



Universidad  
Nacional  
de Loja

# Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos

Naturales no Renovables

Carrera de Ingeniería Electromecánica

Diseño, construcción y desarrollo de guías prácticas con DYOR

Trabajo de Titulación previo a la  
obtención del título de ingeniero  
electromecánico

**AUTOR:**

Jorge Miguel Chávez Retete

**DIRECTOR:**

Ing. Jefferson Fernando Camacho Muñoz, Mg. Sc.

Loja-Ecuador

2022

## Certificación

Loja, 19 de agosto de 2022

Ing. Jefferson Fernando Camacho Muñoz  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

### **CERTIFICO:**

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Titulación denominado: **Diseño, construcción y desarrollo de guías prácticas con DYOR**, previo a la obtención del título de **Ingeniero Electromecánico**, de la autoría de la estudiante **Jorge Miguel Chávez Retete**, con cédula de identidad Nro.**0705480192**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

:



Firmado electrónicamente por:  
**JEFFERSON  
FERNANDO CAMACHO  
MUNOZ**

---

**Ing. Jefferson Camacho Muñoz, MSc.**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## **Autoría**

Yo, **Jorge Miguel Chávez Retete**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual.

**Firma:**



**Cédula de Identidad:** 0705480192

**Fecha:** 02 de Diciembre del 2022

**Correo electrónico:** jorge.chavez@unl.edu.ec

**Teléfono:** 0988218760

**Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Titulación.**

Yo, **Jorge Miguel Chávez Retete**, declaro ser autor del Trabajo de Titulación denominado: **Diseño, construcción y desarrollo de guías prácticas con DYOR**, como requisito para optar por el título de **Ingeniero Electromecánico**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional. Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los dos días del mes de diciembre del dos mil veintidós.

**Firma:**



**Autor:** Jorge Miguel Chávez Retete

**Cédula:** 0705480192

**Dirección:** Loja, (ciudad de loja)

**Correo electrónico:** jorge.chavez@unl.edu.ec

**Telefono:** 0988218760

**DATOS COMPLEMENTARIOS:**

**Director del Trabajo de Titulación:** Ing. Jefferson Fernando Camacho Muñoz, Mg. Sc.

## **Dedicatoria**

Mi Trabajo de Titulación la dedico con mucho amor y cariño.

A Dios que me dio la oportunidad de vivir y darme una hermosa familia.

Principalmente a mis padres que me dieron la vida y han estado conmigo en todo momento. Gracias por todo mamá y papá por darme una carrera para mi futuro y por tener la confianza en mí, a pesar de que en la vida hemos pasado por momentos complicados siempre han estado apoyándome y brindándome un amor sincero, por todo esto y más le agradezco de todo corazón el que estén conmigo en este momento de mi vida.

A mi hermana gracias por estar conmigo y apoyarme siempre, por ser mi guía y una motivación más al igual que mi sobrino para seguir adelante.

A mis amigos, compañeros y profesores que han formado parte de este periodo universitario que ayudaron a que la Universidad no sea tan difícil a veces.

*Jorge Miguel Chávez Retete*

## **Agradecimiento**

He llegado al final de mi Trabajo de Titulación y deseo agradecer con cariño y respeto a mis padres por haberme fomentado valores que me han ayudado en mis estudios y en la vida, y lo más importante por darme la vida, quienes me han dedicado la mayor parte de su tiempo, motivándome todos los días a mejorar como persona y en mi camino de formación personal como profesional, y así lograr ser una persona útil para la sociedad. Gracias a mi hermana por su apoyo moral y sincero que me ha brindado de forma incondicional. Principalmente agradecer a Dios por ayudarme a concretar este objetivo que a pesar de los problemas que se presentaron siempre me protegió y me llevo por el buen camino.

Quiero agradecer enormemente a mi tutor Ing. Jefferson Camacho Mg. quien me brindo su ayuda y tiempo para solventar dudas que se presentaron en el transcurso del trabajo y así lograr llegar a la meta de finalizar correctamente mi trabajo de tesis.

Agradezco a la Universidad Nacional de Loja del estado de Ecuador por haberme dado la oportunidad de estudiar y lograr alcanzar mi formación profesional. Para terminar, agradezco a mi familia, amigos, compañeros y maestros por haberme permitido coincidir en esta etapa de mi vida y sean parte de ella.

*Jorge Miguel Chávez Retete*

## Índice de contenido

<b>Portada</b> .....	i
<b>Certificación</b> .....	ii
<b>Autoría</b> .....	iii
<b>Carta de autorización</b> .....	iv
<b>Dedicatoria</b> .....	v
<b>Agradecimiento</b> .....	vi
<b>Índice de contenido</b> .....	vii
Índice de Tablas:.....	xi
Índice de las Figuras:.....	xii
Índice de Anexos .....	xv
<b>1. Título</b> .....	1
<b>2. Resumen</b> .....	2
2.1 Abstract.....	3
<b>3. Introducción</b> .....	4
<b>4. Marco Teórico</b> .....	6
4.1 Capítulo I.....	6
4.1.1 Motivación y Aprendizaje .....	6
4.1.1.1 Motivación.....	6
4.1.1.2 Aprendizaje.....	6
4.1.2 Motivación.....	6
4.1.2.1 Importancia de la motivación en el proceso de aprendizaje.....	6
4.1.3 Tipos.....	7
4.1.3.1 Aprendizaje.....	7
4.1.3.2 Motivación para el Aprendizaje. ....	7
4.1.4 Uso de tecnologías para la motivación del aprendizaje .....	8

4.1.4.1 La robótica como motivación en el proceso del aprendizaje.....	8
4.2 Capítulo II.....	9
4.2.1 Robótica educativa.....	9
4.2.1.1 Definición.....	9
4.2.1.2 Importancia.....	9
4.2.2 Tipos.....	9
4.2.2.1 DYOR.....	9
4.2.2.2 Robótica como potencia de aprendizaje.....	12
4.2.2.3 Robótica diseño de recursos educativos.....	13
4.2.2.4 Guías Prácticas (Aprendizaje).....	13
4.2.2.5 Instrumentos de Validación.....	15
4.3 Capítulo III.....	16
4.3.1 Elaboración del robot y las guías prácticas.....	16
4.3.1.1 Diseño Mecánico.....	16
4.3.1.2 Software de Ingeniería.....	16
4.3.2 Desarrollo de las guías prácticas.....	16
4.3.3 Componentes del robot DYOR.....	17
4.3.3.1 Placas de programación.....	17
4.3.3.2 Placa Arduino UNO.....	18
4.3.3.3 Power Bank / Batería 5v.....	20
4.3.3.4 Ultrasonido HC-SR04.....	21
4.3.3.5 Zumbador de sonido.....	22
4.3.3.6 Servo SG90.....	22
4.3.3.7 Servo Motores de rotación continua.....	23
4.3.3.8 Shields de Arduino.....	23
4.3.3.9 Bluetooth HC-06 y HC-05 Android Arduino.....	25



4.3.3.10	Siguelíneas TCRT5000.....	25
4.3.3.11	Rueda loca. ....	26
4.3.3.12	Matriz de Leds (8x8). ....	26
4.3.4	Entorno de desarrollo para Arduino .....	27
4.3.4.1	Estructura general de un programa Arduino.....	29
4.3.4.2	Bibliotecas para Arduino .....	31
4.3.4.3	Bibliotecas estándar para Arduino.....	32
5.	<b>Metodología</b> .....	34
5.1	Diseño del Robot DYOR.....	34
5.2	Construcción del Robot DYOR.....	36
5.2.1	Transformar un servo motor de posición a rotación continua.....	39
5.3	Conexión de los componentes con la placa Arduino UNO y el Shield.....	41
5.4	Programación.....	43
5.4.1	Arduino.....	43
5.4.1.1	Programación de los elementos y sensores.....	43
5.4.1.2	Programaciones del Robot DYOR. ....	49
5.5	Guías Prácticas .....	49
5.6	Validación de las Guías Prácticas.....	50
5.6.1	Ejercicios aplicados en el curso de programación.....	51
5.6.1.1	Encender un Led y ciclo for. ....	51
6.	<b>Resultados</b> .....	58
6.1	Guía Práctica de diseño del robot DYOR.....	58
6.2	Guías Prácticas de programaciones del robot.....	64
6.2.1	Guía Práctica del Sensor Siguelínea.....	65
6.2.2	Guía Práctica del sensor Ultrasónico .....	72
6.2.3	Guían Práctica del control a distancia del robot.....	79

6.3	Cuestionario aplicado a los estudiantes .....	87
7.	<b>Discusión</b> .....	99
8.	<b>Conclusiones</b> .....	101
9.	<b>Recomendaciones</b> .....	103
10.	<b>Bibliografía</b> .....	104
11.	<b>Anexos</b> .....	107

## **Índice de tablas:**

<b>Tabla 1.</b> Conceptualización .....	11
<b>Tabla 2.</b> Comparación entre algunas placas controladoras.....	17
<b>Tabla 3.</b> Bibliotecas comunes para Arduino.....	32
<b>Tabla 4.</b> Conexión de cada componente al Shield de la placa Arduino UNO.....	41
<b>Tabla 5.</b> Relación del ancho de Pulso con los ángulos de posición.....	44
<b>Tabla 6.</b> Relación del ángulo con el movimiento de giro.....	46

## Índice de las figuras:

<b>Figura 1.</b> Elementos de la placa Arduino Uno. ....	19
<b>Figura 2.</b> Power Bank. ....	20
<b>Figura 3.</b> Batería de 9V. ....	21
<b>Figura 4.</b> Sensor Ultrasonido HC-SR04. ....	21
<b>Figura 5.</b> Zumbador de sonido. ....	22
<b>Figura 6.</b> Servo Motor G90 (de posición). ....	23
<b>Figura 7.</b> Servo Motor FS90R (rotación continua). ....	23
<b>Figura 8.</b> Shield para servomotores. ....	24
<b>Figura 9.</b> Bluetooth HC-05 y el HC-06 (6 pines) y (4 pines). ....	25
<b>Figura 10.</b> Siguelínea TCRT5000. ....	26
<b>Figura 11.</b> Rueda Loca. ....	26
<b>Figura 12.</b> Matriz de Leds de 8x8. ....	27
<b>Figura 13.</b> Plataforma de Arduino. ....	28
<b>Figura 14.</b> Nuevo Sketch en Arduino. ....	28
<b>Figura 15.</b> Partes de una programación en Arduino. ....	29
<b>Figura 16.</b> Declaración de Variables. ....	30
<b>Figura 17.</b> Void setup (). ....	30
<b>Figura 18.</b> void loop (). ....	31
<b>Figura 19.</b> Biblioteca de Arduino. ....	32
<b>Figura 20.</b> Plano de la base y el cuerpo del robot. ....	34
<b>Figura 21.</b> Base y cuerpo del robot en SolidWorks. ....	35
<b>Figura 22.</b> Robot DYOR realizado en SolidWorks. ....	35
<b>Figura 23.</b> Frente del Robot DYOR. ....	36
<b>Figura 25.</b> Batería y placa Arduino. ....	37
<b>Figura 27.</b> Rueda Loca ensamblada en la base. ....	37

<b>Figura 29.</b> Cuerpo del robot con los elementos montados. ....	38
<b>Figura 30.</b> Sensor Siguelínea.....	38
<b>Figura 31.</b> Piñón del servo motor SG90. ....	39
<b>Figura 33.</b> Servo motor con sus respectivas resistencias soldadas. ....	40
<b>Figura 34.</b> Pulsos para darle el giro y paro al servo motor modificado.....	40
<b>Figura 35.</b> Ancho de pulso del sensor HC-SR04.....	43
<b>Figura 36.</b> Encender y apagar un Led en un periodo de 1 s. ....	51
<b>Figura 37.</b> Circuito de Led para el ejercicio en protoboard.....	52
<b>Figura 38.</b> Ciclo For de parpadeo de un Led (10 veces). ....	52
<b>Figura 39.</b> Ejercicio de encender 10 veces un Led en el void setup (). ....	53
<b>Figura 40.</b> Ciclo for en servo motor. ....	53
<b>Figura 41.</b> Robot DYOR – Siguelínea.....	71
<b>Figura 42.</b> Robot DYOR – Sensor Ultrasónico.....	78
<b>Figura 43.</b> Demostración del mando a distancia y los movimientos del Robot DYOR.	87
<b>Figura 44.</b> Porcentaje de hombres y mujeres.....	88
<b>Figura 45.</b> Porcentaje de edades. ....	88
<b>Figura 46.</b> Porcentaje de conocimiento sobre la Robótica Educativa. ....	89
<b>Figura 47.</b> Porcentaje del conocimiento del Robot DYOR. ....	89
<b>Figura 48.</b> Porcentaje de dificultad de aprender programación.....	90
<b>Figura 49.</b> Porcentaje de motivación al aprender programación.....	91
<b>Figura 50.</b> Porcentaje del conocimiento de la placa Arduino.....	92
<b>Figura 51.</b> Porcentaje sobre el uso del software Arduino.....	92
<b>Figura 52.</b> Porcentaje el conocimiento de bloques de programación. ....	93
<b>Figura 53.</b> Porcentaje en el interés por aprender a programar con esta metodología....	94
<b>Figura 54.</b> Porcentaje respecto a saber sobre la construcción de un robot. ....	95
<b>Figura 55.</b> Porcentaje sobre el conocimiento de crear app en Android.....	95

<b>Figura 56.</b> Porcentaje de la capacidad de crear un mando a distancia. ....	96
<b>Figura 57.</b> Estudiantes impartidos el curso.....	107
<b>Figura 58.</b> Explicando sobre el robot DYOR y el lenguaje de programación de Arduino. .....	107
<b>Figura 59.</b> Robot DYOR – Vista Lateral izquierda.....	108
<b>Figura 60.</b> Robot DYOR – Vista Lateral Derecha. ....	108
<b>Figura 61.</b> Robot DYOR – Vista Frontal.....	109
<b>Figura 62.</b> Robot DYOR – Sensor Siguelínea.....	109

## **Índice de anexos:**

<b>Anexo 1.</b> Salón de clases al cual se dio el curso de Programación. ....	107
<b>Anexo 2.</b> Exponiendo sobre el lenguaje de Arduino.....	107
<b>Anexo 3.</b> Vistas del Robot DYOR. ....	108
<b>Anexo 4.</b> DYOR – Detección de Obstáculos. ....	110
<b>Anexo 5.</b> DYOR – Detección de Obstáculos con Alarma. ....	114
<b>Anexo 6.</b> DYOR – Controles básicos (derecha – izquierda – Avanzar – Retroceder) a robot a distancia.....	118
<b>Anexo 7.</b> Certificación de traducción del resumen.....	122

## **1. Título**

**Diseño, construcción y desarrollo de guías prácticas con DYOR**



## 2. Resumen

El presente proyecto se basa en aplicar la robótica educativa como un mecanismo de aprendizaje a través de un robot DYOR, que permita al estudiante motivarse para desarrollar de una manera más efectiva su capacidad intelectual.

Para definir el proceso de trabajo se realizó una constante búsqueda bibliográfica con el fin de comprender la importancia de la motivación en el proceso de aprendizaje. También se describe el lenguaje de programación en Arduino que se ejecutara en un robot llamado DYOR, que es un robot diseñado por uno mismo y conocer los componentes que formarán parte del robot y darán movimiento al mismo. Además de desarrollar guías prácticas tanto del diseño y construcción como algunas programaciones que tendrá dicho robot.

Establecidas las guías prácticas, construido el robot y teniendo las programaciones pertinentes, se desarrolló un pequeño curso a estudiantes de primer ciclo de la carrera de electricidad de la Universidad Nacional de Loja, donde se impartió un pequeño curso de programación en Arduino, donde se dio ejercicios desde aprender a encender un Led, hasta programar un robot DYOR.

La metodología que se aplicó para validar las guías prácticas se llevó a cabo de encuestar a los estudiantes contestando un cuestionario en donde se establecieron preguntas concretas sobre lenguaje de programación, robot DYOR y la robótica educativa, con el fin de conocer el pensamiento de los estudiantes al terminar el curso de programación y la capacidad de adquirir conocimiento de temas expuestos de forma interesante que les motiva conocer y aprender de manera innovadora.

**Palabras claves:** Guías prácticas con DYOR, lenguaje de programación, diseño de robots, Arduino, aprender a programar.

## 2.1 Abstract

The following project has been based on the application of educational robotics as a learning mechanism through a DYOR robot, which allows students to motivate themselves to develop their intellectual capacity efficiently.

To define the work process, a constant bibliographic search was carried out in order to understand the importance of motivation in the learning process. It also describes the programming language in Arduino to be executed in a robot called DYOR, which is a robot designed by myself, and also to know the components that will be part of the robot and give movement to it. Furthermore, to develop practical guidelines for the design and construction as well as some programming that the robot will have.

Once the practical guidelines were established, as well as the relevant programming, a small course was developed and addressed to the first cycle students of the electrical engineering college of the National University of Loja, where a small programming course was taught about Arduino language. Additionally, exercises were developed by the students, who first learned to turn on a Led light, and eventually to program a DYOR robot.

The methodology applied to validate the practical guides was carried out by surveying students, who answered a questionnaire, where specific questions about programming language, DYOR robot and educational robotics were established, the survey helped collect information about the thoughts of the students regarding the programming course and the ability to acquire knowledge of topics exposed in an interesting way, which motivates them to learn and learn in an innovative way.

**Keywords:** practical guides whit DYOR, programming language, robots design, Arduino, learning to program.

### 3. Introducción

A pesar de que en los últimos años se empezó a encontrar más información e investigación sobre la robótica educativa, aún queda visible la falta de investigación y aportaciones científicas a este ámbito desde su justificación pedagógica hasta su evaluación y fundamentaciones metodológicas. Es por ello que este trabajo busca aportar en encontrar la relación que hay entre la robótica educativa y los factores que influye en la motivación del estudiante por aprender un lenguaje de programación.

El presente trabajo tiene como objetivo principal “Desarrollar guías prácticas para el diseño, construcción y ejecución de movimientos, de un robot DYOR”, donde se define un mecanismo de aprendizaje para motivar al estudiante a que aprenda y entienda un lenguaje de programación, incentivándolo a programar un robot que puede ser construido por uno mismo.

Como planteamiento del primer capítulo del presenta trabajo, se recolectará diferentes fuentes bibliográficas como artículos científicos, tesis y sitios webs relacionados con el tema, la información necesaria y confiable para definir la motivación y el aprendizaje que permitirá conocer qué ventajas tiene una buena motivación para tener la capacidad de adquirir un aprendizaje con respeto al estudiante.

El segundo capítulo se expone los conceptos de la robótica educativa y como se ha venido tomando en cuenta como un método de enseñanza para hacer que algunas materias que son complicadas para los estudiantes, se presenten como una manera más interesante y se facilite su comprensión. Además, de conocer sobre el robot DYOR, que es un robot que se tomó como guía del artículo “Diseña, Fabrica y Programa Tu Propio Robot” de Leopoldo Ernesto, donde nos da la iniciativa de cómo construir y programar tu propio robot.

El tercer capítulo del trabajo hace referencia a la construcción del robot donde se da la iniciativa de los componentes que formarán parte y darán los movimientos del robot y la placa principal que se instalará que es Arduino. Se da a conocer lo que es la placa programable y el lenguaje de programación que utiliza en el software Arduino. Además de conocer sobre el desarrollo de guías practicas; las partes de la conforman y la determinación de una metodología que permitan validar dichas guías prácticas.

Para desarrollar el siguiente proyecto de grado se plantearon los siguientes objetivos:

- Objetivo general:

Desarrollar guías prácticas para el diseño, construcción y ejecución de movimientos, de un robot DYOR.

- Objetivos específicos:
  - Desarrollar guías prácticas para el diseño y construcción de un robot DYOR.
  - Desarrollar guías prácticas para aplicar estructuras de programación aplicando DYOR.
  - Validar las guías prácticas desarrolladas.

## **4. Marco Teórico**

### **4.1 Capítulo I**

#### **4.1.1 *Motivación y Aprendizaje***

##### **4.1.1.1 Motivación.**

La motivación es el motor de la conducta humana. El interés por una actividad es “despertado” por una necesidad, la misma que es un mecanismo que incita a la persona a la acción, y que puede ser de origen fisiológico o psicológico. La motivación determina el nivel con que energía y en qué dirección actuamos (Carrillo & Padilla, 2009).

La motivación se presenta como una actitud interna y positiva frente al nuevo aprendizaje, es lo que mueve al sujeto a aprender, es por tanto un proceso endógeno, donde la motivación depende de cada persona y que es lo que lo incentiva a tener esa actitud indudable (Carrillo, 2009).

##### **4.1.1.2 Aprendizaje.**

El objeto de aprendizaje debe ser funcional, integrable, potencialmente significativo e internamente coherente, mientras que el sujeto que aprende debe disponer de las estructuras cognitivas necesarias que le permitan establecer relaciones con el nuevo conocimiento (Carrillo, 2009).

El cerebro humano se encuentra programado para la supervivencia, por lo que está potencialmente preparado para aprender, este aprendizaje implica un proceso dual, por un lado, necesita y registra lo conocido y, por el otro busca lo novedoso para aprender (Carrillo, 2009).

#### **4.1.2 *Motivación***

##### **4.1.2.1 Importancia de la motivación en el proceso de aprendizaje.**

La motivación es un factor notable en el proceso de aprendizaje para el estudiante, un punto fundamental es:

Interés por el tema de trabajo: Este aspecto es básico y obvio, el interés que tenga el estudiante por el tema concreto de estudio interviene en su motivación para el aprendizaje, un tema interesante desencadena con facilidad el esfuerzo necesario para aprenderlo (Carrillo, 2009).

Se hace evidente que la motivación influye sobre el pensamiento del estudiante y, por consecuencia, en el resultado del aprendizaje. En este sentido, se relaciona con las posibilidades reales de que el alumno consiga las metas, conozca cómo actuar para afrontar con éxito las tareas y problemas y maneje los conocimientos e ideas previas sobre los contenidos por aprender, su significado y utilidad (Ospina Rodríguez, 2006).

### **4.1.3 Tipos**

#### **4.1.3.1 Aprendizaje.**

En la actualidad, los procesos de enseñanza y aprendizaje nacen a partir de nuevas tendencias apoyadas en la tecnología y medios informáticos, herramientas de gran utilidad en los procesos de información. Estas presentan novedosas formas de aprender y ofrecen una educación más personalizada y de calidad (García J. , 2018). Entre los tipos de aprendizaje se tiene:

- A. Implícito
- A. Explícito
- A. Asociativo
- A. No asociativo (Sensibilización)
- A. Significativo
- A. Cooperativo
- A. Emocional
- A. Observacional
- A. Experimental
- A. Por descubrimiento
- A. Memorístico
- A. Receptivo

En el libro “Psicología y Mente” en la sección Psicología educativa y del desarrollo de Jonathan García se encuentra detallado cada aprendizaje descrito.

#### **4.1.3.2 Motivación para el Aprendizaje.**

Una primera distinción, es reconocer el lugar o sitio de origen de esa fuerza que impulsa al estudiante. De esta manera puede ser interna (intrínseca) o externa (extrínseca) a la persona:

- La motivación intrínseca es todo que trae, pone, ejecuta, activa el individuo por sí mismo cuando lo desea, para aquello que le apetece. Es, por tanto, que lleva consigo y la pone en ejecución cuando lo considera oportuno.
- La motivación extrínseca, es provocada desde fuera del individuo, por otras personas o el ambiente, es decir, depende absolutamente del exterior, de que se cumplan una serie de condiciones ambientales o haya alguien dispuesto y capacitado para generar esa motivación (Soriano, 2001).

#### ***4.1.4 Uso de tecnologías para la motivación del aprendizaje***

Las tecnologías forman parte de la vida cotidiana de las comunidades educativas y su introducción como canales de conocimiento compartido está teniendo una incidencia cada vez mayor en los estudios universitarios.

La tecnología involucra un concepto amplio que refiere a una representación del mundo que tiene a la base el diseño de relaciones sociales, económicas, culturales, educativas junto con condiciones generadores de conocimiento y productos a partir de la resolución de problemas en diferentes ámbitos de la vida cotidiana, puesto que son múltiples sus campos de desarrollo, en donde surge el interés por parte del estudiante por aprender, campos como: la industria, la inteligencia artificial, la robótica, la ingeniería genética, la biotecnología, entre otros campos (Córdova, 2017).

##### **4.1.4.1 La robótica como motivación en el proceso del aprendizaje.**

Desde hace unos años, la robótica ha formado parte como una herramienta que incentiva a los estudiantes en el ámbito del aprendizaje. En muchos casos, los robots están presentes en el aula no con el fin de enseñar la disciplina de la robótica propiamente dicha, sino aprovechar su carácter multidisciplinar para activar métodos cognitivos que propicien un aprendizaje significativo y un acercamiento al mundo de la ciencia y la tecnología (Antolí & Pellín, 2019).

En general, un trabajo de robótica integra varias tecnologías y, por ende, propicia un aprendizaje multidisciplinar. Los trabajos de robótica permiten adquirir conceptos con cierta profundidad y permite entrar en una dinámica de aprendizaje autónomo y desarrollo de varias competencias como la iniciativa y la innovación (Pellín, 2019).

## **4.2 Capítulo II**

### **4.2.1 Robótica educativa**

#### **4.2.1.1 Definición.**

El término “robótica educativa” se entiende como una herramienta que apoya los procesos de aprendizaje al poner en acción toda la capacidad de manipulación y exploración del alumno para generar conocimientos a través de su experiencia. Este es un proceso que influye en el alumno desde dos ámbitos, el primero es potenciar las características sensoriales del alumno y el segundo es fomentar su capacidad intelectual (Ocampos, 2017).

#### **4.2.1.2 Importancia.**

La importancia de la aplicar la robótica educativa es despertar el interés de los estudiantes transformando las materias: física, matemática, diseño, lenguaje de programación; en asignaturas más atractivas e integradoras, al crear entornos de aprendizaje propicios que recreen los problemas del ambiente que nos rodea (Quiroga, 2017).

De esta manera se contrarresta la crisis actual en la educación científica, que se debe principalmente a los métodos de enseñanza que hacen a estas asignaturas difíciles y poco interesantes; sembrando en el estudiante una actitud negativa hacia la tecnología, alejándolo del ámbito de la ciencia y la investigación (Quiroga, 2017).

### **4.2.2 Tipos**

Al hablar sobre la robótica educativa según Jacek Malec, 2001 la clasifica en dos tipos:

- Robótica en educación: además de ingeniería y ciencia, es divertida. Convince a los estudiantes de que es algo útil.
- Robótica para la educación: es útil en el proceso educativo y es tomada como una herramienta de aprendizaje en la educación.

#### **4.2.2.1 DYOR.**

El robot DYOR que sus siglas significan “Do Your Own Robot” (en español “Diseña tu propio robot”) es un paquete educativo de robótica, en donde da el inicio al



aprendizaje de la programación y diseño de los robots que son controlados a distancia. Es tipo de robótica educativa consiste en incentivar a los alumnos en aspectos básicos de la ingeniería (Leopoldo, 2016).

- **Elementos**

- Circuitos Electrónicos
- Sensores
- Software de Programación
- Software de Diseño
- Arduino
- Guía instructiva

- **Características**

El robot DYOR hace posible entender de una manera fácil lo que conlleva la robótica, cabe recalcar con un poco de aprendizaje al inicio, conforme se va avanzando se va descubriendo lo que somos capaces de crear (Leopoldo, 2016).

El desarrollo de los robots se hace cada vez más asequible por la reducción de costes en electrónica y mecánica involucradas y existen miles de formas de iniciarse y un montón de información en línea para aquellos que desean empezar (Leopoldo, 2016).

- **Estructura**

Tomando la experiencia de aprendizaje orientado a un robot “cuadrúpedo” también conocido como Oruga, se define:

- a. **Conceptualización:** incluye el definir el marco teórico asociado al cuerpo de conocimiento apropiado, en este caso electricidad y electrónica, física y biología, así como el conocimiento mínimo necesario para el manejo de herramientas tecnológicas, tal como se muestra en la Tabla 1.

**Tabla 1.** Conceptualización.

FICHA DE LA EXPERIENCIA	
Robot: ORUGA	
Electricidad y Electrónica	Variables eléctricas, lo digital y lo analógico
	Sensores y Actuadores
	Motores
	Controladore
Física	Cinemática y Dinámica
Programación	Valores binarios, los bits
	Algoritmos, ejemplos cotidianos
Biología	Qué es una oruga
	Características Biométricas
	Para qué está hecha

Fuente: Bruno Valencia – Artículo de Robótica.

Cabe aclarar que la parte bilógica dependerá mucho del tipo de robot a diseñar.

- b. Diseño y construcción: en cada una de estas etapas se debe:
- Revisar como una herramienta CAD es capaz de generar archivos para la fabricación de piezas a partir de sus dimensiones geométricas, en donde, se podrá observar las piezas previamente fabricadas.

- En la etapa de construcción, se unen las partes diseñadas en la herramienta CAD con las partes electrónicas y se procede el ensamblaje del robot, en donde el estudiante entenderá el concepto de articulación, asociando los tipos de movimiento con la definición de grado de libertad.
- c. Construcción del conocimiento: basado fundamentalmente en la programación del robot, el alumno debe recrear lo que observa al ver funcionar robots y partes semejantes, entendiendo el funcionamiento del robot.
- d. Evaluación de lo aprendido: es necesario evaluar el trabajo realizado por los estudiantes, para aquello, influye dos aspectos:
  - Tarea asignada: buscando el robot con más eficiencia (velocidad, movimientos, menos errores).
  - Creatividad: crear funciones personalizadas con la experiencia adquirida durante el transcurso del trabajo (Valencia, Rivera, & Zegarra, 2020).

- **Estructura del Robot DYOR**

La estructura de los robots DYOR habitualmente son de una complejidad simple, por la cual el ensamblaje y la unión de todas las piezas electrónicas se facilita, la estructura con la cual se construirá el robot dependerá del diseñador, el cual puede usar:

- Madera
- Playbond
- Cartón prensado
- Resina
- Acrílico
- Fibra de vidrio
- Entre otros materiales similares.

Lo cual será la base y forma de cada robot DYOR donde se encontrarán los elementos electrónicos para su funcionamiento (Leopoldo, 2016).

#### **4.2.2.2 Robótica como potencia de aprendizaje.**

Es importante propiciar el trabajo de la robótica educativa teniendo en cuenta cuatro conceptos que deben estar presentes en el desarrollo de cada trabajo: “Imaginar,

Diseñar, Construir, Programar, palabras, frases o etapas, pero que no tienen una organización lineal, sino que pueden superponerse una con otra durante todo el proceso y desarrollo del proyecto. “Robótica Educativa no es construir o programar, es un proceso de aprendizaje en el que, según como se mire, los robots son casi una excusa” (García & Castrillejo, 2011).

#### **4.2.2.3 Robótica diseño de recursos educativos.**

Las actividades experimentales como la construcción y programación de un robot de bajo coste garantizan un aprendizaje interdisciplinar de la robótica. De este modo, el estudiante aprende robótica como una combinación de conocimiento de diferentes campos como la electrónica, la ingeniería de software, que le van a permitir a solucionar problemas. Además, es constructivista porque el estudiante desarrolla sus propios dispositivos y estos interactúan con el mundo para llevar a cabo una tarea definida (Plaza, 2019).

La robótica cada vez está tomando más importancia en el mundo, en proyecto de investigación de Pedro Plaza (2019), redacta algunos recursos educativos tales como:

- Talleres de STREAM (Ciencia, Tecnología, Robótica, Ingeniería, Arte y Matemáticas) donde se crea una serie de módulos de robótica diseñados para capturar la imaginación de los estudiantes escolares, grandes y pequeños.
- Utilización de programas y laboratorios virtuales de participación para alentar y retener a los estudiantes en las disciplinas de ciencias, tecnología, ingeniería y matemáticas (Plaza, 2019).

El laboratorio virtual de la Universidad de Valencia, es un sitio web creado por esta institución que motiva al estudiante a inclinarse por las ciencias de la programación y diseño, dando a conocer que la robótica es simple y didáctica para el aprendizaje aplicando el robot DYOR.

#### **4.2.2.4 Guías Prácticas (Aprendizaje).**

Las guías prácticas son un conjunto de actividades en el cual su objetivo es guiar a los estudiantes a realizar un trabajo experimental y práctico en el cual obtendrán, en base de sus conocimientos adquiridos, un aprendizaje significativo.

- **Estructura**

Es un formato general en el cual se van a documentar las guías de prácticas, este formato cuenta con una estructura ordenada para la ejecución de un guía, en la cual destacan en el orden respectivo los siguientes puntos, (Aguilar & Loaiza, 2020):

- **Presentación:** es donde se indica el inicio de la guía práctica y se presenta el título en este apartado.
- **Objetivos:** se presentan los objetivos a cumplir en todo el proceso de la práctica.
- **Requisitos:** se describen los requisitos mínimos para la ejecución de la práctica.
- **Conocimientos previos:** se encuentran descritos los conocimientos que ya deben tener adquiridos los estudiantes para desarrollar la práctica.
- **Equipos e instrumentos:** presentados en la guía práctica con el fin de tener presente los instrumentos y equipos que se usaran.
- **Exposición:** se da una pequeña introducción de lo que tratara la práctica.
- **Proceso:** Se describe los pasos y la metodología a aplicar de forma clara para que no exista problemas de entendimiento por parte de los estudiantes.
- **Actividades a desarrollar:** se presentan las actividades que tienen que desarrollar cada estudiante con el fin de cumplir con los objetivos.
- **Resultados:** esta etapa es desarrollada por el estudiante, describiendo de forma cualitativa y/o cuantitativa los resultados obtenidos en la práctica.
- **Conclusiones:** es desarrollado por el estudiante, demostrando el cumplimiento de cada objetivo propuesto en la guía práctica.
- **Recomendaciones:** el estudiante describe, en base de la experiencia obtenida en la ejecución de la práctica, recomendaciones al momento de realizar la práctica.
- **Referencias:** se presentan las citas bibliográficas que se investigaron en la guía práctica.

Estos puntos de la estructura están encaminados para el correcto entendimiento y cumplimiento de los objetivos de las guías prácticas.

- **Características**

Las guías prácticas permiten afianzar los conocimientos adquiridos en el transcurso de la materia por parte de cada estudiante, que, con la motivación de diseñar y construir su propio robot, obtendrán un aprendizaje significativo. Además, para realizar las prácticas se toma en cuenta el nivel académico del estudiante, debido a que, se requiere que el estudiante tenga conocimientos previos de programación y diseño (Lema, 2012).

#### **4.2.2.5 Instrumentos de Validación.**

En función de los resultados obtenidos se determina si el método aplicado es el adecuado para los estudiantes. Debido a que no siempre se encuentran disponibles datos comparativos se debe tener presente las actividades específicas de nuestro método y determinar:

- Cumplimiento de los objetivos.
- Tiempo de trabajo.
- Interés por parte del estudiante, entre otras.

Un instrumento de validación utilizado en el trabajo de investigación de un “Desarrollo de guías prácticas orientados al aprendizaje de la robótica industrial” por parte de Adrián Aguilar (2020) fue el test:

- Test: Realizar un test compuesto por enunciados dirigidos en tres diferentes puntos de vista: visual, procedimental y consecución de objetivos.

En la aprobación de cada test, se tabula los resultados y se realiza una evaluación porcentual. Las prácticas son válidas cuando encajen las siguientes restricciones:

- Las preguntas que tengan un porcentaje de respuesta favorable igual o mayor al 80%, se consideran respuestas favorables.
- Se hayan realizado acciones para edificar las guías de prácticas con las críticas y sugerencias manifestadas por los estudiantes (Aguilar & Loaiza, 2020).

### **4.3 Capítulo III**

#### **4.3.1 *Elaboración del robot y las guías prácticas***

##### **4.3.1.1 Diseño Mecánico.**

El diseño mecánico de la estructura del robot será una de las actividades primordiales para este trabajo. Para el diseño mecánico existen programas con los cuales se puede determinar la forma final que tendrá el robot DYOR y hasta se puede simular los movimientos.

##### **4.3.1.2 Software de Ingeniería.**

Existen una variedad de software de ingeniería que permiten desarrollar piezas y mecanismo para la construcción de máquinas y piezas, te permite tener una visualización y un análisis previo de cómo se presenta cada modelo diseñado, uno de los softwares de ingeniería de este campo conocido es:

- **SolidWorks**

Es una aplicación de automatización de diseño mecánico que les permite a los diseñadores croquizar ideas con rapidez, experimentar con operaciones y cotas, y producir modelos y dibujos detallados (Hirschtick, 2015).

#### **4.3.2 *Desarrollo de las guías prácticas***

Las guías prácticas deben cumplir las siguientes características generales (Carrasquilla, 2014):

- Ser Válidas y reproducibles.
- De aplicación fácil.
- Actividades claras y sin ambigüedad.
- Flexibles.
- Origen multidisciplinario.
- Documentación sólida.
- Proveer procesos de revisión.

Teniendo en cuenta estos aspectos, el proceso de desarrollo de guías prácticas es variable entre diferentes grupos, su desarrollo va desde lo sencillo a lo complejo, lo cual

alimentara la motivación del estudiante por ir adquiriendo conocimiento e ir resolviendo problemas ya con una base sólida y experimental, obteniendo así una destreza en el diseño y programación de la robótica educativa (Carrasquilla, 2014).

La elaboración de guías para prácticas, están orientadas para que el estudiante adquiera destrezas y se encuentre totalmente preparado para poder solventar inconvenientes o retos que encuentre en su vida profesional.

La metodología empleada para encontrar la solución de los distintos escenarios expuestos en cada una de las guías son piezas claves para que cada estudiante obtenga habilidades profesionales y ejercite la destreza de resolver problemas en el ámbito de la programación, la experiencia adquirida puede ser dirigida a situaciones específicas que se encuentran en procesos reales (Carrasquilla, 2014).

### 4.3.3 Componentes del robot DYOR

#### 4.3.3.1 Placas de programación.

En la actualidad existen una variedad de placas de programación que han ayudado en el aprendizaje al nivel de la educación como un apoyo metodológico de estudio. En el proceso de validar la placa Arduino, como la tarjeta de programación a utilizar en este trabajo, se realiza una comparación entre las placas más utilizadas y que son ampliamente comercializadas, entre ellas se encuentran descritas en la Tabla 2 con sus respectivas propiedades pertinentes.

**Tabla 2.** Comparación entre algunas placas controladoras.

PLACAS Propiedades	Arduino Uno	Raspberry Pi	BeagleBone	STM32 F4 Discovery
Procesador	ATMega 328	Arm11	AM335x	ARM-Córtex
Velocidad	16 MHz	700 MHz	1 GHz	40 KHz



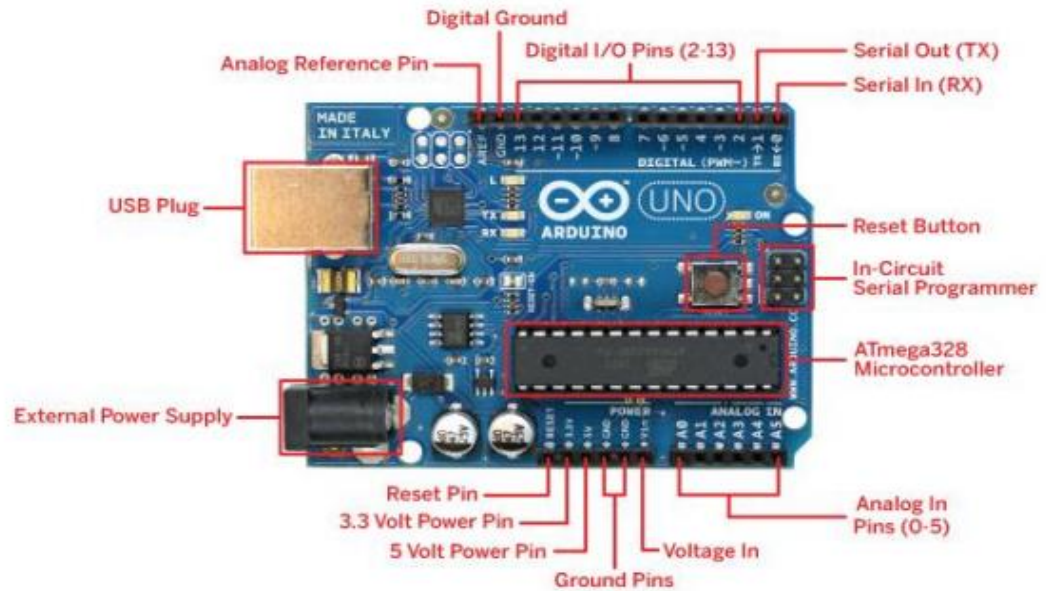
<b>RAM</b>	2 KB	512 MB	512 MB	192 KB
<b>USB</b>	n/a	2	1	2
<b>I/O</b>	14 GPIO, 6-10 bits analógicos	8 GPIO	69 GPIO, LCD, GPMC, MMCI, MMC2, 7AIN, 4 temporizadores, 4 puertos seriales, CAN0	24 GPIO, 5 bits analógicos, 16 DMA
<b>Tamaño</b>	2.95"x2.1"	3.37"x2.125"	3.4"x2.1"	7.87"x3.94"
<b>Entorno</b>	Arduino IDE, MATLAB®	Linux, IDLE, Open Embedded, QEMU, Scratchbox, Eclipse	Python, Linux, Scratch, Eclipse, Android ADK	STM32CubeMX, MATLAB®, Linux, OpenOCD
<b>Costo</b>	20 USD	45 USD	50 USD	50 USD

**Fuente:** El Autor.

#### 4.3.3.2 Placa Arduino UNO.

Es un modelo de la familia Arduino diseñado y distribuido por dicha comunidad. La placa tiene un tamaño de 68x53 mm. Su unidad de procesamiento consiste en un microcontrolador ATmega328. Puede ser alimentada mediante USB plug o alimentación externa y contiene pines tanto analógicos como digitales (Sánchez, 2012).

Los conectores y pines de la placa Arduino UNO se encuentran distribuidos de una forma en la cual facilita la conexión de los componentes electrónicos, así como se indica en la Figura 1.



**Figura 1.** Elementos de la placa Arduino Uno.

**Fuente:** De Emilio Sánchez, 2012.

Características Técnicas de la placa Arduino UNO (Torrente Artero, 2013):

- Cuenta con un microcontrolador Atmega328P.
- Voltaje de entrada de 7 a 12v (recomendado).
- Voltaje de entrada de 8 a 20v (limite).
- Terminales de E / S digitales 14 (de las cuales 6 proporcionan salida PWM) “Pulse Width Modulation” (Modulación de Ancho de Pulsos).
- Terminales de E / S digitales de PWM 6.
- Terminales de entrada analógica 6.
- Corriente DC por Terminales E / S 20 mA.
- Corriente DC para 3.3V Terminal 50 mA.
- Memoria flash 32 KB (ATmega328P) de los cuales 0,5 KB utilizados por el gestor de arranque.
- SRAM 2 KB (ATmega328P).
- EEPROM 1 KB (ATmega328P).
- Velocidad de reloj 16 MHz.
- Led en terminal 13.
- Longitud 68.6 mm.

- Anchura 53.4 mm.
- Peso 25 g.
- RX (Recepción de Datos) y TX (Trasmisión de Datos): Se usan para transmisiones serie de señales TTL (Time To Live).
- Interrupciones externas: Las Terminales 2 y 3 están configurados para generar una interrupción en el ATmega. Las interrupciones pueden dispararse cuando se encuentra un valor bajo en estas entradas y con flancos de subida o bajada de la entrada.

#### ➤ Alimentación del Arduino

Para la alimentación del Arduino lo más utilizado son el PowerBank o una batería ordinaria de 5v:

#### 4.3.3.3 Power Bank / Batería 5v.

En el caso del Power Bank según (Llamas, 2016): tienen como ventaja que proporcionan 5V regulados, por lo que se puede alimentar Arduino a través del USB, sin preocuparnos de la necesidad de regular el voltaje.



**Figura 2.** Power Bank.

**Fuente:** Luis Llamas – Tecnología, informática y diseño.

En cambio, con un batería dicho por (Llamas, 2016): Usar una pila de 9V es una de las opciones más extendidas, especialmente para usuarios que se están iniciando y en proyectos de pequeño tamaño. El voltaje de 9V es apropiado para alimentar Arduino. Tienen la ventaja de ser fáciles de encontrar y usar. Además, hay disponibles cables y porta pilas, que incluso incorporan un conector Jack tipo Arduino, lo que hace que sean fáciles de usar.



**Figura 3.** Batería de 9V.

**Fuente:** Luis Llamas – Tecnología, informática y diseño.

#### 4.3.3.4 Ultrasonido HC-SR04.

El sensor HC-SR04 es un sensor de distancia que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Presenta un pequeño tamaño, bajo consumo energético, buena precisión, ideal para usar con Arduino y excelente precio. Es muy utilizado dentro de los sensores tipo ultrasonido en proyectos de robótica como; robots exploradores, laberinto y en proyectos de automatización como sistemas de medición de nivel o distancia (Mechatronics, 2017).



**Figura 4.** Sensor Ultrasonido HC-SR04.

**Fuente:** Mechatronics, 2017.

#### 4.3.3.5 Zumbador de sonido.

Este módulo es un zumbador activo para Arduino y emite diferentes sonidos a varias frecuencias. No acepta fuente de oscilación, sólo recibe ondas cuadradas para conducir el sonido. Presenta las siguientes características (Solectro, 2021):

- Voltaje de funcionamiento: 3.3-5.5 VDC.
- Consumo de corriente: <25mA.
- Conexión de 3 pines:
  - VCC: Voltaje de alta potencia.
  - GND: Conexión a tierra.
  - S: Señal de control del zumbador.
- Frecuencias de funcionamiento: 2kHz-5kHz.
- Tamaño de PCB: 18.5x15 mm.



**Figura 5.** Zumbador de sonido.

**Fuente:** Solectro, 2021.

#### 4.3.3.6 Servo SG90.

Es un dispositivo pequeño que tiene un eje de giro controlado. Este puede ser girado a posiciones angulares específicas, al enviar una señal codificada. Al no existir una señal codificada de entrada el servo mantendrá su posición angular sin tener una reacción en el engranaje. Cuando hay una señal codificada de entrada, la posición angular de los engranes cambia y el eje gira a la posición asignada, su rango de rotación es de 0° a 180° (90° en cada dirección) (Carrod, 2014).



**Figura 6.** Servo Motor G90 (de posición).

**Fuente:** Carrod Electrónica, 2014.

#### **4.3.3.7 Servo Motores de rotación continua.**

Micro Servo Motor de Rotación Continua – FS90R la empresa (Geek, 2021) describe que: gira 360 grados completamente hacia adelante o hacia atrás, en lugar de moverse a una sola posición, se conectado a Arduino y se logra controlar su movimiento.



**Figura 7.** Servo Motor FS90R (rotación continua).

**Fuente:** Geek Factory, 2021.

#### **4.3.3.8 Shields de Arduino.**

Una shield (“Escudo” en español) es un circuito modular impreso en una placa para ser apilada sobre la parte superior de Arduino o de otra shield, lo cual ayuda a expandir la funcionalidad de la unidad de control. Logrando la comunicación entre ellas mediante conexión USB o simplemente interactuando con sus terminales de entrada y salida (Torrente Artero, 2013). Se debe tomar en cuenta que al montar una shield sobre una placa Arduino debe de tener la misma forma para poder integrarse a ella.

Existen dos tipos de Shield, los oficiales y los no oficiales, alguno de ellos se los describe a continuación (Torrente Artero, 2013):

- **Shields Oficiales:** son placas armadas y distribuidas especialmente por Arduino, las más conocidas son:
  - **Arduino Ethernet Shield:** es una tarjeta que se apila a la placa Arduino UNO con la característica de ser conectado a una red cableada TCP/IP.
  - **Arduino Wireless SD Shield:** es compatible con Arduino UNO permitiendo la comunicación inalámbrica mediante dispositivos XBee acoplándose con otros mecanismos XBee en interiores a 100m y en forma exterior hasta 300 m, además contiene una ranura para colocar y usar una Micro SD.
  - **Arduino Wi-Fi Shield:** este escudo aumenta la capacidad de la placa Arduino UNO si se desea conectar a redes TCP/IP, puesto que incorpora un microcontrolador HDG104 y contiene una antena que concede conectar a redes Wi-Fi versión 802.11b y 802.11g.
  - **Arduino Motor Shield:** es una shield que permite el control de servomotore, motores a pasos, relevadores entre otros dispositivos, en Arduino UNO. Su desempeño de este escudo es gracias a que integra un microcontrolador L298P ayudando a controlar la velocidad y dirección de mecanismo.
- **Shield no Oficiales (prototipos):** son placas preensambladas y construidas por gran parte de individuos aficionados a Arduino, de otro modo estos escudos pueden ser llamados prototipos, un ejemplo es el shield para servomotores de la Figura 8.

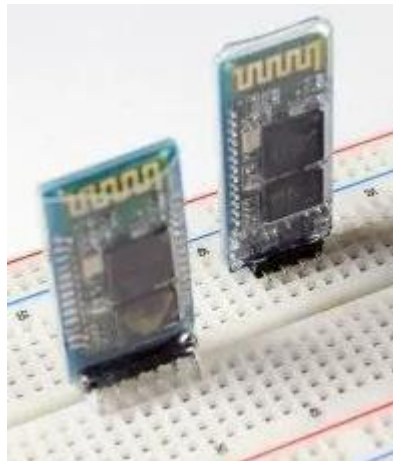


**Figura 8.** Shield para servomotores.

**Fuente:** El Autor.

#### 4.3.3.9 Bluetooth HC-06 y HC-05 Android Arduino.

Se usa en una comunicación con una App de Android a Arduino. En primer lugar, el dispositivo HC-06 y el HC-05, gracias a sus puertos TxD y RxD nos permite realizar comunicaciones inalámbricas a una distancia de 10 metros máximo. Teniendo en cuenta que estos dispositivos son fáciles de manejar, por sus comandos AT. Dichos comandos se envían por un puerto serial. También es de bajo consumo ya que trabajan a 3.3V. Los módulos son para aplicaciones sobre todo con microcontroladores, PIC o tarjetas Arduino. Sacando ventaja de su tamaño 12.7mmx27mm (puede variar dependiendo el tamaño) (HETPRO, 2015).



**Figura 9.** Bluetooth HC-05 y el HC-06 (6 pines) y (4 pines).

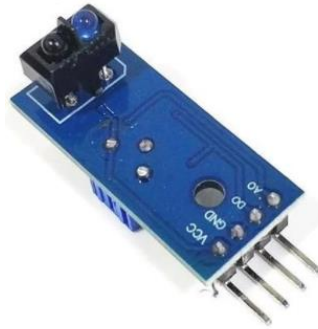
**Fuente:** HETPRO, 2015.

#### 4.3.3.10 Siguelíneas TCRT5000.

El sensor de infrarrojos TCRT5000 permite detectar luz reflejada de forma sencilla y es ideal para proyectos en los que se tiene que detectar objetos o seguir líneas. Además, cuenta con (Robótica Fácil, 2020):

- TCRT5000 IR reflex sensors.
- Comparador analógico LM393.
- Potenciómetro digital para ajuste de sensibilidad.
- Agujero para fijación.





**Figura 10.** Siguelínea TCRT5000.

**Fuente:** Tienda de Robótica Fácil, 2020.

#### 4.3.3.11 Rueda loca.

Según (SanDoRobotics, 2019) la Rueda loca de nylon cuenta con base de metal color blanco para robot o coche, soporta hasta 12 Kg de carga, presenta las siguientes dimensiones:

- Diámetro: 25mm x 13mm de ancho.
- Altura: 34mm.
- Base: 38 x 34 mm.



**Figura 11.** Rueda Loca.

**Fuente:** SanDoRobotics, 2019.

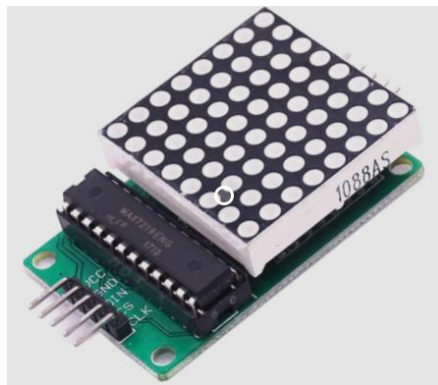
#### 4.3.3.12 Matriz de Leds (8x8).

Crea increíbles animaciones en tu proyecto con la matriz LED de 8×8, puedes proyectar caracteres, números o figuras.

Módulo de Matriz de Led 8×8 es operado con un MAX7219 que permite controlar el módulo usando solamente 3 pines del microcontrolador. El módulo dispone de una interfaz serial similar a SPI, por lo que es compatible con una gran cantidad de microcontroladores (TecnoTienda, 2021).

Características generales:

- Voltaje de operación: 4.0V-5V
- Corriente: 330mA
- Corriente por segmento: 40mA
- CLK: max.10Mhz
- Scan rate: 500-1300Hz (800Hz típico).



**Figura 12.** Matriz de Leds de 8x8.

**Fuente:** Tecno-Tienda, 2021.

#### ***4.3.4 Entorno de desarrollo para Arduino***

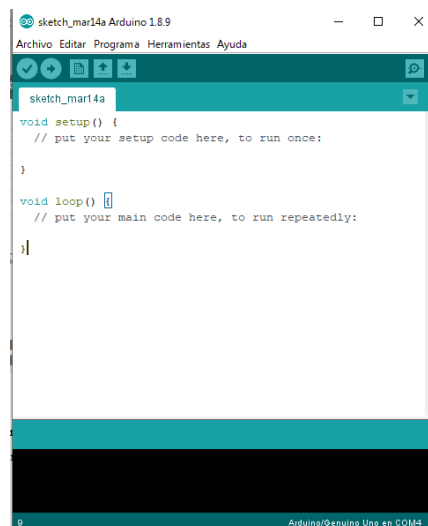
Entorno de Desarrollo Integrado (IDE por sus siglas en inglés Integrated Development Environment), es un software que agrupa ciertas herramientas que ayudan y permiten a desarrolladores a escribir, editar, simular y detectar errores en un programa como se observa en la Figura 13 (Torrente Artero, 2013).



**Figura 13.** Plataforma de Arduino.

**Fuente:** El Autor.

Al describir un programa (Sketch) como muestra la Figura 14 se dice que está integrado por una serie de instrucciones ordenadas en grupos adecuadamente sin tener algún tipo de confusión. Un Sketch es usado para ser grabado en la memoria flash del microcontrolador de Arduino gracias al uso de un IDE.



**Figura 14.** Nuevo Sketch en Arduino.

**Fuente:** El Autor.

Si se desea empezar a trabajar con Arduino, crear o probar un Sketch se necesitará del IDE de desarrollo oficial, que puede ser descargado gratis en la página <https://www.arduino.cc/>, pero no es la única opción a utilizar, existen otras plataformas. Por diversas razones, muchos usuarios no quedan satisfechos con la versión oficial, a continuación, se enlistan algunas opciones (Torrente Artero, 2013):

- CodeBlocs

- Codebender
- Visualmicro
- Atom
- Platomformio
- SublimeText

#### 4.3.4.1 Estructura general de un programa Arduino.

En el lenguaje de programación de Arduino, la estructura básica de un “sketch” es muy fácil (como se indica en la Figura 15), se conforma de tres partes principales: el primer plano lo conforman las variables globales, y las otras dos son funciones esenciales que están declaradas por bloques que agrupan las instrucciones de la programación (Torrente Artero, 2013).

```

buletooh_y_servo Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
buletooh_y_servo
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial mybluetooth(2,4); //RX y TX
Servo myservo2; // crea el objeto servo
Servo myservol;

int PINSERVO1 = 5;
int PULSOMIN1 = 520;
int PULSOMAX1 = 1500;
int PINSERVO2 = 9;
int PULSOMIN2 = 520;
int PULSOMAX2 = 1500;

void setup(){
  mybluetooth.begin(9600);
  myservol.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);
  myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);
  myservol.write(90);
  myservo2.write(90);
}

void loop(){

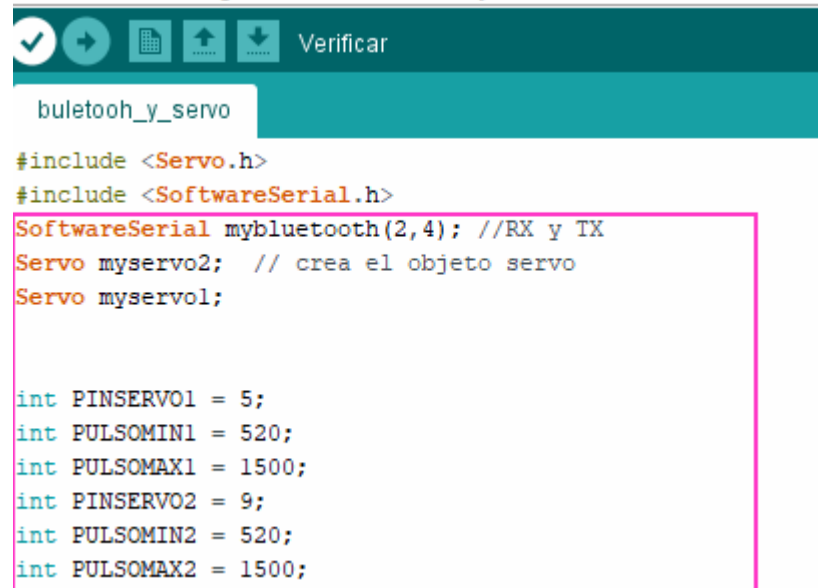
  if (mybluetooth.available()>0){
    char letra=mybluetooth.read();
    if (letra=='W'){ //hacia delante
      myservol.write(60);
      myservo2.write(180);
      delay(4000);
      myservol.write(90);
      myservo2.write(90);
    }
    if (letra=='F'){ // parada
      myservol.write(90);
      myservo2.write(90);
    }
    if (letra=='S'){ // hacia atras
      myservol.write(120);
      myservo2.write(0);
      delay(4000);
      myservol.write(90);
      myservo2.write(90);
    }
  }
}

```

Figura 15. Partes de una programación en Arduino.

Fuente: El Autor.

**Declaración de variables globales:** se declara al inicio de la programación como la configuración de terminales de entrada y salida, constante con un valor predeterminado o lo que se desee declarar ya que no tienen una delimitación, a diferencia de las funciones setup y loop. La Figura 16 muestra un ejemplo.



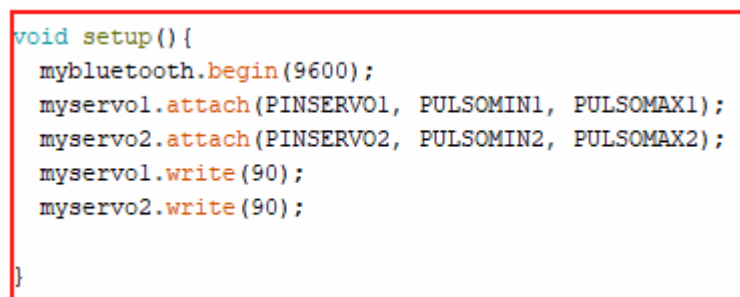
```
buletooh_y_servo
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial mybluetooth(2,4); //RX y TX
Servo myservo2; // crea el objeto servo
Servo myservol;

int PINSERVO1 = 5;
int PULSOMIN1 = 520;
int PULSOMAX1 = 1500;
int PINSERVO2 = 9;
int PULSOMIN2 = 520;
int PULSOMAX2 = 1500;
```

**Figura 16.** Declaración de Variables.

**Fuente:** El Autor.

**void setup ():** es la parte encargada de recoger la configuración, se delimita por inicio y cierre de llaves. En este apartado las instrucciones solo son ejecutadas una vez al iniciar la placa Arduino como se muestra en la Figura 17.



```
void setup() {
  mybluetooth.begin(9600);
  myservol.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);
  myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);
  myservol.write(90);
  myservo2.write(90);
}
```

**Figura 17.** Void setup ().

**Fuente:** El Autor.

**void loop ():** es la función del programa que se ejecutará repetidamente (en un loop – bucle), al igual que el setup es limitado por llaves de inicio y cierre. En esta parte del Sketch, las instrucciones del bloque son repetidas una y otra vez hasta que se ponga un punto de paro o una instrucción que detenga el ciclo en un lapso de tiempo. Un ejemplo se indica en la Figura 18.

```
void loop() {  
  
  if (mybluetooth.available()>0) {  
    char letra=mybluetooth.read();  
    if(letra=='W'){ //hacia delante  
      myservol.write(60);  
      myservo2.write(180);  
      delay(4000);  
      myservol.write(90);  
      myservo2.write(90);  
    }  
    if(letra=='F'){ // parada  
      myservol.write(90);  
      myservo2.write(90);  
    }  
    if(letra=='S'){ // hacia atras  
      myservol.write(120);  
      myservo2.write(0);  
      delay(4000);  
      myservol.write(90);  
      myservo2.write(90);  
    }  
  }  
}
```

**Figura 18.** void loop ().

**Fuente:** El Autor.

#### 4.3.4.2 Bibliotecas para Arduino

El uso de bibliotecas en el entorno de programación Arduino es normal al igual que en otras plataformas de desarrollo (Arduino, 2018). Su objetivo es proporcionar ayuda extra en los sketches, manejo de hardware y datos, si se requiere integrar una nueva biblioteca a un programa se realiza tal como se expone en la Figura 19.

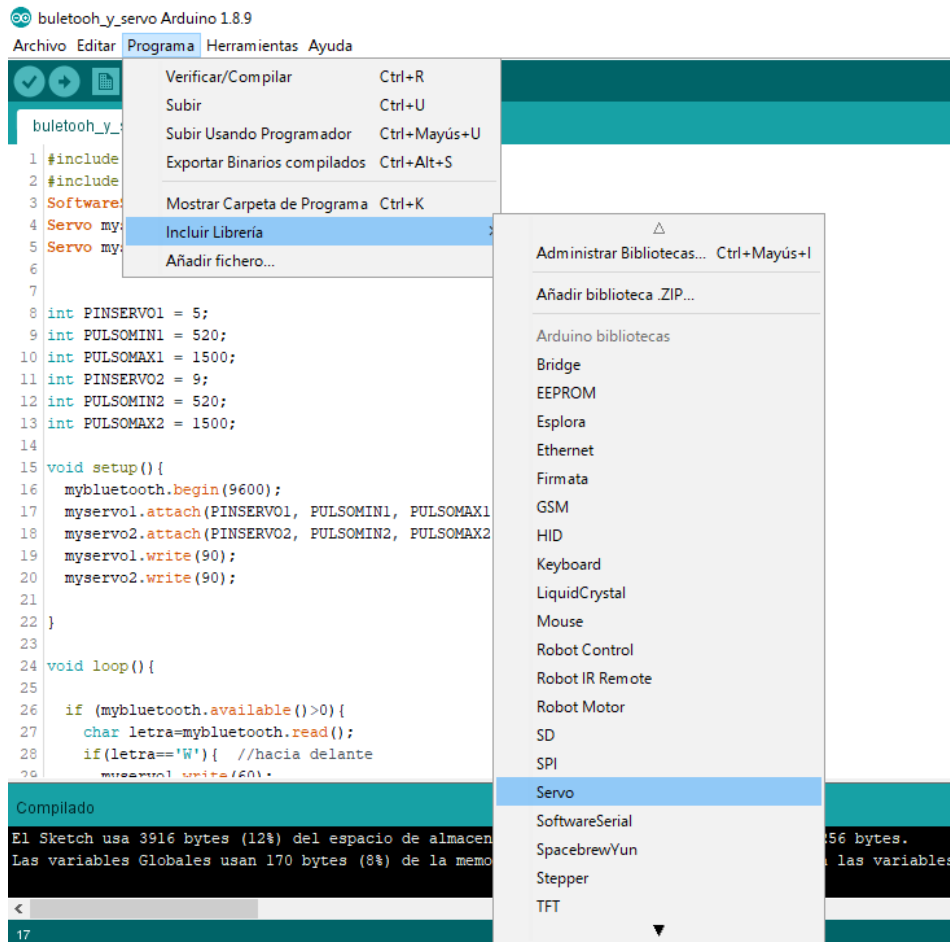


Figura 19. Biblioteca de Arduino.

Fuente: El Autor.

#### 4.3.4.3 Bibliotecas estándar para Arduino.

Las bibliotecas más comunes para Arduino se observan en la Tabla 3.

Tabla 3. Bibliotecas comunes para Arduino.

NOMBRE	FUNCIÓN
<b>EEPROM</b>	Lectura y escritura
<b>Ethernet/ethernet2</b>	Conexión a internet

<b>Firmata</b>	Comunicación de aplicaciones con protocolo serie estándar
<b>GSM</b>	Conexión a las redes GSM/GRPS
<b>LiquidCrystal</b>	Control de pantallas LCD
<b>SD</b>	Lectura y escritura de tarjetas SD.
<b>Servo</b>	Control de servo motores.
<b>SPI</b>	Comunicación de dispositivos que usan el bus Interfaz Periférica Serial (SPI).
<b>SoftwareSerial</b>	Comunicación en cualquier terminal digital.
<b>Stepper</b>	Control de motores a pasos.
<b>TFT</b>	Para dibujar texto, imágenes entre otros.
<b>WiFe</b>	Uso del escudo Arduino para conectarse a Internet.
<b>Wire</b>	Enviar y recibir datos a través de una red de dispositivos y sensores.

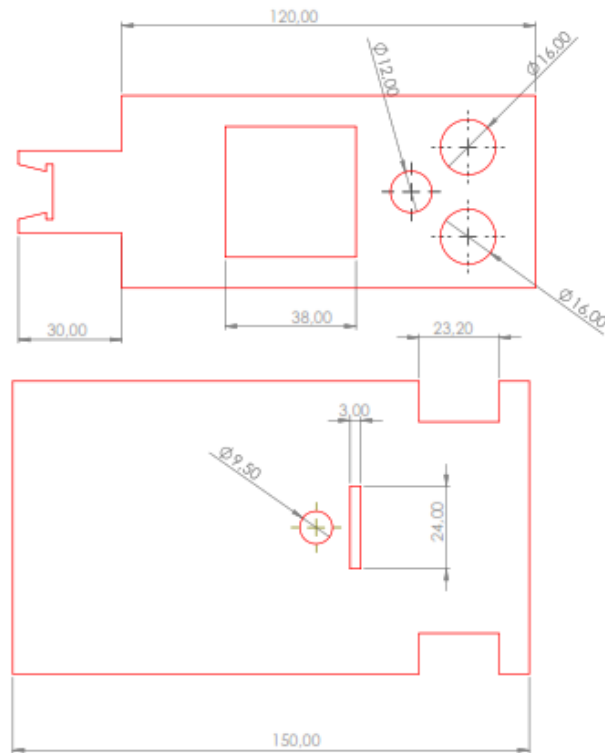
**Fuente:** Fredi Cedeño Enríquez, 2018.



## 5. Metodología

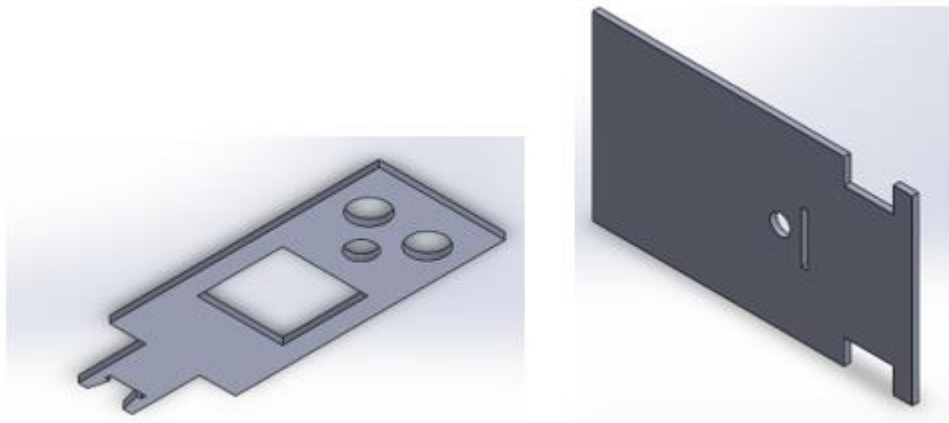
### 5.1 Diseño del Robot DYOR

En el proceso del diseño del robot DYOR se realizó en primera instancia en el programa de SolidWorks, se tomó un ejemplo de (Leopoldo, 2016) “Diseña, Fabrica y Programa tu propio Robot” para el diseño. Se diseña la base y el tronco como se muestra en la Figura 20 y Figura 21, respectivamente, las cuales posteriormente serian impresas en acrílico por una impresora láser.



**Figura 20.** Plano de la base y el cuerpo del robot.

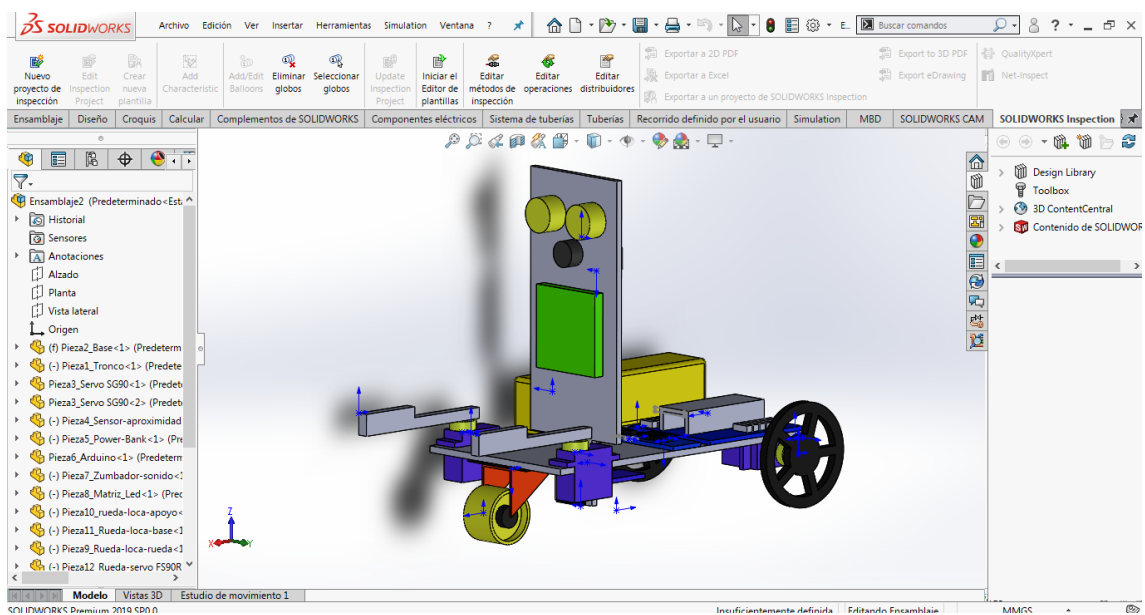
**Fuente:** El Autor.



**Figura 21.** Base y cuerpo del robot en SolidWorks.

**Fuente:** El Autor.

Se diseña cada componente en el programa para lograr hacer un ensamblaje tal como se indica en la Figura 22, aquí se encuentra como quedara el robot con los componentes electrónicos como lo son: la placa Arduino UNO, la fuente de alimentación, sensor Ultrasónico, zumbador, matriz de Led 8X8, los servomotores, seguidor de líneas y el módulo bluetooth.

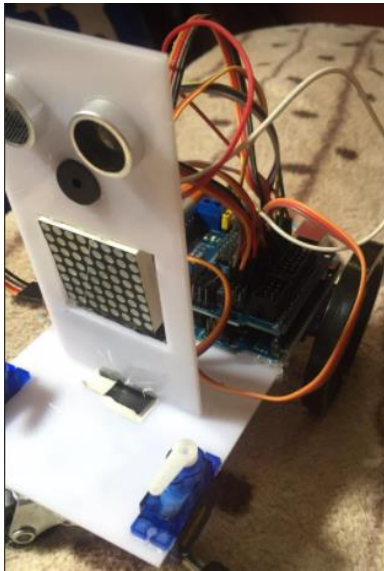


**Figura 22.** Robot DYOR realizado en SolidWorks.

**Fuente:** El Autor.

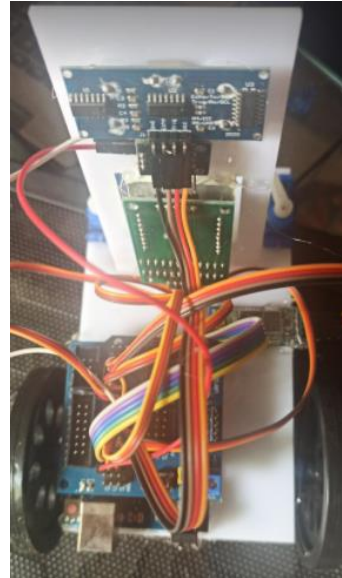
## 5.2 Construcción del Robot DYOR

La construcción del robot DYOR se la realizo tal cual como se lo diseño en la Figura 22, contando con todos los componentes electrónicos descritos en el Capítulo III, y teniendo ya impreso la base y el tronco del robot en acrílico, montando las piezas y los componentes en sus respectivos sitios, así como se observa la Figura 23 y Figura 23.



**Figura 23.** Frente del Robot DYOR.

**Fuente:** El Autor.



**Figura 24.** Detrás del Robot

DYOR.

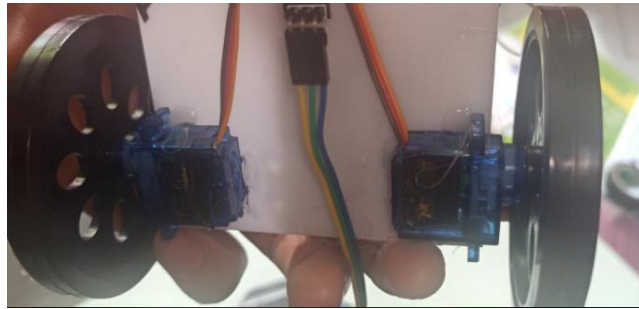
**Fuente:** El Autor.

**En la base:** aquí es montado el Arduino UNO, el módulo Bluetooth HC-06 y la fuente de alimentación y por debajo se encuentran los dos servomotores con sus respectivas ruedas las cuales harán que el robot se mueva, una rueda loca para darle estabilidad al robot y los servomotores que serán interpretados como las manos del robot, tal cual como se aprecia en las siguientes figuras:



**Figura 25.** Batería y placa Arduino.

**Fuente:** El Autor.



**Figura 26.** Servos motores de rotación continua.

**Fuente:** El Autor.



**Figura 27.** Rueda Loca ensamblada en la base.  
posición.

**Fuente:** El Autor.



**Figura 28.** Servos motores de posición.

**Fuente:** El Autor.

**En el tronco:** en esta parte se encuentra el sensor ultrasónico HC-SR04, zumbador, matriz Led 8x8 y por debajo el seguidor de línea, en esta parte va puesto como el tronco y cabeza del robot, haciendo notar que el sensor ultrasónico serán los ojos del robot, el zumbador será la nariz del robot y la matriz de Leds será la boca del robot dándole un buen aspecto al robot DYOR tal como se indica en las siguientes figuras:



**Figura 29.** Cuerpo del robot con los elementos montados.

**Fuente:** El Autor.



**Figura 30.** Sensor Siguelínea.

**Fuente:** El Autor.

### 5.2.1 Transformar un servo motor de posición a rotación continua

Existe la posibilidad que en el mercado de la localidad no exista la venta de los servomotores de rotación continua y, por ende, se tenga que acudir a otras opciones, una de ellas es transformar un servomotor de la familia SG90 (de posición) a rotación continua. Una manera factible de hacerla es desconectando el potenciómetro de la tarjeta controladora y ubicarle dos resistencias (mayores o iguales a 1K) de mismo valor en vez del potenciómetro.

- **Primer paso:** abrir el servomotor sacando para cortar los dos frenos que contiene, estos frenos son los que les dan el límite de llegar a 180°, cada servomotor tiene dos frenos, uno en el piñón superior (Figura 312) y el otro se presenta como una “cepita” en el potenciómetro (Figura 31).



**Figura 31.** Piñón del servo motor SG90.  
potenciómetro.

**Fuente:** El Autor.



**Figura 32.** Módulo de frenado del potenciómetro.

**Fuente:** El Autor.

- **Segundo paso:** desconectar el potenciómetro de la tarjeta controladora, en el centro de la tarjeta ubicar las dos puntas de las dos resistencias y en los otros dos sobrantes las puntas de cada resistencia de cada lado, para lograr que el centro tenga un punto común y las otras distribuidas con el fin de tener el giro a la izquierda, el giro a la derecha y la parada del servomotor, así como se observa en la Figura 33.

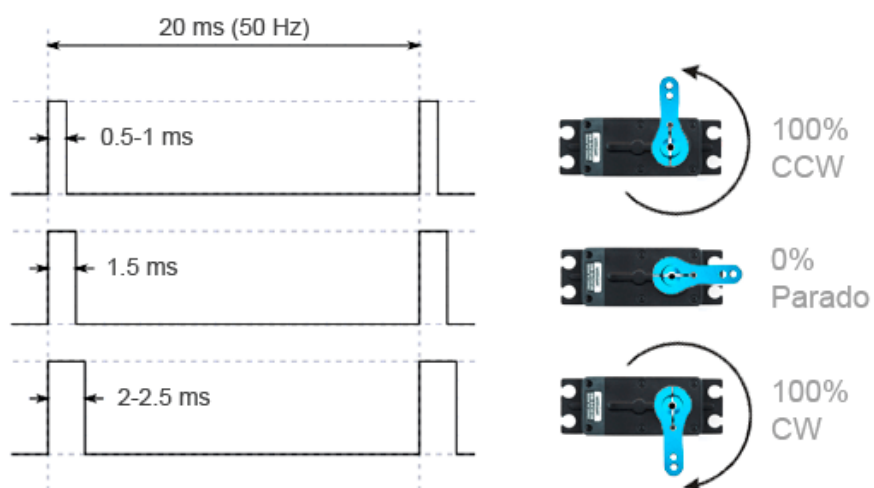


**Figura 33.** Servo motor con sus respectivas resistencias soldadas.

**Fuente:** El Autor.

- **Tercer paso:** armar el servomotor tanto en la parte superior como al inferior, ubicando cada uno de sus tornillos para que este seguro para proceder a probar el servomotor con una pequeña programación con el fin de observar los dos giros y la fase de parada.

Para la programación se debe tener en cuenta que el control de un servo de rotación continua es idéntico al de un servo convencional, sólo varía el significado de la señal de control, que en lugar de transformarse en ángulo de posición se interpreta como velocidad angular, en ambos sentidos de giro, así como se señala en la Figura 34.



**Figura 34.** Pulsos para darle el giro y paro al servo motor modificado.

**Fuente:** Luis Llamas – Ingeniería, informática y diseño.

Adaptando los 3 ángulos de posición a cada pulso de la Figura 34 para cada giro del servo, se tiene que:

- 0° el servo girara hacia el lado izquierdo.
- 90° el servo permanecerá detenido.
- 180° el servo girara hacia el lado derecho.

### 5.3 Conexión de los componentes con la placa Arduino UNO y el Shield

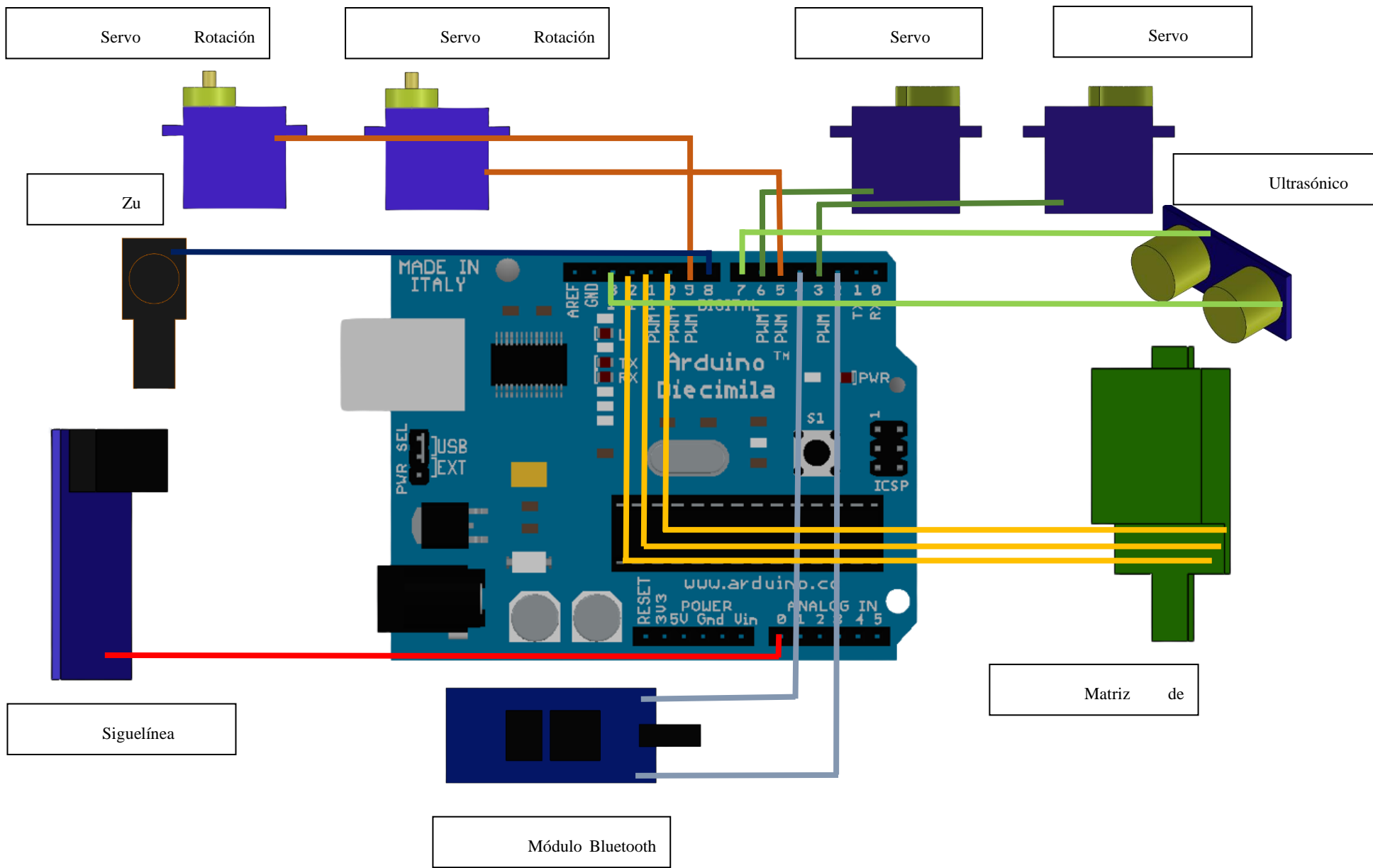
Para la conexión de cada componente del robot DYOR se propone la siguiente configuración como se muestra en la Tabla 4.

**Tabla 4.** Conexión de cada componente al Shield de la placa Arduino UNO.

COMPONENTE ELECTRÓNICO	PINES
Servo rotación izquierdo	9
Servo rotación derecho	5
Servo de posición izquierdo	6
Servo de posición derecho	3
Módulo Bluetooth	2 (RX – Arduino) – 4(TX – Arduino)
Matriz de LEDs 8x8	10(CS), 11(CLK), 12(DIN)
Sensor Ultrasónico SR-HC04	7(ECHO), 13 (TRIGGER)
Zumbador	8
Módulo Siguelíneas	A0

Fuente: El Autor.





## 5.4 Programación

En el proceso de programación del robot, se aplica un lenguaje de programación por bloque. Es muy sutil al momento de entenderlo y la tendencia al aprendizaje es muy rápida. Los estudiantes se adaptarán a este método de programación utilizando software que le permitirán desarrollar la escritura de una secuencia lógica y ordenada de instrucciones mediante el uso y aplicación de comandos fáciles de adaptar para hallar una solución efectiva y correcta a un problema determinado. El software que se aplicara es:

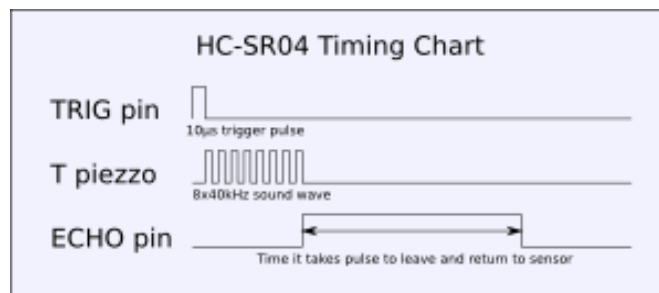
- Arduino

### 5.4.1 Arduino

El software Arduino es un programa libre que facilita la escritura de código y la carga del mismo en la placa. Se realiza la programación que va desde lo sencillo a lo complejo para posteriormente desarrollar las guías prácticas. A continuación, se presenta la metodología de programar los elementos del robot y las funciones que se han programado para los movimientos de robot DYOR:

#### 5.4.1.1 Programación de los elementos y sensores.

- Sensor Ultrasónico HC-SR04



**Figura 35.** Ancho de pulso del sensor HC-SR04.

**Fuente:** PanamaHitek – Arduino Sensor Ultrasónico HC-SR04.

Posee 4 pines de los cuales 2 son de alimentación (VCC Y GND) y 2 de las señales digitales (Trig y Echo). Para generar la medición en el pin del Trig se debe dar un pulso de 10 us, el sensor comienza a generar un pulso a 40kHz para iniciar la ecolocalización, cuando ya detecta la distancia por el pin del Echo nos vuelve un pulso que corresponde al doble de la distancia (de ida y de venida).

1. En la declaración de variables se define los dos puertos:
  - #define Echo "pin conectado al Arduino"
  - #define Trig "pin conectado al Arduino"
2. Por las características del sensor, se declara variable que permitan medir con decimales:
  - long duración, distancia;
3. En el void setup () se define que puerto será la salida y entrada del Arduino:
  - pinMode(Echo, INPUT);
  - pinMode(Trig, OUTPUT);
4. En la parte del void loop () se define el pulso que se enviara desde el Trig con el fin de medir la distancia de los objetos, ubicándole un delay en microsegundos:
  - digitalWrite(Trig, LOW);
  - delayMicroseconds(2);
  - digitalWrite(Trig, HIGH);
  - delayMicroseconds(10);
  - digitalWrite(Trig, LOW);

Tener en cuenta de trabajar con la velocidad del sonido igual a  $340 \text{ m/s} = 1/29 \text{ cm/us}$ , se incluye la formula en la programación para obtener valores en centímetros y con condicionales, dependiendo lo que se desee crear, se definen en la programación.

- **Servo Motor de Posición**

El servo motor de posición se facilita su programación con una librería que nos brinda el software de Arduino. El tipo de señal que cuentan es de dar pulso de en ms, sabiendo la igualdad en el pulso que se da y en los ángulos que corresponde:

**Tabla 5.** Relación del ancho de Pulso con los ángulos de posición.

Ancho de Pulso (ms)	Ángulo (°)
1	0
1.5	90
2	180

Fuente: El Autor.

1. Declaración de variables y la biblioteca del servo motor:
  - #include <Servo.h>
  - Servo servo1;
  - Servo servo2;
2. En el void setup () se declara los puertos conectados del servo al Arduino.
  - servo1.attach(“pin conectado al Arduino”);
  - servo2.attach(“pin conectado al Arduino”);
3. En void loop () se define los movimientos del servo, según el ángulo se le da el movimiento del servo, recordando que su rango es de 0° a 180°:
  - Servo1.write (“ángulo de movimiento”);
  - Servo1.write (“ángulo de movimiento”);

- **Servo Motor de rotación continua**

El caso del servo motor de rotación continua fue efecto de la modificación de uno de posición, este proceso esta descrito en el marco teórico. Su programación es similar al de un servo de posición, teniendo en cuenta la siguiente tabla:

**Tabla 6.** Relación del ángulo con el movimiento de giro.

Ángulo (°)	Movimiento (Sentido de Giro)
0	izquierda
90	detenido
180	derecha

**Fuente:** El Autor.

Se siguen los mismos pasos de los servos de posición, con la diferencia de que en la declaración de variable se ubica el ancho de pulso en un mínimo y máximo que; haciendo la metodología de prueba y error para lograr equilibrar los dos giros y el estado de paro, se obtuvo los siguientes datos:

- `int PINSERVO1 = 5;`
- `int PULSOMIN1 = 520;`
- `int PULSOMAX1 = 1500;`

Del libro (Torrente Artero, 2013) “ARDUINO - Curso Práctico de formación” se ubica la función *attach*, así el lenguaje en el void setup ():

- `myservo1.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);`

Es otra manera de programar el servo motor que, en este caso, beneficia el movimiento del servo motor modificado. De esta manera se sigue la misma metodología de programación del servo motor de posición, teniendo en cuenta los datos de la Tabla 6 para darle los movimientos al servo.

- **Módulo Bluetooth HC – 06**

Para la programación de este módulo se debe tener en cuenta que el pin de transmisor Tx del módulo debe estar conectado al pin Rx del Arduino; de la misma

manera el Rx del módulo conectado en el Tx de la placa. Esta conexión se realiza debido a que lo que transmita el módulo debe recibir la placa y viceversa. Ubicando la biblioteca y funciones que indica (Torrente Artero, 2013) del módulo su programación es la siguiente:

1. Se declara las variables y se incluye la bibliografía del módulo:
  - `#include <SoftwareSerial.h>`
  - `SoftwareSerial mybluetooth ("pin RX", "pin TX"); //RX y TX`
2. Se define la frecuencia de transmisión y recepción de datos en el void setup ():
  - `mybluetooth.begin (9600);`
3. En el void loop () se define la función de tendrá el módulo, puede ser de obtener datos o a definir una función a través de un "if", por ejemplo:
  - `if (mybluetooth.available()){`
  - `char letra=mybluetooth.read();`
  - `if(letra=='W')`

Al momento de entrelazar el dispositivo con el módulo bluetooth se necesitar una clave la cual por defecto es "1234" que viene establecida, se la puede modificar tanto la clave como el nombre.

- **Seguimiento TCRT5000**

El sensor seguimiento nos dos valores, un alto (cuando detecta un color negro) y un bajo (cuando detecta la claridad), teniendo en cuenta su lectura, se programa de la siguiente manera:

1. Se declara variables del sensor:
  - `int SENSOR = ("pin de conexión a la placa");`
  - `int VALOR; // se declara esta variable para guardar los datos`
  - `int ANTERIOR = 1; // para iniciar con un valor de 1`
2. Se define que forma será el sensor, en este caso es una señal de entrada, entonces en el void setup ():
  - `pinMode (SENSOR, INPUT);`
3. Se define los datos de lectura del sensor en un void loop () con la ayuda de un condicional "if":

- VALOR = digitalRead (SENSOR);
- if (VALOR = ANTERIOR) {
- if (VALOR == HIGH)
- Serial.println("Línea");
- }else{
- Serial.println("Fuera");
- ANTERIOR = VALOR;}

- **Zumbador**

También llamado Buzzer, es el elemento digital que puede emitir pitidos como frecuencias específicas (melodías). La programación de este elemento es variada, pero tiene bibliotecas que mejoran y facilitan su programación.

1. Se declara las variables y las bibliotecas a usar:
  - #include <EasyBuzzer.h>
2. Se define el pin en el cual estará conectado en void setup () de la placa Arduino.
  - EasyBuzzer.setPin (8);
3. Se declara el sonido que se desea programar, por ejemplo, un pitido, donde se ubica la frecuencia y la duración beep en ms.
  - EasyBuzzer.singleBeep (2000, 100)

Además, se puede programar melodías sabiendo la equivalencia de la frecuencia del sonido en el zumbador con las notas musicales.

- **Matriz de Leds 8x8 (con integrador):**

La programación de este dispositivo puede ser de distintas maneras, las librerías con las que cuenta Arduino son amplia y facilita la programación del dispositivo.

1. Se define la librería y se declara las variables:
  - #include <MatrizLed.h>
  - MatrizLed pantalla;
2. Se define los pines de la matriz en el orden correspondiente en el void setup ():
  - pantalla.begin (12,11,10,1); // (DIN, CLK, CS, Nro.Dispositivo)
3. Se ubica el texto que desea que se muestre en la matriz en un void loop ():

- pantalla.borrar();
- pantalla.escribirFraseScroll("INGENIERIA Y FUTBOL",100); //(Texto, milisegundos)

Con otras bibliotecas es posible programar cada led de la matriz, como por ejemplo crear una secuencia de forma vertical u horizontal, cada programación siempre será dependiendo el fin que se quiera obtener.

#### **5.4.1.2 Programaciones del Robot DYOR.**

Con las funciones y bibliotecas pertinentes para la programación de cada elemento del robot DYOR, se realiza programaciones en las cuales se da movimiento al robot que serán ejecutadas dependiendo de los sensores (Ultrasónico y/o Siguelínea) o con un mando a distancia comunicado al módulo bluetooth; las programaciones principales que se realizaron se encuentran a continuación:

- Anexo 4. DYOR – Detección de Obstáculos.
- Anexo 5. DYOR – Detección de Obstáculos con Alarma.
- Anexo 6. DYOR – Controles básicos (derecha – izquierda – Avanzar – Retroceder) a robot a distancia.

### **5.5 Guías Prácticas**

El desarrollo de las guías prácticas se define de acorde a la experiencia obtenida al ejecutarlas tanto en la construcción del robot, como en la programación. El modelo de las guías prácticas se realiza de acuerdo a la investigación efectuada anteriormente, ubicando cada una de las partes pertinentes en cada una de ellas. Se detalla la los objetivos, conocimientos previos, equipos e instrumentos a utilizar, una corta exposición de la práctica y el procedimiento a seguir, donde provoca al estudiante a interesarse sobre la robótica educativa, comprendiendo de una manera diferente la construcción y programación de un robot DYOR. Además, de un apartado donde los estudiantes redacten sus resultados y a que conclusión llegan al momento de ejecutar las guías prácticas.

El modelo se plantea de esta manera:



## **PORTADA**

- A. Tema:**
- B. Objetivos:**
- C. Conocimientos Previos:**
- D. Equipos e instrumentos:**
- E. Exposición:**
- F. Proceso:**
- G. Resultados (elaborado por el estudiante).**
- H. Conclusiones (elaborado por el estudiante).**
- I. Discusión (elaborado por el estudiante).**
- J. Recomendaciones (elaborado por el estudiante).**
- K. Referencias.**

### **5.6 Validación de las Guías Prácticas**

Las guías prácticas se las desarrolla para establecer métodos confiables y válidos que permita a los estudiantes encargarse de realizar un proceso informado de diseño, construcción y programación, teniendo un mayor aprendizaje con la motivación de experimentar de manera digital y físico el proceso de programar y las ventajas que tiene aplicando la robótica educativa.

Estas guías prácticas son validadas siguiendo una metodología que consiste en hacer un pequeño curso de programación en Arduino a estudiantes del primer ciclo de la carrera de electricidad, comenzando desde lo más básico de encender un led, hasta desarrollar ciclos “for” y condicionales mostrando las funciones que se puede crear y las diferentes actividades que se pueden realizar, además de cómo se puede resolver

problemas de la vida real aplicando estas funciones de programación tanto en led como en los elementos del robot DYOR.

### 5.6.1 Ejercicios aplicados en el curso de programación

#### 5.6.1.1 Encender un Led y ciclo for.

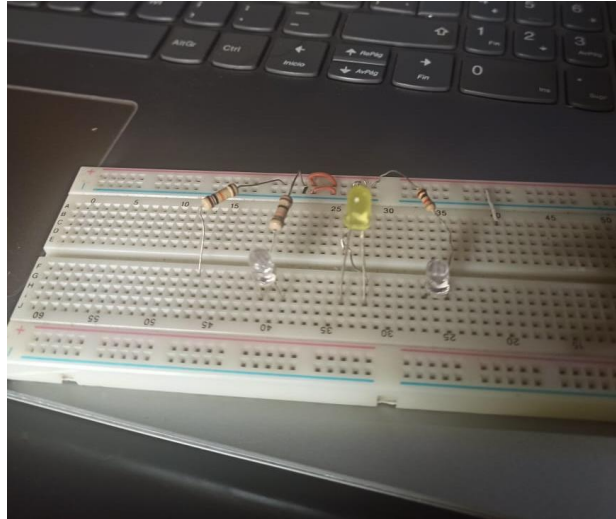
El primer ejercicio básico es encender un led con Arduino, ejecutando su programación, siguiendo todos los pasos descritos anteriormente, respetando las partes y el orden de programar en Arduino:

- **Encender un Led:**

```
1 int led = 13;
2
3 void setup() {
4   pinMode(led, OUTPUT);
5
6 }
7
8 void loop() {
9   digitalWrite(led, HIGH);
10  delay(1000);
11  digitalWrite(led, LOW);
12  delay(1000);
13
14 }
```

**Figura 36.** Encender y apagar un Led en un periodo de 1 s.

**Fuente:** El Autor.



**Figura 37.** Circuito de Led para el ejercicio en protoboard.

**Fuente:** El Autor.

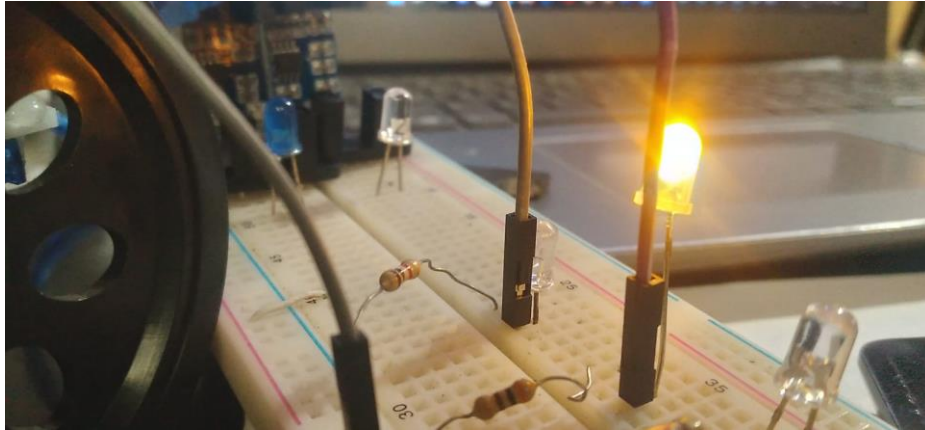
En este ejercicio se presenta el funcionamiento de encender un led con un circuito en un protoboard como se observa la Figura 37, explicando la importancia de las resistencias en los leds y que es posible usar el pin 13 del Arduino con un led directo, además de poder usar el led del Arduino para esta clase de ejercicios.

- **Parpadear un Led 10 veces:**

```
LEDS_EJERCICIO_1 $
1 int led1 = 8;
2 int led2 = 11;
3 int x ;
4
5 void setup() {
6
7   pinMode(8, OUTPUT);
8   pinMode (11, OUTPUT);
9
10  for(x = 0 ; x < 10; x++){
11    digitalWrite(led1, HIGH);
12    delay(500);
13    digitalWrite(led1, LOW);
14    delay(500);
15  }
16
17 }
18
19
20 void loop() {
21
22 }
```

**Figura 38.** Ciclo For de parpadeo de un Led (10 veces).

**Fuente:** El Autor.



**Figura 39.** Ejercicio de encender 10 veces un Led en el void setup ().

**Fuente:** El Autor.

En esta ocasión se aplica el ciclo for, demostrando que se puede simplificar la programación y hasta facilitar su función, se puede aplicar en un caso real como, por ejemplo: Un contador de un lanzamiento de una nave al espacio, donde el led debe indicar los 10 segundos antes del lanzamiento. Y así, muchos casos de la vida real se pueden ejercer con la programación básica de los leds.

- **Ciclo for con servo motor**

```
EJEMPLO_SERVOMOTOR_FOR
1 #include <Servo.h>
2
3 Servo servol;
4 Servo servo2;
5
6
7 void setup() {
8   servol.attach(6);
9   servo2.attach(3);
10
11 }
12
13 void loop() {
14   for(int x=90; x < 180; x++){
15     servol.write(x);
16     servo2.write(x);
17     delay(20);
18   }
19   for(int x=180; x >=90; x--){
20     servol.write(x);
21     servo2.write(x);
22     delay(20);
23   }
24
25 }
```

**Figura 40.** Ciclo for en servo motor.

**Fuente:** El Autor.

De este ejercicio se expone que no solo los ciclos for se pueden usar en leds, sino también en servo motores, controlando la velocidad de movimiento de estos, obteniendo una diferente manera de programarlos, además de visualizar los movimientos más uniformes y la aplicación de un segundo for de forma inversa como lo muestra la programación de la Figura 40.

Por último, se aplica la encuesta a los estudiantes con el fin de validar las guías prácticas, además de conocer el pensamiento del estudiante; que es lo que le motiva aprender y cuál es el aprendizaje que obtiene al observar una metodología diferente de enseñanza como lo es la robótica educativa, comprendiendo que las matemáticas nos simplifican la comprensión en el ámbito de la programación y que llega a un grado de dar movimiento a los robots. El modelo de cuestionario que se tomó es el siguiente:

# UNIVERSIDAD NACIONAL DE LOJA

## **Cuestionario para Determinar la Influencia del robot DYOR como motivación para el aprendizaje del lenguaje de programación.**

*Jorge Miguel Chávez Retete*

Con este cuestionario se pretende analizar la motivación que ejerce el robot DYOR en los estudiantes al momento de adquirir el aprendizaje del lenguaje de programación, construyendo y programando su propio robot.

### **INFORMACIÓN PERSONAL (MARCA LA OPCIÓN CORRECTA)**

#### **Sexo:**

- Hombre
- Mujer

#### **Edad:**

- 20 años o menos
- 21-24 años
- 25 o más

#### **Ha escuchado sobre la Robótica Educativa:**

- Si
- No

**Ha escuchado sobre el robot DYOR “Do Your Own Robot” (en español “Diseña tu propio robot”):**

- Si
- No

**NIVEL DE CONOCIMIENTO SOBRE LENGUAJE DE PROGRAMACIÓN Y ROBÓTICA EDUCATIVA**

**¿Qué dificultad vez aprender lenguajes de programación?**

- Alta
- Media
- Baja

**¿Te motivaría aprender un lenguaje de programación a través de la construcción de tu propio robot?**

- Si
- No

**Has trabajado con Placas de Arduino**

- Si
- No

**Has usado el Software de Arduino**

- Si
- No

**Sabes lo que son bloques de programación**

- No
- Si

**¿Crees que comenzar a aprender a programar sería más interesante aplicando en programar robots?**

- Si
- No

**¿Crees que puedes construir tu propio robot?**

- Si
- No

**Has creado alguna aplicación en el sistema Android**

- Si
- No

**¿Te consideras capaz de crear un mando a distancia para el robot DYOR?**

- Si
- No

**¿Qué funciones has observado que realizan los robots?**

---

---



## **6. Resultados**

### **6.1 Guía Práctica de diseño del robot DYOR.**

Siguiendo la metodología propuesta, se desarrolló la guía práctica del diseño del robot DYOR, mostrando los pasos a utilizar, los conocimientos previos que se deben tener y el aprendizaje que obtendrán al aplicar este proceso que va desde el diseño del robot en software (como SolidWorks) hasta la construcción del mismo. La motivación de diseñar un robot propio es enorme, el aprendizaje que obtendrán será mayor ya que se encontrarán con problemas que serán capaces de resolver por si solos, logrando así a obtener el conocimiento suficiente para hacer su propio robot DYOR.

La Guía Práctica que se desarrollo es la siguiente:

# **UNIVERSIDAD NACIONAL DE LOJA**

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

## **ROBOT DYOR**

CICLO # 1

GUÍA PRÁCTICA - ROBÓTICA EDUCATIVA



PERIODO ACADÉMICO: ABRIL – SEPTIEMBRE /2022

TUTOR ENCARGADO: ING. JEFFERSON CAMACHO

TESISTA: JORGE MIGUEL CHÁVEZ R.

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**A. Tema:**

Diseño y construcción de un Robot DYOR.

**B. Objetivos:**

- Aplicar un software de diseño ingenieril (SolidWorks) para la elaboración del robot DYOR.
- Emplear el hardware Arduino UNO y conocer más los componentes electrónicos programables.
- Desarrollar la destreza de crear robots DYOR propios.

**C. Conocimientos Previos:**

- Uso del Software SolidWorks.
- Conocimientos sobre sensores, servomotores, módulos entre otros componentes electrónicos.
- Impresiones 3D y a Laser.

**D. Equipos e instrumentos**

- Computadora
- Arduino UNO
- Shield para servomotores
- Fuente de Alimentación para Arduino UNO
- Sensor de Proximidad HC-SR04

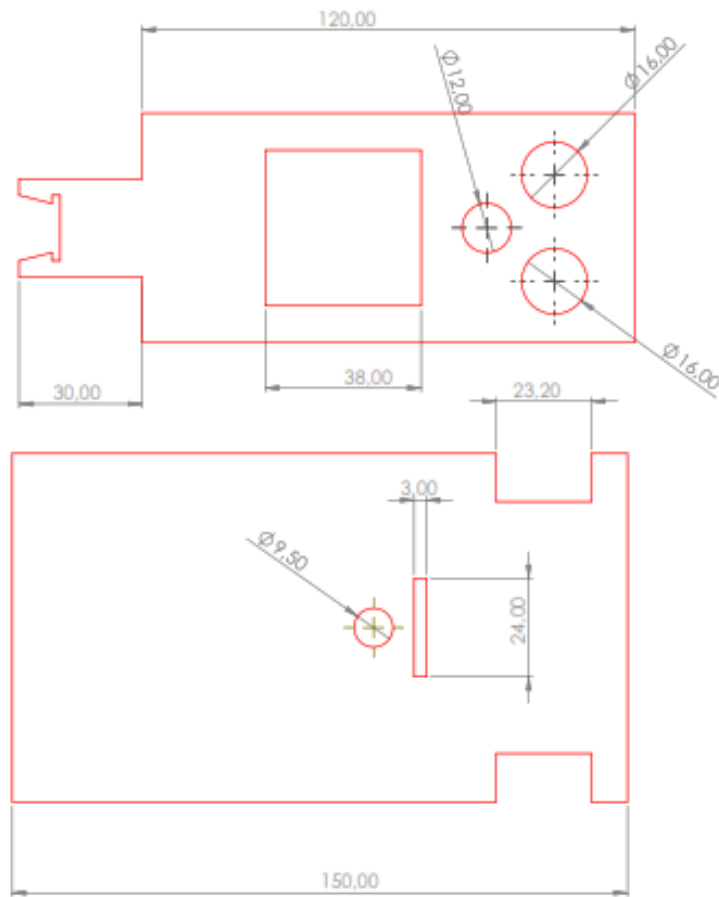
- Módulo Bluetooth HC-06
- Servomotores de posición y rotación continua (con ruedas
- Zumbador
- Siguelíneas TCRT5000
- Rueda Loca
- Matriz de Leds 8x8 (con integrador)

### **E. Exposición**

La robótica educativa ayuda a despertar el interés del estudiante motivándolo a desarrollar las destrezas de construcción y con los conocimientos previos adquirir un mayor aprendizaje en el ámbito de los robots.

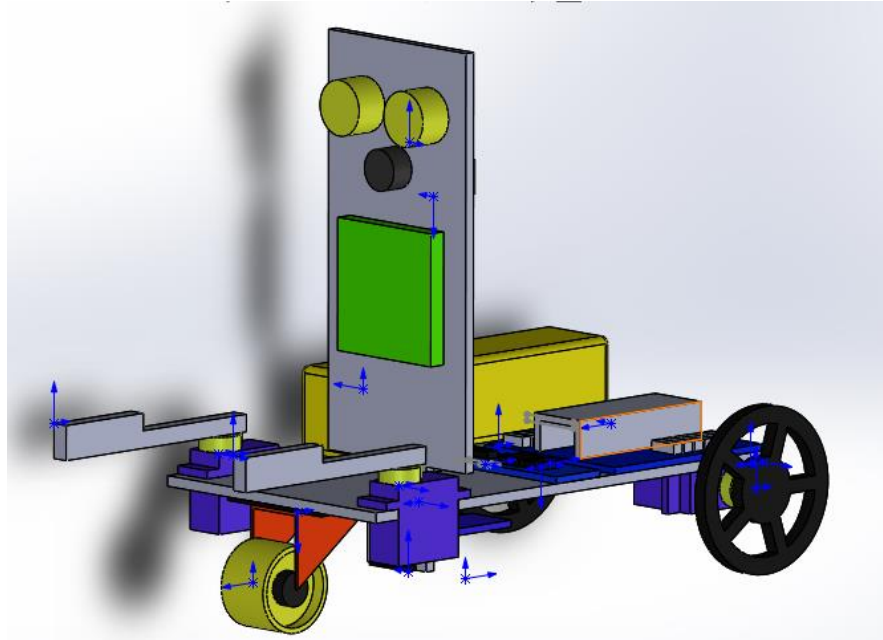
### **F. Proceso**

**Primer paso:** Con el uso del software SolidWorks se ejecuta el diseño del robot. Deberá crear un diseño del cual al robot se le logre implementar los componentes electrónicos programables, teniendo en cuenta que funciones se le podrá programar a nuestro robot para sus movimientos, como ejemplo se plantea la siguiente figura que se compone de una base y el tronco del robot.



**Figura 1.** Base y Tronco del robot DYOR.

**Segundo Paso:** Diseñar los componentes electrónicos en el software SolidWorks, para que posteriormente con la opción de “Ensamblaje” se complete el diseño del robot DYOR obteniendo así una visualización previa de cómo será nuestro robot, como el ejemplo que se muestra en la figura 2.



**Figura 2.** Robot DYOR ensamblado en SolidWorks.

**Tercer paso:** Construir el robot DYOR que se la realizo en el software SolidWorks, contando con todos los componentes electrónicos descritos anteriormente, y teniendo ya impreso la base y el tronco del robot en el material que más les sea factible (que puede ser en 3D o Laser), montando las piezas y los componentes en sus respectivos sitios.

**G. Resultados (elaborado por el estudiante).**

**H. Conclusiones (elaborado por el estudiante).**

**I. Discusión (elaborado por el estudiante).**

**J. Recomendaciones (elaborado por el estudiante).**

**K. Referencias**

- <http://dyor.roboticafacil.es/>
- [https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfluT0RS9C\\_260KP/view?usp=sharing](https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfluT0RS9C_260KP/view?usp=sharing)

Con esta Guía Práctica se demuestra que la idea de construir nuestro propio robot no es tan complicada o no está tan lejos, motivando a los estudiantes a adquirir el conocimiento necesario para crear cosas increíbles, teniendo confianza en sí mismo y despertar el interés hacia la robótica y conocer el lenguaje de programación que cada vez es más utilizado con respecto a Arduino.

También el estudiante podrá exponer sus resultados y se observará la creatividad de cada uno al momento de diseñar su robot, concluirá con cada objetivo de la práctica, realizará un contraste sobre la experiencia que ha ganado al diseñar el robot y aportarán con recomendaciones para mejorar la metodología si es el caso.

## **6.2 Guías Prácticas de programaciones del robot**

El desarrollo de las guías prácticas de programación se realizaron de la misma manera que la de las guías del diseño del robot DYOR. Se usaron cada elemento del robot para crear los movimientos del mismo, explicado la programación de cada elemento en la plataforma Arduino, se aplicaron 3 principales programaciones y de cada una se llevó a cabo una guía práctica:

**6.2.1 Guía Práctica del Sensor Siguelínea.**

**UNIVERSIDAD NACIONAL DE LOJA**

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**ROBOT DYOR**

CICLO # 1

GUÍA PRÁCTICA - ROBÓTICA EDUCATIVA



PERIODO ACADÉMICO: ABRIL – SEPTIEMBRE / 2022

TUTOR ENCARGADO: ING. JEFFERSON CAMACHO

TESISTA: JORGE MIGUEL CHÁVEZ R.



ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**A. Tema:**

Programación del robot DYOR – Seguidor de Línea.

**B. Objetivos:**

- Aplicar la plataforma de Arduino.
- Conocer las bibliotecas del Sensor Siguelínea y los servos motores de Arduino.
- Conocer sobre la estructura de programación en el entorno de Arduino.

**C. Conocimientos Previos:**

- Robot DYOR.
- Plataforma del entorno Arduino.
- Arduino UNO.
- Electrónica Digital.

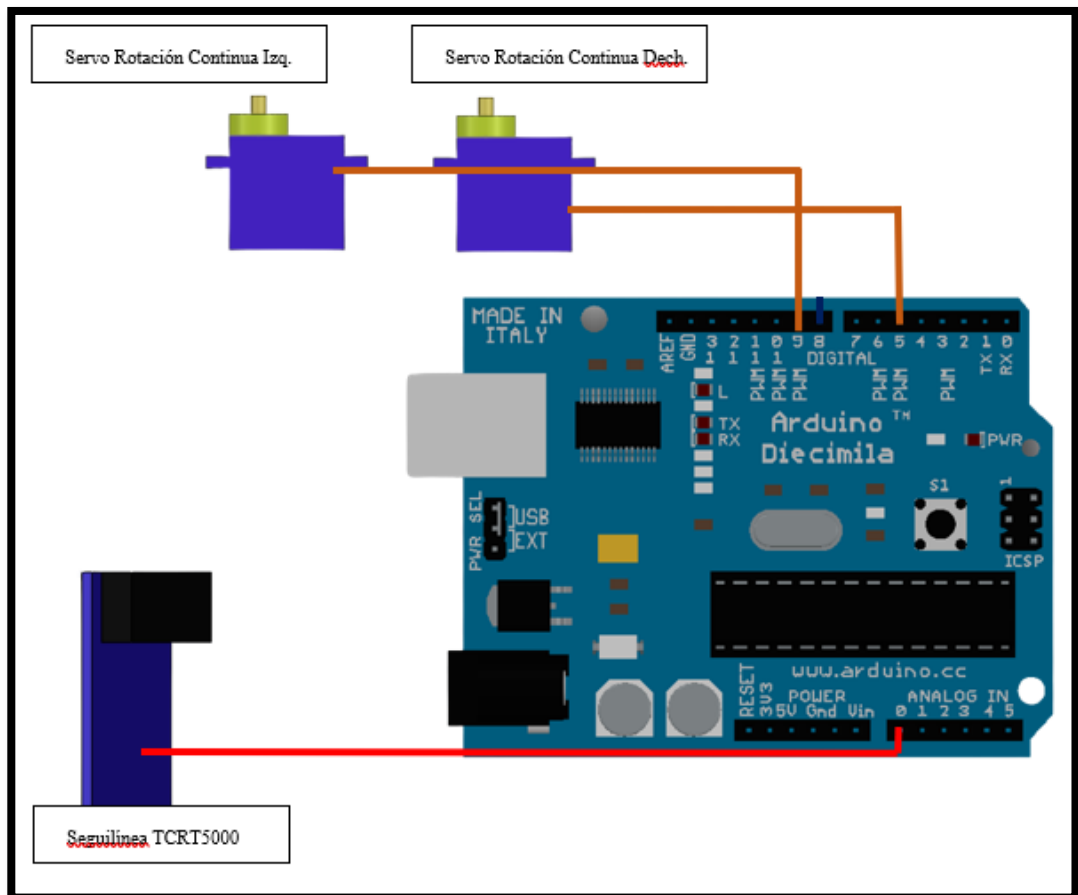
**D. Equipos e instrumentos:**

- Computadora
- Robot DYOR:
- Arduino UNO
- Servo Motores de rotación continua
- Módulo Siguelínea

**E. Exposición:**

La programación de un robot DYOR ayudará al estudiante a sostener las bases para un buen lenguaje de programación en el entorno de Arduino, creando una programación que provoque que el robot siga una línea negra de principio a fin y resolver los problemas que se vayan creando como por ejemplo cuando el robot se pierda de la trayectoria y hacer que vuelva a la línea.

#### F. Proceso:



**Figura 1.** Propuesta de conexión de los elementos digitales.

**Fuente:** El Autor.

**Primer paso:** declarar las variables globales al inicio de la programación como la configuración de terminales de entrada y salida, constante con un valor predeterminado o lo que se desee declarar sin limitaciones.

Biblioteca de los servomotores:

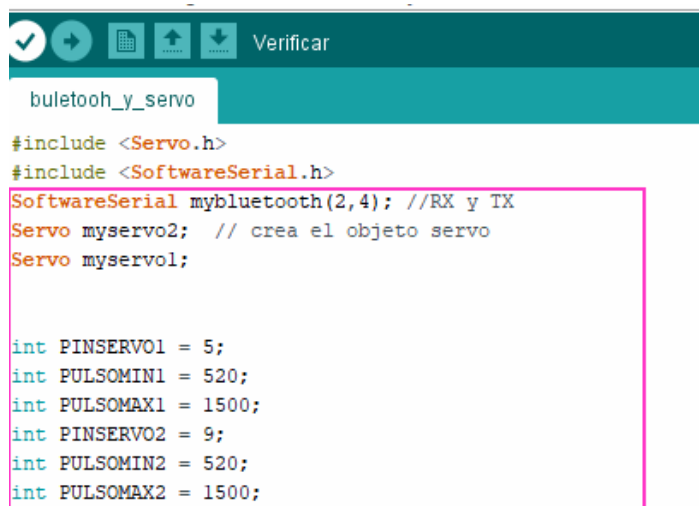
- `#include <Servo.h>`

Declaración de puertos del sensor Siguelínea y los servomotores:

- `int SENSOR = A0;`

// crea el objeto servo

- `Servo myservo2;`
- `Servo myservo1;`

A screenshot of an IDE window titled 'buletooh\_y\_servo'. The code is as follows:

```
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial mybluetooth(2,4); //RX y TX
Servo myservo2; // crea el objeto servo
Servo myservo1;

int PINSERVO1 = 5;
int PULSOMIN1 = 520;
int PULSOMAX1 = 1500;
int PINSERVO2 = 9;
int PULSOMIN2 = 520;
int PULSOMAX2 = 1500;
```

The code is color-coded: keywords in orange, comments in grey, and variable names in blue. A pink rectangular box highlights the servo-related code from the `Servo` declarations to the pin/pulse constant declarations.

**Figura 2.** Ejemplo de declaración de variables globales.

**Fuente:** El Autor.

**NOTA:** Tener en cuenta las bibliotecas a utilizar para no tener errores en la programación.

**Segundo paso:** programar las variables en el void `setup ()`: es la parte encargada de recoger la configuración, se delimita por inicio y cierre de llaves. En este apartado las instrucciones solo son ejecutadas una vez al iniciar la placa Arduino.

```

void setup() {
  mybluetooth.begin(9600);
  myservol.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);
  myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);
  myservol.write(90);
  myservo2.write(90);
}

```

**Figura 3.** Ejemplo de una programación en el void setup ().

**Fuente:** El Autor.

**Tercer paso:** programar en el void loop ():es la función del programa que se ejecutará repetidamente (en un loop – bucle), al igual que el setup es limitado por llaves de inicio y cierre. En esta parte del Sketch, las instrucciones del bloque son repetidas una y otra vez hasta que se ponga un punto de paro o una instrucción que detenga el ciclo en un lapso de tiempo.

```

void loop() {

  if (mybluetooth.available()>0) {
    char letra=mybluetooth.read();
    if(letra=='W'){ //hacia delante
      myservol.write(60);
      myservo2.write(180);
      delay(4000);
      myservol.write(90);
      myservo2.write(90);
    }
    if(letra=='F'){ // parada
      myservol.write(90);
      myservo2.write(90);
    }
    if(letra=='S'){ // hacia atras
      myservol.write(120);
      myservo2.write(0);
      delay(4000);
      myservol.write(90);
      myservo2.write(90);
    }
  }
}

```

**Figura 4.** Ejemplo de programación en el void loop ().

**Fuente:** El Autor.

**Cuarto paso:** compilar la programación, y cuando no exista ningún error en las líneas de programación, enviar al Arduino UNO y observar el comportamiento del robot DYOR.

**G. Resultados (elaborado por el estudiante).**

**H. Conclusiones (elaborado por el estudiante).**

**I. Discusión (elaborado por el estudiante).**

**J. Recomendaciones (elaborado por el estudiante).**

**K. Referencias:**

- [https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfIuT0RS9C\\_260KP/view?usp=sharing](https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfIuT0RS9C_260KP/view?usp=sharing)
- [https://drive.google.com/file/d/17L4P\\_WauDTh8ux4QQxVp\\_zlpsXCij9zz/view?usp=sharing](https://drive.google.com/file/d/17L4P_WauDTh8ux4QQxVp_zlpsXCij9zz/view?usp=sharing)

La programación del siguelínea nos indica como el robot se guía a través de una trayectoria trazada por una cinta negra, el problema el cual mejora la habilidad del estudiante es aplicar la condicional correcta para que cuando el robot se pierda de la línea, vuelva a la trayectoria y continúe su camino trazado. El resultado de la programación se puede observar:

“<https://drive.google.com/drive/folders/1NJV6ocxVBeL821gQc6jT7BCqbaeMxsJS?usp=sharing>”, donde se demuestra como sigue su trayectoria y como vuelve a la trayectoria después de perderse por un momento.



**Figura 41.** Robot DYOR – Siguelínea.

**Fuente:** El Autor.

6.2.2 *Guía Práctica del sensor Ultrasónico*

**UNIVERSIDAD NACIONAL DE LOJA**

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**ROBOT DYOR**

CICLO # 1

GUÍA PRÁCTICA - ROBÓTICA EDUCATIVA



TUTOR ENCARGADO: ING. JEFFERSON CAMACHO

TESISTA: JORGE MIGUEL CHÁVEZ R.

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**A. Tema:**

Programación del robot DYOR – Sensor Ultrasónico (HC-SR04).

**B. Objetivos:**

- Aplicar la plataforma de Arduino.
- Conocer las bibliotecas del Sensor Siguelínea y los servos motores de Arduino.
- Conocer sobre la estructura de programación en el entorno de Arduino.

**C. Conocimientos Previos:**

- Robot DYOR
- Plataforma del entorno Arduino
- Arduino UNO
- Electrónica Digital

**D. Equipos e instrumentos:**

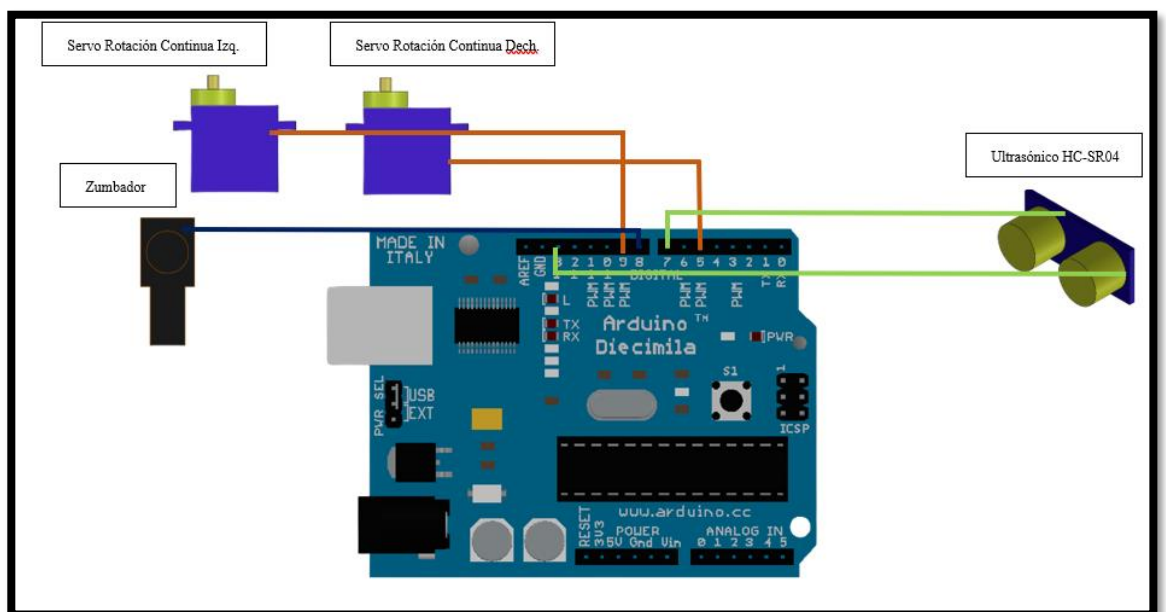
- Computadora
- Robot DYOR:
  - Servos motores
  - Sensor Ultrasónico
  - Zumbador
  - Matriz Led 8x8



## E. Exposición:

La interacción con el robot DYOR incentivara al estudiante a querer saber más de como programar el robot y que cumpla la función de avanzar y esquivar obstáculos, desarrollando una programación en el cual, con el sensor Ultrasónico, detecte un obstáculo a una determinada distancia que provoque que el robot se detenga y de la vuelta hacia otra dirección y en todo este proceso suene una alarma (con el zumbador), alertando al robot que se presenta un obstáculo y debe dar la vuelta.

## F. Proceso:



**Figura 1.** Propuesta de conexión de los elementos digitales a la placa Arduino.

**Fuente:** El Autor.

**Primer paso:** declarar las variables globales al inicio de la programación como la configuración de terminales de entrada y salida, constante con un valor predeterminado o lo que se desee declarar sin limitaciones.

Biblioteca de los servomotores:

- `#include <EasyBuzzer.h>`
- `#include <Servo.h>`

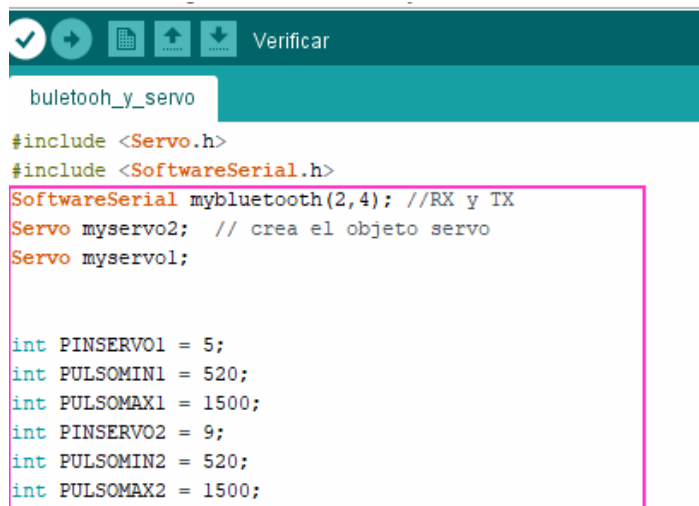
- #include <SoftwareSerial.h>

Declaración de puertos del sensor Ultrasónico y los servomotores:

- #define Pecho 7
- #define Ptrig 13
- long duracion, distancia;

crear el objeto servo

- Servo myservo2;
- Servo myservo1;



```

buletooh_y_servo
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial mybluetooth(2,4); //RX y TX
Servo myservo2; // crea el objeto servo
Servo myservol;

int PINSERVO1 = 5;
int PULSOMIN1 = 520;
int PULSOMAX1 = 1500;
int PINSERVO2 = 9;
int PULSOMIN2 = 520;
int PULSOMAX2 = 1500;

```

**Figura 2.** Ejemplo de declaración de variables globales.

**Fuente:** El Autor.

**NOTA:** Tener en cuenta las bibliotecas a utilizar para no tener errores en la programación.

**Segundo paso:** programar las variables en el void setup (): es la parte encargada de recoger la configuración, se delimita por inicio y cierre de llaves. En este apartado las instrucciones son ejecutadas una vez al iniciar la placa Arduino.

```

void setup() {
  mybluetooth.begin(9600);
  myservol.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);
  myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);
  myservol.write(90);
  myservo2.write(90);
}

```

**Figura 3.** Ejemplo de una programación en el void setup ().

**Fuente:** El Autor.

**Tercer paso:** programar en el void loop ():es la función del programa que se ejecutará repetidamente (en un loop – bucle), al igual que el setup es limitado por llaves de inicio y cierre. En esta parte del Sketch, las instrucciones del bloque son repetidas una y otra vez hasta que se ponga un punto de paro o una instrucción que detenga el ciclo en un lapso de tiempo.

```

void loop() {

  if (mybluetooth.available()>0) {
    char letra=mybluetooth.read();
    if(letra=='W'){ //hacia delante
      myservol.write(60);
      myservo2.write(180);
      delay(4000);
      myservol.write(90);
      myservo2.write(90);
    }
    if(letra=='F'){ // parada
      myservol.write(90);
      myservo2.write(90);
    }
    if(letra=='S'){ // hacia atras
      myservol.write(120);
      myservo2.write(0);
      delay(4000);
      myservol.write(90);
      myservo2.write(90);
    }
  }
}

```

**Figura 4.** Ejemplo de programación en el void loop.

**Fuente:** El Autor.

**Cuarto paso:** compilar la programación, y cuando no exista ningún error en las líneas de programación, enviar al Arduino UNO y observar el comportamiento del robot DYOR.

**G. Resultados (elaborado por el estudiante).**

**H. Conclusiones (elaborado por el estudiante).**

**I. Discusión (elaborado por el estudiante).**

**J. Recomendaciones (elaborado por el estudiante).**

**K. Referencias:**

- [https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfluT0RS9C\\_260KP/view?usp=sharing](https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfluT0RS9C_260KP/view?usp=sharing)
- [https://drive.google.com/file/d/17L4P\\_WauDTh8ux4QQxVp\\_zlpsXCij9zz/view?usp=sharing](https://drive.google.com/file/d/17L4P_WauDTh8ux4QQxVp_zlpsXCij9zz/view?usp=sharing)

Evadir obstáculos es una manera de prevenir que el robot se impacte con cualquier cosa que detecte a una distancia determinada, a través de condicionales, se aplica una programación en la cual permita que el robot al detectar un obstáculo a una distancia que le permita detenerse provoque active un sonido de alarma, muestre en su matriz de led una palabra común de obstáculo “PARED” y además haga un movimiento en donde retroceda y se dirija a otra dirección libre de obstáculos. La simulación en el robot de esta práctica se la puede observar:

<https://drive.google.com/drive/folders/1NJV6ocxVBeL821gQc6jT7BCqbaeMxsJS?usp=sharing>, donde muestra el movimiento del robot que ejecuta al momento de detectar un obstáculo a través del sensor ultrasónico.



**Figura 42.** Robot DYOR – Sensor Ultrasónico.

**Fuente:** El Autor.

6.2.3 *Guían Práctica del control a distancia del robot*

**UNIVERSIDAD NACIONAL DE LOJA**

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**ROBOT DYOR**

CICLO # 1

GUÍA PRÁCTICA - ROBÓTICA EDUCATIVA



TUTOR ENCARGADO: ING. JEFFERSON CAMACHO

TESISTA: JORGE MIGUEL CHÁVEZ R.

ÁREA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**A. Tema:**

Programación del robot DYOR – Control del robot a distancia (smartphone).

**B. Objetivos:**

- Aplicar la plataforma de Arduino.
- Conocer las bibliotecas de los servos motores de Arduino para darle movimiento al robot (de las palas y de los neumáticos).
- Conocer sobre la estructura de programación en el entorno de Arduino.

**C. Conocimientos Previos:**

- Robot DYOR
- Plataforma del entorno Arduino
- Arduino UNO
- Electrónica Digital

**D. Equipos e instrumentos:**

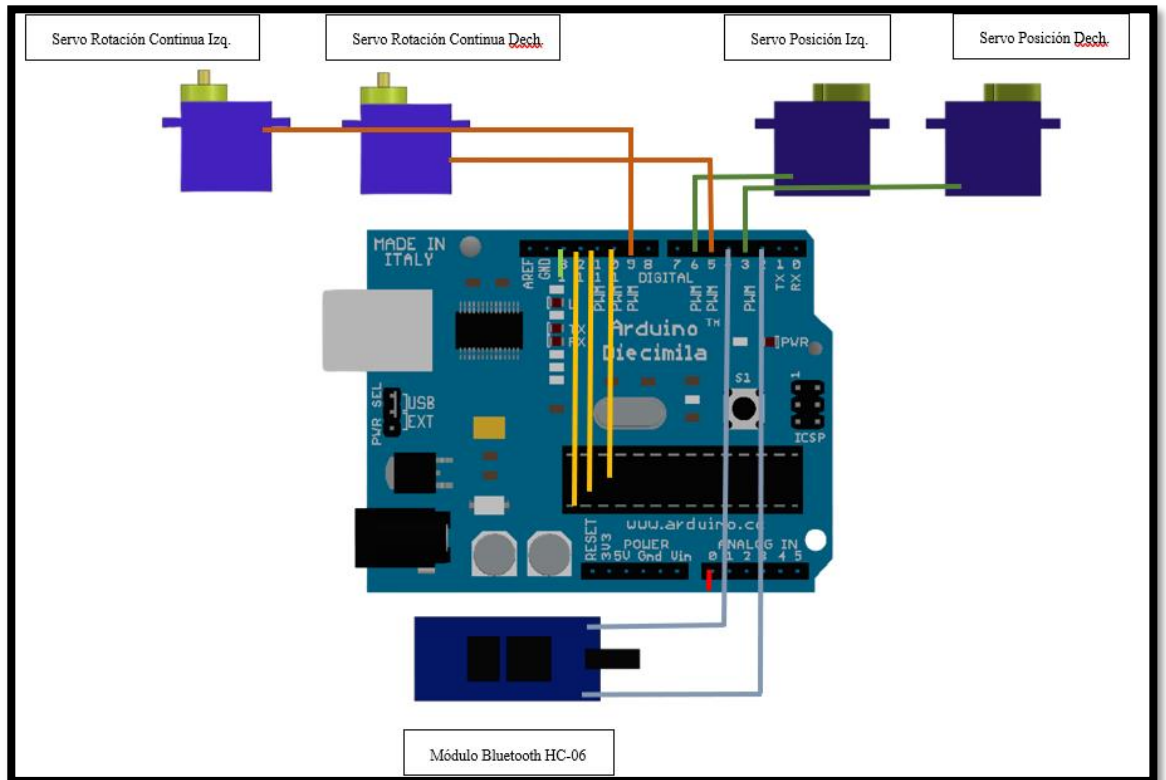
- Computadora
- Smartphone
- Robot DYOR:
  - Arduino UNO
  - Servo Motores (rotación continua – posición)
  - Módulo Bluetooth

**E. Exposición:**

La programación de un robot DYOR causará en el estudiante la capacidad de obtener de una manera más interesante el aprendizaje de la programación en el entorno

de Arduino, desarrollando una programación con el objetivo que le robot sea controlado por un usuario a distancia desde un smartphone.

## F. Proceso:



**Figura 1.** Propuesta de conexión de los elementos digitales del robot.

**Fuente:** El Autor.

**Primer paso:** declarar las variables globales al inicio de la programación como la configuración de terminales de entrada y salida, constante con un valor predeterminado o lo que se desee declarar sin limitaciones.

Biblioteca de los servomotores:

- `#include <Servo.h>`
- `#include <SoftwareSerial.h>`

Declaración de puertos del sensor Siguelínea y los servomotores:



- SoftwareSerial mybluetooth(2,4); //RX y TX
- Servo myservo2; // crea el objeto servo
- Servo myservo1;
- Servo myservo3;
- Servo myservo4;

```

1 #include <EasyBuzzer.h>
2 #include <Servo.h>
3 #include <SoftwareSerial.h>
4 #define Pecho 7
5 #define Ptrig 13
6 long duracion, distancia;
7 Servo myservo2; // crea el objeto servo
8 Servo myservol;

```

**Figura 2.** Ejemplo de declaración de variables globales.

**Fuente:** El Autor.

**NOTA:** Tener en cuenta las bibliotecas a utilizar para no tener errores en la programación.

**Segundo paso:** programar las variables en el void setup (): es la parte encargada de recoger la configuración, se delimita por inicio y cierre de llaves. En este apartado las instrucciones solo son ejecutadas una vez al iniciar la placa Arduino.

```

17 void setup() {
18   pinMode (SENSOR, INPUT) ;
19   myservol.attach(PINSERVO1, PULSOMIN1, PULSOMAX1) ;
20   myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2) ;
21   myservol.write (90) ;
22   myservo2.write (90) ;
23 }

```

**Figura 3.** Ejemplo de una programación en el void setup ().

**Fuente:** El Autor.

**Tercer paso:** programar en el void loop ():es la función del programa que se ejecutará repetidamente (en un loop – bucle), al igual que el setup es limitado por llaves de inicio y cierre. En esta parte del Sketch, las instrucciones del bloque son repetidas una y otra vez hasta que se ponga un punto de paro o una instrucción que detenga el ciclo en un lapso de tiempo.

```
25 void loop() {
26   VALOR = digitalRead(SENSOR);
27   myservol.write(90);
28   myservo2.write(90);
29   if (VALOR == HIGH) {
30     myservol.write(160);
31     myservo2.write(60);
32   }else{
33     myservol.write(90);
34     myservo2.write(0);
35     ANTERIOR = VALOR;
36   }
37 }
```

**Figura 4.** Ejemplo de programación en el void loop.

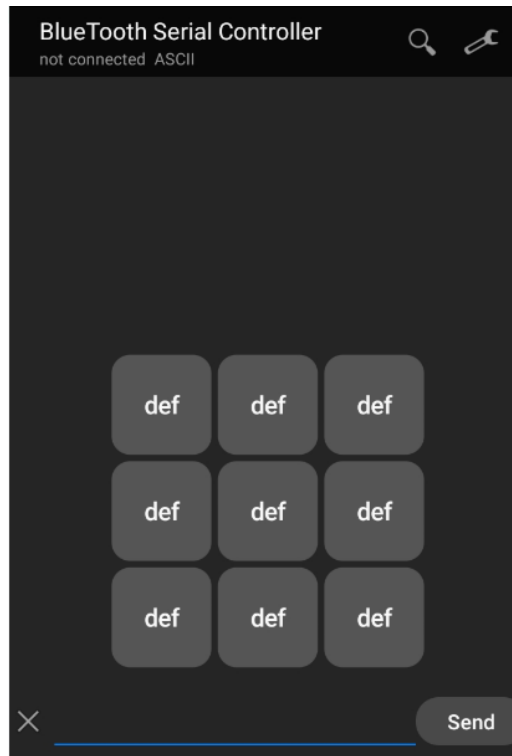
**Fuente:** El Autor.

**Cuarto paso:** compilar la programación, y cuando no exista ningún error en las líneas de programación, enviar al Arduino UNO.

**Quinto paso:** para controlar al robot DYOR, se utilizará las siguientes aplicaciones para el smartphone:

De esta manera es el entorno de cada aplicación:

**Aplicación 1:**



**Figura 7.** Entrono de la Aplicación 1.

**Fuente:** El Autor.

Esta aplicación permite definir el comando de cada botón, ubicando así el comando propuesto en la programación de la placa Arduino, además esta aplicación permite cambiar el aspecto de cada botón (def) para darle un mejor aspecto al controlador, adicionalmente permite enviar la señal escribiendo el comando en la línea inferior de la aplicación.

### **Aplicación 2:**



**Figura 8.** Entorno de la Aplicación 2.

**Fuente:** El Autor.

A diferencia de la aplicación 1, ésta da un aspecto diseñado del controlador (tipo el mando de una consola de Play Station) en donde se debe ubicar los controladores de cada botón ya previamente propuestos en la programación.

**Nota:** Puede elegir una de las aplicaciones descritas anteriormente y en cada una de ellas al terminar de configurarla se debe conectar al módulo bluetooth del robot DYOR.

- G. Resultados (elaborado por el estudiante).**
- H. Conclusiones (elaborado por el estudiante).**
- I. Discusión (elaborado por el estudiante).**
- J. Recomendaciones (elaborado por el estudiante).**
- K. Referencias:**

- [https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfluT0RS9C\\_260KP/view?usp=sharing](https://drive.google.com/file/d/1ICWw4GAPTbL8NIHVfluT0RS9C_260KP/view?usp=sharing)
- [https://drive.google.com/file/d/17L4P\\_WauDTh8ux4QQxVp\\_zlpsXCij9zz/view?usp=sharing](https://drive.google.com/file/d/17L4P_WauDTh8ux4QQxVp_zlpsXCij9zz/view?usp=sharing)

En total se realizaron 4 guías prácticas, a parte de los ejercicios puntuales que se expusieron en el curso de programación con los leds y los servos motores, una de ellas es del diseño y construcción del robot DYOR y las restantes son de programaciones que se han realizado al mismo, todas en la plataforma de Arduino.

Con el desarrollo de estos ejercicios básicos de programación en Arduino, se hizo práctico para cada estudiante, se despertó el interés a cada uno y captará de una manera rápida la metodología de programar en Arduino. Aplicando la metodología de declarar variables, definir el void setup () y de aplicar el bucle void loop () la compilación del programa no dará errores y se subirá correctamente a la placa.

Además, en el curso se expuso la guía práctica del robot DYOR – Control del robot a distancia (smartphone), indicando la programación utilizada y los movimientos que puede hacer el robot controlado desde el smartphone. Las líneas de programación se encuentran en la sección anterior “Programación del Robot DYOR”.



**Figura 43.** Demostración del mando a distancia y los movimientos del Robot DYOR.

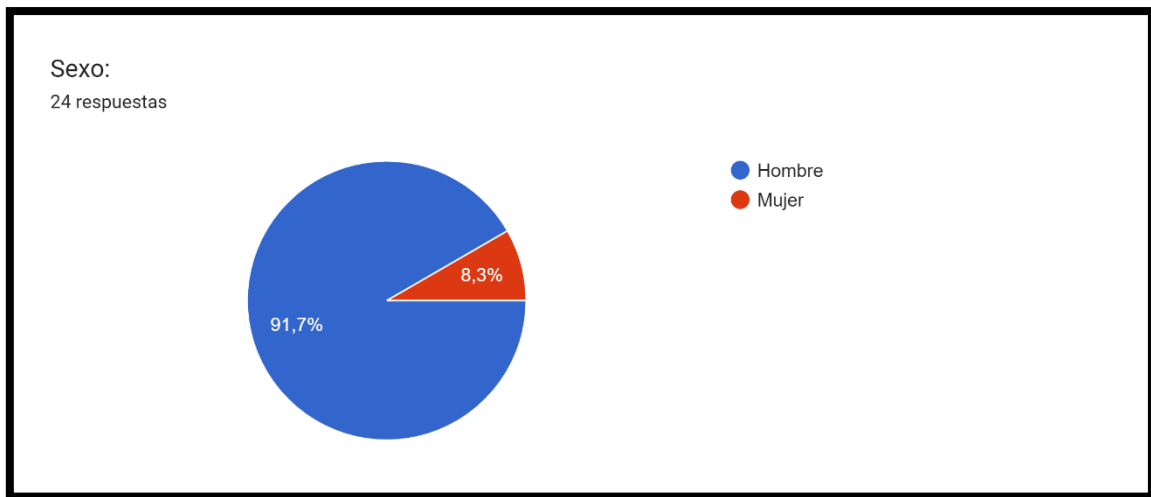
**Fuente:** El Autor.

### **6.3 Cuestionario aplicado a los estudiantes**

Al final el curso de programación, se tomó a cada uno el cuestionario en donde los estudiantes respondieron cada una de las preguntas propuestas, ya con un aprendizaje más amplio sobre la robótica educativa, cada pregunta se propuso con el fin de demostrar que la robótica educativa motiva al estudiante a captar de una manera más eficiente la programación.

En las preguntas de información personal se obtuvo:

- **Sexo**

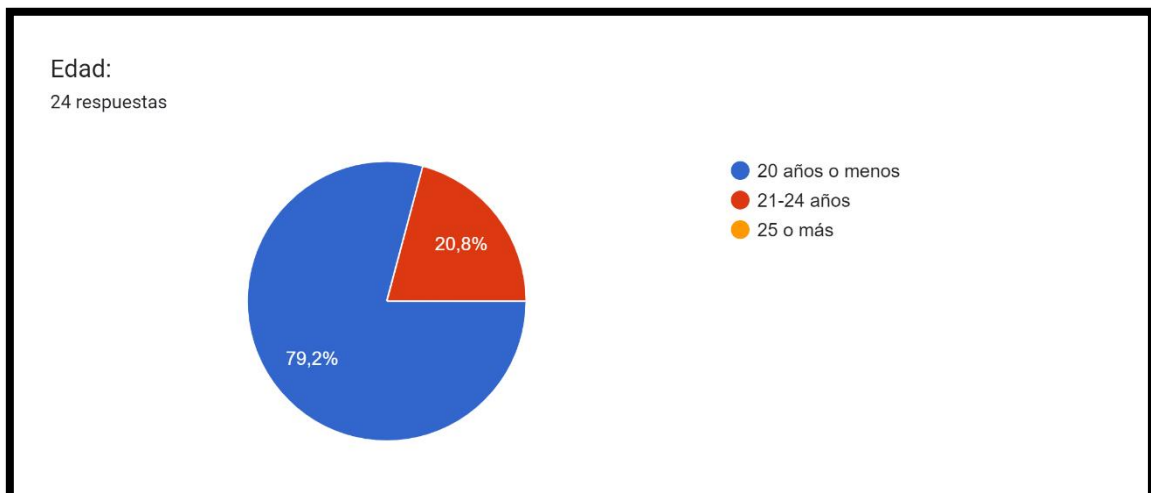


**Figura 44.** Porcentaje de hombres y mujeres.

**Fuente:** Formulario creado en Google.

En este curso se presenta un 92% de hombres que equivale a 22 hombres y un 8% que el restante son 2 mujeres.

- **Edad**

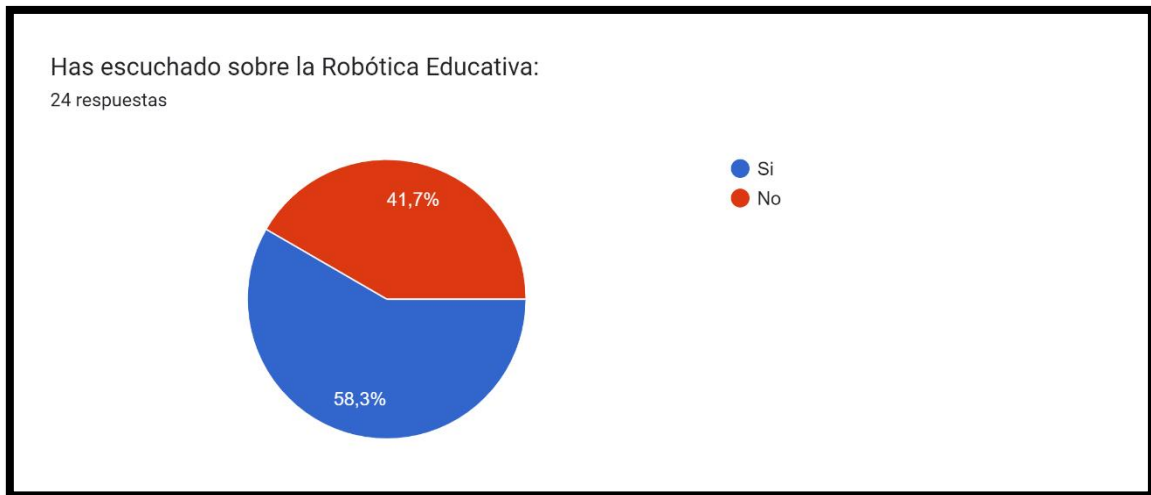


**Figura 45.** Porcentaje de edades.

**Fuente:** Formulario creado en Google.

La mayoría de los estudiantes ronda los 20 años y un poco menos, es normal al ser estudiantes de primer ciclo.

- **Ha escuchado sobre la Robótica Educativa**

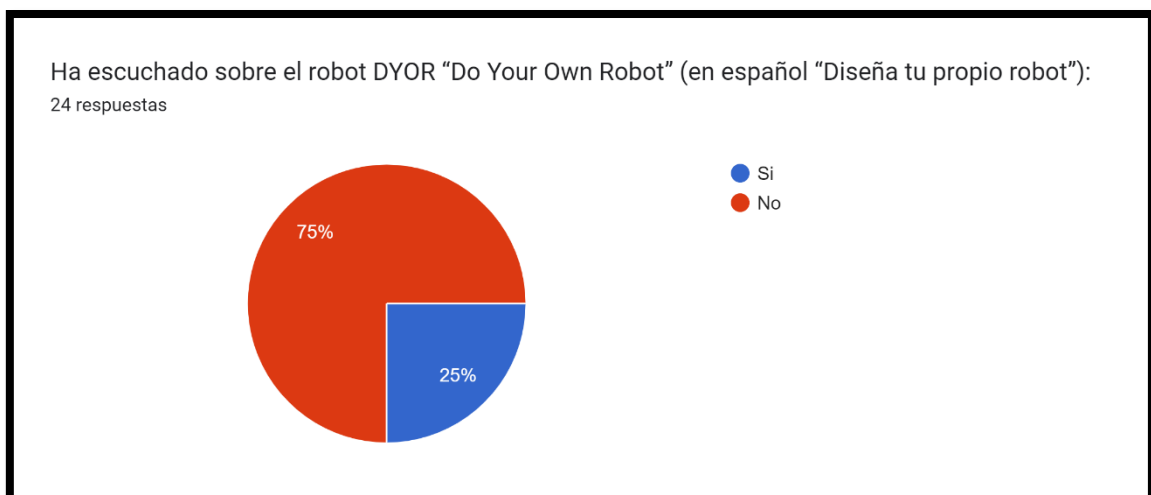


**Figura 46.** Porcentaje de conocimiento sobre la Robótica Educativa.

**Fuente:** Formulario creado en Google.

La robótica educativa está comenzando a ampliarse en el mundo, un 58.3% de los estudiantes conoce sobre este tema, además de la exposición que se dio en el curso, donde todos obtuvieron el conocimiento sobre el aprendizaje con robótica.

- **Ha escuchado sobre el robot DYOR “Do Your Own Robot” (en español “Diseña tu propio robot”)**



**Figura 47.** Porcentaje del conocimiento del Robot DYOR.

**Fuente:** Formulario creado en Google.



El conocimiento sobre el robot DYOR aún es escaso, un 75% de los estudiantes no lo conoce, y el 25% que contestó que “sí” es por el curso impartido. Al exponer el robot DYOR despertó el interés de los estudiantes y más verlo al robot construido físicamente.

Preguntas sobre el nivel de conocimiento sobre lenguaje de programación y robótica educativa.

- **¿Qué dificultad vez aprender lenguajes de programación?**

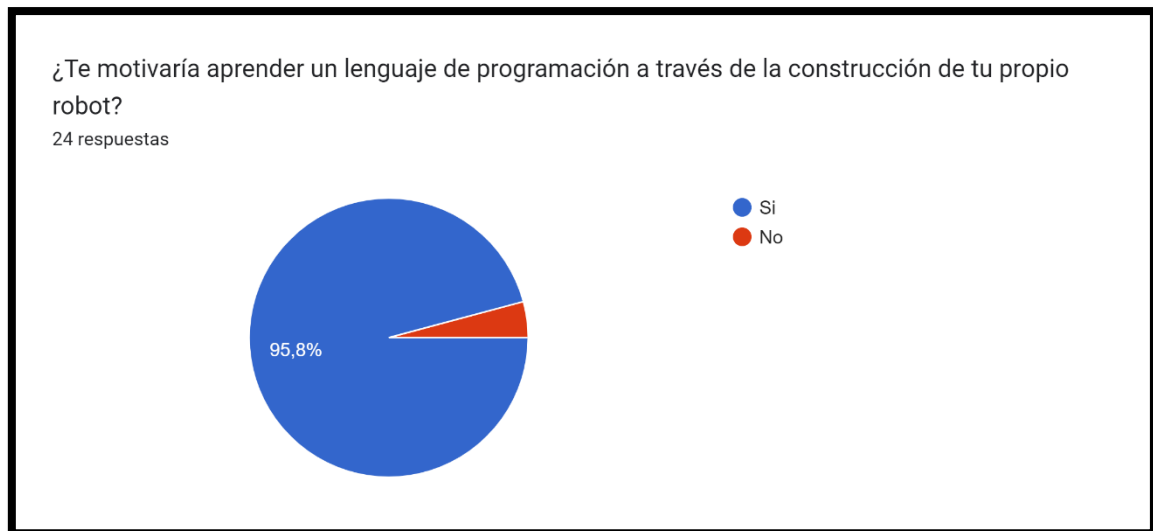


**Figura 48.** Porcentaje de dificultad de aprender programación.

**Fuente:** Formulario creado en Google.

Un 87.5% de los estudiantes contestó que aprender lenguajes de programación no es tan complicado, es decir, 21 estudiantes ven a la programación como un nivel medio para aprender, 2 que el nivel es alto y 1 que lo ve baja. Con el curso impartido ven a la programación como un nivel accesible, que a los inicios de la etapa de universidad se puede aprender programación.

- **¿Te motivaría aprender un lenguaje de programación a través de la construcción de tu propio robot?**

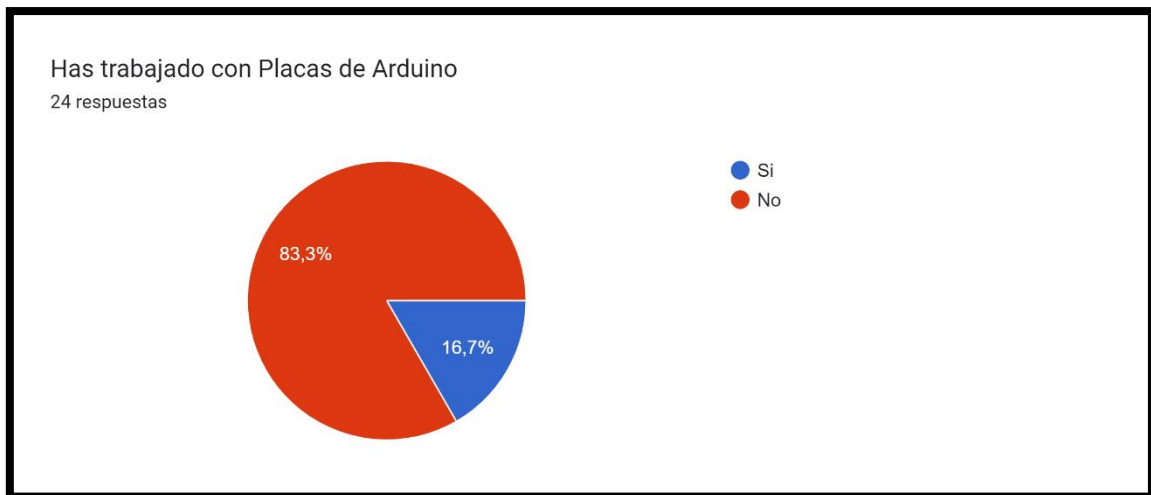


**Figura 49.** Porcentaje de motivación al aprender programación.

**Fuente:** Formulario creado en Google.

Un 95.8% se motiva a aprender un lenguaje de programación a través de los robots, con el curso impartido de programación con DYOR, al observar las programaciones que puede tener el robot 23 estudiantes les motivaría aprender el lenguaje de programación en Arduino. El interés por aprender es más susceptible cuando el estudiante ve algo que le motiva a hacerlo, al ver la programación de forma digital y física, provoca que se entusiasme más por aprender y se incline en este ámbito, al crear una programación simple de Leds aplicando los ciclo “for” y las condicionales “if”.

- **Has trabajado con Placas de Arduino**

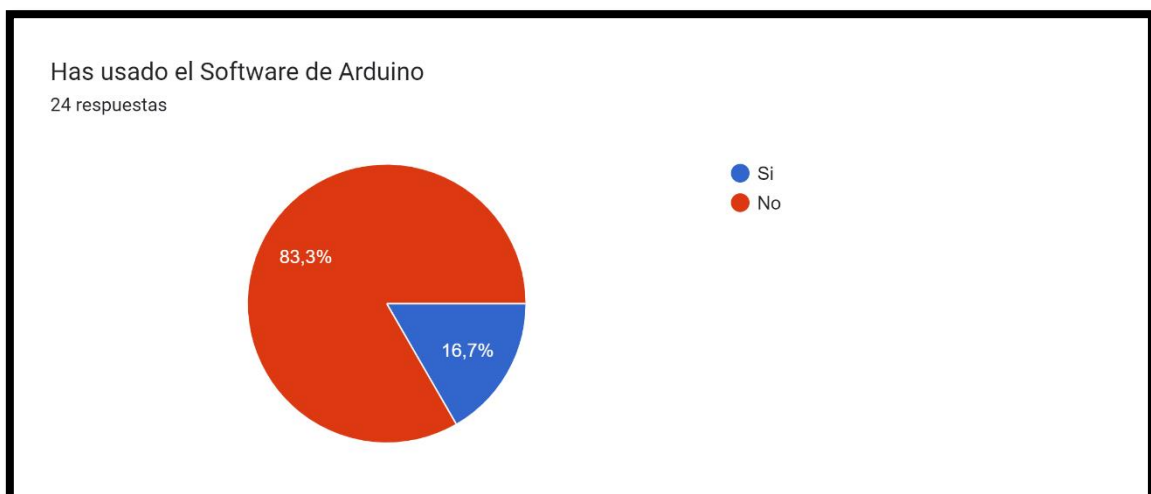


**Figura 50.** Porcentaje del conocimiento de la placa Arduino.

**Fuente:** Formulario creado en Google.

El 83.3% de los estudiantes no ha trabajado con la placa Arduino, es a primera vez que observaron cómo se trabaja con Arduino, desde la compilación de la programación hasta subirla a la placa para observar lo ejecutado en la programación. Y el 16.7% si ha trabajado con estas placas programables.

- **Has usado el Software de Arduino**



**Figura 51.** Porcentaje sobre el uso del software Arduino.

**Fuente:** Formulario creado en Google.

El 83.3% que equivale a 20 estudiantes que no han usado el software Arduino, están empezando desde 0 a aprender y de una metodología totalmente diferente donde observa la programación en funcionamiento en una placa Arduino, con ejercicios que van desde encender un led hasta programar un robot DYOR, el restante de los estudiantes ha trabajado con Arduino pero muy poco y el interés por aprender con esta metodología fue igual que el de los estudiantes que aún no han trabajado con el software Arduino.

- **Sabes lo que son bloques de programación**



**Figura 52.** Porcentaje el conocimiento de bloques de programación.

**Fuente:** Formulario creado en Google.

Los bloques de programación son las estructuras que se rige cada lenguaje de programación, en este caso, las estructuras de programación que se dio en el curso se dividen en 3: declaración de variables, void setup () y el ciclo void loop (). En el curso de programación se explicó esta estructura. En el cuestionario el 83.3% de los estudiantes no conoce los bloques de programación, en el curso fueron aprendiendo el orden del cual se rige la programación en Arduino y la estructura que debe tener para que la compilación se ejecute sin errores en ninguna línea.

- **¿Crees qué comenzar a aprender a programar sería más interesante aplicando en programar robots?**

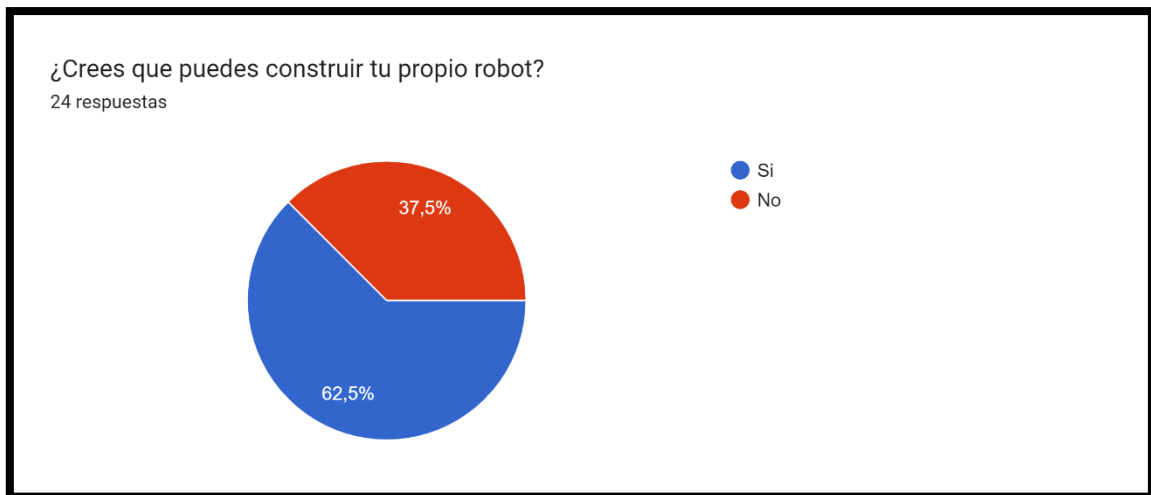


**Figura 53.** Porcentaje en el interés por aprender a programar con esta metodología.

**Fuente:** Formulario creado en Google.

El 95.8% de los estudiantes está interesado en aprender programación con robótica educativa, es decir, 23 estudiantes de los 24 les interesó esta metodología de crear las líneas de programación en Arduino y enviarlas a la placa para observar físicamente la función de su programación. La motivación es un aspecto fundamental para obtener un mayor aprendizaje en un tema determinado, en el curso los estudiantes se motivaron al ver como una simple estructura de programación puede encender un led por etapas y hasta llegar a darle movimientos a un robot.

- **¿Crees que puedes construir tu propio robot?**

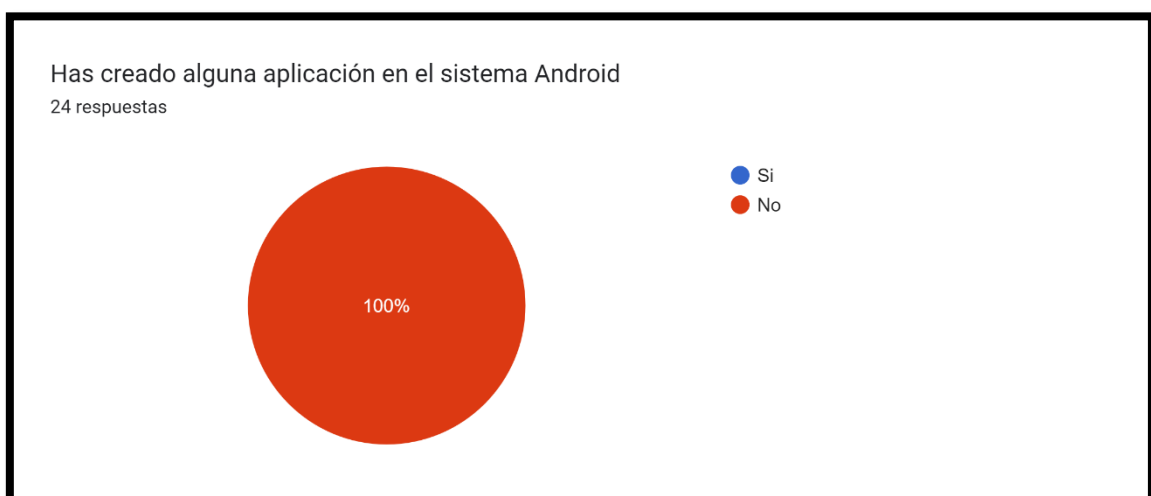


**Figura 54.** Porcentaje respecto a saber sobre la construcción de un robot.

**Fuente:** Formulario creado en Google.

Antes de iniciar con el curso, a los estudiantes se les pregunto si se creen capaces de construir un robot y todos respondieron que no, al terminar el curso se les hizo la misma pregunta al cual el 62.5% contestaron que, si están en la capacidad de construir su propio robot, que pueden darle una programación y hacer que su robot se mueva de forma automática o ser controlado.

- **Has creado alguna aplicación en el sistema Android**

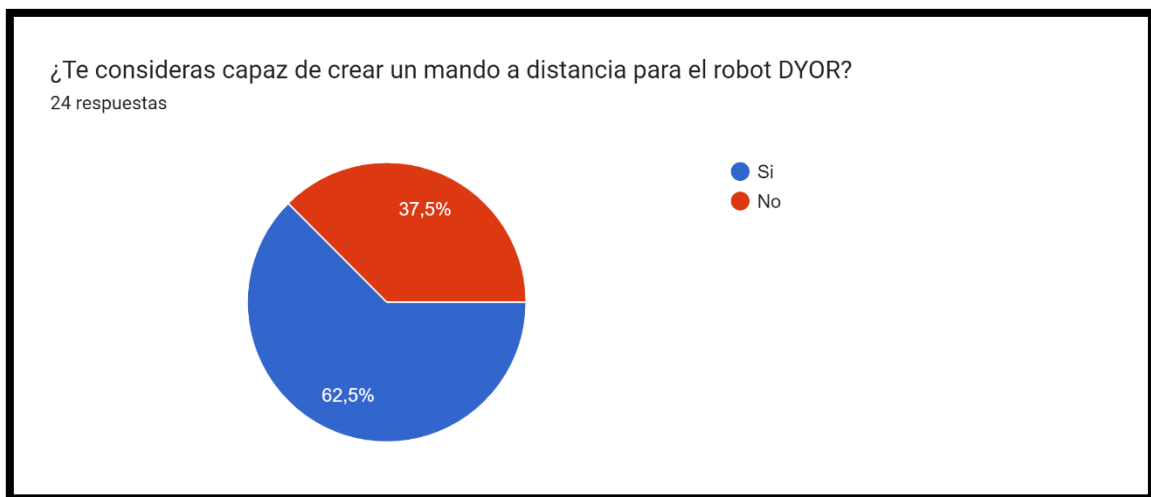


**Figura 55.** Porcentaje sobre el conocimiento de crear app en Android.

**Fuente:** Formulario creado en Google.

El asunto de crear una aplicación Android es con el fin de hacer mucho más interesante el controlar el robot con un mando a distancia que en este caso es un smartphone, todos los estudiantes respondieron que no han creado aun una aplicación para un sistema Android, es normal, el sistema educativo del país aún no explica en los colegios estos temas de tecnología, en la actualidad existe un sin número de software que nos permiten crear aplicaciones, en este caso, diseñar un mando a distancia no es tan complicado, y más si la programación para controlar el robot se rige por condicionales que puede ser un número o una letra y su comunicación es por bluetooth.

- **¿Te consideras capaz de crear un mando a distancia para el robot DYOR?**



**Figura 56.** Porcentaje de la capacidad de crear un mando a distancia.

**Fuente:** Formulario creado en Google.

Un 62.5% de los estudiantes se consideran capaz de crear un mando a distancia para el robot DYOR, al observa cómo funciona la aplicación “BlueTooth Serial Controller” y “Arduino JoysTick” que nos permiten crear una comunicación con el módulo bluetooth del robot y que su funcionamiento consiste en declarar las letras o números en los controladores que fueron programados en el robot desde Arduino.

- **¿Qué funciones has observado que realizan los robots?**

- Mayormente de sensores, realizar movimientos, seguir líneas, detectar obstáculos y buscar rutas.

- Sensores, movimiento.
- Moverse por sí solo.
- Disparar.
- Trabajar.
- Caminar, detectar obstáculos.
- Los robots pueden movilizarse mediante algún tipo de control a distancia.
- Todo tipo de funciones mecánicas.
- Todo bajo un código y la elaboración.
- Sensor de Movimiento, Sensor de Pared.
- Mando y señales.
- Todas las funciones que nosotros le mandemos a hacer.
- Realizan lo que el usuario pide.
- Movimientos, obstáculos.
- Giros, trasladarse, abrir, cerrar, etc.
- Adelante, atrás, girar, abrir o cerrar, sensor de obstáculos.
- Reciben señales.
- Se mueven por medio de algoritmos.
- Movimientos inteligentes.
- Mandos.
- Detectar obstáculos, manejar, abrir y cerrar las manos.
- Puede ser manejado de forma inalámbrica.
- Utiliza sensores.
- Moverse a través de mandos.
- Caminar en toda dirección, detectar líneas, detectar obstáculos, abrir, cerrar.
- Esquivar obstáculos.
- Que programando hacemos mover algún robot.

Al terminar el curso los estudiantes obtuvieron una visión más amplia de lo que es el lenguaje de programación, particularmente en Arduino. Al ver los movimientos del robot DYOR conocieron los movimientos que puede realizar, entre las respuestas de los estudiantes; se encuentran los sensores que permiten obtener datos cuantitativas capaces de infringir en los movimientos de un robot, detectar obstáculos, caminar en direcciones que se controlen, y un dato curioso el de moverse con algoritmos, es posible, un ejemplo



de ello son los robots de laberintos, que su programación se rige en algoritmos que le permitan llegar a desafiar el laberinto y cruzarlo.

La encuesta nos da la información de que conocimiento obtuvo el estudiante a través del curso de programación en Arduino, ellos están conociendo un lenguaje de programación distinto a este, el cual lo supieron comparar y tener una mejor idea de cuál es el que les motivan más aprender un lenguaje de programación con esta metodología. El aprendizaje que se consigue es amplio y concreto, el cerebro humano se encuentra programado para adquirir conocimiento para la supervivencia, por lo que esta potencialmente preparado para aprender, este es factible cuando el interés del estudiante es despertado por la necesidad de crear, incita a comprender mejor las cosas. Recordando que la motivación determina el nivel con que energía y en qué dirección actuamos, al aplicar la robótica educativa y la metodología de enseñanza el estudiante tendrá todos sus sentidos atentos a cada detalle al momento de aprender en la clase y al momento de practicar lo aprendido en clase.

## 7. Discusión

El desarrollo de las guías prácticas tiene como finalidad plantear un modelo conceptual de un aprendizaje que incite a los estudiantes a comprender mejor la programación, especialmente el lenguaje de Arduino.

El robot DYOR está basado en el artículo de Leopoldo Armesto de “Diseña, Fabrica y Programa Tu Propio Robot”, donde fue creado en la Universidad Politécnica de Valencia - España con el propósito de ser implementado como actividad dentro del currículum de secundaria. Este proyecto se diferencia en la programación del robot; donde se realizó toda su programación en el software de Arduino, caso contrario que en el artículo investigado se realizó en Facilino. Además del desarrollo de guías prácticas para las diferentes programaciones aplicadas al robot DYOR, donde se plantea un mecanismo de aprendizaje que sea muy efectivo para los estudiantes.

El diseño del robot DYOR propuesto se tomó como guía al artículo de Leopoldo Ernesto, el robot posee elementos de bajo costes y la construcción de una complejidad accesible para un estudiante de inicios de Universidad y que también puede ser construido por estudiantes de colegio. La base y el tronco del robot se realizó en acrílico del cual su impresión se realizó en laser, el ensamblaje del robot se realizó con silicona en los lugares previamente realizados en el diseño digital del robot. La programación del robot se ejecutó en base de la información obtenida de cada elemento del robot: sensor ultrasónico, servos motores, entre otros. Es por ello que existen 3 guías prácticas de programación que son claves para la ejecución de los movimientos del robot donde: se puede dirigir con un mando a distancia, puede detectar obstáculos a una distancia determinada y evadirlos y por último el robot sigue una trayectoria trazada por una línea negra. Todo esto con el fin de generar conocimiento a través de la experiencia y fomentar la capacidad intelectual del estudiante.

El pequeño curso impartido a los estudiantes de primer ciclo de la carrera de electricidad determinó que la metodología de aprender el lenguaje de programación que utiliza Arduino compilando y enviando a la placa para ver la ejecución del mismo, motiva al estudiante a querer aprender más sobre este tema, al ver que pueden llegar a diseñar y construir su propio robot y programarlo para darle movimientos. El trabajo tiene un gran aporte en la robótica educativa, tales como se describió en los artículos investigados que;

la robótica educativa aún carece de investigaciones y aportaciones científicas, desde justificaciones pedagógicas hasta fundamentaciones metodológicas. A través del cuestionario planteado se conoció el pensamiento y el aprendizaje que obtuvieron, también como se potenció las características sensoriales del alumno al momento de observar como una pequeña programación influye en la capacidad de movimiento de un robot.

## 8. Conclusiones

En este trabajo se presentó un mecanismo que motiva al estudiante en el ámbito de enseñar un lenguaje de programación, en donde, se despertó el interés del estudiante por aprender este tema obteniendo así un aprendizaje significativo. Utilizando la robótica educativa y el robot DYOR como una herramienta base en la metodología de aprender un lenguaje de programación en el sistema Arduino

El diseño y construcción del robot DYOR dio firmeza a la adquisición de un aprendizaje autónomo en el proceso de poner en práctica toda la capacidad del alumno al momento de manipular y explorar sus capacidades sensoriales aprendiendo a diseñar en el software SolidWorks y a todo lo que conlleva la construcción del robot DYOR, utilizando cada componente electrónico seleccionado para darle los movimientos a través de diferentes programaciones, todo esto aplicando la guía práctica referente al diseño y construir del robot

La metodología de enseñar un lenguaje de programación utilizando la placa Arduino UNO causó que los estudiantes entraran en una dinámica, en donde, se aviva el interés por conocer aspectos de la ingeniería en el ámbito de los circuitos electrónicos, tecnología e innovación, aprendiendo así, como se usa y como se programa cada componente que forma parte del robot, con las guías prácticas desarrolladas, aprendieron a aplicar estructuras de programación en el entorno de Arduino, logrando fomentar habilidades que les permiten integrar un trabajo especializado en la programación de: controlar el robot a distancia desde un smartphone, que siga una trayectoria y que evada obstáculo a una distancia pertinente.

La aplicación del cuestionario realizado en los estudiantes de primer ciclo de la carrera de Electricidad de la Universidad Nacional de Loja, afirmó el gran interés que se presenta en el estudiante al momento de aprender un lenguaje de programación utilizando la robótica educativa y el robot DYOR como una herramienta fundamental en este proceso, en el cual, un 95.8% está de acuerdo en que, aprender a programar es más interesante aplicando la programación en robots, donde compilan y comprueban si no existe errores y después envían la programación a la placa de Arduino UNO para observar la ejecución de la misma en los componentes que están conectados y que se han programado en dicha placa. Validando así las guías prácticas propuestas. Además de que

más de la mitad (alrededor de un 62%) se siente con la capacidad y la motivación de construir y programar su propio robot, provocando que el estudiante adquiriera los conocimientos necesarios para resolver problemas reales fomentando su capacidad intelectual.

## **9. Recomendaciones**

Las guías prácticas y el curso que se dio fueron impartidos a estudiantes de primer ciclo. Investigar si es factible aplicar a estudiantes que estén cursando colegio, más precisamente el bachillerato, para que desde una temprana edad ya sepan programar y hasta construir sus propios robots.

Programar en un software diferente para aprender un lenguaje de programación diferente a Arduino, un software que sea compatible con la placa Arduino, así, tener un conocimiento más amplio en el ámbito de la programación, algunos de los softwares que se puede aplicar son: MATLAB® y Facilino, este último fue utilizado en el artículo de Leopoldo Ernesto.

Aplicar una programación en la cual se obtengan datos a distancia, adquiriendo más sensores que nos demuestren datos como: velocidad, distancia recorrida, porcentaje de batería usada, hasta un sistema donde se puede visualizar el frente a través de una cámara, entre otros datos consiguiendo así en tiempo real o almacenados en una base de datos.

## 10. Bibliografía

- Aguilar, A., & Loaiza, E. (2020). *Desarrollo de un manual de guías de prácticas orientado al aprendizaje de la robótica industrial*. Cuenca.
- Amazon. (2017). *Amazon.es*. Obtenido de <https://www.amazon.es/drv8833-Puente-Driver-controlador-Arduino/dp/B077KLNNF7>
- Antolí, J., & Pellín, N. (2019). *Diseño de un Sistema de Aprendizaje Basado en Proyecto para el Máster Universitario en Automática y Robótica*. San Vicente del Raspeig: Rosabel Roig-Vila.
- Arduino. (07 de Abril de 2018). *Arduino - Libraries*. Obtenido de <https://www.arduino.cc/en/Reference/Libraries>
- Carrasquilla, G. (2014). *Guía Metodológica para la elaboración de Guías de Práctica Clínica con Evaluación Económica en el Sistema General de Seguridad Social en Salud*. Bogotá.
- Carrillo, M., & Padilla, J. (2009). *Revista de Educación. La Motivación y el Aprendizaje*, 20-33.
- Carrod. (2014). *Carrod Electrónica*. Obtenido de <https://www.carrod.mx/products/servo-motor-torque-13-kg-cm-20023>
- Córdova, L. A. (2017). *Elaboración de prácticas de aprendizaje de programación con software libre aplicado a la plataforma Raspberry Pi 3, orientado a estudiantes de bachillerato*. Guayaquil.
- García , J. (2018). *Sección Psicología educativa y del desarrollo*. Obtenido de <https://educrea.cl/los-13-tipos-de-aprendizaje-cuales-son/>
- García, J., & Castrillejo, D. (2011). *Los Robots como excusa*. Obtenido de <https://www.um.es/ead/red/46/garcia.pdf>
- Geek. (2021). *Geek Factory*. Obtenido de <https://www.geekfactory.mx/tienda/robotica/motores-actuadores-servos-y-accesorios/micro-servo-motor-de-rotacion-continua-fs90r/>

- HETPRO. (2015). *Hetpro Store*. Obtenido de <https://hetpro-store.com/TUTORIALES/bluetooth-hc-06-app-arduino/>
- Hirschtick, J. (2015). *INTRODUCCIÓN A SOLIDWORKS*. Massachusetts. Obtenido de [https://my.solidworks.com/solidworks/guide/SOLIDWORKS\\_Introduction\\_ES.pdf](https://my.solidworks.com/solidworks/guide/SOLIDWORKS_Introduction_ES.pdf)
- Lema, O. (2012). Actualización del sistema operativo, manual de operación y guía de prácticas para el sistema didáctico robótico móvil del Laboratorio de mecatrónica de la FIMCP-ESPOL. 50-62.
- Leopoldo, A. Á. (2016). *DYOR: Do Your Own Robot*. Obtenido de Aprende a diseñar, construir, programar y personalizar tu robot DYOR: <http://dyor.roboticafacil.es/>
- Llamas, L. (5 de Marzo de 2016). *INGENIERÍA, INFORMÁTICA Y DISEÑO*. Obtenido de Opciones para alimentar arduino con Baterías: <https://www.luisllamas.es/alimentar-arduino-baterias/>
- Mechatronics, N. (2017). *Naylamp Mechatronics SAC*. Obtenido de Tienda de productos ingenieril : <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- Ocampos, J. L. (2017). DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO“ROBOT SEGUIDOR DE LÍNEA VELOCISTA. *Pistas Educativas*, 208-2018.
- Ospina Rodríguez, J. (2006). Artículo de Reflexión. *La motivación, motor del aprendizaje*, 158-160.
- Plaza, M. (2019). *LABORATORIO DUAL DE ROBÓTICA EDUCATIVA*. Madrid.
- Quiroga, L. P. (2017). La robótica educativa y la educación . *Revista de educación y pensamiento*, 70-75.
- Robótica Fácil. (2020). *Tienda de róbica, Arduino y ESP*. Obtenido de <https://roboticafacil.es/prod/tcrt5000/>



- Sánchez, E. L. (Diciembre de 2012). *Diseño de un sistema de control domótico basado en la plataforma Arduino*. Obtenido de <https://riunet.upv.es/bitstream/handle/10251/18228/Memoria.pdf?sequence=1&isAllowed=y>
- SanDoRobotics. (2019). *SDR - Tecnología electrónica*. Obtenido de <https://sandorobotics.com/producto/sd-r14/>
- Solectro. (2021). *SOLECTROSHOP*. Obtenido de <https://solectroshop.com/es/modulos-arduino/1378-zumbador-pcb-activo-5v-negro.html>
- Soriano, M. (2001). La Motivación, Pilar Básico de todo tipo de esfuerzo. *Revista de relaciones Laborables*.
- TecnoTienda. (2021). *Tecnología - Electrónica - Tecniservicios*. Obtenido de <https://mitecnotienda.net/producto/max7219-interfas-p-arduino-matriz-led-8x8-hn/>
- Torrente Artero, O. (2013). *ARDUINO Curso práctico de formación*. México: Alfaomega.
- Valencia, B., Rivera, D., & Zegarra, H. (2020). Construcción de experiencias de aprendizaje en robótica educativa, para alumnos de secundaria basadas en gamificación e integración de tecnologías de la información y la comunicación. *Ingeniería, Educación y tecnología*.

## 11. Anexos

- **Anexo 1.** Salón de clases al cual se dio el curso de Programación.



**Figura 57.** Estudiantes impartidos el curso.

**Fuente:** El Autor.

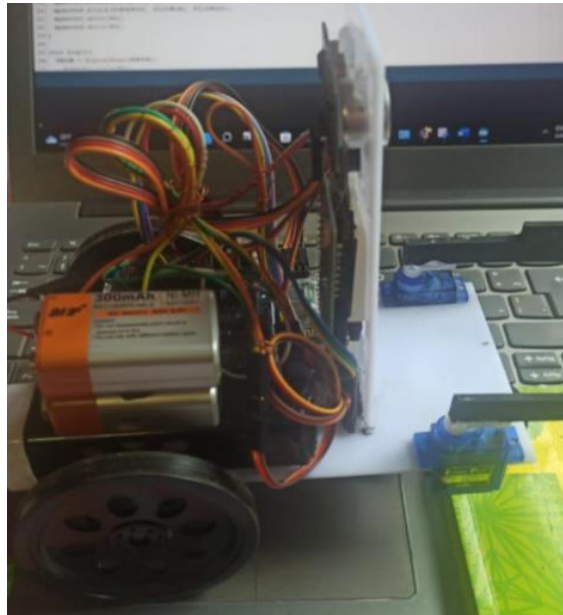
- **Anexo 2.** Exponiendo sobre el lenguaje de Arduino.



**Figura 58.** Explicando sobre el robot DYOR y el lenguaje de programación de Arduino.

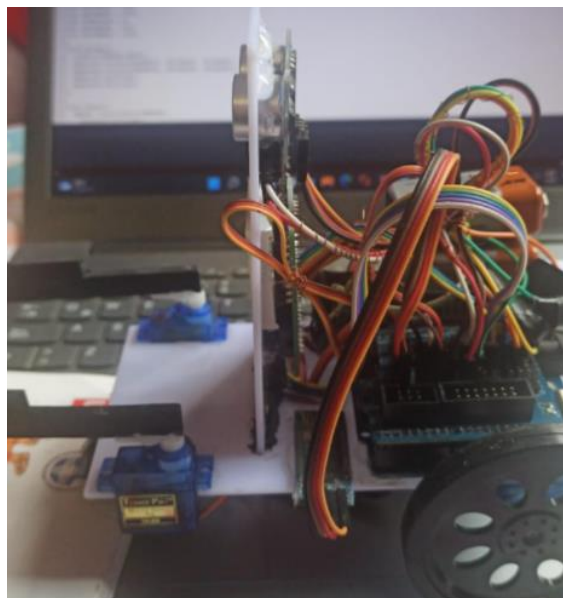
**Fuente:** El Autor.

- **Anexo 3.** Vistas del Robot DYOR.



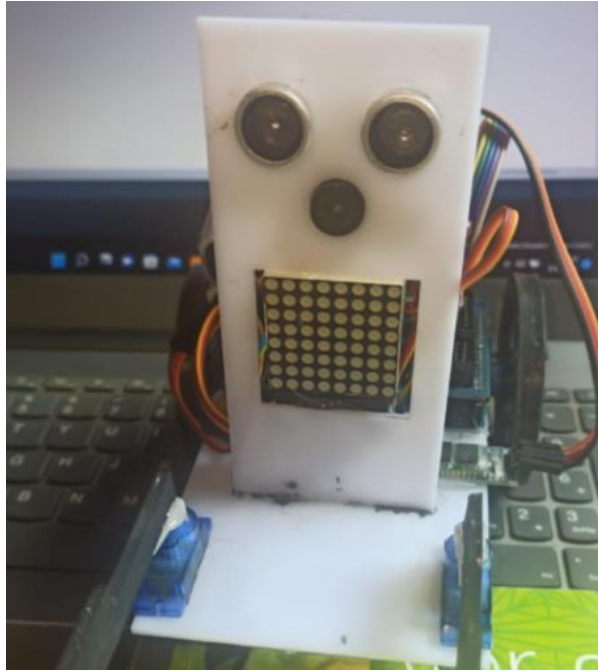
**Figura 59.** Robot DYOR – Vista Lateral izquierda.

**Fuente:** El Autor.



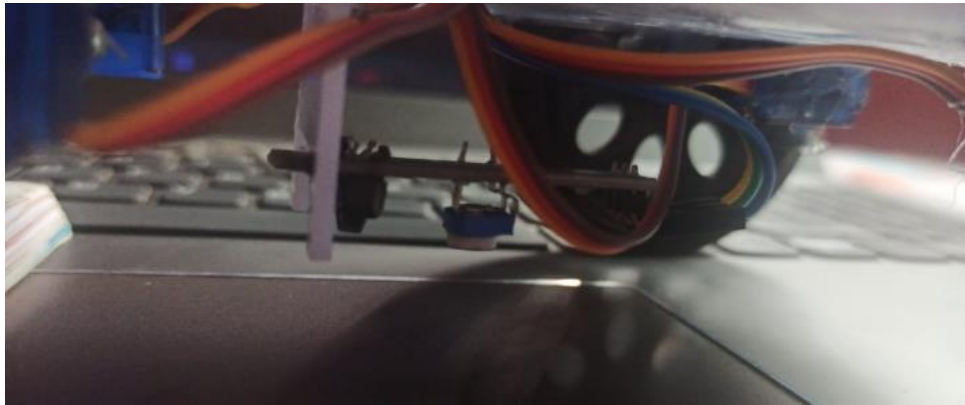
**Figura 60.** Robot DYOR – Vista Lateral Derecha.

**Fuente:** El Autor.



**Figura 61.** Robot DYOR – Vista Frontal.

**Fuente:** El Autor.



**Figura 62.** Robot DYOR – Sensor Siguelínea.

**Fuente:** El Autor.

- **Anexo 4.** DYOR – Detección de Obstáculos.

```
#include <Servo.h>

#include <SoftwareSerial.h>

#define Pecho 7

#define Ptrig 13

long duración, distancia;

// crea el objeto servo

Servo myservo2;

Servo myservo1;

int PINSERVO1 = 5;

int PULSOMIN1 = 520;

int PULSOMAX1 = 1500;

int PINSERVO2 = 9;

int PULSOMIN2 = 520;

int PULSOMAX2 = 1500;

void setup () {

    Serial.begin (9600);    // inicializa el puerto serial a 9600 baudios

    pinMode(Pecho, INPUT); // define el pin 6 como entrada (echo)

    pinMode(Ptrig, OUTPUT); // define el pin 7 como salida (trigger)
```

```

myservo1.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);

myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);

myservo1.write(90);

myservo2.write(90);

}

void loop () {

  digitalWrite (Ptrig, LOW);

  delayMicroseconds(2);

  digitalWrite (Ptrig, HIGH); // genera el pulso de trigger por 10ms

  delayMicroseconds(10);

  digitalWrite(Ptrig, LOW);

  duración = pulseIn (Pecho, HIGH);

  distancia = (duración/2) / 29; // calcula la distancia en centímetros

  if (distancia >= 500 || distancia <= 0){ // si la distancia es mayor a 500cm o
menor a 0cm

  }

  else {

    myservo1.write(60);

```

```
myservo2.write(180);
```

```
}
```

```
if (distancia <= 10 && distancia >= 1){
```

```
myservo1.write(90);
```

```
myservo2.write(90);
```

```
delay(1500);
```

```
myservo1.write(120);
```

```
myservo2.write(0);
```

```
delay(1500);
```

```
myservo1.write(90);
```

```
myservo2.write(90);
```

```
delay(1500);
```

```
myservo1.write(60);
```

```
myservo2.write(90);
```

```
delay(2000);
```

```
Serial.println("PARED....."); // envía la palabra Alarma por el puerto
```

```
serial
```

```
}
```

```
        delay(400);           // espera 400ms para que se logre ver la distancia
en la consola
    }
```



- **Anexo 5.** DYOR – Detección de Obstáculos con Alarma.

```
#include <EasyBuzzer.h>

#include <Servo.h>

#include <SoftwareSerial.h>

#define Pecho 7

#define Ptrig 13

long duracion, distancia;

Servo myservo2; // crea el objeto servo

Servo myservo1;

int PINSERVO1 = 5;

int PULSOMIN1 = 520;

int PULSOMAX1 = 1500;

int PINSERVO2 = 9;

int PULSOMIN2 = 520;

int PULSOMAX2 = 1500;

void setup() {

    Serial.begin (9600);    // inicializa el puerto seria a 9600 baudios

    pinMode(Pecho, INPUT); // define el pin 6 como entrada (echo)
```

```

pinMode(Ptrig, OUTPUT); // define el pin 7 como salida (triger)

myservo1.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);

myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);

myservo1.write(90);

myservo2.write(90);

}

void loop() {

digitalWrite(Ptrig, LOW);

delayMicroseconds(2);

digitalWrite(Ptrig, HIGH); // genera el pulso de triger por 10ms

delayMicroseconds(10);

digitalWrite(Ptrig, LOW);

duracion = pulseIn(Pecho, HIGH);

distancia = (duracion/2) / 29; // calcula la distancia en centimetros

if (distancia >= 500 || distancia <= 0){ // si la distancia es mayor a 500cm o
menor a 0cm

Serial.println("---"); // no mide nada

}

```

```
else {  
  
    Serial.print(distancia);    // envia el valor de la distancia por el puerto serial  
  
    Serial.println("cm");      // le coloca a la distancia los centímetros "cm"  
  
    EasyBuzzer.update();  
  
    myservo1.write(60);  
  
    myservo2.write(180);  
  
}
```

```
if (distancia <= 10 && distancia >= 1){
```

```
    myservo1.write(90);  
  
    myservo2.write(90);  
  
    delay(1500);  
  
    myservo1.write(120);  
  
    myservo2.write(0);  
  
    delay(1500);  
  
    myservo1.write(90);  
  
    myservo2.write(90);  
  
    delay(1500);  
  
    myservo1.write(60);  
  
    myservo2.write(90);
```

```

delay(2000);

Serial.println("PARED.....");    // envia la palabra Alarma por el puerto
serial

EasyBuzzer.beep(

2000,    // Frecuencia en hercios

100,    // Duración beep en ms

100,    // Duración silencio en ms

2,    // Números de beeps por ciclos

300,    // Duración de la pausa

1    // Número de ciclos; // zumbador low apagado    // suena la alarma

);

}

delay(400);    // espera 400ms para que se logre ver la distancia
en la consola

}

```

- **Anexo 6.** DYOR – Controles básicos (derecha – izquierda – Avanzar – Retroceder) a robot a distancia.

```
#include <Servo.h>
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial mybluetooth(2,4); //RX y TX
```

```
Servo myservo2; // crea el objeto servo
```

```
Servo myservo1;
```

```
Servo myservo3;
```

```
Servo myservo4;
```

```
int PINSERVO1 = 5;
```

```
int PULSOMIN1 = 520;
```

```
int PULSOMAX1 = 1500;
```

```
int PINSERVO2 = 9;
```

```
int PULSOMIN2 = 520;
```

```
int PULSOMAX2 = 1500;
```

```
int PINSERVO3 = 3;
```

```
int PULSOMIN3 = 520;
```

```
int PULSOMAX3 = 1500;
```

```
int PINSERVO4 = 6;
```

```

int PULSOMIN4 = 520;

int PULSOMAX4 = 1500;

void setup(){

  mybluetooth.begin(9600);

  myservo1.attach(PINSERVO1, PULSOMIN1, PULSOMAX1);

  myservo2.attach(PINSERVO2, PULSOMIN2, PULSOMAX2);

  myservo3.attach(PINSERVO3, PULSOMIN3, PULSOMAX3);

  myservo4.attach(PINSERVO4, PULSOMIN4, PULSOMAX4);

  myservo1.write(90);

  myservo2.write(90);

  myservo3.write(90);

  myservo4.write(90);

}

void loop(){

  if (mybluetooth.available()){

    char letra=mybluetooth.read();

    if(letra=='W'){ //hacia delante

      myservo1.write(0);

```

```
myservo2.write(180);

}

if(letra=='F'){ // parada

    myservo1.write(90);

    myservo2.write(90);

}

if(letra=='S'){ // hacia atras

    myservo1.write(180);

    myservo2.write(0);

}

if(letra=='A'){ // GIRO A LA DRECHA

    myservo1.write(0);

    myservo2.write(90);

}

if(letra=='D'){ // GIRO A LA IZQUIERDA

    myservo1.write(90);

    myservo2.write(180);

}

if(letra=='T'){ // MANOS RECTAS
```

```
myservo3.write(90);

myservo4.write(90);

}

if(letra=='Y'){ // MANOS ABIERTAS

myservo3.write(70);

myservo4.write(70);

}

if(letra=='M'){ // CERRADAS

myservo3.write(180);

myservo4.write(180);

}

}

}
```



- **Anexo 7.** Certificación de traducción del resumen.



*Mg. Yanina Quizhpe Espinoza*  
*Licenciada en Ciencias de Educación mención*  
*Inglés*  
*Magister en Traducción y mediación cultural*

Celular: +593989805087  
Email: [yaniges@icloud.com](mailto:yaniges@icloud.com)  
Loja, Ecuador 110104

Loja, 19 de agosto de 2022

Yo, Lic. Yanina Quizhpe Espinoza, con cédula de identidad 1104337553, docente del Instituto de Idiomas de la Universidad Nacional de Loja, y con licencia como traductora registrada en el Ministerio de trabajo del Ecuador **MDT-3104-CCL-252640**, certifico:

Que tengo el conocimiento y dominio de los idiomas español e inglés y que la traducción del resumen de trabajo de integración curricular, **DISEÑO, CONSTRUCCIÓN Y DESARROLLO DE GUÍAS PRÁCTICAS CON DYOR**, cuya autoría del estudiante Jorge Miguel Chávez Retete, con cédula 0705480192, es verdadero y correcto a mi mejor saber y entender.

Atentamente

YANINA  
BELEN  
QUIZHPE  
ESPINOZA  
Firmado digitalmente por  
YANINA BELEN  
QUIZHPE  
ESPINOZA  
Fecha: 2022.08.19  
14:01:45 -05'00'

Yanina Quizhpe Espinoza.

**Traductora**