DIR_A,value);
        digitalWrite(MOTORRIGHT_DIR_B,LOW);
    }
    else
    {
        digitalWrite(MOTORRIGHT_DIR_A,LOW);
        analogWrite(MOTORRIGHT_DIR_B,value);
        value *= -1;
    }
}
}

```

*Figura 74.* Manual de Uso. Código seguidor de línea P1. Fuente. Autores del proyecto.

```

// Función para girar hacia la derecha
void setMotorRight(int value)
{
  if ( value >= 0 )
  {
    analogWrite(MOTORLEFT_DIR_A,value);
    digitalWrite(MOTORLEFT_DIR_B,LOW);
  }
  else
  {
    digitalWrite(MOTORLEFT_DIR_A,LOW);
    analogWrite(MOTORLEFT_DIR_B,value);
    value *= -1;
  }
}

// Función de arranque para motores
void setMotors(int left, int right)
{
  setMotorLeft(left);
  setMotorRight(right);
}

// Función de frenado de motores
void setBrake(boolean left, boolean right, int value)
{
  if ( left )
  {
    analogWrite(MOTORRIGHT_DIR_A,value);
    analogWrite(MOTORRIGHT_DIR_B,value);
  }

  if ( right )
  {
    analogWrite(MOTORLEFT_DIR_A,value);
    analogWrite(MOTORLEFT_DIR_B,value);
  }
}

```

**Figura 75.** Manual de Uso. Código seguidor de línea P2. Fuente. Autores del proyecto.

```

void setup()
{
  //Configuración de pines de salida y entrada
  pinMode(LEDPIN          ,OUTPUT);
  pinMode(MOTORRIGHT_DIR_A ,OUTPUT);
  pinMode(MOTORRIGHT_DIR_B ,OUTPUT);
  pinMode(MOTORLEFT_DIR_A  ,OUTPUT);
  pinMode(MOTORLEFT_DIR_B  ,OUTPUT);
  pinMode(BUTTONPIN        ,INPUT);
  digitalWrite(BUTTONPIN,HIGH);

  //Espera a que el pulsador sea accionado para empezar la calibración
  while ( digitalRead(BUTTONPIN) );

  //Calibración QTR8-A
  for ( int i=0; i<70; i++)
  {
    digitalWrite(LEDPIN, HIGH); delay(20);
    qtra.calibrate();
    digitalWrite(LEDPIN, LOW);  delay(20);
  }

  digitalWrite(LEDPIN, LOW);

  //Espera a que el pulsador sea nuevamente accionado para iniciar el robot
  while ( digitalRead(BUTTONPIN) );

  delay(500); //Tiempo de espera antes de arrancar

  setMotors(90, 90);

  delay(300);
}
//Inicialización de constantes de algoritmo PID
unsigned int position = 0;
int derivative = 0;
int proportional = 0;
int power_difference = 0;

int max = 90;

int last_proportional;

float KP = 0.15;
float KD = 2.2;

#define RANGEBRAKE 3500

```

*Figura 76.* Manual de Uso. Código seguidor de línea P3. Fuente. Autores del proyecto.

```

//Bucle principal
void loop()
{
  //Posición de la línea
  position = qtra.readLine(sensorValues);

  //Cálculo de posición relativa
  proportional = ((int)position) - 3500;

  //Activación de función de frenado si la línea se encuentra a los extremos
  if ( proportional <= -RANGEBRAKE )
  {
    setMotorRight(0);
    setBrake(true,false,255);
    delay(1);
  }
  else if ( proportional >= RANGEBRAKE )
  {
    setMotorLeft(0);
    setBrake(false,true,255);
    delay(1);
  }

  //Aplicación de algoritmo PID para encontrar el error con respecto
  //al centro de la línea
  derivative = proportional - last_proportional;

  last_proportional = proportional;

  int power_difference = ( proportional * KP ) + ( derivative * KD );

  if ( power_difference > max ) power_difference = max;
  else if ( power_difference < -max ) power_difference = -max;

  ( power_difference < 0 ) ?
    setMotors(max+power_difference, max) : setMotors(max, max-power_difference);
}

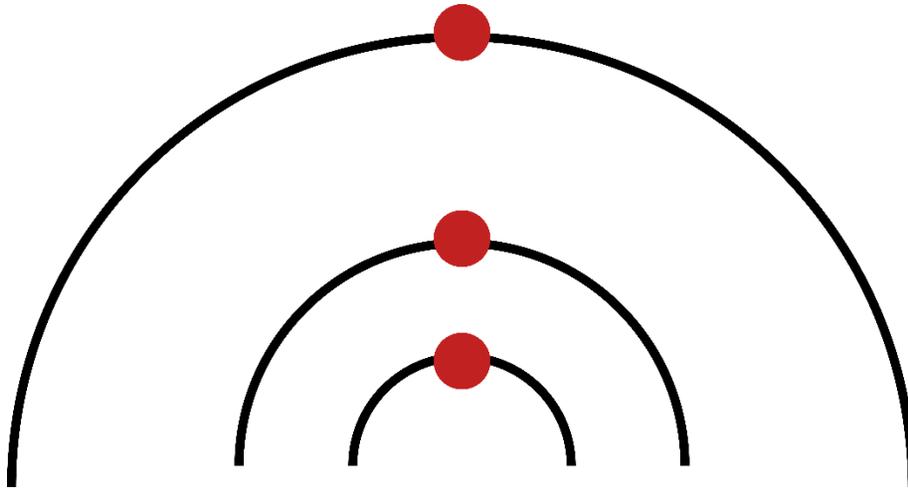
```

*Figura 77.* Manual de Uso. Código seguidor de línea P4. Fuente. Autores del proyecto.

## 3.5 Anotaciones Importantes

### 3.4.1 Rueda Loca

Es necesario ubicar una rueda loca en la cabeza del robot para que facilite el giro y movimiento del robot, sin embargo, se debe tener en cuenta que, a mayor distancia entre los motores principales y la rueda loca, mayor es la distancia que recorre en un giro determinado.



*Figura 78.* Manual de Uso. Rueda loca. Fuente. Autores del proyecto.

### 3.4.2 Tasa de descarga de batería

La tasa de descarga nos indica cuanta corriente puede proveer la batería en un determinado momento. Esta debe ser alta puesto que los micro reductores a máxima velocidad tendrán un consumo de energía bastante alto y una batería normal no podría suministrársela. Además de ello el uso de una batería con poca tasa de descarga no podrá aprovechar al máximo las características de la tarjeta, la cual está diseñada para soportar consumos altos de corriente.

Para calcular la tasa de descarga puedes usar esta fórmula:

*Tasa de descarga = Capacidad de la batería \* Razón de descarga*

La batería presente en el kit tiene una capacidad de 1500mAh y una razón de descarga de 25c, por lo tanto, su tasa de descarga será de 7200 mA.

Para este robot el consumo máximo teórico es alrededor de 3500mA, de esta forma podemos estar seguros de que no habrá cuellos de botella con relación a la batería.

### **3.4.3 Qtr8-A**

El sensor Qtr8-A es ampliamente utilizado en la elaboración de robots seguidores de línea por su forma compacta que integra más de 6 sensores individuales, además de ello cuenta con una librería oficial que facilita su uso y acoplamiento a un robot autónomo.

Estos sensores de reflectancia deben conectarse a los pines analógicos A0 a A7. El pin de control del emisor de los sensores LEDON se puede conectar opcionalmente a un pin digital, o se puede dejar desconectado y eliminar la llamada a setEmitterPin ().

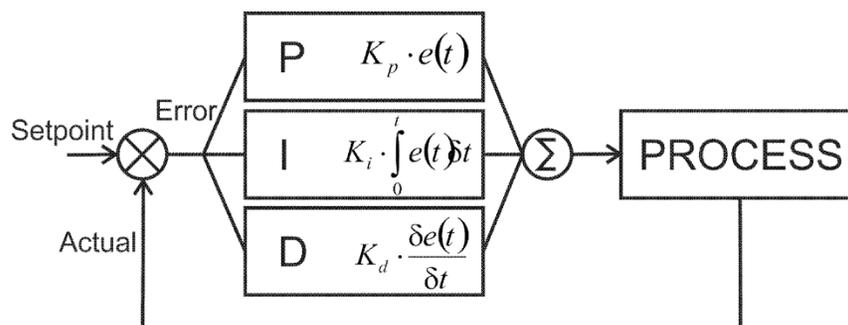
Para su funcionamiento correcto es necesario llamar a una función de calibración. Durante esta fase, es necesario exponer cada sensor de reflectancia a las lecturas más claras y oscuras que haya. Por ejemplo, si se está haciendo un seguidor de línea, se debe deslizar los sensores a lo largo de la línea durante la fase de calibración para que cada sensor pueda obtener una lectura de qué tan oscura es la línea y qué tan claro es el suelo. Una calibración incorrecta resultará en lecturas deficientes.

La librería de estos sensores permite leer los valores del sensor calibrado y usarlos para estimar la posición de una línea. Esta ubicación estimada está representada como un número de 0 a 5000.

### 3.4.3 PID

El PID (control proporcional, integral y derivativo) es un mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor que se quiere obtener (set point, target position o posición objetivo), para aplicar una acción correctora que ajuste el proceso.

En el caso del robot velocista, el controlador PID, (que es una rutina basada matemáticamente), procesará los datos del sensor, y los utilizará para controlar la dirección (velocidad de cada motor), para de esta forma mantenerlo en curso.



**Figura 79.** Manual de Uso. PID. Fuente. Autores del proyecto.

**Error:** Llamamos error a la diferencia entre la posición objetivo y la posición medida del error (qué tan lejos del punto de consigna se encuentra el sensor, en nuestro caso el objetivo es tener los sensores centrados sobre la línea).



*Figura 80.* Manual de Uso. Corrección de error PID. Fuente. Autores del proyecto.

**Posición objetivo:** Este es el punto en el cual el error es cero. En el caso del robot velocista el valor que indica un error cero es 3500, correspondiente a la lectura del sensor con la línea centrada. Este es el punto ideal, siempre buscaremos alinear el robot en la línea.

### Parámetros

- **Proporcional:** Es la respuesta al error que se tiene que entregar de manera inmediata, es decir, si nos encontramos en el centro de la línea, los motores tendrán en respuesta una velocidad de igual valor, si nos alejamos del centro, uno de los motores reducirá su velocidad y el otro aumentará.

$$\text{Proporcional} = (\text{posición}) - \text{posición objetivo}$$

- **Integral:** La integral es la sumatoria de los errores acumulados, tiene como propósito el disminuir y eliminar el error en estado estacionario provocado por el modo proporcional, en otras palabras, si el robot velocista se encuentra mucho tiempo alejado del centro (ocurre muchas veces cuando se encuentra en curvas), la acción integral se irá acumulando e irá disminuyendo el error hasta llegar al punto de consigna,

$$\text{Integral} = \text{Integral} + \text{proporcional pasado}$$

- Derivativo: Es la derivada del error, su función es mantener el error al mínimo, corrigiéndolo proporcionalmente con la misma velocidad que se produce, de esta manera evita que el error se incremente, en otra palabra, anticipará la acción, evitando así las oscilaciones excesivas.

$$\text{Derivativo} = \text{proporcional} - \text{proporcional pasado}$$

### Constantes

- Factor (Kp) - Es un valor constante utilizado para aumentar o reducir el impacto de proporcional. Si el valor es excesivo, el robot tenderá a responder inestablemente, oscilando excesivamente. Si el valor es muy pequeño, el robot responderá muy lentamente, tendiendo a salirse de las curvas.
- Factor (Ki) - Es un valor constante utilizado para aumentar o reducir el impacto de la Integral, El valor excesivo de este provocará oscilaciones excesivas. Un valor demasiado bajo no causará impacto alguno.
- Factor (Kd) - Es un valor constante utilizado para aumentar o reducir el impacto de la Derivada. Un valor excesivo provocará una sobre amortiguación, provocando a su vez inestabilidad.

$$\text{salida\_pwm} = (\text{proporcional} * Kp) + (\text{derivativo} * Kd) + (\text{integral} * Ki);$$

## Calibración PID

Aquí viene el reto, la calibración PID, es aquí donde se tendrá que buscar las constantes que correspondan a las características físicas del robot, la forma más fácil de hacerlo es por ensayo y error, hasta obtener el valor deseado. Aquí hay unos pasos que ayudarán mucho a buscar esas constantes:

1. Comience con  $K_p$ ,  $K_i$  y  $K_d$  igualando 0 y trabajar con  $K_p$  primero. Pruebe establecer  $K_p$  a un valor de 1 y observar el robot. El objetivo es conseguir que el robot siga la línea, incluso si es muy inestable. Si el robot llega más allá y pierde la línea, reducir el valor de  $K_p$ . Si el robot no puede navegar por una vez, o parece lenta, aumente el valor  $K_p$ .
2. Una vez que el robot es capaz de seguir un poco la línea, asignar un valor de 1 a  $K_d$ . Intente aumentar este valor hasta que vea menos oscilaciones.
3. Una vez que el robot es bastante estable en la línea siguiente, asigne un valor de 0,5 a 1,0 a  $K_i$ . Si el valor de  $K_i$  es demasiado alto, el robot se sacudirá izquierda y derecha rápidamente. Si es demasiado baja, no se verá ninguna diferencia perceptible. El Integral es acumulativo por lo tanto el valor  $K_i$  tiene un impacto significativo. puede terminar ajustando por 0,01 incrementos.
4. Una vez que el robot está siguiendo la línea con una buena precisión, se puede aumentar la velocidad y ver si todavía es capaz de seguir la línea. La velocidad afecta el controlador PID y requerirá ajustar nuevamente las variables.

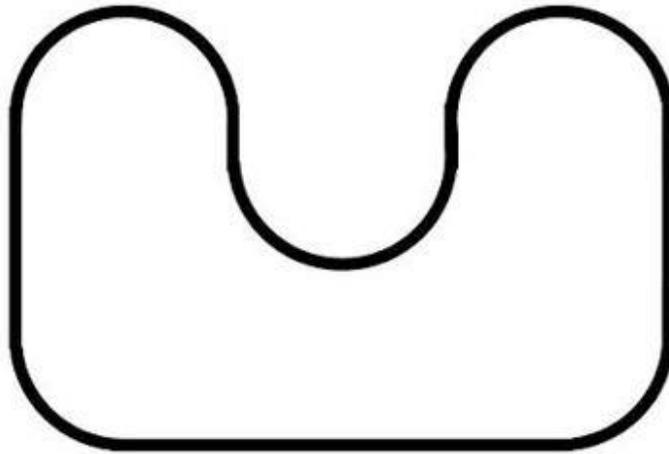
### **3.4.4 Frenado Micromotores**

Los frenos son sin duda alguna muy importantes cuando se trata de obtener un mayor control en los robots velocistas, nos permiten tener una segunda opción cuando nos salimos del rango de visión que nos muestran los sensores infrarrojos (estos eventos tienen mucha probabilidad de ocurrir en curvas obtusas o de 90°, haciendo de verdad muy difícil el control sin la opción de los frenos).

Cuando se aplican, es necesario tomar en cuenta que debemos hacerlo dándole dirección a la inercia que ocasiona el frenado hacia la toma correcta de la curva, para ello debemos frenar la rueda interna y liberar la rueda externa.

### **3.6 Cómo hacer tu propia pista**

Puedes construir tu propia pista de manera fácil utilizando un pliego de papel cartulina y cinta negra aislante. Solo debes procurar que las curvas no sean demasiado cerradas, de otro modo es probable que el robot se desvíe.



*Figura 81.* Manual de Uso. Pista. Fuente. Autores del proyecto.

### **3.7 Funcionamiento y recomendaciones**

Antes de poner a prueba la velocidad de tu robot asegúrate de seguir estas recomendaciones:

- Al iniciar encender tu robot la función de calibración automática se activará, por lo que te recomendamos que desplaces durante unos segundos la barra de sensores por la línea negra y la superficie blanca, de esta forma se establecerán los valores mínimos y máximos de lectura de los sensores.
- Asegúrate de mantener limpia la superficie de los sensores para evitar lecturas erróneas.
- Procura ajustar la distancia de los sensores lo más cercana al suelo posible, de esta manera las lecturas serán más precisas y menos vulnerables a luz del ambiente.