



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE
MÉXICO**

CENTRO UNIVERSITARIO UAEM ECATEPEC

**“DISEÑO Y DESARROLLO DE UN ROBOT MÓVIL A
BAJO COSTO PARA NIÑOS: ECATEBOT”**

TESIS

**PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTA EL C.

LICONA ALVITER JUAN ALBERTO

ASESOR: DR. EN C. RODOLFO ZOLÁ GARCÍA LOZANO

REVISORAS: M. EN I.S.C. ALEJANDRA MORALES RAMÍREZ

M. EN A. LAURA EDITH ALVITER ROJAS

ECATEPEC DE MORELOS, ESTADO DE MÉXICO, NOVIEMBRE 2019.

Página en blanco

Página en blanco

Página en blanco

Página en blanco

Página en blanco

ÍNDICE

INTRODUCCIÓN.	10
CAPÍTULO I. MARCO TEÓRICO.	12
1.1 Robots móviles y robótica.	12
1.1.1 Definición de robótica.....	12
1.1.2 Definición de robot.....	14
1.1.3 Definición de robot móvil.	15
1.2 Antecedentes de los robots móviles	16
1.3 Características de los robots móviles.	17
1.3.1 Sistemas de locomoción de un robot móvil.	18
1.3.2 Robots móviles con ruedas.	18
1.4 Motores de corriente continua.	23
1.4.1 Motor reductor.	23
1.4.2 Motor reductor 1:120.	24
1.5 Puente h.	25
1.5.1 Puentes en H integrados. L293 y L293D.	26
1.5.2 Diagrama lógico puente h L293NE.	27
1.6 Módulo buzzer.	28
1.7 Módulo LED RGB.	29
1.7.1 LED RGB cátodo común.	29
1.8 Comunicación inalámbrica bluetooth.	30
1.8.1 HC serial bluetooth.	33
1.9 Arduino.	35
1.9.1 Arduino nano.	37
1.10 MIT App Inventor 2	40
CAPÍTULO 2. METODOLOGÍA DE DESARROLLO.	42
2.1 Metodología para el desarrollo de proyectos de electrónica.	42
2.2 Etapa de análisis.	45
2.3 Fases de desarrollo.	48
2.4 Conclusiones.	49

CAPÍTULO 3. CONSTRUCCIÓN DEL ROBOT.	50
3.1 Análisis de requerimientos del EcateBot.	50
3.3 Diseño modular.	53
3.3.1 Diagramas del robot móvil.	54
3.3.2 Codificación de los módulos.	57
3.4 Desarrollo de proyecto.	60
Iteración 1. Movimiento.	61
Iteración 2. Controlar el movimiento automático.	62
Iteración 3. Ensamblaje y programación de módulos Buzzer y LED RGB.	64
Iteración 4. Diseño y desarrollo de aplicación para móviles con plataforma App Inventor 2.	66
Iteración 5. Configuración de la comunicación entre el dispositivo móvil y la tarjeta Arduino mediante el módulo Bluetooth/WIFI.	68
Iteración 6. Interfaz de la aplicación.	71
Análisis de costos.	71
 CONCLUSIONES	 75
 ANEXOS.	 76
Anexo 1. Manual de ensamblaje módulo de movimiento.	76
Anexo 2. Módulo de movimiento automático	79
Anexo 3. Módulo de control inalámbrico.	83
Anexo 4. Desarrollo aplicación móvil.	85
Anexo 5. Consumo de energía EcateBot.	88
Anexo 6. Esquemáticos EcateBot.	88
Anexo 7. Diseño de placas modulares y control EcateBot.	89
Anexo 8. Códigos Arduino.	90
 REFERENCIAS.	 97

ÍNDICE DE FIGURAS Y TABLAS

ÍNDICE DE FIGURAS

Figura 1. Tipos de locomoción	18
Figura 2. Configuración de los robots móviles.	19
Figura 3. Robot Terregator	21
Figura 4. Esquema motor reductor.	25
Figura 5. Diagrama lógico puente h.	27
Figura 6. Diagrama lógico puente h.	27
Figura 7. Esquema LED RGB.	30
Figura 8. Diagrama Arduino nano.	37
Figura 9. Proceso metodológico para el desarrollo de proyectos de electrónica	43
Figura 10. Esquema planta baja del robot.	54
Figura 11. Diagrama EcateBot primer nivel.	55
Figura 12. Interfaz gráfica de la aplicación móvil EcateBot.	56

ÍNDICE DE TABLAS

Tabla 1. Evolución de los robots móviles a través de la historia.	17
Tabla 2. Dimensiones puente h	26
Tabla 3. Funcionamiento de los pines puente H.	28
Tabla 4. Clasificación y capacidad de las clases Bluetooth.	33
Tabla 5. Diferencia entre módulo HC-05 y HC-06.	34
Tabla 6. Especificaciones técnicas del Arduino Nano.	38
Tabla 7. Requerimientos usuario-desarrollador.	52
Tabla 8. Descripción modular del EcateBot.	53
Tabla 9. Costo del EcateBot.	72

Introducción.

En el ámbito educativo, se pueden presentar diversos problemas que obstaculizan el complejo camino de la enseñanza [1], debido a que cada estudiante posee características particulares, como los rasgos afectivos, fisiológicos y cognitivos, que sirven como indicadores relativamente estables, de cómo éstos perciben, interaccionan y responden a sus ambientes de aprendizaje [2]. Además, como consecuencia a la hora de aprender, lo hacen de manera distinta y organizan ese proceso de acuerdo con la forma de cómo asimilan los contenidos que estudian [3].

Hoy en día el uso de la tecnología es un motivador para los estudiantes y brinda encuentros de aprendizaje más activos, esto debido a que puede ofrecer simulaciones o modelados que animen y provoquen respuestas más activas y relacionadas con el aprendizaje por exploración por parte de los estudiantes. Además, dicha tecnología crea situaciones de aprendizaje que estimula a los estudiantes a desafiar su propio conocimiento y construir nuevos marcos conceptuales [4].

Por tanto, los ambientes para el aprendizaje desarrollados a partir de la robótica educativa, surge como nueva propuesta didáctica que responde a los requerimientos de sociedades contemporáneas que reclaman nuevos métodos de enseñanza. Sin embargo, los robots móviles de la actualidad que son implementados en el ámbito educativo como una herramienta que induce al aprendizaje, tienen un alto costo en el mercado actual y no son de fácil alcance para los interesados en utilizarlos. Algunos ejemplos de estos robots enfocados a la educación son: el Finch robot, mBot, Thymio y Edison estos robots móviles cuentan con sensores de temperatura, sensores para la evasión de obstáculos, programación con base en bloques, comunicación inalámbrica, entre otras funciones de control [4]. Su finalidad es inspirar a los

estudiantes aprender ciencia computacional, electrónica y robótica, proporcionándoles una herramienta tangible para representar el conocimiento.

Es así que el objetivo de este trabajo es diseñar y fabricar un robot móvil de aplicación educativa para niños a un bajo costo.

En suma, la importancia de la construcción del EcateBot gira en torno a dos ejes principales:

El primero, es que el EcateBot explotará el deseo de los educandos por interactuar de manera muy fácil y sencilla con un robot. Este robot móvil presentará una gran ventaja para su construcción, la cual es su bajo costo de fabricación, en contraste con los que se encuentran en el mercado, los cuales su desarrollo oscila entre los \$2000 a \$3000 pesos [4]; no obstante, el EcateBot ofrece funciones como control de luz, sonido y movimiento inalámbrico comandado por una aplicación móvil.

Y en segunda instancia se desarrollaron instructivos de ensamblaje para que los estudiantes puedan elaborar el EcateBot, y a la par apliquen conceptos de las áreas de física, matemáticas y programación, promoviendo así sus habilidades y competencias.

Por tanto, el desarrollo del EcateBot busca fomentar y despertar la curiosidad, estimular la motivación, promover la creatividad; incentivar el pensamiento abstracto y computacional; y desarrollar la autonomía del estudiante.

El capítulo uno habla de la composición de los robots móviles y el EcateBot. En el capítulo dos se menciona como se desarrollara nuestro robot móvil, a su vez en el capítulo 3 se trabaja en la construcción del robot.

CAPÍTULO I

Marco teórico.

1.1 Robots móviles y robótica.

Tradicionalmente las aplicaciones de la robótica estaban centradas en los sectores manufactureros enfocados a la producción masiva: industria del automóvil, transformaciones metálicas, industria química, etc. [8]. Sin embargo, hoy en día las aplicaciones de los robots móviles son muy amplias como, por ejemplo: exploración minera, exploración planetaria, misiones de búsqueda y rescate de personas, limpieza de desechos peligrosos, vigilancia, reconocimiento de terreno, así mismo con asistencia médica, exploración marítima, investigación y desarrollo, investigación militar, agricultura, inspección, transporte, educación, etc. [9].

Para entender que es un robot móvil es necesario conocer su funcionamiento y como están contruidos. Dichos temas se describen a continuación.

1.1.1 Definición de robótica.

Para [4] la robótica es el conjunto de conocimientos teóricos y prácticos que permiten concebir, realizar y automatizar sistemas basados en estructuras metálicas con múltiples articulaciones, dotadas de un determinado grado de “inteligencia” y designados principalmente a la sustitución del ser humano en diversas labores, o a la producción industrial.

Por otro lado, Sosa [5] menciona que es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia.

A su vez Riccillo [6] indica que la robótica es un área interdisciplinaria, bastante divertida, que tiene un montón de posibilidades. Básicamente, es el desarrollo de entes que poseen cierta autonomía.

Oxford [7] define la robótica como una técnica que se utiliza en el diseño y la construcción de robots y aparatos que realizan operaciones o trabajos, generalmente en instalaciones industriales y en sustitución de la mano de obra humana.

Se considera a la robótica como una técnica que aplica informática al diseño y empleo de aparatos que, en sustitución de personas, realizan operaciones o trabajos, por lo general en instalaciones industriales [8].

Por último, Barrientos [9] dice que la robótica es la técnica utilizada para el diseño y construcción de los robots.

En suma y con base en lo anterior, se define a la robótica como un conjunto de conocimientos teóricos y prácticos, multidisciplinarios, los cuales son aplicados para el desarrollo de máquinas con cierto grado de inteligencia artificial y autonomía, capaces de desempeñar tareas mecánicas o repetitivas realizadas por el ser humano en ambientes industriales o cotidianos. Un aspecto importante es que las posibilidades de desarrollo que ofrece la robótica hacen que sea de gran interés para personas de todas las edades.

1.1.2 Definición de robot.

Siguiendo un proceso similar al que se siguió en la definición de la robótica, en esta sección se presentan inicialmente diferentes definiciones de robot, para posteriormente construir una definición propia a partir de estos aspectos.

La Asociación Japonesa de Robótica Industrial como se citó en Lemus [10] antepone la definición de robot como dispositivos capaces de moverse de modo flexible y análogo al que poseen los organismos vivos, con o sin funciones intelectuales, permitiendo operaciones en respuesta a las órdenes humanas.

Robot Industries Association (RIA, 2005) como se cita en Lemus [10] comenta que un robot es un manipulador funcional reprogramable, capaz de mover material, piezas, herramientas o dispositivos especializados mediante movimientos variables programados, con el fin de realizar tareas diversas.

Barrientos [9] menciona que un robot puede denotarse como un sistema electromecánico reprogramable, que permite realizar diferentes tareas representativas que requieren un grado elevado de precisión.

Finalmente, Flores y Huerta [11] consideran al robot como un artefacto de forma mecánica o virtual, que por lo general es un sistema electromecánico que tiene un propósito específico. Clasificándolos según sus capacidades en: Androides, Móviles, Zoomórficos, Poli articulados.

En [12] se dice que un robot es una máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.

Por lo que se concluye, que un robot se define como un sistema con componentes electrónicos y mecánicos el cuál es reprogramable, capaz de moverse de forma flexible en el entorno que este especializado, con el fin de realizar tareas diversas que requieren un grado precisión.

1.1.3 Definición de robot móvil.

En el presente trabajo se desarrolla un tipo de robot particular denominado robot móvil. Por esta razón, al final de esta sección, se presenta la definición de robot móvil construida a partir de las definiciones reportadas en diversas fuentes bibliográficas que se listan a continuación:

El Instituto de Robótica de América (1979) como se citó en Flores y Huerta [11] definió el robot móvil como un manipulador reprogramable y multifuncional diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de varios movimientos programados para el desarrollo de una variedad de tareas.

La definición de robot móvil parte de la teoría de la robótica clásica descrita por [13] mencionando que un robot móvil es un vehículo de propulsión autónoma y locomoción (re) programada bajo control automático enfocado a desarrollar una cierta tarea.

Así mismo Barrientos [9] define un robot móvil como un sistema electromecánico capaz de desplazarse autónomamente sin estar sujeto a un punto en específico. Este posee sensores los cuales monitorean su posición, su desplazamiento está dado por dispositivos de locomoción, tales como motores, ruedas, patas, etc.

Por su parte Bambino [8] menciona que los robots móviles son dispositivos de transporte automático, es decir, una plataforma mecánica dotada de un sistema de locomoción capaz de navegar a través de un determinado ambiente de trabajo, el cual contiene un cierto nivel de autonomía para su desplazamiento portando cargas. De manera que un robot móvil es una máquina automática que es capaz de tener movimiento en un medio ambiente determinado.

En suma, se concluye que un robot móvil, es un dispositivo electromecánico, autónomo y reprogramable dotado de un sistema de locomoción que le permite navegar a través de un determinado ambiente de trabajo para desarrollar actividades específicas.

1.2 Antecedentes de los robots móviles

La diferencia entre los robots móviles e industriales es que los últimos están ensamblados en una máquina fija. Por lo general tiene uno o dos brazos articulados que son lo único que se mueve; por otro lado, los robots móviles son un sistema electromecánico que opera de manera automática y que es capaz de moverse en una zona determinada. Los medios en los cuales estos dispositivos se pueden mover son muy variados, desde superficies comunes de hogar y oficina; hasta por los aires. Por tanto, en la tabla 1 se puede apreciar la evolución de los robots móviles a través de la historia.

Tabla 1. Evolución de los robots móviles a través de la historia.

Nombre	Año	Autor	Características
Shakey	1970	Nils Nilsson	El primer robot con propósito general con cámara, sensores y radio localización. Siendo el primer mecanismo en utilizar la inteligencia artificial (IA).
Newt	1971	Ralph Hollis	Robot móvil cognitivo controlado por una microcomputadora que contiene sensores de proximidad, imagen y un manipulador.
Lunar Rover Vehicle	1971	NASA	Diseñado para operar en la gravedad de la luna. Era alimentado de una batería que accionaba un motor de 200W, vida útil de 180 km.
Stanford Cart	1979	Hans Moravec	Diseñado para poner moverse en un entorno de manera autónoma, su navegación podía detectar objetos en 3D utilizando una cámara deslizante para la visión estereoscópica.
Dante II	1994	El Instituto de Robótica CMU	Sistema de seis patas cuyo propósito fue tomar muestras de gases en el volcán Spurr, situado en Alaska.
Sojourner Rover	1997	NASA	Dedicado a enviar fotografías del entorno, muestras químicas de las rocas y aceite, así como del clima del planeta Marte.

Fuente: Bambino [8].

1.3 Características de los robots móviles.

Los robots móviles pueden ser diseñados y configurados de diferentes maneras dependiendo el ámbito y función que se pretenda alcanzar, como lo puede ser en ambientes de investigación, en labores de rescate o en lugares en los cuales el ser humano no puede alcanzar o es de alto riesgo. Las características de estos pueden ser muy variadas, a continuación se presentan

algunas de ellas, las cuales son funcionales para la elaboración de este proyecto.

1.3.1 Sistemas de locomoción de un robot móvil.

La locomoción del robot es una de las principales características que está condicionada por su entorno, siendo el estudio de los mecanismos de control para permitirles moverse de forma fluida y eficiente.

Siendo de tal manera que los robots móviles se pueden clasificar dependiendo su tipo de locomoción, es decir, la capacidad de los robots para desplazarse de un lugar a otro; en general, los tres medios de movimiento son: por ruedas, por patas y por orugas (Figura 1).

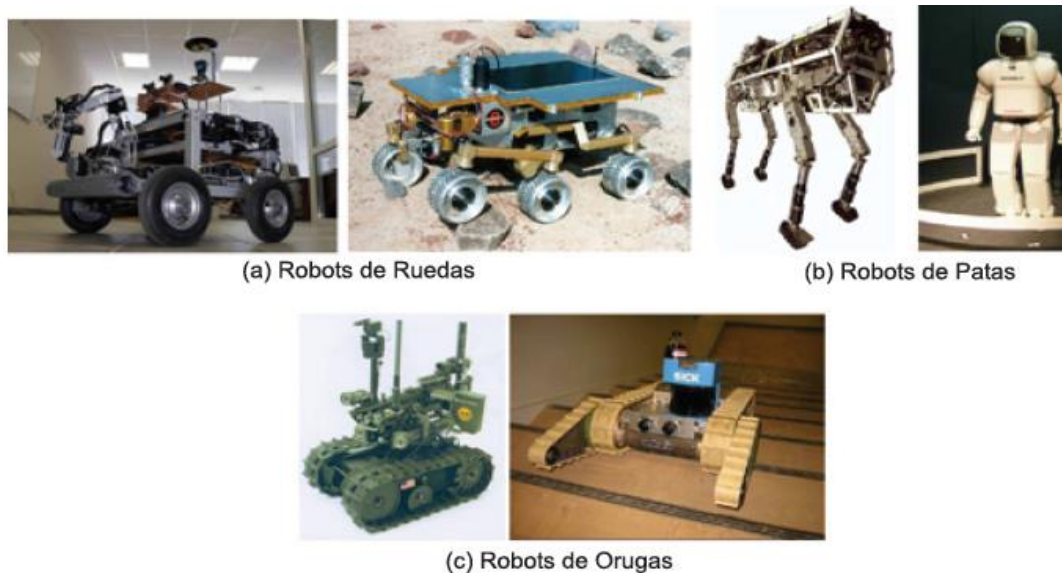


Figura 1. Tipos de locomoción

1.3.2 Robots móviles con ruedas.

Los robots móviles con ruedas (RMR) es la rama con mayor desarrollo, puesto que las ruedas ofrecen mayores ventajas en comparación al de patas y orugas, gracias a los siguientes atributos: eficiencia en el ahorro de energía en

superficies lisas y firmes, no hay desgaste en el espacio en que ruedan, además de no requerir muchas piezas para su ensamblaje, lo que facilita en gran manera su construcción.

Sin embargo, también cuenta con desventajas en configuraciones muy especiales, esto es porque no es posible alterar el margen de estabilidad para adaptarse al estado del terreno, lo que confina de forma importante los caminos aceptables en los que puede deslizarse [14].

Dentro de los RMR, Ollero [15] comenta que existen distintas configuraciones las cuales nos dan características específicas de funcionamiento respecto a las necesidades del usuario dentro de las cuales las más generales son la configuración "Ackerman o tipo coche", "Triciclo clásico", "Skid Steer", "Pistas de deslizamiento", "Síncronas" y la última "Tracción diferencial" (Figura 2). Todas están se refieren a la forma en la cual se colocan las ruedas, número de ruedas del robot móvil, tipo de tracción, etcétera.

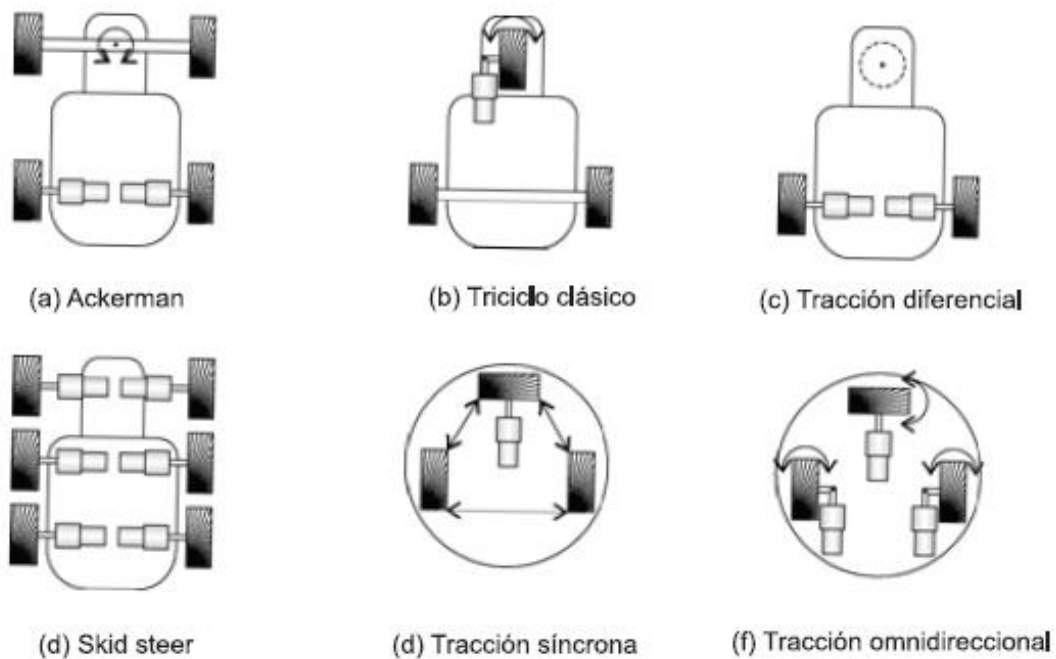


Figura 2. Configuración de los robots móviles.

Ackerman o tipo coche.

Es el utilizado en vehículos de cuatro ruedas convencionales. Los vehículos robóticos para exteriores resultan normalmente de la modificación de automóviles. En este sistema de locomoción, las ruedas delanteras giran un ángulo que permite eliminar el deslizamiento y controlar así la dirección de movimiento.

El sistema se basa en dos ruedas traseras tractoras que se montan de forma paralela en el chasis principal del vehículo, mientras que las ruedas delanteras son para el direccionamiento, y se utilizan para seguir la trayectoria del robot [14].

Triciclo clásico.

En esta configuración la rueda delantera sirve tanto para la tracción como para el direccionamiento. El eje trasero, con dos ruedas laterales, es pasivo y sus ruedas se mueven libremente. La maniobrabilidad es mayor, pero puede presentar problemas de estabilidad en terrenos difíciles. El centro de gravedad tiende a desplazarse cuando el vehículo se encuentra en una pendiente, causando pérdida de tracción [14].

Skid Steer.

Dicho diseño se compone de varias ruedas en cada lado del vehículo que actúan de forma simultánea. El movimiento es el resultado de combinar las velocidades de las ruedas de la izquierda con las de la derecha. Este tipo de configuración se utilizó en el robot "Terregator" (Figura 3), un vehículo robótico desarrollado en el Robotics Institute de la Carnegie Mellon University para aplicaciones de exteriores tales como la minería. Este robot se ha aplicado

también para inspección y obtención de mapas de tuberías enterradas empleando para ello un sistema radar (“Ground Penetrating Radar”) [16].



Figura 3. Robot Terregator

Pistas de deslizamiento.

Son vehículos tipo oruga en los que tanto la impulsión como el direccionamiento se consiguen mediante pistas de deslizamiento. Pueden considerarse parecidas al funcionamiento del Skid Steer. Debido a que el direccionamiento está dado por la combinación de velocidades de las pistas de la izquierda con las de la derecha. Su aplicación está destinada a terrenos hostiles, donde la velocidad de translación es menos importante que una adherencia al terreno [16].

Tracción síncrona.

En este diseño las ruedas son de dirección y movilidad, las ruedas están enclavadas de tal forma que siempre apuntan en la misma dirección. Para

cambiar de trayectoria el robot gira simultáneamente todas sus ruedas alrededor de un eje vertical, pero su chasis sigue apuntando en la misma orientación, de modo que se tendrá que arbitrar un procedimiento para que su cuerpo se oriente al mismo sentido que sus ruedas. Cabe mencionar que tiene una mayor complejidad mecánica [15].

Tracción diferencial.

En este tipo de locomoción su direccionamiento está dado por la diferencia de velocidades de las ruedas laterales. La tracción es dotada por las mismas ruedas. Dos ruedas montadas en un único eje son controladas y propulsadas independientemente proporcionando ambas direccionamiento y tracción. Este sistema es muy útil si se considera la habilidad del movimiento del vehículo, haciendo posible cambiar su orientación sin movimientos de traslación. Las variables de control de este sistema son las velocidades angulares de las ruedas izquierda y derecha. Los modelos cinemáticos nos proporcionan trayectorias definidas, de tal manera que su principal función sea para interiores [14].

Tracción omnidireccional.

Estos robots tienen máxima maniobrabilidad en el plano; esto significa que pueden moverse en cualquier trayectoria. En contraste las otras configuraciones cuentan con una movilidad restringida. De acuerdo a la rotación de cada una de las ruedas el robot puede avanzar, girar, o desplazarse lateralmente sin necesidad de reorientarse. Este sistema combina su direccionamiento con el diferencial, que se consigue actuando de forma independiente las ruedas que impulsan el vehículo [16].

1.4 Motores de corriente continua.

Pozueta [17] nos dice que la máquina de corriente continua o CC fue ideada por el belga Gramme en 1860. Esta empleaba un enrollador de rotor especial para lograr la rectificación del voltaje alterno generado. El físico W. Siemens entre otros contribuyeron al desarrollo de estas máquinas hasta llegar a lo que se conoce actualmente.

No obstante, la máquina de C.C. tiene múltiples aplicaciones, especialmente como motor, debido principalmente a:

- Amplio rango de velocidades, ajustables de modo continuo y controlable con alta precisión.
- Característica de torque-velocidad variable.
- Rápida aceleración, desaceleración y cambio de sentido de giro.
- Posibilidad de frenado regenerativo.

Se constituyen principalmente de un inductor, el cuál es de polos salientes y se encuentra en el estator. El inducido está en el rotor y su devanado se aloja en ranuras. Los cuales, al trabajar en conjunto con el campo magnético, generan el movimiento dependiendo su polarización [17].

1.4.1 Motor reductor.

Un motor reductor se constituye por un motor eléctrico y una serie de engranajes que van acoplados a la flecha de un motor eléctrico y sirve para reducir el número de revoluciones por minuto (rpm) del motor y mantenerlo en una velocidad constante [18].

Ventajas del motor reductor.

- Buena regularidad de marcha tanto en la velocidad como en la potencia transmitida.
- Mayor seguridad en la transmisión.
- Bajos costos de mantenimiento.
- Menor espacio requerido y mayor rigidez en el montaje.
- Menor tiempo demandado para su instalación.

Desventajas del motor reductor.

- Mayores pérdidas de potencia que otros sistemas.
- A grandes potencias se debe tener muy en cuenta la evacuación del calor del sistema.
- Mayores costos de instalación y montaje con equipos de alto rendimiento.

1.4.2 Motor reductor 1:120.

Este tipo de motor es el más utilizado en la robótica móvil, ya que brinda un control preciso respecto a su velocidad [18]. Su voltaje de alimentación para su funcionamiento esta dado de 3V a 12V. A continuación se nombran las principales características de funcionamiento con 5V:

- Torque de 4.5 Kg F*cm.
- Velocidad 100 RPM (Revolución por minuto).
- Consumo de corriente sin carga 75mA.
- Consumo de corriente atracando 670mA.
- Rango de voltaje de alimentación: 3 a 12V DC.
- Tamaño: 70.50 mm x 23.00 mm.

- Peso de 32 gr.
- Material plástico.
- Engranaje de plástico.
- Color Amarillo.

Por tanto, el motor reductor 1:120 se adecuan al proyecto, por su manejo practico y facilidad adquisitiva en relación costo-beneficio, además del consumo de energía que este nos proporciona (Figura 4).

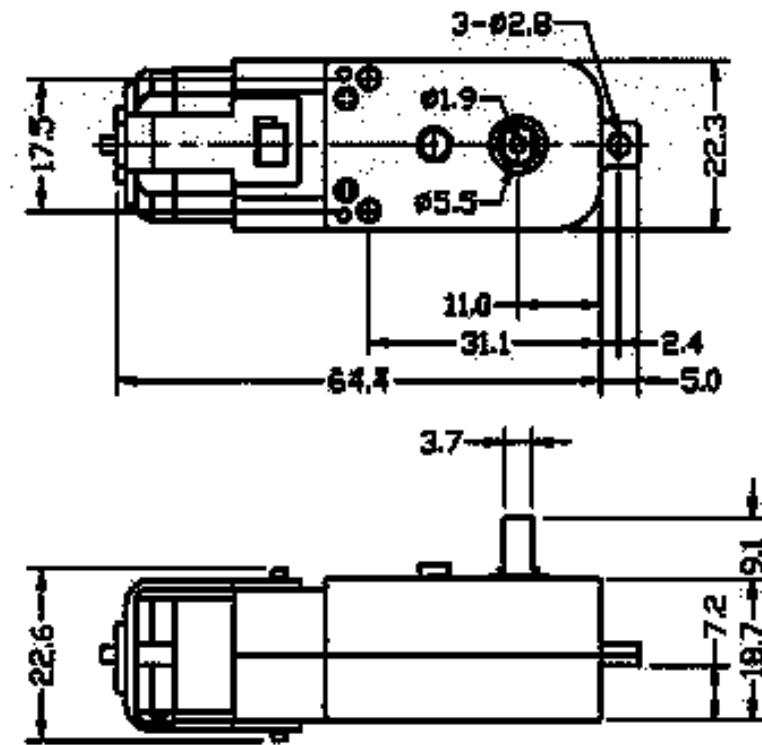


Figura 4. Esquema motor reductor.

1.5 Puente h.

Según Texas instruments [19] un puente h es un controlador cuádruple de alta-corriente diseñado para proveer accionamiento bidireccional, es decir, permite manejar cargas de potencia media, así como pequeños motores y cargas inductivas.

Es un circuito electrónico que permite a un motor eléctrico DC (Corriente Directa) girar en ambos sentidos, avanzar y retroceder. Se encuentran ya ensamblados o pueden ser contruidos con otros componentes eléctricos y/o electrónicos.

De acuerdo a lo anterior un puente h es un componente electrónico cuya principal función es controlar el movimiento de los motores de forma bidireccional, suministrando la energía suficiente para el movimiento de los motores.

1.5.1 Puentes en H integrados. L293 y L293D.

Los circuitos integrados L293 y L293D son dispositivos conductores de alta corriente de media H, es decir, enfocados a múltiples salidas de corriente. Están enfocados para proveer control bidireccional de corrientes de 1 A con voltajes de 4.5V hasta 36 V. Ambos dispositivos son generados para impulsar cargas inductivas de distintos tipos de motores, senoidales, motor a pasos, de CD y bipolares, así como otras cargas de alta corriente / alto voltaje en positivo, como suministro en distintas aplicaciones [19].

En la tabla 2 se puede observar la información física de los puentes para el ensamblaje de la placa, cabe mencionar que estos dispositivos funcionan a una temperatura de 0 a 70 grados.

Tabla 2. Dimensiones puente h

Puente H	Número de pines	Dimensiones
L293NE	16 pines	19.8mm * 6.35mm
L293DNE	16 pines	19.8mm * 6.35mm

Fuente. Texas instruments [19]

1.5.2 Diagrama lógico puente h L293NE.

Cada salida es un circuito de accionamiento de control. Los controladores están capacitados en pares: 1 y 2 habilitados por 1,2 EN; 3 y 4 están autorizados por 3,4EN (Figura 5).

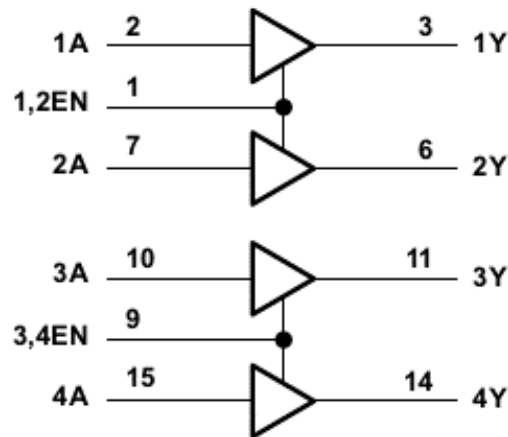


Figura 5. Diagrama lógico puente h.

En la Figura 6 se puede observar la vista del componente desde arriba, este elemento tiene 16 pines que proporcionan una función específica descrita en la tabla 3, la cual se utilizó como referencia para poder proporcionar el funcionamiento adecuado para los motores reductores.

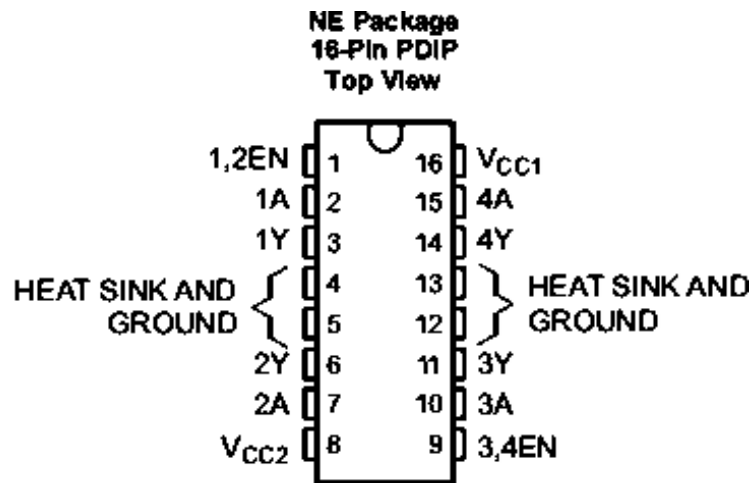


Figura 6. Diagrama lógico puente h.

Tabla 3. Funcionamiento de los pines puente H.

PIN		Tipo	Descripción
Nombre	No.		
1,2 EN	1	I	Habilita los canales de controlador 1 y 2 (entrada activa alta)
<1:4>A	2,7,10,15	I	Entradas del controlador no inversoras
<1:4>Y	3,6,11,14	O	Controladores de salida
3,4 EN	9	I	Habilita los canales de controlador 3 y 4 (entrada activa alta)
GND	4,5,12,13	-	Dispositivo de tierra y pin disipador de calor. Conectar al plano de tierra del circuito impreso con múltiples vías sólidas.
VCC	16	-	Suministro de 5 V para traducción lógica interna
VCC2	8	-	Potencia VCC para controladores de 4.5 V a 36 V

Fuente: Texas instruments [19]

El puente H L293 nos brinda los requerimientos necesarios para cubrir con el trabajo demandado de este proyecto entre los cuales se encuentra el control de dirección de los motores (Movimiento hacia delante, atrás, izquierda y derecha).

1.6 Módulo buzzer.

El Buzzer o zumbador es un dispositivo de señales de audio, el cuál puede ser mecánico, electromecánico, o piezoeléctrico. Entre sus principales aplicaciones se encuentra el de la integración con alarmas, contadores y confirmación de señales como clics o pulsadas de teclado [20].

Este dispositivo está integrado con transductores electrónicos, corriente directa, usado en computadoras, impresoras, copiadoras, alarmas, juguetes electrónicos, teléfonos, y dispositivos electrónicos para la transmisión de sonidos. Para este proyecto se desarrolló un módulo que facilita el uso y

ensamblaje del Buzzer en la placa de componentes. Este dispositivo funciona a 5V, como un simple circuito de “insertar y utilizar”.

Especificaciones:

- Buzzer pasivo.
- Puede ser controlado por un chip sencillo de IO directamente.
- Voltaje de trabajo: 5V.
- Tamaño de tarjeta 22 (mm) x 12(mm).

Configuración de los pines:

1. VCC
2. GND/ Señal.

1.7 Módulo LED RGB.

En este apartado, se describe como un led combina los colores rojo, verde y azul para formar tonos de luz.

1.7.1 LED RGB cátodo común.

El diodo emisor de luz rojo, verde y azul, por sus siglas en ingles Red Green Blue, respectivamente. En este dispositivo están integrados tres LED de estos colores. Con la intensidad de corriente de cada uno de estos LED es posible controlar una amplia variedad de colores. De la misma forma en que se utilizarían estos dispositivos de forma individual, es necesario conectar una resistencia que limite el paso de la corriente para cada pin de LED. La diferencia en la utilización del LED RGB es que el cátodo de los tres LEDS internos están conectados a una terminal llamada cátodo común. Estos diodos son utilizados para indicar estatus, uso comercial, señalamientos y luz de fondo [21].

El módulo LED RGB está compuesto de una placa que integra el circuito diseñado para “insertar y utilizar” sobre la placa de control brindando una funcionalidad óptima.

Características:

- Luz de salida uniforme.
- Bajo consumo de energía y soldabilidad longeva.

Dimensiones del diodo.

En la figura 7 se observar el esquema de un LED RGB con medidas en milímetros.

1. Rojo.
2. Cátodo común.
3. Azul.
4. Verde.

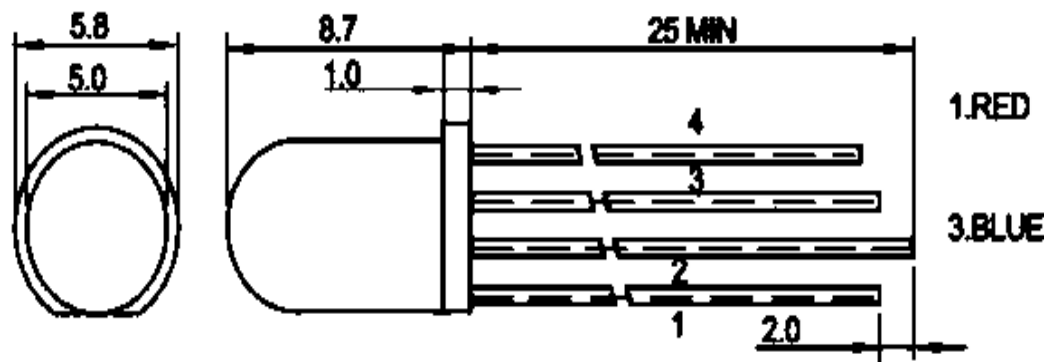


Figura 7. Esquema LED RGB.

1.8 Comunicación inalámbrica bluetooth.

En 1994, Ericsson Mobile Communications como se citó en [22] inició un estudio para investigar la posibilidad de crear una interfaz de radio de baja

potencia y bajo costo entre los teléfonos móviles y sus accesorios. Como principal objetivo tiene el eliminar cables que se utilizan para la comunicación en PC, así como facilitar la comunicación a corto alcance, tanto en el área de trabajo como en los espacios públicos [22].

La tecnología Bluetooth es un estándar abierto para la comunicación de datos que utiliza señales cortas de enlaces de radio que reemplazan a los cables entre las computadoras y las unidades conectadas a ella. Propiamente es un estándar para la conectividad inalámbrica, mayormente con soporte para las computadoras y la industria del teléfono celular. Su principal función en el mercado es la transferencia de voz y datos, entre dispositivos de comunicación y PC; entre sus principales características, pueden nombrarse su robustez, baja complejidad, bajo consumo y bajo costo [23].

La principal fortaleza de la tecnología Bluetooth es la habilidad de manejar simultáneamente transmisiones de voz y datos. Es capaz de soportar un canal asíncrono de datos y hasta tres canales de voz síncronos, o un solo canal de voz y datos. Esta capacidad combinada con la conexión ad hoc del dispositivo y el descubrimiento automático del servicio, lo convierten en una solución superior para dispositivos móviles y aplicaciones de Internet. La combinación permite la solución innovadora como un móvil de manos libres para llamadas de voz, imprimir a distancia, sincronización automática con un teléfono celular y conexión con laptop.

Con base en lo anterior se concluye que la tecnología Bluetooth es una forma de comunicación a corta distancia, con bajo costo, baja complejidad de uso y un consumo de energía mínimo, lo cual lo hace óptimo para desarrollo de proyectos.

Arquitectura general.

El hardware de control de enlace Bluetooth, integrado como un chip o módulo de radio y un módulo de banda base, implementa la radio frecuencia y porciones del administrador de enlaces de la especificación Bluetooth.

Este hardware maneja la transmisión (Tx) y recepción (Rx) de radio, las cuales sirven para hacer la comunicación con la tarjeta Arduino Nano que contiene pines específicos para cumplir con el proceso de captación y envío de datos con Tx y Rx. Además del procesamiento de la señal digital requerida para el protocolo de banda base. Sus funciones incluyen establecer conexiones, soporte para enlaces asíncronos (datos) y síncronos (voz), corrección de errores y autenticación.

Topología de red.

La topología de las redes Bluetooth pueden ser punto-a-punto o punto-a-multipunto.

Los dispositivos Bluetooth, se comunican en redes denominadas piconets. Estas redes tienen la posibilidad de crecer hasta tener 8 conexiones punto a punto. Un dispositivo que pertenece a una piconet también puede pertenecer a otra, ya sea como un esclavo en ambos o como maestro de una piconet y un esclavo en otra.

A un conjunto de 2 o más redes piconets se les denomina scarnetts, con las cuales trabajan los Bluetooth.

Clases de Bluetooth.

Existen distintos tipos de clasificación de los dispositivos Bluetooth como "Clase 1", "Clase 2" o "Clase 3" y estas hacen únicamente referencia a la potencia de transmisión del dispositivo, dichas jerarquías son totalmente compatibles con los dispositivos de una clase y con los de la otra [23]. En la tabla 4 se muestra los rangos de potencia en cada clase:

Tabla 4. Clasificación y capacidad de las clases Bluetooth.

Clase	Potencia máxima permitida	Potencia máxima permitida	Rango aproximado
1	100mw	20dbm	~100 metros
2	2,5mw	4dbm	~20 metros
3	1mw	0dbm	~1 metro

Fuente. López [23].

1.8.1 HC serial bluetooth.

Pahuja [24] menciona que el HC Serial Bluetooth consiste en un módulo de interfaz serial y adaptador Bluetooth. Este es usado para convertir el puerto serial a Bluetooth. Este módulo tiene 2 modos: dispositivo maestro y esclavo. El dispositivo salido de fábrica tiene un número después del nombre que define si es maestro o esclavo y este no puede ser cambiado a otro modo. Pero para el dispositivo nombrado después de los números, los usuarios pueden definir la forma de trabajo (maestro o esclavo) del dispositivo vía comandos AT.

El HC-06 específicamente incluye:

Dispositivo maestro: HC-06-M, M= Maestro

Dispositivo esclavo: HC-06-S, S= Esclavo

La principal función del módulo Bluetooth es reemplazar el puerto serial, así como: La conexión de las señales RX y TX, entre un maestro y un esclavo, para que estas se comuniquen entre sí.

- Cuando el Microcontrolador (MCU) tiene un módulo esclavo Bluetooth, ese se puede comunicar con un adaptador Bluetooth de computadora y teléfonos inteligentes.
- Los dispositivos que cuentan con Bluetooth en el mercado por lo general son esclavos, así que se puede usar un dispositivo maestro y hacer el proceso de comunicación.
- La operación de los módulos Bluetooth no necesita controladores, y pueden comunicarse con los dispositivos con Bluetooth. Dicha comunicación se dará en dos condiciones:

- I. La comunicación debe ser entre maestro y esclavo.
- II. La contraseña debe ser correcta.

En la tabla 5 se muestran las diferencias esenciales entre el módulo HC-05 y HC-06.

Tabla 5. Diferencia entre módulo HC-05 y HC-06.

HC-05	HC-06
Modo maestro y esclavo pueden cambiarse.	Modo maestro y/o esclavo, no pueden cambiarse.
Nombre del Bluetooth: HC-05	Nombre del Bluetooth: HC-06
Contraseña: 1234	Contraseña:1234

Fuente. Pahuja [24].

Para el desarrollo del EcateBot se utilizó el módulo HC-06, el cuál tiene la función de ser esclavo, esto es, capta las señales que el teléfono móvil vía Bluetooth envía a través de las señales Tx y Rx.

En el siguiente apartado se habla de la placa de control de software libre, llamada Arduino.

1.9 Arduino.

Arduino [25] es una plataforma de prototipos de electrónica de código abierto basada en hardware y software flexibles, fáciles de usar. Las placas Arduino son capaces de crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores, controlando diferentes tipos de actuadores, como activar un motor, encender un LED y publicar algo en línea. Se puede enviar un conjunto de instrucciones a la placa, para hacer dichas actividades se utiliza el lenguaje de programación Arduino, el cual es basado en Wiring, este último usado para programar microcontroladores; el entorno de desarrollo integrado de Arduino (IDE) basado en Processing el cual es un lenguaje de programación flexible.

Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Esto debido a una comunidad mundial de fabricantes (estudiantes, artistas, programadores y profesionales) se han reunido en torno a esta plataforma de código abierto y sus contribuciones han añadido una cantidad increíble de conocimiento que puede ser de gran ayuda para principiantes y expertos por igual.

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para la fabricación rápida de prototipo, dirigido a estudiantes sin experiencia en electrónica y programación.

Hay otros microcontroladores y plataformas de éstos, disponibles para la informática física. Por ejemplo Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard de MIT y otros que ofrecen una funcionalidad similar.

Todas estas herramientas toman los detalles desordenados de la programación del microcontrolador y lo envuelven en un paquete fácil de usar. Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para los profesores, estudiantes y aficionados interesados sobre otros sistemas:

- Económico: las placas Arduino son económicas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino se puede ensamblar a mano, e incluso los módulos Arduino montados previamente cuestan menos de \$100 MXN.
- Multiplataforma- El software de Arduino (IDE) se ejecuta en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontroladores están limitados a Windows.
- Entorno de programación sencillo y claro: el software Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible como para que los usuarios avanzados puedan aprovecharlo también. Para los maestros, está convenientemente basado en el entorno de programación de Procesamiento, por lo que los estudiantes que aprenden a programar en ese entorno estarán familiarizados con el funcionamiento del IDE de Arduino.
- Software de código abierto y extensible: el software Arduino se publica como herramientas de código abierto, disponibles para extensión por programadores experimentados. El lenguaje puede expandirse a través de bibliotecas C ++, y las personas que quieran comprender los detalles técnicos pueden dar el salto de Arduino al lenguaje de programación AVR C en el que se basa. Del mismo modo, puede agregar código AVR-C directamente en sus programas Arduino si así lo desea.

Existen diversos tipos de Arduino, cada uno de ellos puede ser utilizado según la necesidad del usuario, para este proyecto se utilizó el Arduino nano el cual nos brinda el funcionamiento descrito a continuación [25].

1.9.1 Arduino nano.

El Arduino nano (Figura 8) es una pequeña, completa y amigable tableta basada en el ATmega328, contiene las mismas funciones que un Arduino UNO, pero en diferentes paquetes. Tiene solo un puerto de alimentación y funciona con un cable mini-B USB [25].

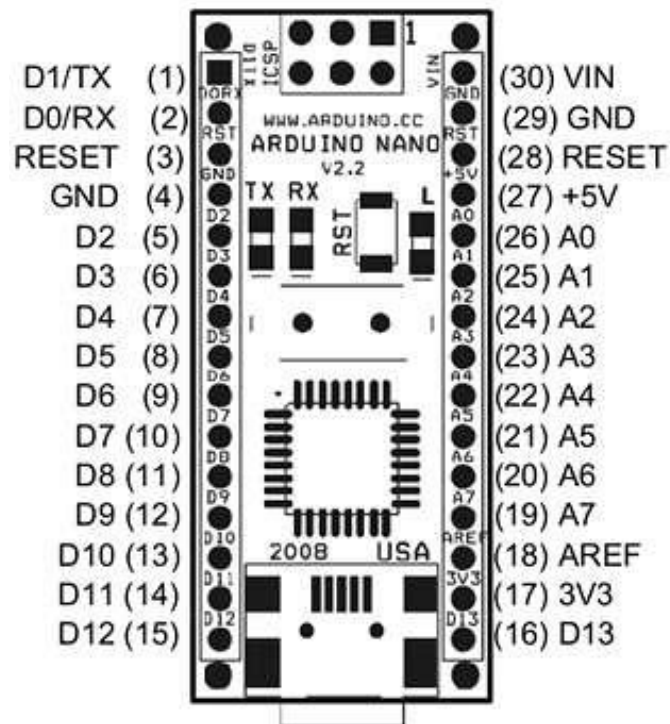


Figura 8. Diagrama Arduino nano.

Esta placa es programada en el IDE de Arduino, configurando en las opciones la tarjeta a utilizar “Arduino Nano” y el programador MCU “ATmega328”. En la tabla 6 se muestran las especificaciones técnicas de la tarjeta.

Tabla 6. Especificaciones técnicas del Arduino Nano.

Arduino Nano	
Microcontrolador	ATmega328
Arquitectura	AVR
Tensión de funcionamiento	5 V
Memoria flash	32 KB de los cuales 2 KB utilizados por el gestor de arranque
SRAM	2 KB
Velocidad de reloj	16 MHz
Clavijas analógicas IN	8 pines
EEPROM	1 KB
Corriente continua por pernos de E / S	40 mA (Pines de E / S)
Voltaje de entrada	7-12 V
Pines de E / S digitales	22 (6 de los cuales son PWM)
Salida PWM	6 pines
El consumo de energía	19 mA
Tamaño de PCB	18 x 45 mm
Peso	7 g

Fuente: Arduino [25].

Puertos de entrada y salida.

Cada uno de los 14 pines digitales en el Arduino Nano se pueden usar como entrada o salida, usando las funciones `pinMode ()`, `digitalWrite ()` y `digitalRead ()`. Las cuales operan a 5 V. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia interna de pull-up (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas como:

- Serial: 0 (RX) y 1 (TX). Se usa para recibir (RX) y transmitir (TX) datos en serie TTL. Estos puertos están conectados a los pines correspondientes del chip serial FTDI (Future Technology Devices International) USB a TTL.

- Interrupciones externas: 2 y 3. Estas clavijas se pueden configurar para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.
- PWM: 3, 5, 6, 9, 10 y 11. Proporcionan salida PWM de 8 bits con la función `analogWrite ()`.
- LED: 13. Hay un LED integrado conectado al pin digital 13. Cuando el pin tiene un valor ALTO, el LED está encendido, cuando el pin está BAJO, está apagado.

El Nano tiene 8 entradas analógicas, cada una de las cuales proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto, miden desde tierra a 5 V. Los pines analógicos 6 y 7 no se pueden usar como pines digitales. Además, algunos pines tienen una funcionalidad especializada.

Hay otro par pines en el tablero:

- AREF. Voltaje de referencia para las entradas analógicas. Se utiliza con la instrucción `analogReference ()`.
- Reiniciar. Para reiniciar el Microcontrolador mantenga el estado en LOW. Normalmente se usa para agregar un botón de reinicio a los escudos que bloquean el que está en el tablero.

Comunicación

La comunicación entre el Arduino nano y una PC se realiza mediante una conexión serial, la cual nos brinda facilidad de uso y ahorro de tiempo en carga de programas, además de la ya mencionada interfaz de programación que nos ayuda a tener un mejor control en la programación.

1.10 MIT App Inventor 2

MIT APP INEVNTOR [26] esta herramienta es un entorno de programación intuitiva y visual que les permite a todos, incluso a los niños, crear aplicaciones totalmente funcionales para teléfonos inteligentes y tabletas. Las personas que comienzan a utilizar MIT App Inventor pueden tener una primera aplicación simple en 30 min. Con un buen funcionamiento. Además, esta herramienta basada en bloques facilita la creación de aplicaciones complejas de alto impacto en mucho menos tiempo a comparación de entornos de programación tradicionales.

MIT App Inventor busca democratizar el desarrollo de software al capacitar a todas las personas especialmente a los jóvenes, para pasar del consumo de tecnología a la creación de tecnología.

Esta herramienta permite realizar las siguientes funciones:

- Configurar instrucciones: Prepara el teléfono o tableta para probar las aplicaciones en tiempo real o utiliza el emulador sino se cuenta con un dispositivo móvil.
- Diseño y programación orientada a eventos basada en bloques.
- Tutoriales para principiantes, dentro de la plataforma.
- Empaquetado de aplicación a tus dispositivos o dispositivos de tus amigos, es decir, puedes obtener el archivo del instalador (.apk) para realizar el ensamblaje en tus dispositivos con SO Android.

Para poder utilizar MIT APP Inventor 2 se necesita una conexión a internet, durante toda la sesión de trabajo. Así mismo, se debe tener una cuenta de Google, de lo contrario no se podrá ingresar a la plataforma.

Por otro lado, para configurar las pruebas de escritorio mientras se construye la app, se cuenta con 3 opciones distintas:

- Si se está utilizando un dispositivo con sistema operativo Android, se pueden construir aplicaciones sin necesidad de instalar algún software en la computadora. Solo se necesita instalar la aplicación “App Inventor Companion” que está disponible en Play Store, en tu dispositivo. Esta opción es la más recomendada, porque permite revisar los cambios en tiempo real.
- Si no se cuenta con un dispositivo Android, es necesario instalar software en la computadora el cual tiene la función de ser un emulador de un dispositivo móvil, dicho software se descarga de la página oficial. Esta opción no es tan recomendable por su complejo uso.
- Si no se tiene conexión inalámbrica a internet, se requiere de igual manera instalar el software correspondiente y una conexión alámbrica de tu dispositivo con la computadora, aunque cabe mencionar que se tienen muchos problemas al detectar los dispositivos vía USB, especialmente en Windows. Se debe utilizar esta opción como último recurso.

Dentro en el entorno de AI2 (App Inventor 2) se pueden implementar múltiples funciones, por ejemplo: sensores, medios, dibujos y animación, conectividad, maps, social, almacenamiento e interfaz de usuario.

Las herramientas y funciones de la AI2 son los óptimos para resolver las necesidades de funcionamiento del EcateBot, respecto al manejo adecuado por parte de los niños y su manejo.

La forma de trabajo para el desarrollo del proyecto se describe en el siguiente capítulo, con base a una metodología.

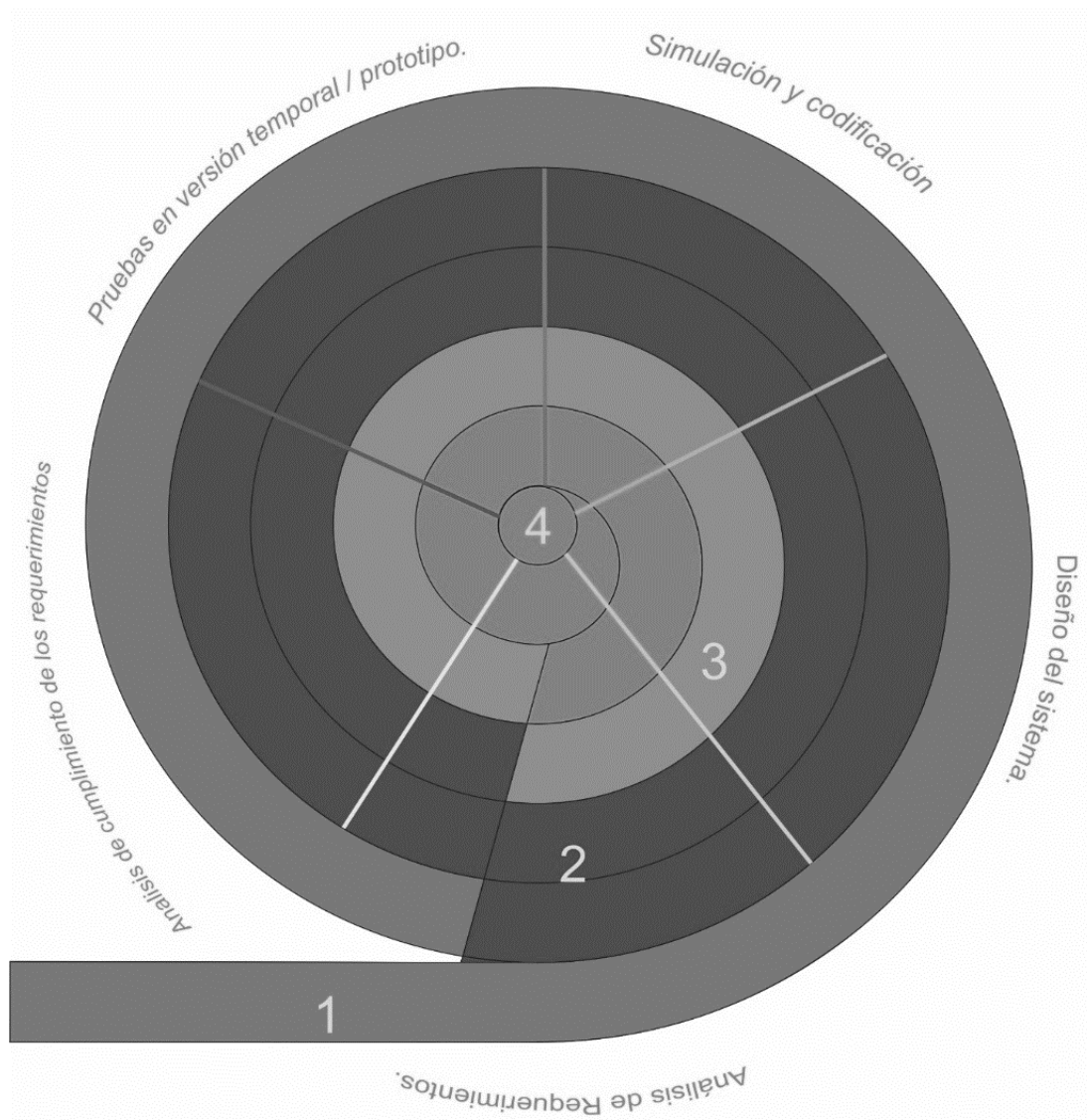
CAPÍTULO 2

Metodología de desarrollo.

2.1 Metodología para el desarrollo de proyectos de electrónica.

Pressman [27], comenta que los modelos de proceso prescriptivo fueron propuestos originalmente para poner orden en los conflictos del desarrollo de software. Los antecedentes indican que estos modelos tradicionales han brindado cierta estructura útil al trabajo de la ingeniería de software y que constituye un plan suficientemente eficaz para los equipos de desarrollo de software. Siendo así que todos los modelos de proceso de software pueden incluir actividades estructurales, pero cada una proporciona un énfasis distinto en ellas y define en forma diferente el flujo de cada proceso que invoca cada actividad estructural.

Partiendo de estos modelos y sus atributos, el modelo en espiral es el que se acopla más a la naturaleza de esta investigación, debido a que es un modelo evolutivo del proceso de software y se ajusta al entorno iterativo de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Además de que se cuenta con la posibilidad de un desarrollo rápido de versiones que a su paso se vuelven cada vez más completas. Un modelo en espiral está constituido por un conjunto de actividades estructurales. Cada una de las actividades representa un segmento de la trayectoria en espiral ilustrada en la Figura 9.



- 1.- Fase inicial de identificación de requerimientos.
- 2.- Iteraciones necesarias para el desarrollo del proyecto.
- 3.- Ciclo para el desarrollo del circuito impreso.
- 4.- Prototipo final e implementación.

Figura 9. Proceso metodológico para el desarrollo de proyectos de electrónica

Al comenzar el proceso evolutivo, se parte del centro, realizando actividades implícitas en un circuito alrededor de la espiral en el sentido de las manecillas del reloj. El primer circuito da como resultado el desarrollo de una especificación del producto; las vueltas sucesivas se usan para obtener un

prototipo, y luego versiones más sofisticadas del software. Cada paso por la región de planeación arroja ajustes en el plan del proyecto. Con base a la retroalimentación dada por el cliente se especifican los costos y programación de las acciones, esto después de su entrega. Además, el líder de proyecto ajusta el número de iteraciones para terminar el software.

El modelo en espiral es un enfoque realista para el desarrollo de proyectos, debido a que este evoluciona a partir del uso de prototipos como mecanismo de reducción de riesgos, permitiendo que en cualquier parte de la evolución del producto sea posible hacer prototipos. Este modelo demanda consideración directa de los riesgos, antes que estos se vuelvan un problema.

Para efectos de este proyecto se hicieron adaptaciones al modelo espiral. A diferencia de la metodología de espiral, se considera que las acciones deben ir encaminadas a converger en un objetivo y es por eso que las iteraciones comienzan alrededor de la espiral en el sentido anti horario, partiendo del exterior. Con esta modificación se pretende evitar, en la medida de lo posible, que el proyecto entre en un ciclo infinito de trabajo.

La metodología a utilizar cuenta con cinco etapas de diseño:

1. Análisis de requerimientos.
2. Diseño del sistema/módulos.
3. Simulación y codificación.
4. Pruebas en versiones temporal / prototipo.
5. Análisis del cumplimiento de los requisitos.

Estas etapas se repetirán en cada una de las iteraciones que se realicen a lo largo del proyecto. Las llamadas iteraciones están encaminadas a cumplir los

requerimientos funcionales de manera modular, es decir, nos ayudarán a converger en el producto final, aportando funciones específicas.

Por otro lado, en esta metodología se establecen de manera preliminar cuatro bloques. Analizando de adentro hacia afuera el desarrollo las fases son:

1. Fase inicial de identificación de imposiciones.
2. Fase de desarrollo modular.
3. Fase de desarrollo del circuito impreso.
4. Fase de prototipo final e implementación.

Aun cuando se presenta solamente para cada fase una iteración, dependiendo de las características del proyecto cada módulo puede requerir un número mayor de iteraciones. Es será definido por el desarrollador respecto a las necesidades del proyecto.

2.2 Etapa de análisis.

1. Análisis de requerimientos:

El objetivo de esta etapa es establecer y documentar los requerimientos del proyecto a desarrollar a través de la comunicación directa entre el cliente y el equipo de desarrollo. Esta etapa se desarrolla en dos momentos diferentes del proyecto. El primero es cuando comienza el proyecto y se establece el contacto inicial entre el cliente y el equipo de desarrolladores. Además, se especifica el problema a solucionar, entablando una comunicación directa entre el solicitante y el desarrollador.

El segundo momento en que se hace el análisis de requerimientos es en los ciclos de desarrollo interior, en donde se trabaja para el desarrollo de los

módulos del sistema. En estos ciclos interiores el desarrollador es quien define las exigencias del sistema en base al diseño general realizado previamente, sin necesidad que el cliente este activo en el mismo. Dichas obligaciones técnicas es necesario describirlas a profundidad.

2. Diseño de módulos:

El objetivo de esta etapa es que el desarrollador exponga, de forma escrita, el funcionamiento y la estructura del sistema a desarrollar. Esta sección está dirigida a un lector con perfil técnico el cual, a partir del documento, evaluará la viabilidad técnica y practica de desarrollo del proyecto. De esta forma, una vez definidas las características del sistema (o de un módulo), el desarrollador, mediante bocetos, diagramas de flujo y circuitos, comienza a idear la estructura general del sistema y determina el número de módulos en los que será dividido el sistema, así como las actividades de desarrollo y/o el número de iteraciones a realizar. Inicialmente el sistema será descrito de manera escrita y utilizando como apoyo diagramas a bloques. En iteraciones posteriores se hará el diseño de cada módulo pasando de la descripción por bloques a una descripción de circuito. En caso de que el sistema requiera módulos de programación de software, estos serán descritos mediante diagramas de flujo hasta llegar a los códigos de programación.

3. Simulación y codificación:

Una vez diseñados los diferentes módulos, el siguiente paso es la simulación de los mismos. El objetivo de esta etapa es comprobar el correcto funcionamiento del sistema (o del módulo) de manera virtual, esta práctica nos ayuda a evitar el desperdicio de material y tiempo de ensamblado. Esta comprobación se realizará con cada uno de los módulos por separado y posteriormente el funcionamiento general del sistema.

4. Pruebas en versión temporal / prototipo:

Cuando las simulaciones de los módulos correspondan al comportamiento deseado, se ejecutarán las pruebas de los circuitos (o programas) en un entorno real. Esto se realizará ensamblando los circuitos en tabletas temporales (Protoboard) en la fase de desarrollo modular, o en circuitos impresos para las fases de desarrollo del circuito impreso y la de prototipo final e implementación.

5. Análisis de cumplimiento de los requerimientos:

Cuando ya se tiene el módulo o el prototipo funcionando, se analiza que el resultado de los módulos y del sistema cumpla con lo establecido con el cliente.

Un aspecto muy importante a incluir es un análisis de costo desde los siguientes puntos de vista:

a) Análisis de costo del desarrollo del proyecto de investigación. En este análisis se debe de incluir una estimación de los materiales y equipos adquiridos para el desarrollo del proyecto. Es muy importante incluir los materiales adquiridos para las pruebas, los desperdicios, etcétera. También se deben de incluir los costos debidos a transportación, así como los de recursos humanos. En otras palabras, en esta sección se analizará el costo de desarrollar el proyecto desde el inicio hasta el prototipo funcional.

b) Análisis de costo de un prototipo. En esta sección se hará una estimación de cuánto costaría producir un prototipo partiendo de la experiencia adquirida hasta el momento y de los prototipos desarrollados. Se deben de incluir costos de material, recursos humanos, y tiempo de desarrollo.

2.3 Fases de desarrollo.

En estas fases se identifican diferentes tiempos necesarios para el desarrollo del proyecto. A continuación, se describe brevemente cada uno de los bloques de desarrollo.

1. Fase inicial de identificación de requerimientos:

Esta iteración se enfoca a las actividades necesarias para la definición del funcionamiento del sistema y su diseño general. En esta fase es en donde se hace la solicitud del sistema al desarrollador, el solicitante da a conocer el problema que quiere solucionar.

2. Iteraciones necesarias para el desarrollo del proyecto:

En este conjunto de iteraciones se desarrollarán los diferentes módulos del sistema. El número de iteraciones necesarias determinará el número de módulos en este bloque. En principio, al término de cada una de estas iteraciones se obtendrán prototipos o circuitos temporales de cada uno de los módulos.

3. Ciclo de desarrollo del circuito impreso:

En esta etapa se realiza el diseño, la impresión y fabricación del circuito final que será utilizado en el sistema. Este diseño ya fue probado de manera temporal en etapas previas y funciona correctamente.

4. **Prototipo final e implementación:**

Una vez funcionando todos los módulos integrados al sistema en general mediante circuitos impresos es necesario construir el prototipo final. A estas alturas, el sistema ya está terminado, restando como última tarea la implementación del mismo en el ámbito al cuál fue desarrollado.

2.4 Conclusiones.

En esta etapa del proyecto se analiza el cumplimiento de los requerimientos del cliente, próximos avances del trabajo y los aspectos que se consideren interesantes desde el punto de vista comercial, técnico y académico.

CAPÍTULO 3.

Construcción del robot.

En este apartado describiremos el proceso de elaboración del EcateBot con base en la metodología descrita previamente.

3.1 Análisis de requerimientos del EcateBot.

En esta actividad se analiza el proyecto desde el punto de vista del usuario, con el fin de identificar sus necesidades. Con miras al cumplimiento de estas imposiciones se estableció una jerarquización para cada una de las necesidades y requerimientos del cliente partiendo de su grado de transcendencia.

Se enlistan los elementos fundamentales que debe contener el EcateBot para que sea de utilidad para los niños:

- Material que resista el uso.
- Un costo asequible para su adquisición y desarrollo.
- Fácil de ensamblar y programar para que se pueda aprender y explorar con el EcateBot.
- Debe captar la atención con aspectos visuales y funcionales, evitando la reluctancia.

Requerimientos críticos que debe contener el EcateBot.

En acuerdo entre el cliente y el desarrollador se establecen los siguientes requerimientos críticos que debe tener el robot móvil:

- El vehículo debe avanzar hacia, delante, atrás, derecha e izquierda.

- Al controlarlo, los giros se deben realizar de manera suave, es decir, que no realice movimientos bruscos.
- La forma en la que se realice el control sea por conexión vía Bluetooth con un teléfono celular, el cual nos brinde las siguientes opciones:
 - Evitar objetos manualmente: cuando este vaya en una trayectoria se brinda al usuario la posibilidad de evitar la colisión.
 - Control de iluminación: como el encendido y apagado de luz del robot.
 - Control de sonido: encendido y apagado de un claxon.
- El celular controlador, indique si el robot está conectado.
- Al perder la conexión el vehículo se detenga.
- El robot cuente con una forma de recargar o sustituir la batería.
- La aplicación móvil cuente con un botón de salir.
- Los alumnos apliquen conocimientos fundamentales como la programación, desarrollo de aplicaciones con programación modular, tratamiento de imágenes, posicionamiento de componentes electrónicos y aplicación de conocimientos lógico-matemáticos, al desarrollar los módulos que conforman el EcateBot.

Otros requerimientos:

- Los motores del vehículo deben tener un torque suficiente para evitar que se atasque y/o exista un deslizamiento en una superficie plana.
- El vehículo deberá estar construido con materiales adecuados para evitar rupturas o desgaste prematuro.
- El diseño del vehículo debe ser ergonómico tanto para su uso, como para su armado.

- La aplicación para controlar el vehículo debe ser desarrollada en plataforma Android diseñada de forma responsiva, es decir, que se pueda utilizar en distintos modelos de teléfonos y se mantenga una buena distribución de los componentes. Al ser software libre se puede distribuir y modificar con mayor facilidad.
- Cualquier dispositivo con plataforma Android superior a la versión 2.1 pueda manejar el vehículo.

Requerimientos técnicos.

En la tabla 7 se concentran los requerimientos establecidos por el cliente. Así mismo, con base a estos requerimientos, el desarrollador los relaciona con componentes electrónicos que darán solución a dicha necesidad, de la manera que mejor convenga.

Tabla 7. Requerimientos usuario-desarrollador.

Tabla de requerimientos	
Usuario	Técnicos
Movilidad hacía, delante, atrás, derecha e izquierda.	Motores de corriente directa CD y microcontrolador.
Sensibilidad de control.	Aplicación Android, Mediante programación.
Control inalámbrico.	Conectar el Arduino nano a dispositivo móvil, vía Bluetooth.
Indicador de conexión con Bluetooth.	Aplicación Android.
Cualquier dispositivo Android puede ser controlador.	Aplicación Android soportable en versión mínima 2.1
Iluminación y sonidos del vehículo.	LEDS, BUZZER y microcontrolador.
Materiales que soporten el uso rudo.	Diseño del vehículo

Fuente: Elaboración propia.

3.3 Diseño modular.

En la tabla 8 se describen los módulos del EcateBot siendo estos la parte fundamental para el desarrollo del mismo, proporcionando una división del proyecto funcional y de calidad.

Tabla 8. Descripción modular del EcateBot.

Módulos propuestos para el desarrollo el vehículo	Descripción
Módulo 1: Sistema de locomoción.	Este módulo será el encargado de que el robot tenga movimiento, dado por una polea con las ruedas y el motor.
Módulo 2: Movimiento automático	<p>El robot móvil será capaz de desplazarse de un punto a otro realizando una rutina de movimientos dada por la programación de la tarjeta Arduino nano y el trabajo del puente h para dotar del movimiento a los motores, en esta rutina el EcateBot se mueve hacia delante, gira derecha, izquierda y reversa durante determinado tiempo cada acción.</p> <p>Esta rutina está en el bucle del código en Arduino y se detendrá al quitar la energía a los motores o apagando la alimentación al Arduino.</p>
Módulo 3: Aplicación móvil con luz y sonido.	El vehículo será capaz de interactuar con su entorno mediante la expulsión de señales provenientes del teléfono móvil en este caso las señales dotar de movimiento en todas direcciones al EcateBot y además accionará los módulos de LED RGB y BUZZER. Las señales del móvil serán captadas por el Arduino mediante la comunicación Bluetooth para la toma de decisiones y así mismo las acciones de los actuadores.

Fuente: Elaboración propia.

Con base al diseño modular, a continuación, se muestran los diagramas de los componentes en el robot móvil EcateBot.

3.3.1 Diagramas del robot móvil.

Con base en la tabla anterior, los siguientes diagramas a bloques vistos desde una proyección ortográfica de planta, establecen la forma en la cual los componentes del EcateBot estarán situados. Para ello es necesario implementar un nivel extra para el acomodo de los módulos del EcateBot. En la planta baja del robot móvil (Figura 10) se encuentran situados los motores reductores 1:120, con una locomoción de tracción diferencial, es decir, se agrega una rueda loca. Así mismo, un banco de baterías alcalinas tipo AA de 4 espacios, además la alimentación a la tarjeta Arduino con una batería de 9V.

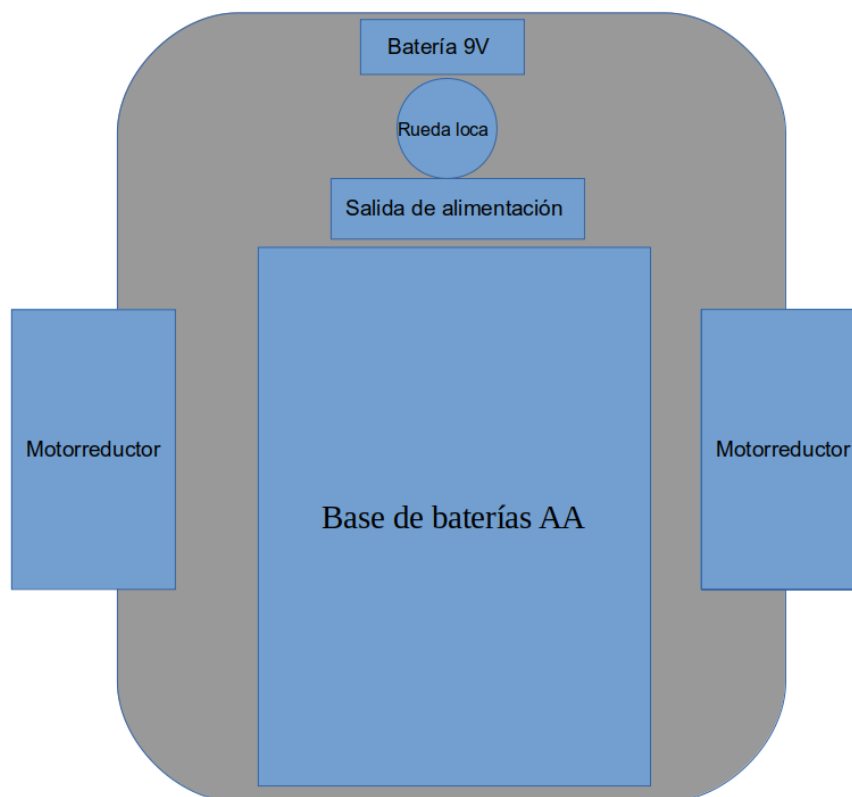


Figura 10. Esquema planta baja del robot.

Por otro lado, en el nivel 1 (Figura 11) sobre la placa de acrílico se encuentra colocado nuestro circuito impreso del EcateBot en el cual se colocan los módulos Bluetooth, LED RGB y BUZZER.

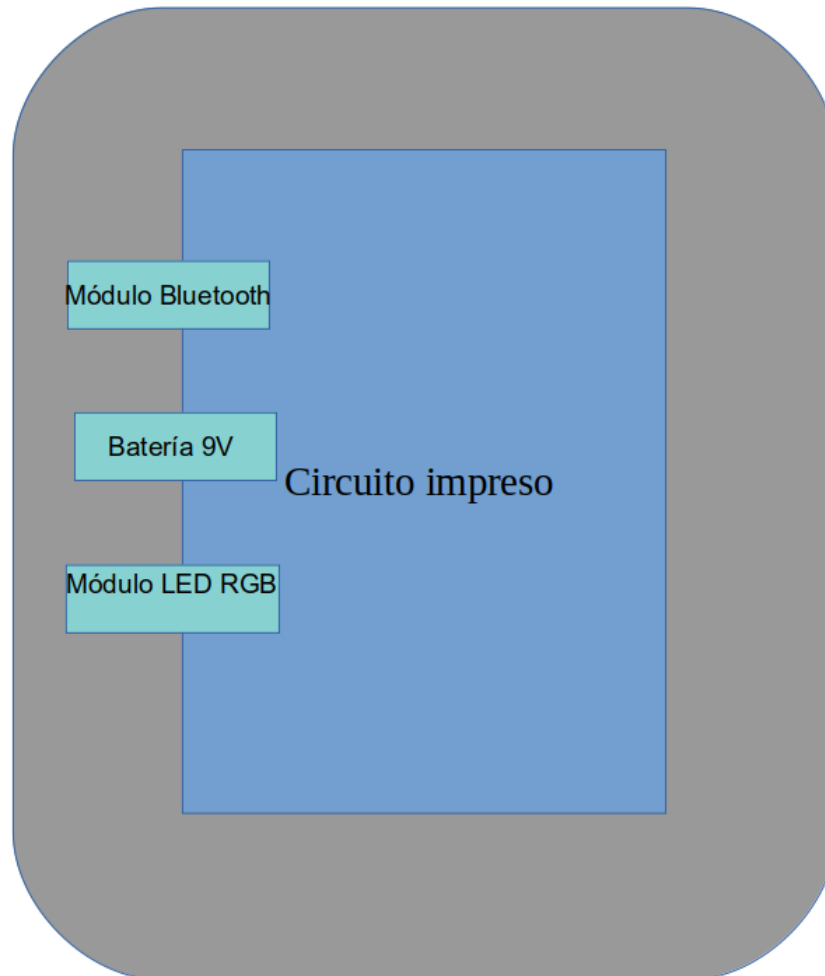


Figura 11. Diagrama EcateBot primer nivel.

Interfaz aplicación.

Con base en los requerimientos dados por el usuario se desarrolló la interfaz de la aplicación (Figura 12). Esta contiene un total de 10 botones los cuales tienen una función en específico y dichos botones son descritos en el anexo 4.



Figura 12. Interfaz gráfica de la aplicación móvil EcateBot.

3.3.2 Codificación de los módulos.

En este apartado se trabaja con el proyecto en la parte de software finalizando y depurando las partes del código que no son necesarias de implementar. Por lo que es indispensable revisar cuales son los requerimientos del EcateBot en sus distintos módulos y resolver las demandas con eficiencia.

Módulo de movimiento.

En este módulo del robot no se tiene codificación solo se necesita alimentación al motor reductor y observar ¿Cuál es el efecto que tiene en el movimiento cuando se invierten las polaridades de positivo a negativo? Siendo la iteración más sencilla del EcateBot por su fácil ensamblaje y manejo de los componentes (ver Anexo1).

Módulo de movimiento automático.

En este apartado se monta el circuito en la placa de acrílico (ver Anexo 2). Para que con ayuda del Arduino nano, se programe el código, el cual muestra una rutina de movimiento para múltiples direcciones por un determinado tiempo de ejecución, para poder observar detenidamente el cambio, se tiene dado que al realizar un cambio de dirección se detenga 2 segundos y continúe con su trayectoria, así sucesivamente, hasta comenzar de nuevo.

En la primera sección se concatenan los pines que van a utilizar los motores para determinar cuál se activará y enviará la señal de activación del movimiento.

Posteriormente en el apartado void loop () se tiene el código que se repetirá como un bucle.

Se parte con el movimiento hacia delante durante 5 segundos se detiene 2 segundos, gira a la izquierda 5 segundos, se detiene 2 segundos, continua la ejecución girando a la derecha durante 5 segundos para después detenerse 2 segundos y terminar con la instrucción para ir en reversa 5 segundos finalizando ese tiempo detiene los motores 5 segundos y repite el ciclo hasta que se desconecta la alimentación.

Se compila el programa antes de subirlo a la placa, para verificar que no haya algún error en el código, debido a que existe la posibilidad de que se olvide agregar algún símbolo o carácter provocando que no logre compilar (ver anexo 8).

Módulo de aplicación móvil con luz y sonido.

Para la comunicación entre el EcateBot y la aplicación móvil es necesario establecer la conexión del módulo Bluetooth, esta acción debe realizar después de cargar el código a la tarjeta Arduino ya que si se conecta el módulo no permitirá subir el programa a la placa (ver anexo 3).

Primeramente, se crea la codificación de la aplicación, la cual su principal función es enviar las instrucciones a ejecutar por medio de letras que el código de la tarjeta Arduino tiene una acción específica.

Código placa Arduino.

Para definir la programación en bloques relacionada con los movimientos del robot, se toma como referencia el código de movimiento automático.

En la parte preliminar de la codificación se determinan las concatenaciones que se utilizarán a lo largo del programa, siendo esta una técnica de

programación para tener un control de las variables y constantes haciendo más visual su objetivo dentro del algoritmo.

Posteriormente se inicia la comunicación serial, la cual es de utilidad para la recepción y transmisión de datos a través del módulo Bluetooth.

Comienza la parte robusta del código con una condición “IF” la cual tiene la consigna de recibir la instrucción de forma serial enviada por la aplicación y asignarla a la variable “estado”.

Se continúa con el código tomando una serie de comparaciones con la instrucción “IF”, la cual compara el valor de estado con las distintas posibles instrucciones que puede recibir por medio de la aplicación como: desplazarse al frente, movimiento a la izquierda, movimiento a la derecha, movimiento hacia atrás, encender las luces o encender Buzzer tipo sirena. Cualquiera de estas instrucciones se puede detener con la señal de estado “Bt_ST”.

Cabe mencionar que las instrucciones de accionamiento de motores pueden ejecutar en paralelo una instrucción de los módulos luz o sonido, pero no así, una instrucción semejante de movimiento, por lo que es secuencial en este sentido la parte de desplazamiento.

La lectura de la variable “estado” permanece ejecutando la última señal recibida hasta que el usuario envía una diferente.

El funcionamiento de los módulos LED RGB y BUZZER se da por el accionamiento de señales. El primero por señales analógicas que enciende y apagan el LED por determinado tiempo dado en milisegundos, es decir, las instrucciones que se están ejecutando se repiten a alta velocidad.

Lo mismo sucede con el módulo BUZZER, ejecuta con rapidez la parte de su codificación dotada por un ciclo "FOR" este nos indica que se mantendrá hasta que llegué al punto de culminación dado por la misma instrucción. Las frecuencias de sonido tienen un máximo y un mínimo las cuales son captadas por el aparato auditivo dando tonalidades de agudas a graves (ver anexo 8).

3.4 Desarrollo de proyecto.

Para llegar al prototipo final, se concluyó que sería conveniente dividir las tareas en iteraciones, cada iteración se desarrolló con base en la metodología propuesta y los requerimientos que demanda el usuario. Es así que procedemos a describir cada una de estas, las cuales a su vez complementan los módulos del EcateBot.

1. Iteración movimiento.
2. Iteración controlar el movimiento automático.
3. Iteración ensamblaje y programación del BUZZER y Leds.
4. Iteración diseño y desarrollo de aplicación para móviles con plataforma App Inventor 2.
5. Iteración configuración de la comunicación entre el dispositivo móvil y la tarjeta Arduino mediante el módulo Bluetooth/WIFI.
6. Interfaz de la aplicación.

En las siguientes secciones se describirán algunas de las características de las iteraciones realizadas a lo largo del desarrollo del proyecto.

Iteración 1. Movimiento.

Requerimientos.

Esta iteración demanda que el EcateBot sea capaz de moverse hacia delante, o atrás, se plantea que, para cumplir dichos requerimientos se utilice un motor reductor que nos proporcione el torque adecuado.

Requerimientos para el desarrollo de la iteración:

- El robot debe moverse adelante o hacia atrás.
- Capacidad de accionamiento sencillo, es decir, al conectar la alimentación se genere el movimiento.

Señales de entrada.

- cableado de alimentación

Señales de salida.

- Movimiento de la polea para dotar de torque a las ruedas.

Implementación.

Para esta etapa se montan los componentes en la placa de acrílico con las respectivas piezas que harán que se obtenga el resultado esperado, como resultado se obtuvo la finalización del primer módulo. El módulo de movimiento (Figura 13) en el anexo 1, se puede observar el proceso de armado de la iteración.

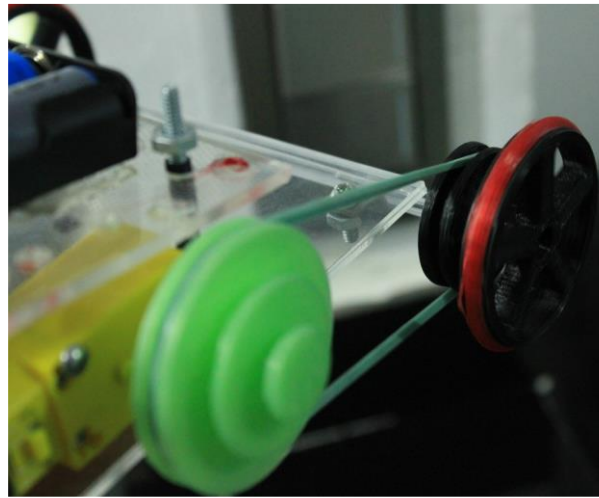


Figura 13. Montura de polea.

Iteración 2. Controlar el movimiento automático.

Requerimientos.

Para esta iteración se necesita que el EcateBot ejecute una rutina de movimiento de forma automática, la cual está dada por la tarjeta Arduino y el puente h que controlan los motores, el usuario necesita encender el carrito para observar dicho movimiento.

- Rutina con movimiento hacia adelante, atrás, izquierda y derecha.
- Sea capaz de observarse la rutina, es decir, que sus movimientos puedan ser visibles distinguiendo el cambio de dirección.

Señales de entrada.

- Alimentación de motores y puente h.
- Suministro de energía para el circuito impreso.

Señales de salida.

Accionamiento de rutina automática, que se repite en el bucle hasta que se desconecta la alimentación de la placa y/o motores.

Desarrollo de circuito.

Para el desarrollo del circuito se utilizó la herramienta de AUTODEKS llamada EAGLE. Autodesk EAGLE es un software para automatización del diseño electrónico (EDA). Permite a los diseñadores de circuitos impresos conectar sin problemas diagramas esquemáticos, ubicación de componentes, enrutamiento de circuitos impresos y contenido completo de bibliotecas.

Este software tiene 3 versiones distintas dependiendo la necesidad de él usuario. La versión gratuita, standard y Premium. En este proyecto se utiliza la versión gratuita que siendo aún limitada es funcional para cubrir con las necesidades de este proyecto.

Los componentes se encuentran en la librería del software o los importamos buscando librerías desarrolladas en internet integrándolas en la carpeta para su uso en el proyecto.

Se toma a consideración los pines con los cuales se programan las acciones del robot móvil, es decir, colocar y hacer la conexión de los componentes con su respectivo puerto de salida o entrada, dependiendo el caso.

Se procura tener un diseño ergonómico con la colocación de los elementos tanto en el esquemático como en el PCB (anexo 6 y 7).

Posteriormente se exporta el circuito PCB como un archivo en formato G-code. Este es un archivo (Nombre del archivo.bat.etch) que contiene las coordenadas de trayectoria en X, Y, Z y otro archivo que contiene las coordenadas para la perforación de los pines sobre los que se soldarán los componentes (Nombre del archivo.batch.ech.drill) estos los utiliza la máquina CNC (Control numérico computarizado), dicha máquina trabaja con motores a pasos que son muy precisos en sus movimientos con ayuda de un entorno de trabajo que a su vez funciona con una PC con sistema operativo Windows, en nuestro caso Windows 7.

Para este diseño es necesario considerar todos los elementos que contendrá el EcateBot, módulos Buzzer, Bluetooth, LED RGB y control de los motores, esto para que la placa sea de utilidad para las iteraciones posteriores. El diseño está plasmado en la sección de anexos.

Implementación.

Una vez terminada la placa se ensamblan los componentes propiamente acomodados en el lugar correspondiente, así mismo es de vital importancia contar con un manual de armado para el usuario, éste para solventar cualquier duda que llegue a surgir en su desarrollo.

Por otro lado, cabe mencionar que el funcionamiento del EcateBot cumple con los requerimientos dados por el usuario

Iteración 3. Ensamblaje y programación de módulos Buzzer y LED RGB.

En esta iteración se trabaja con los módulos BUZZER y LED RGB en conjunto ya que son la parte de captación visual y auditiva del EcateBot.

Requerimientos.

A continuación, se hace el listado de los requerimientos fundamentales de accionamiento de estos módulos, en su programación y en su acoplamiento con la placa.

- La luz debe ser intermitente entre colores azul y rojo.
- El sonido del zumbador debe ir de agudo a grave.
- Ambos accionamientos terminan hasta que llega otra señal.

Señales de entrada.

- Accionamiento de la rutina si la señal de entrada es igual a la letra descrita en la programación.

Señales de salida.

- Intermitencia de luz roja y azul estilo sirena policial.
- Sonido Buzzer estilo sirena policial del módulo BUZZER.

Desarrollo de circuito.

Para poder conformar los respectivos circuitos se utilizan las mismas herramientas previas y sus componentes electrónicos.

Primeramente, se hace uso de Eagle para el diseño de los esquemáticos y PCB, los cuales se pueden observar en los anexos 6 y 7.

Para la producción de la placa fenólica con el circuito, se hizo uso de la máquina CNC cargando los archivos para las pistas y pines de ensamblaje.

Para el módulo LED RGB se utilizaron 3 resistencias de 220 ohm para proteger a los pines rojo, verde y azul respectivamente, se colocaron en el circuito junto

con nodos macho, estos últimos para que pueda ser ensamblado y retirado con facilidad.

Por su parte el módulo BUZZER se monta en la placa implementando además pines macho con la misma finalidad que los pines del módulo LED RGB permitan ergonomía en el armado del robot móvil.

Implementación.

Se prueba el funcionamiento de ambos módulos ejecutando el código de manera secuencial en la placa desarrollada para la iteración de movimiento automático. Se obtuvo un resultado positivo en la iteración, la cual está lista para ser integrada en el producto final.

Es necesario tener un apartado en el instructivo de cómo montar estos módulos en la placa del EcateBot ya que pueden dañarse al ser ensamblados erróneamente.

Iteración 4. Diseño y desarrollo de aplicación para móviles con plataforma App Inventor 2.

En esta iteración se trabaja con el desarrollo de la aplicación en la plataforma App Inventor 2.

Requerimientos.

Como funciones principales de la aplicación móvil se considera que sean las siguientes:

- Fungir como control remoto para el movimiento del EcateBot.

- Accionamiento de LUZ, Sonido en el robot móvil.
- Debe ser capaz de establecer una conexión inalámbrica con el uso del módulo Bluetooth.
- Diseño adaptable a distintos tipos de dispositivos.

Señales de entrada.

La aplicación fungirá como un control remoto por lo consiguiente no tiene señales de entrada, es decir, es el encargado de enviar dichas señales a un receptor.

Señales de salida.

Como se describió previamente el control del EcateBot enviará señales por medio de caracteres al momento que el usuario presiona alguno de sus botones:

- Conexión al módulo Bluetooth.
- Movimiento hacia: adelante, atrás, izquierda, derecha y stop.
- Accionamiento de sonido, luces de sirena policial y sonido dispositivo móvil al presionar un botón.

Desarrollo de la aplicación.

Con base a las iteraciones previas, en esta iteración de igual manera se desarrolló un manual de armado de aplicación, en términos generales se explica cómo llegar al final del desarrollo móvil (anexo 4).

Implementación.

Posterior al desarrollo es forzoso instalar el “archivo.apk” en el teléfono. A su vez, es necesario activar la opción de instalar aplicaciones de terceros para que permita ejecutar el paquete.

Una vez que finaliza la instalación se requiere dar los permisos de uso, para posteriormente continuar con las pruebas usando el robot móvil. Como observaciones posteriores al desarrollo, se identifica que es funcional la personalización de la aplicación, es decir, el usuario aprenderá hacer la distribución de los elementos y la configuración necesaria.

Iteración 5. Configuración de la comunicación entre el dispositivo móvil y la tarjeta Arduino mediante el módulo Bluetooth/WIFI.

Dado que las iteraciones de desarrollo están terminadas y funcionando, es necesario hacer una iteración de comunicación entre el robot móvil EcateBot y la aplicación de control remoto EcateBot.

Requerimientos.

Establecer una comunicación efectiva con el robot móvil, es decir, generar un buen canal de transmisión de la información, con retroalimentación dada por las siguientes características:

- Uso del módulo Bluetooth.
- Conexión con interfaz de la aplicación móvil.
- Al momento del envío de una señal obtener una respuesta, aplicación-robot.

Señales de entrada.

Enlace de comunicación entre Bluetooth, en la cual el módulo Bluetooth del robot móvil EcateBot funge como esclavo y el control remoto EcateBot (Aplicación) es el dispositivo maestro.

Estas señales son los caracteres que envía la aplicación.

Señales de salida.

Accionamiento de los módulos, movimiento de los motores respecto a la señal recibida.

Desarrollo de comunicación.

1. La primera vez que se hace la comunicación entre Bluetooth es necesario ingresar a la configuración dentro de nuestro teléfono móvil, e identificar el módulo Bluetooth HC-06, al momento de seleccionarlo nos pedirá una contraseña la cuál puede ser '0000' o '1234'. Una vez conectados es recomendable cambiar el identificador de la conexión "HC-06" como, por ejemplo. "EcateBot".
2. Dentro de la aplicación al presionar el icono de conexión, se debe seleccionar de la lista que se despliega el identificador de comunicación Bluetooth, en este caso "EcateBot", el cual demorará de 2 a 4 segundos para establecer el enlace.

3. Una vez enlazados se comienza con el envío de instrucciones al gusto (Figura 14).



Figura 14. Proceso de comunicación EcateBot.

Implementación.

Hasta este punto, las iteraciones previas se han utilizado para realizar el control del vehículo, así como la aplicación móvil (Figura 15).

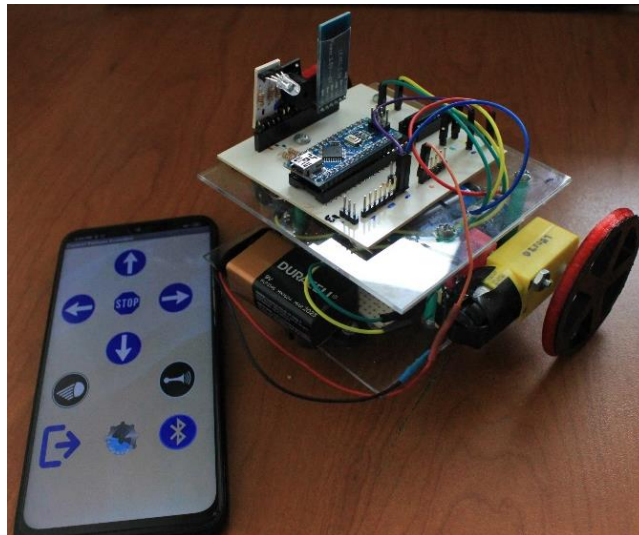


Figura 15. Vehículo y aplicación móvil.

Iteración 6. Interfaz de la aplicación.

En este apartado se trabaja con el diseño gráfico de la aplicación, dentro del cual se manejan aspectos visuales respecto al funcionamiento del robot móvil.

Requerimientos.

- Diseño de interfaz amigable, es decir, que el aspecto visual sea llamativa con colores que no irriten la vista pero que sean atractivos.
- Los botones dentro de la aplicación deben estar bien distribuidos en la pantalla.
- Facilidad de manejo y respuesta rápida.

Desarrollo.

Con base en los requerimientos es necesario contar con los siguientes archivos multimedia, ya sea que se obtengan gratuitamente en internet o se desarrollen por medio de motores de tratamiento de imágenes.

- Imágenes para colocar con los botones y fondo de pantalla.
- Sonidos para colocar en pulsación.

Partiendo de la guía que se encuentra en el anexo 4, desarrollo aplicación móvil, se puede desarrollar la interfaz con un diseño ergonómico.

Análisis de costos.

A continuación se describirán los costos requeridos (tabla 9) para el desarrollo de un EcateBot tomando como base su diseño, el cual fue descrito en las secciones previas.

Tabla 9. Costo del EcateBot.

Tabla costo de proyecto			
Materiales	Cantidad	Costo unitario (\$MXN)	Costo (\$MXN)
Arduino nano	1	80	80
Módulo Bluetooth	1	75	75
Puente H	1	25	25
Base de 40 pines	1	10	10
Base 16 pines	1	5	5
Módulo LED RGB.	1	19	19
Módulo buzzer pza.	1	19	19
Placas acrílica acrílicas	2 de 10x10 cm ² 1 de 15x15 cm ²	32	96
Tornillo con tuercas.	10	1	10
Cilindros de soporte impresora 3D	4	5	20
Rueda loca impresora 3D	2.	15	30
Motores reductores 1:120.	2	20	40
Tuvo acrílico 10 cm.	2	10	20
Ligas	1	0.5	0.5
Llantas para ejes cilíndricos impresora 3D	3	10	30

Llanta con polea impresora 3D	1	10	10
Llanta con entrada Motorreductor impresora 3D	2	10	20
Base T soporte motorreductor impresora 3D	2	10	20
Abrazaderas	4	0.5	2
Placa fenólica de 8x8 cm ²	1	30	30
Pines hembra De 16 pines	1	5	5
Tiras macho de 16 pines	2	5	10
Jumper Hembra	10	1	10
Fusible americano	2	2.5	5
Porta fusible	2	1	2
Led base pilas	1	1	1
Resistencias 220 ohm	3	0.33	1
Porta Pilas AA	1	12	12
Broche pila 9V	1	10	10
Estaño para soldar	1	10	10
Total			647.5

Por tanto, para hacer un estimado del costo total de producción se considera que en una jornada de trabajo de un técnico se podrían fabricar 10 productos.

Se considera que el pago de una persona con perfil técnico en electrónica recibiría un salario diario de \$200.00 MXN. En base a esta información el costo de un robot estará en \$667.50 MNX.

Con estos costos de materiales, mano de obra y asumiendo una ganancia del 30%, el precio de venta del EcateBot es de \$867.00 MN.

Sin embargo, es importante resaltar que antes de poner en el mercado este producto es necesario hacer una investigación de proveedores y costos de insumos considerando adquisiciones al mayoreo. De la experiencia alcanzada hasta el momento se considera que esto podría reducir los costos de materiales hasta en un 50%.

Conclusiones

A continuación se listan las conclusiones alcanzadas como resultado del desarrollo del EcateBot.

- El objetivo de este proyecto fue alcanzado, debido a que, se diseñó y fabricó un robot móvil de aplicación educativa para que niñas y niños de educación básica sean capaces de ensamblar el robot. Sin embargo, en la práctica se ha observado que el robot es de interés para todas las edades.
- El diseño del robot está constituido de forma modular. De manera que el usuario puede aplicar su creatividad para el ensamblado y aplicación del mismo. Por lo que adicionalmente está abierta la posibilidad para incluir nuevos módulos.
- El EcateBot permite construir cuatro versiones diferentes del robot que van desde un carro con tracción mecánica por polea, un carro con tracción diferencial, un carro con trayectoria programable y finalmente un carro controlado por un dispositivo móvil mediante conexión Bluetooth.
- Los costos de fabricación y el precio de venta del producto son competitivos respecto a los precios de productos similares en el mercado. Sin embargo las características del EcateBot son superiores a las de otros productos, en relación al costo-beneficio.
- La programación de una trayectoria del robot requiere de la aplicación de conocimientos matemáticos tales como el cálculo de la distancia recorrida a partir del tiempo. Por esta razón se considera que el EcateBot puede ser una herramienta en el aprendizaje y aplicación de estos conceptos.

Anexos.

Anexo 1. Manual de ensamblaje módulo de movimiento.

Material:

1. Tornillos-tuerca y abrazaderas (4x).



2. Tornillos-tuercas para base (4x).



3. Cilindros (4x).



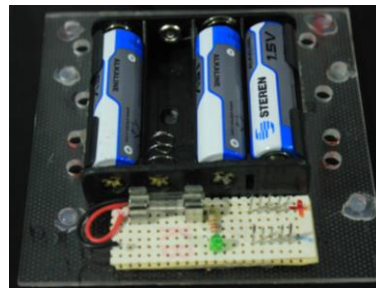
4. Rueda con tracción.



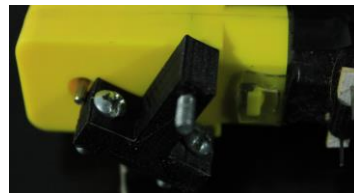
5. Tubo de plástico (2X).



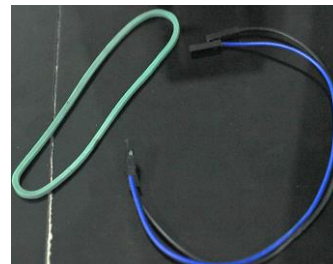
6. Porta pilas y baterías AA (4x).



7. Soporte "T" y motor reductor.



8. Liga y Jumper tipo hembra-hembra.



9. Llantas para tuvo (4X).

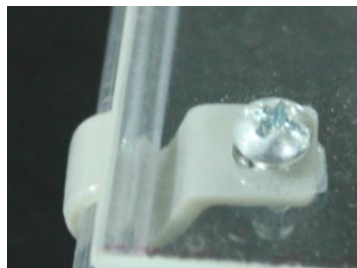


Es hora de poner manos a la obra

1. Coloca las cuatro pilas AA en la porta pilas que se encuentra en el acrílico1.

NOTA: No olvides que las pilas tienen un sentido. La parte plana de la pila va de lado del resorte.

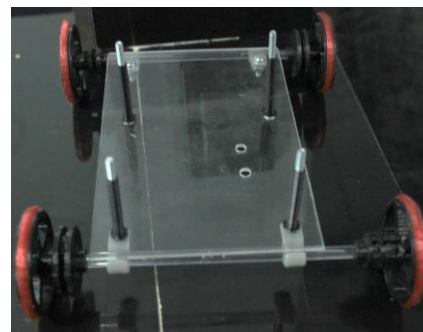
2. Ensambla las 4 abrazaderas sujetando los tubos de plástico en los agujeros que se encuentran en el superior.



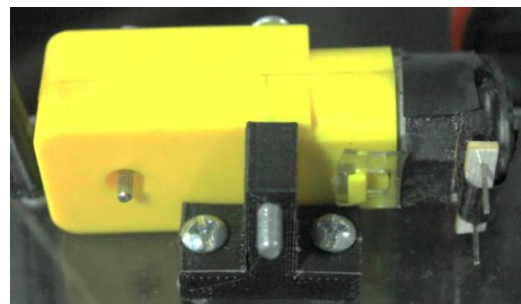
3. Coloca las 4 ruedas en la punta correspondiente de cada tubo.



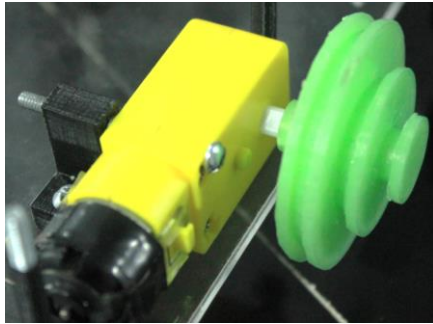
4. posteriormente coloca los cuatro tornillos y sus respectivos cilindros.



5. Coloca el tornillo mediano en la base "T" y los tornillos pequeños en la parte superior.

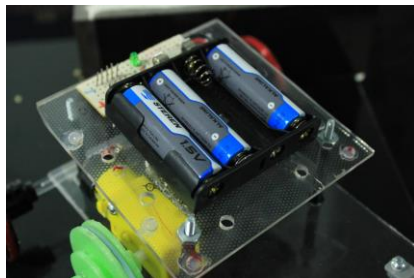


6. Monta el motor sobre la base con las ruedas asegurándolo con las tuercas.



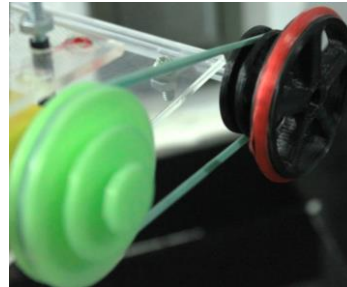
7. Coloca la rueda en la junta del motor hasta que embone por completo.

8. Acomoda el acrílico con las baterías sobre los tornillos con cilindro y asegura la parte frontal con tuercas.

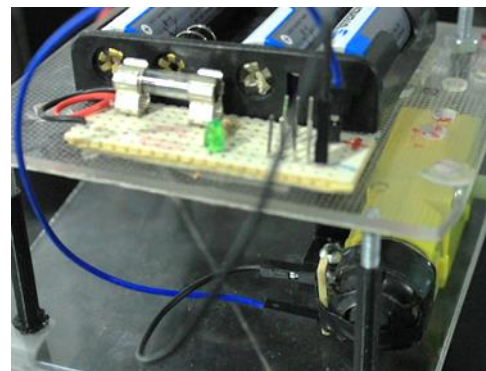


9. Con la ayuda de un desarmador y unas pinzas, gira la tuerca hacia la derecha y ajusta hasta que no se muevan los componentes en los acrílicos.

10. Utiliza la liga para sujetar la rueda del motor y la rueda.



11. inserta los Jumpers hembra en los pines del motor. Después coloca las puntas sobrantes en positivo y negativo. Observa el movimiento del motor.



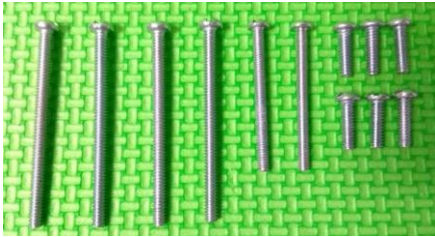
Observaciones:

- ¿Qué sucede si invertimos los cables al conectarlos en el motor?
- ¿Qué sucede si muevo la liga a un punto más tenso?
- ¿Para qué sirve el fusible? ¿Qué sucede si lo quito?

Anexo 2. Módulo de movimiento automático

1. Material:

1. Tornillos de diferentes tamaños.
+Grande (6x) +Mediano (2x)
+Chico (4x)



2. Cilindros (4x).



3. Tuercas (16x).



4. Llave.



5. Jumpers tipo hembra-hembra



(7x).

6. Motores (x2).



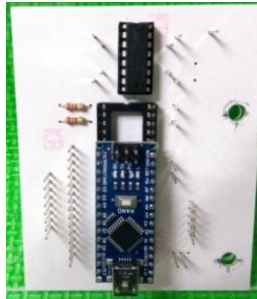
7. Base T (x2).



8. Porta pilas.



9. Acrílico y placa con Arduino.



10. Llantas (x2).



11. Rueda loca.



2. Ensamblaje:

1. Coloca las cuatro pilas AA en la porta pilas.

NOTA: No olvides que las pilas tienen un sentido. La parte plana de la pila va de lado del resorte.



2. Con ayuda de un desarmador de cruz coloca uno de los tornillos

medianos dentro del agujero del motor. (Repite para el otro motor).

NOTA: El tornillo deberá salir de lado donde están las dos conexiones del motor.



3. Coloca la base T dentro del tornillo en la entrada base de la T. (Repite para la otra base T).



4. Coloca una tuerca en el tornillo y ajusta.

NOTA: Observa que el tornillo queda muy cerca de las pilas y es difícil introducir una tuerca para el tornillo. Para sujetar el tornillo, saca un poco el tornillo éste e incorpora la tuerca poco a poco, hasta que se ajuste bien.

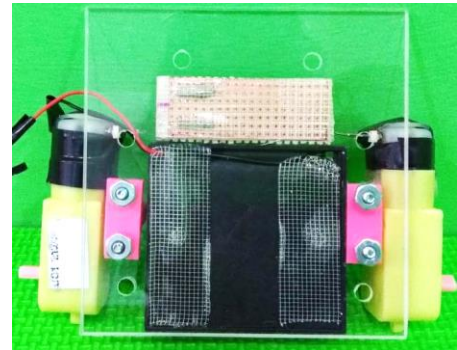


5. Toma dos tornillos pequeños e ingresa cada uno en los agujeros restantes de la base T. (Repite para la otra base T).

6. Posiciona la pieza armada en la porta pilas, en los dos agujeros del centro. (Repite para la otra pieza).

7. En cada agujero coloca un tornillo pequeño y sujétalo con una tuerca. (Repite con los demás agujeros de las bases T).

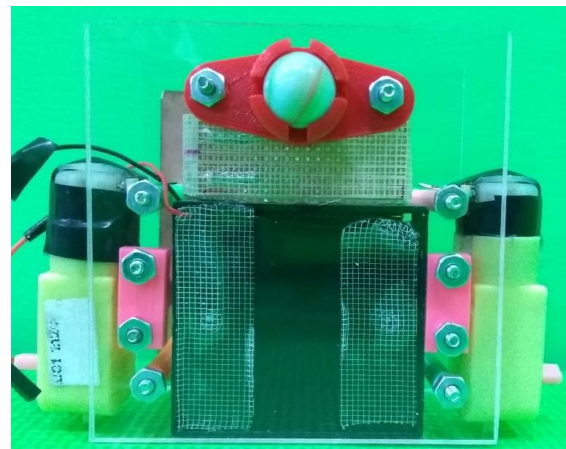
NOTA: Las tuercas deberán ir por debajo de la porta pilas.



8. Toma la rueda loca y colócala por debajo de la porta pilas, en los agujeros que están frente a las conexiones del motor.

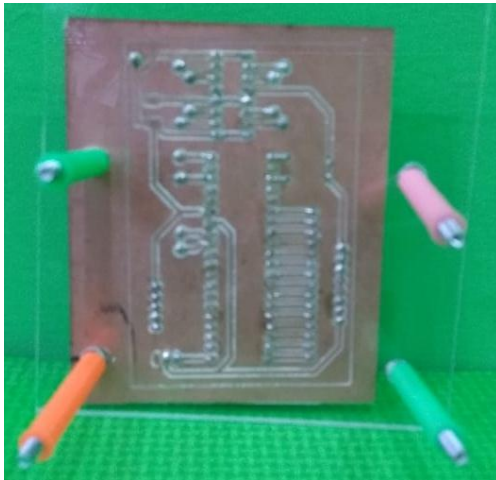
9. Ingresas dos tornillos pequeños para cada agujero de la rueda loca.

10. Sujétalas con una tuerca para cada tornillo.

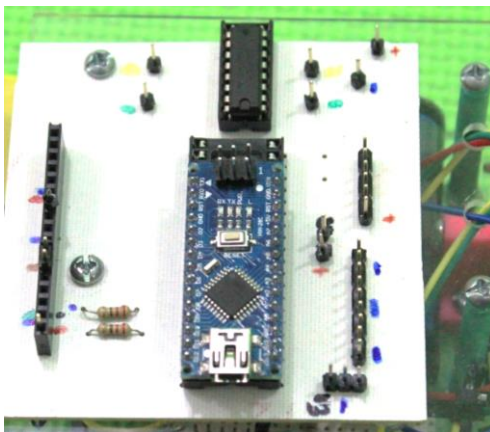


11. Toma el Acrílico/placa y coloca los cuatro tornillos grandes en cada uno de los agujeros.

12. Embona cada uno de los cilindros en los tornillos.



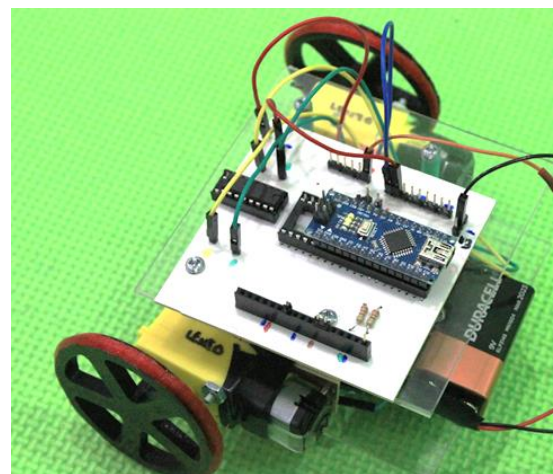
13. Instala el Acrílico/placa, colócalo sobre la porta pilas y



embona los tornillos en los agujeros.

14. Sujeta cada tornillo con una tuerca y ajusta.

15. Embona las ruedas a cada extremo de los motores. Así estará listo el EcateBot.



Observaciones:

¿Cuánto tiempo demora en terminar la rutina de movimiento?

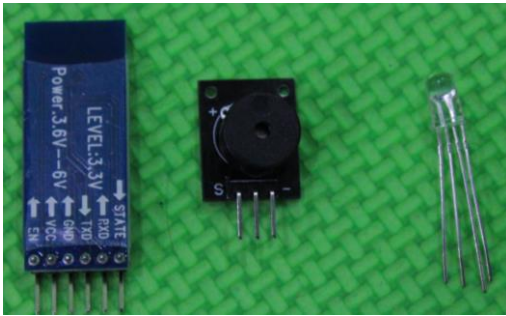
¿Cómo puedo bajar la velocidad de los motores?

¿Podrías hacer una rutina de movimientos?

Anexo 3. Módulo de control inalámbrico

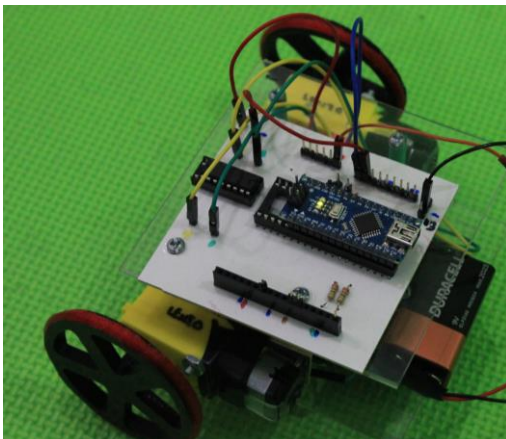
1. Material:

1. Módulo Bluetooth HC-05
2. Placa Buzzer
3. LED RGD



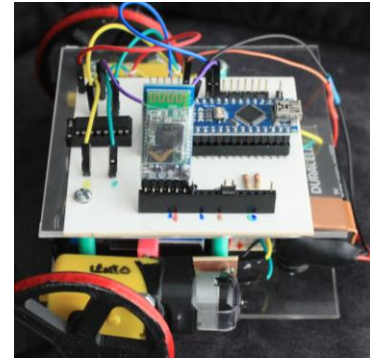
2. Ensamblaje:

Partiendo de la configuración que se utilizó en el módulo de movimiento automático se agregan los componentes antes descritos para montarlos en el circuito base.

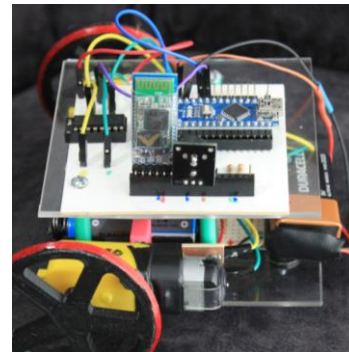


Primero coloca el módulo Bluetooth en los 6 primeros pines de la línea hembra. La parte de circuito del

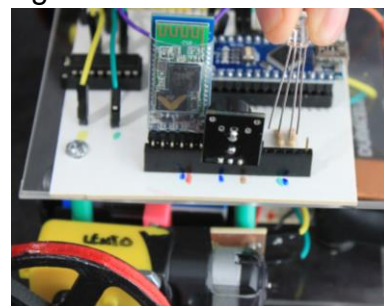
Bluetooth tendrá la vista por fuera del carrito.



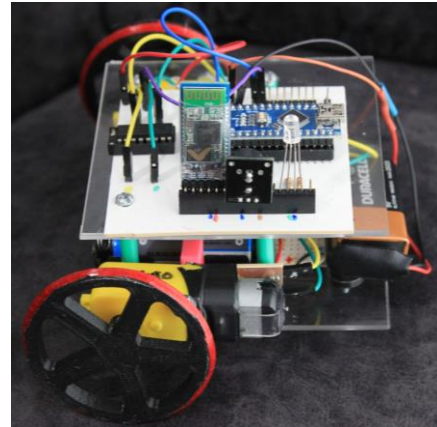
Repetimos el proceso con el Buzzer, los 3 pines inmediatos servirán para ensamblarlo.



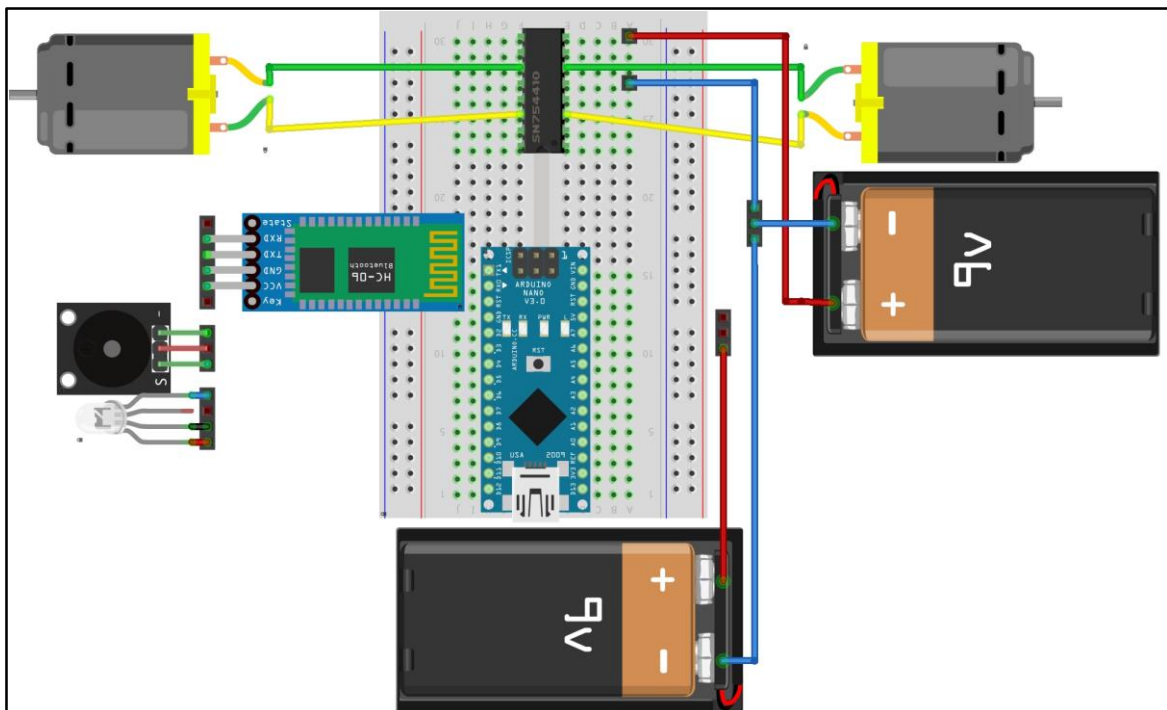
Por último, se coloca en los 4 pines restantes el LED RGB de izquierda a derecha, como se muestra en la imagen.



Finalmente se utilizan los jumper para realizar la conexión como se indica en el diagrama. Si la polaridad del motor reacciona diferente intercambia los cables. Así finaliza la conexión del EcateBot.



3. Diagrama de conexión.



Anexo 4. Desarrollo aplicación móvil.

Materiales:

Dispositivo móvil con sistema operativo Android en su versión superior a 2.2.
Computadora con acceso a internet.

Con la finalidad de que se pueda visualizar y dar una explicación más detallada de cómo desarrollar la aplicación móvil se generó un video, el cual se publicó en la plataforma YouTube, así se tendrá acceso al mismo las veces que sea necesario.

Sin embargo, se requirió que el video se creara en dos sesiones, esto debido a que en una primera sesión se genera el armado de la interfaz gráfica, donde se muestra paso a paso como se colocan los componentes y se administran los archivos multimedia.

Link de acceso: <https://youtu.be/KkBBMfGvRA>

En una segunda sesión se plantea la configuración de la conexión Bluetooth y la programación a bloques de la aplicación.

Link de acceso: <https://youtu.be/J5AFObERCBU>

El código de la aplicación trabaja de la siguiente manera:

1. En el primer bloque prepara la conexión Bluetooth del móvil habilitando la sincronización.
2. Después de que habilita el Bluetooth al hacer click en el botón despliega una lista de dispositivos que se vincularon con el teléfono móvil, permitiendo seleccionar nuestro Módulo HC-06.
3. Una vez conectado el teléfono al módulo Bluetooth. Se mantiene esperando a que el usuario seleccione una acción por realizar, al momento de presionar B_FW (Movimiento hacia delante) envía una letra 'a' al Arduino.
4. Al momento de presionar B_LF (Movimiento hacia la izquierda) envía la letra 'b' a la placa Arduino.
5. Al instante de pulsar B_ST (Detiene los motores) envía la letra 'c' a la placa Arduino.
6. Cuando presionar B_RG (Movimiento hacia la derecha) envía la letra 'd' a la placa Arduino.
7. Presionando B_BW (Movimiento hacia atrás) envía la letra 'e' a la placa Arduino.
8. Pulsando el icono de "engrane" manda llamar a "TechInno sound" el cuál reproducirá en el celular una tonada.
9. Seleccionando B_Clax (Sonido estilo sirena) envía la letra 'h' a la placa Arduino.
10. Presionando B_Luz (Luz roja-azul estilo sirena) envía la letra 'i' a la placa Arduino.
11. Pulsando B_Salir (Desconecta el Bluetooth y sale de la aplicación) envía la letra 'c' a la placa Arduino, la cual a su vez detendrá los motores dejando estático al EcateBot.

Se propone que los niños sean capaces de realizar su propio diseño respecto a la imagen de la aplicación fondo, imágenes, sonidos y luces utilizando la programación en el IDE de Arduino, visualizándose de la siguiente manera:

```

when IpConect .BeforePicking
do
  if BluetoothClient1 . Available
  then
    set IpConect . Elements to BluetoothClient1 . AddressesAndNames
1

when IpConect .AfterPicking
do
  evaluate but ignore result call BluetoothClient1 .Connect
  address IpConect . Selection
2
  if BluetoothClient1 . IsConnected
  then
    set IpConect . TextColor to
3

when B_FW .Click
do
  call BluetoothClient1 .SendText
  text " a "
4

when B_LF .Click
do
  call BluetoothClient1 .SendText
  text " b "
5

when B_ST .Click
do
  call BluetoothClient1 .SendText
  text " c "
6

when B_RG .Click
do
  call BluetoothClient1 .SendText
  text " d "
7

when B_BW .Click
do
  call BluetoothClient1 .SendText
  text " e "
8

when Techinno .Click
do
  call TechInnoSound .Play
9

when B_Clax .Click
do
  call BluetoothClient1 .SendText
  text " h "
10

when B_luz .Click
do
  call BluetoothClient1 .SendText
  text " i "
11

when B_Salir .Click
do
  call BluetoothClient1 .SendText
  text " c "
  close application
  
```

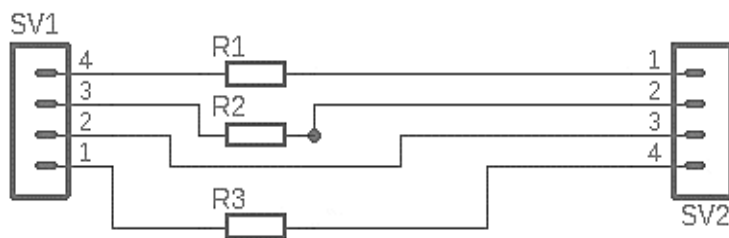
Código de la aplicación.

Anexo 5. Consumo de energía EcateBot.

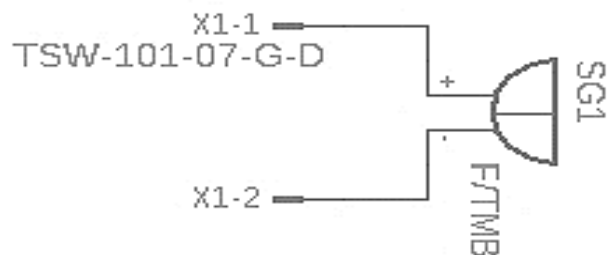
Controlando el EcateBot mediante la aplicación móvil, el consumo de energía al momento de accionar las distintas funciones del mismo, es la siguiente:

Consumo de energía EcateBot	
Función	Consumo
Circuito encendido	8mA-10mA
Conexión Bluetooth	11mA-12mA
Accionamiento módulo BUZZER	13mA
Accionamiento módulo LED RGB	11mA-12mA
Movimiento adelante	45mA >> 35mA
Movimiento izquierdo	25mA >> 20mA
Movimiento derecho	25 mA >> 20mA
Movimiento reverso	53mA >>40mA

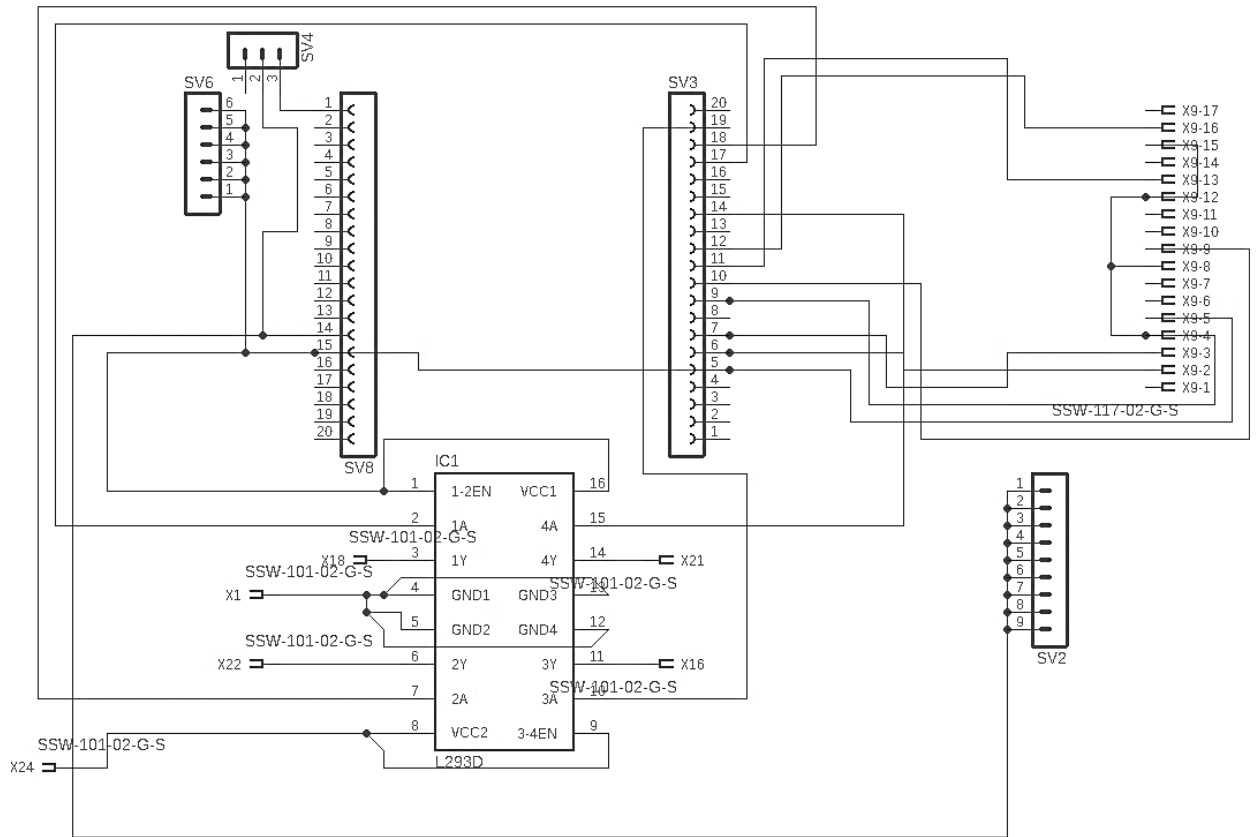
Anexo 6. Esquemáticos EcateBot.



Esquemático Módulo LED RGB.

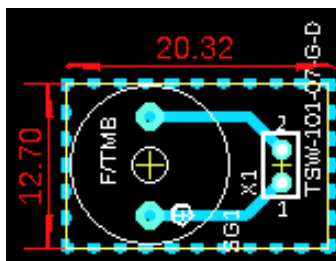


Esquemático Módulo Buzzer.

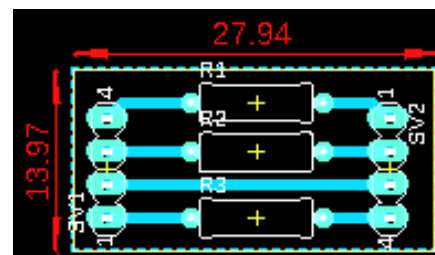


Esquemático del circuito EcateBot.

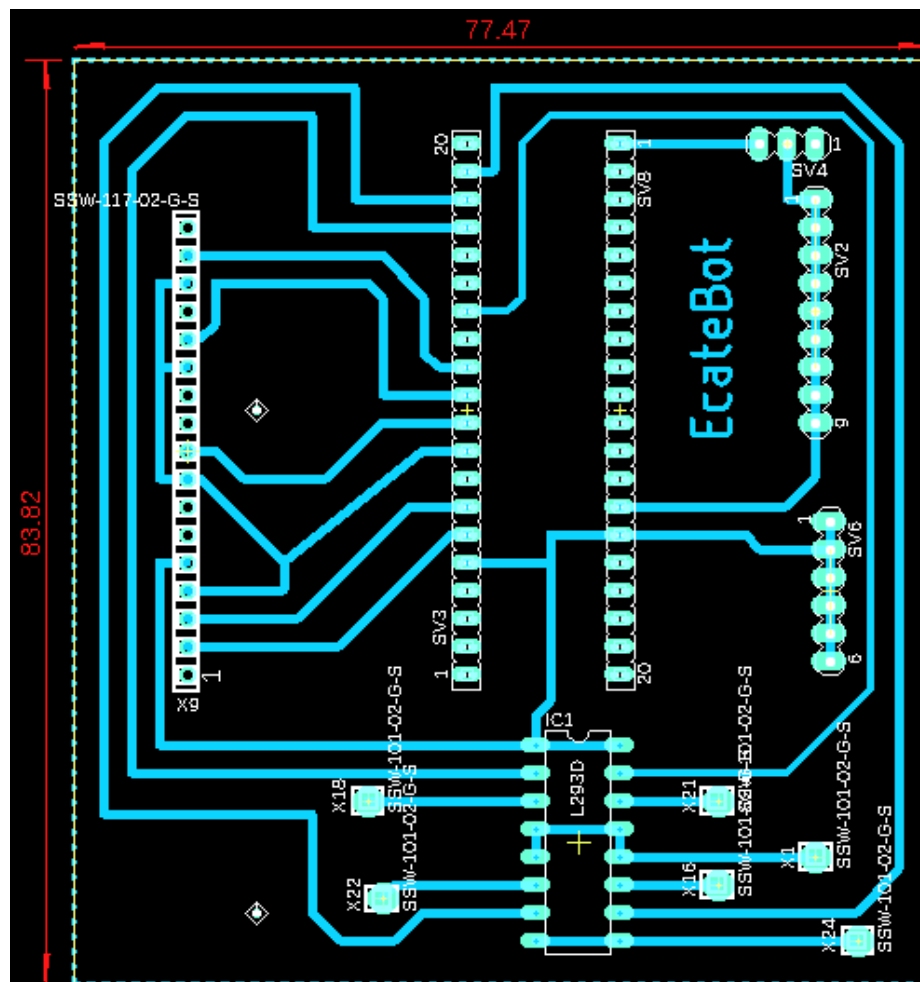
Anexo 7. Diseño de placas modulares y control EcateBot.



Diseño PCB Módulo
BUZZER



Diseño PCB Módulo LED
RGB



Esquema PCB EcateBot.

Anexo 8. Códigos Arduino.

En este apartado se colocan los códigos utilizados en los módulos movimiento automático y control remoto.

Módulo movimiento automático.

El siguiente código desarrollado en el IDE de Arduino, brinda la funcionalidad completa de este módulo.

```
int izqA = 11;//definimos el pin 11 de Arduino para habilitar el movimiento hacia
delante de la rueda izquierda
int izqB = 6;//definimos el pin 11 de Arduino para habilitar el movimiento hacia
delante de la rueda izquierda
int derA = 10;//definimos el pin 11 de Arduino para habilitar el movimiento hacia
delante de la rueda izquierda
int derB = 9;//definimos el pin 11 de Arduino para habilitar el movimiento hacia
delante de la rueda izquierda
int vel = 255; //Concatenación 'vel' con 255 definiendo la velocidad de la moto
void setup () {
}
void loop () {//-----Rutina de movimiento
// ejecuta avanzar.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (derA, vel);
analogWrite (izqA, vel);
delay (5000);
//-----ejecuta parar.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (derA, 0);
analogWrite (izqA, 0);
delay (2000);
//-----ejecuta girar a la izquierda.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (izqA, 0);
analogWrite (derA, vel);
delay (5000);
```

```
//-----ejecuta parar.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (derA, 0);
analogWrite (izqA, 0);
delay (2000);
//-----ejecuta girar a la derecha.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (derA, 0);
analogWrite (izqA, vel);
delay (5000);
//----- ejecuta parar.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (derA, 0);
analogWrite (izqA, 0);
delay (2000);
//-----ejecuta reversa.
analogWrite (derA, 0);
analogWrite (izqA, 0);
analogWrite (derB, vel);
analogWrite (izqB, vel);
delay (5000);
//----- ejecuta parar.
analogWrite (derB, 0);
analogWrite (izqB, 0);
analogWrite (derA, 0);
analogWrite (izqA, 0);
delay (5000);    }
```

Módulo control remoto.

A continuación, las líneas de código necesarias para el módulo de control remoto utilizando la aplicación móvil.

```

int izqA = 11;//definimos el pin 11 de Arduino para habilitar el movimiento hacia
delante de la rueda izquierda
int izqB = 6;//definimos el pin 6 de Arduino para habilitar el movimiento hacia
atrás de la rueda izquierda
int derA = 10;//definimos el pin 10 de Arduino para habilitar el movimiento hacia
delante de la rueda derecha
int derB = 9;//definimos el pin 9 de Arduino para habilitar el movimiento hacia
atrás de la rueda derecha
int vel = 255;// Velocidad de los motores (0-255)
int estado = 'g'; // inicia detenido
//-----
int durac = 250; //Duración del sonido
int fMin = 2000; //Frecuencia más baja que queremos emitir
int fMax = 4000; //Frecuencia más alta que queremos emitir
int t = 0;
int buzzer = 2; //declaro pin 2 Arduino para buzzer DIGITAL
//-----
int L_rojo = 3; //declaro pin 3 Arduino L_rojo PWM
int L_azul = 4; //declaro pin 4 Arduino L_azul DIGITAL
//-----
Void setup () {
  // put your setup code here, to run once:
  Serial. Begin (9600);
  PinMode (L_rojo, OUTPUT);
  PinMode (L_azul, OUTPUT);

```

```
PinMode (buzzer, OUTPUT);
}
void loop () {
  //----- Lectura de Botón presionado
  if (Serial.available() > 0)
  {
    estado = Serial.read();
  }
  //-----control app
  if (estado == 'a') { // Botón desplazar al Frente
    analogWrite (derB, 0);
    analogWrite (izqB, 0);
    analogWrite (derA, vel);
    analogWrite (izqA, vel);
  }
  //-----
  if (estado == 'b') { // Botón IZQ
    analogWrite (derB, 0);
    analogWrite (izqB, 0);
    analogWrite (izqA, 0);
    analogWrite (derA, vel);
  }
  //-----
  if (estado == 'c') { // Botón Parar
    analogWrite (derB, 0);
    analogWrite (izqB, 0);
    analogWrite (derA, 0);
    analogWrite (izqA, 0);
  }
  //-----
}
```

```
if (estado == 'd') { // Botón DER
    analogWrite (derB, 0);
    analogWrite (izqB, 0);
    analogWrite (derA, 0);
    analogWrite (izqA, vel);
}
//-----

if (estado == 'e') { // Botón Reversa
    analogWrite (derA, 0);
    analogWrite (izqA, 0);
    analogWrite (derB, vel);
    analogWrite (izqB, vel);
}
//-----

if (estado == 'g') { // Botón OFF, detiene los motores
    analogWrite (derB, 0);
    analogWrite (izqB, 0);
    analogWrite (derA, 0);
    analogWrite (izqA, 0);
}
//-----

if (estado == 'h') //Botón claxon
{
    for (t = fMin; t <= fMax; t++) //ejecuta el ciclo hasta llegara su tono máximo
        tone (buzzer, t, durac) ;//realiza la acción de sonido
    //sonido más grave
    for (t = fMax; t >= fMin; t--) //Desde el tono mayor, revierte
        tone (buzzer, t, durac);
}
//-----
```

```
if (estado == 'i') //Botón Luz
{
  analogWrite (L_rojo, 255) ;//enciende
  delay (100) ;//espera
  analogWrite (L_rojo, 0) ;//apaga
  delay (100) ;//espera
  analogWrite (L_azul, 255) ;//enciende
  delay (100) ;//espera
  analogWrite (L_azul, 0) ;//apaga
  delay (100) ;//espera
}
}
```


Referencias.

- [1] D. M Gravini, "Estilos de aprendizaje de los estudiantes de primer semestre de los programas de Psicología e Ingeniería Industrial de la Universidad Simón Bolívar", *Psicogente*, vol.11 no.19, pp.24-33, 2008.
- [2] J. K. Keefe, "Profiling and utilizing learning styles". Virginia: NASSP. 1988.
Linnebrink, E.A., and Pintrich, P.R. (2003). The role of self-efficacy beliefs in student engagement and learning in the classroom. *Reading & Writing Quarterly*, 19.
- [3] L. J. Bolívar y V. F. Rojas. "Los estilos de aprendizaje y el locus de control en estudiantes que inician estudios superiores y su vinculación con el rendimiento académico, *REVINPOST*". Vol. 3 no.23, pp.199-215. 2008.
- [4] J. Morrissey. "El uso de las TIC en la enseñanza y el aprendizaje. Cuestiones y desafíos". 2008.
- [5] C. M. Sosa, "La robótica presente en la UACSTY" *ISSN 2007-7521*. 1(2): pp. 30-31. 2006.
- [6] M. Riccillo, " De la ficción a la realidad". Pp.1-5. 2013.
- [7] Léxico, Oxford, (2018, noviembre 14). "Definition Of Robótica" Available: <https://www.lexico.com/es/definicion/robotica>
- [8] Bambino, "Una introducción a los robots móviles", pp. 35-37. 2008.
- [9] Barrientos, "Fundamentos de robótica. México". McGraw-Hill, Segunda Edición. 2002.

- [10] D. Lemus. "Definiendo la robótica", *Saber más revista de divulgación, Web*. 2018.
- [11] 2012, B. Flores, H. Flores y A. Huerta. "Desarrollo de robot móvil de exploración dirigido mediante transferencia de video". *Sistemas, cibernética e informática*, Vol. 9, No. 1. 2012.
- [12] RAE (2018, junio 17), "Definición de robot" Available: <https://dle.rae.es/?id=WYRIhzm>
- [13] Gh. Lazea, E. Lupu. "Aspects on path planning for mobile robots", *Technical University of Bluj-Napoca, Automation department*. Pp. 1. 2001.
- [14] A. Bañó, "Análisis y diseño del control de posición de un robot móvil con tracción diferencial". *Memoria, Universitat Rovira virgili*, 2003.
- [15] Ollero, "Robótica: Manipuladores y robots móviles." *España. Marcombo, Alfaomega*, 2001.
- [16] I. Hernández, "Control de navegación en superficies planas, de un tractor no tripulado. *Instituto Politécnico Nacional, MCC*. 2008.
- [17] M. A. Pozueta, "Máquinas de corriente continua". *Universidad de Cantabria*. Pp. 72. 2017.
- [18] CHAPMAN. "Máquinas eléctricas". *Madrid: McGraw-Hill Interamericana*. 2005.

-
- [19] Texas Instruments, “L293x Quadruple Half-H Drivers”, *SLRS008D*, Pp. 18 ,2016.
- [20] Components 101 (2018, Julio 20) “Active Passive Buzzer”, Available: <https://components101.com/buzzer-pinout-working-datasheet>.
- [21] Kingbright, “T-1 3/4 (5mm) FULL COLOR LED LAMP, *rev no. V.7* 2010.
- [22] TICS (2018, agosto 23), “Uso de dispositivos digitales”, *Blogger*, Available: <https://hunartz.blogspot.com/2018/10/semana-3.html>
- [23] C. López y C. Cofré, “Estándar de comunicación inalámbrica Bluetooth.” *Universidad técnica Federico Santa María. Pp. 11.* 2012.
- [24] R. Pahuja y N. Kumar, “Android Mobile Phone controlled Bluetooth robot using 8051 Microcontroller”, *international journal of scientific engineering and research (ISJER)*, vol. 2 no. 7. 2014.
- [25] Arduino (2018, agosto 25), “Arduino y Arduino nano”. Available: <http://www.Arduino.cc/>.
- [26] MIT APP INVENTOR (2018, septiembre 10), “Acerca del producto”. Available: <https://appinventor.mit.edu/about-us>.
- [27] R. Pressman, “Modelo en espiral. En INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO”, *New York: McGraw-Hill*, Pp.33-39. 2010