



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE INGENIERÍA EN SISTEMAS**  
**COMPUTACIONALES E INFORMÁTICOS**

**Tema:**

---

“PROTOTIPO ENFOCADO AL APRENDIZAJE DE LÓGICA DE PROGRAMACIÓN EN  
NIÑOS DE EDADES COMPRENDIDAS ENTRE 4 A 10 AÑOS.”

---

Trabajo de Graduación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

**Sublínea de Investigación:** Aplicaciones para dispositivos móviles

**Autor:** Caicedo Barreno Christian Patricio

**Tutor:** Ing. Edison Homero Álvarez Mayorga Mg.

AMBATO – ECUADOR

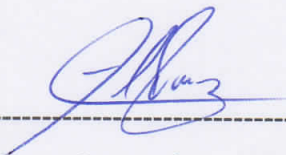
2018

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: PROTOTIPO ENFOCADO AL APRENDIZAJE DE LÓGICA DE PROGRAMACIÓN EN NIÑOS DE EDADES COMPRENDIDAS ENTRE 4 A 10 AÑOS, del señor Caicedo Barreno Christian Patricio estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato Junio, 2018

EL TUTOR



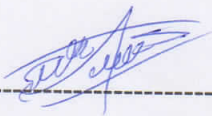
Ing. Édison Álvarez Mg.

CC: 180122596-0

## AUTORÍA

El presente Proyecto de Investigación titulado: PROTOTIPO ENFOCADO AL APRENDIZAJE DE LÓGICA DE PROGRAMACIÓN EN NIÑOS DE EDADES COMPRENDIDAS ENTRE 4 A 10 AÑOS, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Junio, 2018



---

Caicedo Barreno Christian Patricio

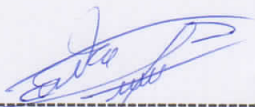
CC: 180422649-4

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato Junio, 2018



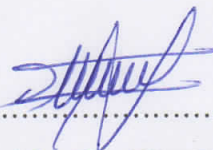
---

Caicedo Barreno Christian Patricio

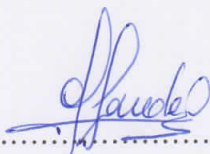
CC#: 1804226494

## APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes: Ing. Mg. Oswaldo Paredes, Ing. Mg. Hernando Buenaño, revisó y aprobó el Informe Final del Proyecto de Investigación titulado: PROTOTIPO ENFOCADO AL APRENDIZAJE DE LÓGICA DE PROGRAMACIÓN EN NIÑOS DE EDADES COMPRENDIDAS ENTRE 4 A 10 AÑOS, presentado por el señor Christian Patricio Caicedo Barreno de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



.....  
Ing. Mg. Elsa Pilar Urrutia Urrutia  
PRESIDENTA DEL TRIBUNAL



.....  
Ing. Mg. Oswaldo Paredes  
DOCENTE CALIFICADOR



.....  
Ing. Mg. Hernando Buenaño  
DOCENTE CALIFICADOR

## **DEDICATORIA**

*Un padre supone más que un centenar de profesores, por ello el presente proyecto de titulación tiene un especial mérito dirigido a mis padres, quienes siempre supieron brindarme su apoyo incondicional en cada decisión que he tomado.*

***Caicedo Barreno Christian Patricio***

## **AGRADECIMIENTO**

*A mis padres por ser un pilar importante en todo momento para poder lograr mis objetivos, a mi hermano por ser un ejemplo a seguir dándome apoyo y motivación constante.*

*A mi tutor por ser una guía importante con sus conocimientos, experiencia y paciencia para así poder culminar con éxito una de las metas trazadas.*

*Son muchas las personas que han formado parte de mi vida estudiantil a las que me encantaría agradecer su amistad, consejos, apoyo, ánimo y compañía en cada momento mi carrera y mi vida.*

***Caicedo Barreno Christian Patricio***

# ÍNDICE

<b>APROBACIÓN DEL TUTOR</b> .....	ii
<b>AUTORÍA</b> .....	iii
<b>DERECHOS DE AUTOR</b> .....	iv
<b>APROBACIÓN DE LA COMISIÓN CALIFICADORA</b> .....	v
<b>DEDICATORIA</b> .....	vi
<b>AGRADECIMIENTO</b> .....	vii
<b>INTRODUCCIÓN</b> .....	xvii
<b>CAPITULO 1 El Problema</b> .....	1
1.1. Tema.....	1
1.2. Planteamiento del problema .....	1
1.3. Delimitación .....	3
1.4. Justificación .....	3
1.5. Objetivos.....	4
1.5.1.    Objetivo general.....	4
1.5.2.    Objetivos específicos .....	4
<b>CAPITULO 2 Marco Teórico</b> .....	5
2.1. Antecedentes investigativos.....	5
2.2. Fundamentación Teórica .....	6
2.2.1.    Lenguaje de programación.....	6
2.2.2.    Lógica de programación .....	7
Los algoritmos informales .....	7
Los algoritmos computacionales .....	7
2.2.3.    Secuencias.....	7
2.2.4.    Decisiones.....	8
2.2.5.    Ciclos .....	8



2.2.6.	Microcontroladores.....	9
2.2.7.	Prototipos.....	9
2.2.8.	Metodologías tradicionales de desarrollo de software.....	10
2.2.9.	Metodologías Ágiles.....	11
2.2.10.	Tipos de Modelo de Prototipos.....	12
2.2.11.	Modelos para la construcción de hardware.....	12
2.3.	Propuesta de solución.....	13
<b>CAPITULO 3 Metodología.....</b>		<b>14</b>
3.1.	Modalidad de la investigación.....	14
3.2.	Población y muestra.....	14
3.3.	Recolección de la información.....	15
3.4.	Procesamiento y análisis de datos.....	15
3.5.	Desarrollo del Proyecto.....	15
<b>CAPITULO 4 Desarrollo de la propuesta.....</b>		<b>17</b>
4.1.	Análisis del funcionamiento de prototipos programables con fines educativos en el área de informática dentro del país.....	17
4.1.1.	Investigación para encontrar diversos prototipos programables en las principales ciudades del Ecuador.....	17
4.1.2.	Documentación de características físicas y lógicas utilizadas por estos prototipos para su funcionamiento.....	18
4.2.	Selección del framework para el desarrollo del prototipo y la aplicación.....	20
4.2.1.	Elaboración y aplicación de encuestas.....	20
4.2.2.	Selección de un framework para el desarrollo de una aplicación móvil.....	24
4.2.3.	Selección de hardware para el desarrollo de la propuesta.....	29
4.2.4.	Metodología utilizada en el desarrollo de la propuesta.....	33
4.3.	Desarrollo de la propuesta de la (aplicación y prototipo).....	37
4.3.1.	Desarrollo de la aplicación móvil.....	37
4.3.2.	Ensamblaje y programación del prototipo.....	53
4.3.3.	Fusión del proyecto.....	62
4.3.4.	Análisis del entorno donde funciona el prototipo.....	68

4.4.	Implementación del prototipo que facilita el aprendizaje de lógica de programación en niños de 4 a 10 años de edad .....	69
4.4.1.	Evaluación sobre conocimientos de lógica de programación a niños.....	69
4.4.2.	Resultado de evaluación previa a capacitación con Rubus.....	75
4.4.3.	Capacitación a niños en lógica de programación con Rubus.....	75
4.4.4.	Evaluación sobre conocimientos de lógica de programación a niños posterior a la capacitación utilizando a Rubus como herramienta de apoyo .....	78
4.4.5.	Resultado de evaluación posterior a capacitación con Rubus .....	83
<b>CAPITULO 5</b>	<b>Conclusiones y Recomendaciones .....</b>	<b>84</b>
	Conclusiones.....	84
	Recomendaciones .....	85
	<b>Bibliografía.....</b>	<b>86</b>
	Anexos.....	89

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Algoritmo de secuencias .....	7
<b>Figura 2</b> Algoritmo de decisiones.....	8
<b>Figura 3</b> Algoritmo de ciclos.....	8
<b>Figura 4</b> Microcontroladores .....	9
<b>Figura 5</b> Ejemplo de prototipo.....	10
<b>Figura 6.</b> Robot Teebot 100% Ecuatoriano .....	19
<b>Figura 7</b> Tipo de dispositivos en donde les gustaría que aprendan a programar .....	21
<b>Figura 8</b> Tipo de dispositivo que poseen los padres de familia.....	22
<b>Figura 9</b> Sistema operativo en dispositivos móviles en Ecuador.....	23
<b>Figura 10</b> Evaluación de características de framework .....	28
<b>Figura 11</b> Evaluación de selección de hardware para el desarrollo de la propuesta.....	32
<b>Figura 12</b> Metodología desarrollo por prototipos.....	34
<b>Figura 13</b> Metodología desarrollo de hardware.....	36
<b>Figura 14</b> Icono de Android Studio .....	37
<b>Figura 15</b> Icono de Rubus en el celular .....	38
<b>Figura 16</b> Pantalla del programador .....	39
<b>Figura 17</b> Pantalla del programador 2do Prototipo.....	40
<b>Figura 18</b> Botón Bases.....	40
<b>Figura 19</b> Botón ciclos.....	41
<b>Figura 20</b> Etiqueta If Pantalla del programador .....	42
<b>Figura 21</b> Colores que sensa Rubus.....	42
<b>Figura 22</b> Pantalla Inicial del prototipo 5 .....	43
<b>Figura 23</b> Icono de la aplicación del prototipo final.....	44
<b>Figura 24</b> Pantalla Inicial del prototipo 6 .....	45
<b>Figura 25</b> Pantalla Botón voz .....	45
<b>Figura 26</b> Pantalla Inicial del prototipo final.....	46
<b>Figura 27</b> Pantalla del botón bases .....	47
<b>Figura 28</b> Pantalla del botón condicionales .....	47
<b>Figura 29</b> Pantalla del botón ciclos.....	48
<b>Figura 30</b> Pantalla del botón voz .....	48
<b>Figura 31</b> Interfaz de trabajo .....	49
<b>Figura 32</b> Interfaz de trabajo donde se elimina ordenes no deseadas.....	49
<b>Figura 33</b> Interfaz de trabajo donde se elimina todas las ordenes .....	50
<b>Figura 34</b> Botones que envían órdenes a Rubus.....	50
<b>Figura 35</b> Programación botón Bases.....	51
<b>Figura 36</b> Programación de las ordenes de Rubus.....	51
<b>Figura 37</b> Continuación Programación de las ordenes de Rubus .....	52

<b>Figura 38</b> Emparejamiento de dispositivos.....	52
<b>Figura 39</b> Librerías utilizadas en Arduino .....	53
<b>Figura 40</b> Sintaxis objeto motors.....	54
<b>Figura 41</b> Pines de comunicación.....	54
<b>Figura 42</b> Pines de sensorización.....	54
<b>Figura 43</b> Sintaxis principal.....	55
<b>Figura 44</b> Configuración de entradas y salidas.....	55
<b>Figura 45</b> Metodo de comunicación serial .....	56
<b>Figura 46</b> Procesamiento de datos .....	56
<b>Figura 47</b> Diseño Bottom-Up aplicado en Rubus.....	57
<b>Figura 48</b> Diseño Rubus mediante circuitos .....	58
<b>Figura 49</b> Superficie de trabajo Rubus .....	68
<b>Figura 50</b> gráfico pregunta 1 .....	70
<b>Figura 51</b> Resultados pregunta 1 previo Rubus.....	70
<b>Figura 52</b> gráfico pregunta 2 .....	71
<b>Figura 53</b> Resultados pregunta 2 previo Rubus.....	71
<b>Figura 54</b> gráfico pregunta 3 .....	72
<b>Figura 55</b> Resultados pregunta 3 previo Rubus.....	72
<b>Figura 56</b> Resultados pregunta 4 previo Rubus.....	73
<b>Figura 57</b> Resultados pregunta 5 previo Rubus.....	74
<b>Figura 58</b> Capacitación previa a los niños.....	75
<b>Figura 59</b> Presentación por grupos la App de Rubus .....	76
<b>Figura 60</b> Niños utilizando a Rubus .....	76
<b>Figura 61</b> Resultados de la interacción entre los niños y el prototipo .....	77
<b>Figura 62</b> Gráfico pregunta 1.....	78
<b>Figura 63</b> Resultados pregunta 1 posterior a Rubus .....	78
<b>Figura 64</b> Gráfico pregunta 2.....	79
<b>Figura 65</b> Resultados pregunta 2 posterior a Rubus .....	79
<b>Figura 66</b> Gráfico pregunta 3.....	80
<b>Figura 67</b> Resultados pregunta 3 posterior a Rubus .....	80
<b>Figura 68</b> Resultados pregunta 4 posterior a Rubus .....	81
<b>Figura 69</b> Resultados pregunta 5 posterior a Rubus .....	82

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Evaluación de características de framework Android Studio .....	25
<b>Tabla 2.</b> Evaluación de características de framework eclipse.....	26
<b>Tabla 3.</b> Evaluación de características de framework App Inventor .....	27
<b>Tabla 4.</b> Evaluación de características de framework para hardware Microcontrolador PIC .....	29
<b>Tabla 5.</b> Características de Arduino.....	30
<b>Tabla 6.</b> Evaluación de características de framework para hardware arduino .....	30
<b>Tabla 7.</b> Características de Pololu Baby Orangután .....	31
<b>Tabla 8.</b> Evaluación de características de framework para hardware Pololu Baby Orangután .....	32
<b>Tabla 9.</b> Materiales Electrónicos .....	59
<b>Tabla 10.</b> Partes Case Rubus .....	60
<b>Tabla 11.</b> Costos en la elaboración de Rubus. ....	62
<b>Tabla 12.</b> Procedimiento Trabajo Rubus .....	64
<b>Tabla 13</b> Fotografías de presentación Rubus con los clientes Capacitación Previa .....	109
<b>Tabla 14</b> Fotografías de presentación Rubus con los clientes demostración y ejercicios.....	110
<b>Tabla 15</b> Fotografías de presentación Rubus con los clientes culminación de la presentación .....	111

## **RESUMEN EJECUTIVO**

El presente trabajo tiene como objetivo construir un prototipo electrónico enfocado en el aprendizaje de lógica de programación en niños de edades comprendidas entre 4 y 10 años de edad. Para ello, el proyecto final consta de: un robot móvil, un entorno de desplazamiento y una aplicación móvil de control. La aplicación fue desarrollada en Android Studio con lenguaje de programación Java bajo la metodología de desarrollo por prototipos, mientras que el robot denominado Rubus a través del framework Arduino en lenguaje C utilizando el microcontrolador Pololu Baby Orangután bajo la metodología Bottom up.

El funcionamiento de Rubus se basa en la recepción de instrucciones de movimiento seleccionadas por cada niño en la aplicación Android, las cuales son enviadas por medio tecnología inalámbrica bluetooth hacia el robot, el mismo que las ejecuta de manera instantánea movilizándose dentro de una alfombra de desplazamiento realizada especialmente para este fin, el objetivo del robot es llegar a la cuadrícula morada evitando por completo el atravesar las cuadrículas rojas y los bordes celestes de la alfombra mencionada, puesto que si esto ocurre el robot detiene su funcionamiento.

## **ABSTRACT**

The objective of this work is to build an electronic prototype focused on learning programming logic in children between 4 and 10 years of age. To do this, the final project consists of: a mobile robot, a moving environment and a mobile control application. The application was developed in Android Studio with Java programming language under the prototype development methodology, while the robot named Rubus through the Arduino framework in C language using the Pololu Baby Orangutan microcontroller under Bottom up methodology.

The operation of Rubus is based on the reception of movement instructions selected by each child in the Android application, which are sent by means of wireless bluetooth technology to the robot, the same that executes them in an instantaneous way, moving within a displacement carpet made especially for this purpose, the goal of the robot is to reach the purple grid by completely avoiding crossing the red grids and the celestial edges of the aforementioned carpet, since if this happens the robot stops its operation

## GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

**Arduino.** - Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar.

**App.** - Aplicación de software que se instala en dispositivos móviles o Tablet para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento, a diferencia de una web App que no es instalable.

**Framework.** - Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

**Hardware.** - Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

**Lógica.** - Método o razonamiento en el que las ideas o la sucesión de los hechos se manifiestan o se desarrollan de forma coherente y sin que haya contradicciones entre ellas.

**Metodología.** – Grupo de mecanismos o procedimientos racionales, empleados para el logro de un objetivo, o serie de objetivos que dirige una investigación científica.

**Programación.** - La programación informática o programación algorítmica, acortada como programación, es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas de computadora.

**Software.** -Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.



## **INTRODUCCIÓN**

El siguiente proyecto de investigación denominado: “PROTOTIPO ENFOCADO AL APRENDIZAJE DE LÓGICA DE PROGRAMACIÓN EN NIÑOS DE EDADES COMPRENDIDAS ENTRE 4 A 10 AÑOS” se encuentra dividido en los siguientes capítulos:

CAPÍTULO I. “EL PROBLEMA”, se identifica el problema a investigar, se plantea la justificación del porqué se realiza la investigación, además se define los objetivos necesarios que estarán inmersos en la consecución de la investigación del proyecto.

CAPÍTULO II. “MARCO TEÓRICO”, se indaga la información necesaria que sustenta el presente proyecto, también se conocen investigaciones previas, las mismas que ayudan de soporte al proyecto de investigación, además se presenta la propuesta de solución ante el problema definido en el capítulo anterior.

CAPÍTULO III. “METODOLOGÍA”, incluye la metodología que se utilizará definiendo las técnicas e instrumentos que serán de ayuda para la recolección de la información necesaria, para lo cual también se definen las etapas que ayudan a la consecución de la presente investigación.

CAPÍTULO IV. “DESARROLLO DE LA PROPUESTA”, se detalla de una manera clara el desarrollo de las etapas propuestas en el capítulo anterior, se comparan diferentes frameworks, se definen las metodologías para desarrollo de la aplicación móvil, programación de Rubus y de la construcción del prototipo que cumplan con los requerimientos del proyecto.

CAPÍTULO V. “CONCLUSIONES Y RECOMENDACIONES”, se plantean conclusiones a las que llega el investigador luego de la culminación del proyecto, también se describe las recomendaciones que pueden servir de ayuda a futuras investigaciones.

# **CAPITULO 1**

## **EL PROBLEMA**

### **1.1. Tema**

Prototipo enfocado al aprendizaje de lógica de programación en niños de edades comprendidas entre 4 a 10 años.

### **1.2. Planteamiento del problema**

Los expertos coinciden que los niños desde tempranas edades se encuentran aptos para aprender las bases de lógica de programación con lo cual estos aprenden el desglosamiento de problemas grandes en problemas más simples [1].

Finlandia e Israel son los dos países que más han avanzado en la enseñanza de programación en escuelas. Ambos sistemas educativos empezaron a introducir la programación en el currículo escolar en la pasada década. En Corea del Sur las rutas están repletas de propuestas para adquirir al menos conocimientos básicos, mientras que Nueva Zelanda y Grecia ya han comenzado a integrarlas en programas piloto [2].

En Estados Unidos hay una fuerte corriente de opinión organizándose para influir en los programas educativos a favor de la programación. Magnates de la industria como Bill Gates, Mark Zuckerberg, Arianna Huffington, Gabe Newell o Jack Dorsey apoyan la asociación sin ánimo de lucro Code.org que persigue la concienciación de alumnos y profesores sobre las posibilidades que ofrecería enseñar programación en la escuela [2].

Los niños deberían tener la oportunidad de aprender ciencias de la computación, empezando en la escuela. Enseñamos física básica a cada niño, no con el objetivo principal de educar físicos si no porque todos ellos viven en un mundo gobernado por sistemas físicos. De la misma manera, todos los niños deberían aprender un poco de informática desde temprana edad porque van a vivir en un mundo en el que la computación está en todas partes [3].

Los niños deben tener la oportunidad de familiarizarse con ciencias computacionales desde temprana edad debido a que el mundo que los rodea está encaminado a ser netamente tecnológico; mas los países tercermundistas no han enfocado su atención en este ámbito por diferentes factores entre los que destacan: el ámbito económico, educativo y social.

En Ecuador ésta realidad no cambia, las unidades educativas no contemplan programación dentro de su programa académico, siendo hoy en día un ámbito sumamente demandante en la sociedad actual, por el continuo avance tecnológico, a esto se suma el hecho de que la mayoría de docentes no tienen una preparación adecuada respecto a este ámbito y desconocen los beneficios que ello conlleva.

En la ciudad de Ambato no existe un mayor enfoque en el desarrollo de este ámbito a más de matemática, lo cual permite crear una base en el pensamiento lógico de los niños, pero presenta una falencia: la escasa motivación que se brinda al impartir este conocimiento, lo que impulsa un interés minoritario en este campo a través del tiempo.

El no contar con métodos de enfoque hacia el desarrollo lógico desde edades tempranas limita sobremanera el pensamiento y desenvolvimiento futuro respecto al aprendizaje de ciencias tecnológicas, por lo que para poder incentivar el interés en matemática y por concatenación en programación es menester captar la atención de los niños mediante entornos interactivos mostrando lo que a futuro se puede conseguir mediante el aprendizaje de ciencias exactas, por lo cual dentro de nuestro medio es imperativo buscar la manera de lograr que los estudiantes tengan la predisposición de participar en eventos que desarrollen su intelecto, beneficiándolos para poder manejar los distintos problemas que se les presente usando maneras innovadoras de enseñanza y añadiendo módulos formativos a sus respectivas clases en los establecimientos educativos de la ciudad; por lo que finalmente se deduce que centrándose en la ciudad de

Ambato, los niños no disponen de un método eficiente que impulse el proceso enseñanza - aprendizaje de programación.

### **1.3. Delimitación**

**Área Académica:** Software.

**Línea de Investigación:** Desarrollo de software.

**Sublínea de Investigación:** Aplicaciones para dispositivos móviles.

**Delimitación espacial:** Las pruebas correspondientes a la interacción y funcionamiento del prototipo con niños fueron realizadas en Unidad Educativa Luis A. Martínez de la ciudad de Ambato.

**Delimitación temporal:** La presente investigación se desarrolló a partir del 26 de septiembre del 2017 hasta el 20 de febrero de 2018.

### **1.4. Justificación**

La adquisición de una lógica base de programación a edades tempranas permite incrementar el desarrollo cognitivo de manera creciente a través del tiempo, lo cual representa una ventaja considerable puesto que incrementa las opciones de solución de problemas sin importar su nivel de complejidad o el entorno en el que estén centrados.

Lógica de programación es un conocimiento sumamente necesario, ya que puede ser aplicado en todo aspecto de la vida diaria, pues supone: analizar un problema, dividirlo, generar alternativas de solución y seleccionar el método más eficaz para resolver dicho problema, todo esto, desarrollado de manera lógica y secuencial para llegar a un objetivo claro.

Los niños tienen un potencial de imaginación y asimilación al máximo, por lo que es necesario explotar estas cualidades a través de medios adecuados que incrementen el conocimiento y a su vez representen alternativas didácticas y llamativas a los juegos tradicionales para captar así la atención de los niños. Al enseñar conceptos de programación en edades tempranas se asegura una asimilación prácticamente total, lo que no siempre sucede en caso de que se

imparta el mismo conocimiento a personas de mayor edad, en donde además, la metodología de enseñanza está limitada al profesor y al grado de interés de la persona. Es por ello que el método de enseñanza-aprendizaje en edades tempranas supone un sinnúmero de ventajas; es necesario encontrar varias alternativas para que los niños tengan algún tipo de acercamiento con el mundo de la tecnología más allá de que únicamente sean miembros pasivos de ella, y puedan utilizar su creatividad e imaginación para adquirir conocimiento mientras se divierten.

Al adquirir lógica de programación las alternativas que pueden plantear los niños no solo son eficientes sino también ingeniosas y creativas.

El presente proyecto es factible puesto que se centra en niños de edades comprendidas entre 4 y 10 años, para potencializar sus habilidades y capacidades viso-espaciales, razonamiento perceptivo, memoria de trabajo y velocidad de procesamiento mediante la aplicación de nuevas tecnologías en este caso, la utilización del robot Rubus.

## **1.5. Objetivos**

### **1.5.1. Objetivo general**

- Diseñar un prototipo que facilite el aprendizaje de lógica de programación en niños entre 4 a 10 años de la ciudad de Ambato.

### **1.5.2. Objetivos específicos**

- Analizar juguetes con fines educativos en el área de informática y afines en las principales ciudades del país.
- Seleccionar una metodología adecuada para el desarrollo de la aplicación móvil en un framework que se ajuste a los requerimientos del programador.
- Diseñar, construir e implementar un prototipo electrónico aplicando una metodología que permita dividir en módulos funcionales la estructura total del sistema.
- Realizar pruebas de funcionalidad del sistema final con niños cuya edad se encuentra en el rango de 4 a 10 años en la Unidad Educativa Luis A. Martínez.

## **CAPITULO 2**

### **MARCO TEÓRICO**

#### **2.1. Antecedentes investigativos**

Que un niño aprenda a desglosar un problema grande en pasos pequeños, y luego años después conforme vaya aprendiendo le sirva para tener una base sobre la que aprender programación de una manera muy eficiente, se puede empezar a hacer a los 6 u 8 años [4].

La programación facilita la resolución de problemas complejos de una forma sistemática, por niveles de abstracción, partiendo desde lo general hacia lo concreto; por lo tanto, puede ser usado en cualquier otra rama que implique resolver problemas.

Centrados en los conceptos mencionados, la empresa pública Yachay propuso una consolidación del ecosistema de desarrollo investigativo en el país creando el programa “Retos Yachay 2013” cuyo fin principal era obtener ofertas de desarrollo tecnológico en el Campus Party 2013; en donde tanto miembros del club de robótica de Yachay asociados con Clear Minds Consultores presentaron un juguete programable a través de un ordenador que pretendía enseñar programación a niños, el cual que tiene una interfaz de software desarrollada en su totalidad por ecuatorianos, y un 90% de componentes electrónicos fabricados también en Ecuador “Teebot, combina diversión, tecnología y conocimiento, es fácil de usar y amigable con los niños desde los 4 años en adelante. Ellos/as aprenden fundamentos de robótica, electrónica y programación mientras juegan con Teebot; al jugar con Teebot, programarlo y

controlarlo el/la niñ@ se empodera de tal manera que se siente capaz de resolver cualquier reto” [5].

El paper “Novice Programmers Talking about Projects: What Automated Text Analysis Reveals about Online Scratch Users’ Comments” hace especial referencia a Scratch, la cual es una aplicación web utilizada para programar pequeños y divertidos juegos visualmente con bloques que se colocan uno sobre otro, fue desarrollada por el Grupo Lifelong Kindergarten del MIT Media Lab y su utilización parte desde el año 2007. [6].

Danny Yaroslavski realizó el videojuego educativo Lightbot para el aprendizaje de conceptos de programación de software. Este videojuego se ha jugado 7 millones de veces, y es muy valorado en iTunes y Google Play Store, está disponible como un juego flash en línea, y una aplicación para teléfonos móviles Android y iOS. Lightbot se ha construido con Flash y OpenFL. El objetivo de Lightbot es mover a un pequeño robot para navegar por un laberinto usando diferentes comandos que se basan en conceptos de programación básicos y así poder encender unas luces y avanzar al siguiente nivel [7].

El presente proyecto toma como base el videojuego desarrollado como proyecto de tesis por Hussein Rahman en 2016 ,titulado: “VIDEOJUEGO EDUCATIVO EN 3D PARA DISPOSITIVOS MÓVILES ANDROID, ENFOCADO AL APRENDIZAJE DE LA LÓGICA DE PROGRAMACIÓN PARA USUARIOS ENTRE LOS 5 A 18 AÑOS DE EDAD” realizado en la Universidad Técnica de Ambato y corresponde a un videojuego multiplataforma enfocado en enseñar lógica de programación a niños y jóvenes [8].

## **2.2. Fundamentación Teórica**

### **2.2.1. Lenguaje de programación**

Es un lenguaje formal diseñado para ser utilizados por máquinas tales como ordenadores , dispositivos móviles, microcontroladores entre otros para que estos puedan realizar procesos controlando aspectos físicos y lógicos dentro de su estructura [9].

### 2.2.2. Lógica de programación

Se define como la capacidad del ser humano para generar instrucciones ordenadas y coherentes es decir algoritmos. En la actualidad los algoritmos se clasifican en:

**Los algoritmos informales.** - Son aquellos que no se realizan para ser ejecutados por una computadora, sino se diseñan para ser ejecutados por el ser humano. Todos los días se ejecutan algoritmos informales para realizar actividades: al prepararse para ir al trabajo, al vestirse, al comer, al regresar a casa, etc. [10].

**Los algoritmos computacionales.** - Son aquellos que son ejecutados por una computadora. Se aprovecha la velocidad de procesamiento del ordenador para obtener un resultado mucho más confiable, por ejemplo, el cálculo de una raíz cuadrada. [10].

Analizando desde muchos ángulos del pensamiento humano y teniendo en cuenta los conceptos de algoritmo informal y algoritmo computacional se llegó a la conclusión de que dicho pensamiento se mueve entre tres estructuras básicas: Secuencias, Decisiones, Ciclos [8].

### 2.2.3. Secuencias

Para escribir una secuencia de ordenes o acciones todo lo que tiene que hacer es colocar una nueva orden o una nueva acción después una última. De esta manera se entiende la secuencialidad y la cordialidad en la ejecución de esas acciones [11].

El siguiente algoritmo permite abrir una puerta, asumiendo que la puerta se encuentra en línea visible. El algoritmo quedaría como se muestra en la Figura 1.

1	Algoritmo para abrir la puerta
2	Inicio
3	Ubicar la puerta que se quiere abrir
4	Moverse de la posición actual
5	Avanzar hacia la puerta
6	Llegar hasta tener la puerta al alcance de la mano
7	Abrir la puerta

**Figura 1.** Algoritmo de secuencias

Fuente: [11]



Se puede apreciar cada acción en su orden lógico y para poder ejecutar el algoritmo con éxito se lo debe realizar al pie de la letra [11].

#### 2.2.4. Decisiones

Este tipo de sentencia es utilizado para controlar el flujo de la información evaluando una condición y seleccionando la mejor instrucción a seguir[12]. A continuación, en la Figura 2, se muestra el mismo algoritmo de abrir una puerta, pero esta vez se añadieron líneas de código que permitirán crear la estructura de Decisiones, así:

1	Algoritmo para abrir una puerta
2	Inicio
3	Ubicar la puerta que queremos abrir
4	Si nos encontramos recostados
5	Levantarnos del lugar
6	Dirigirnos hacia la puerta
7	Caso contrario
8	Avanzar hacia la puerta
9	Llegar hasta tener la puerta al alcance de la mano
10	De estar cerrado abrir la puerta

**Figura 2.** Algoritmo de decisiones  
Fuente: [12]

Después de cada Si condicional se evalúa una acción de ser cierta se realiza una tarea caso contrario se realiza otra [12].

#### 2.2.5. Ciclos

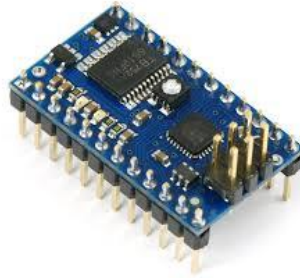
Instrucción que se ejecuta la cantidad de veces que sea necesaria hasta que una condición asignada a dicho bucle deje de cumplirse.[13]. En el evento que realiza el ciclo se deben programar los procesos que se efectúen durante el ciclo. En la Figura 3 se muestra un algoritmo para solucionar el chillido en una puerta.

1	Algoritmo solucionar el chillido en una puerta.
2	Inicio
3	Ubicar en qué lugar de la puerta se encuentra el chillido
4	Mientras la puerta siga con el chillido
5	Aplicar aceite
6	Fin Mientras
7	Fin

**Figura 3.** Algoritmo de ciclos.  
Fuente: [13]

### 2.2.6. Microcontroladores

Dicho así a un circuito integrado programable, capaz de producir los comandos enviados a su memoria. Está compuesto de varios bloques funcionales, que cumplen una tarea específica [14] . En la Figura 4 se muestra un ejemplo de microcontrolador.



**Figura 4.** Microcontroladores  
Fuente: [14]

Al momento de la fabricación del microcontrolador, no contiene datos en la memoria ROM<sup>1</sup>. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM<sup>2</sup> o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores [14].

### 2.2.7. Prototipos

Es una representación limitada de un producto, permite a las partes probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad. Un prototipo puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software [15].

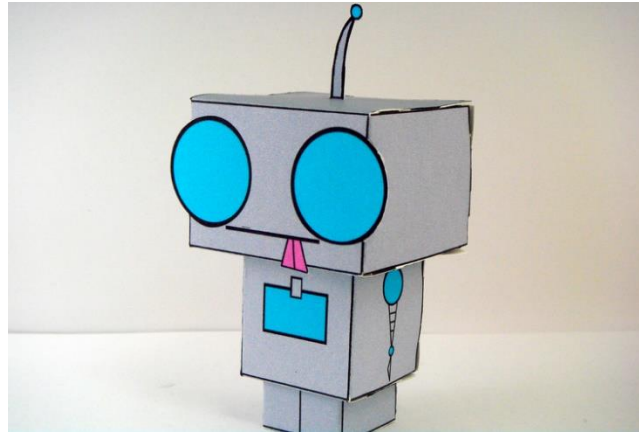
Porque son útiles para comunicar, discutir y definir ideas entre los diseñadores y las partes responsables. Los prototipos apoyan la evaluación de productos, clarifican requisitos de usuario y definen alternativas. Utilizan materiales distintos al del producto final, son baratos, simples y fáciles de producir. Son particularmente útiles en las fases iniciales del desarrollo,

---

<sup>1</sup> ROM: Read Only Memory

<sup>2</sup> EEPROM: Electrically Erasable Programmable Read-Only Memory

durante el diseño conceptual. Son aquellos que se parecen al producto final y utiliza sus mismos materiales [15]. En la Figura 5 tenemos un ejemplo de prototipo.



**Figura 5.** Ejemplo de prototipo  
Fuente: [15]

### **2.2.8. Metodologías tradicionales de desarrollo de software**

Las metodologías tradicionales son denominadas, como metodologías pesadas, centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. [16].

**RUP (RATIONAL UNIFIED PROCESS) y MSF (MICROSOFT SOLUTION FRAMEWORK)** entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

#### **RUP (RATIONAL UNIFIED PROCESS)**

Es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para

satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación [16].

### **MSF (MICROSOFT SOLUTION FRAMEWORK)**

Es un enfoque personalizable para entregar con éxito soluciones tecnológicas de manera más rápida, con menos recursos humanos y menos riesgos, pero con resultados de más calidad. MSF ayuda a los equipos a enfrentarse directamente a las causas más habituales de fracaso de los proyectos tecnológicos y mejorar así las tasas de éxito, la calidad de las soluciones y el impacto comercial [16].

#### **2.2.9. Metodologías Ágiles**

El desarrollo ágil de software se refiere a métodos de Ingeniería del Software basados en el desarrollo iterativo e incremental, estas metodologías son imprescindibles en un mundo en el que nos exponemos a cambios recurrentemente. Siempre hay que tener en cuenta como programadores que lo que es la última tendencia hoy puede que no exista mañana y por esto existe la metodología ágil donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios [17].

### **SCRUM**

Proceso de la Metodología Ágil que se usa para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa. Entre las ventajas se encuentran la productividad, calidad y que se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes estén unidos, comunicados y que el cliente vaya viendo los avances.

### **XP (Extreme Programming)**

La “programación extrema” es un proceso de la Metodología Ágil que se aplica en equipos con muy pocos programadores quienes llevan muy pocos procesos en paralelo. Consiste entonces en diseñar, implementar y programar lo más rápido posible, hasta en casos se recomienda saltar la documentación y los procedimientos tradicionales. Se fundamenta en la capacidad del equipo para comunicarse entre sí y las ganas de aprender de los errores propios inherentes en un programador [17].

### 2.2.10. Tipos de Modelo de Prototipos

**Modelo de Prototipos rápido:** Metodología de diseño que desarrolla rápidamente nuevos diseños, los evalúa y los actualiza cuando el nuevo modelo está listo [18].

**Modelo de Prototipos reutilizable:** También conocido como "Evolutionary Prototyping"; no se pierde el esfuerzo efectuado en la construcción del prototipo pues sus partes o el conjunto pueden ser utilizados para construir el producto real. Mayormente es utilizado en el desarrollo de software, si bien determinados productos de hardware pueden hacer uso del prototipo como la base del diseño de moldes en la fabricación con plásticos o en el diseño de carrocerías de automóviles [18].

**Modelo de Prototipos Modular:** También conocido como Prototipado Incremental (Incremental prototyping); se añaden nuevos elementos sobre el prototipo a medida que el ciclo de diseño progresa.

**Modelo de Prototipos de Baja-fidelidad:** El prototipo se implementa con papel y lápiz, emulando la función del producto real sin mostrar el aspecto real del mismo. Resulta muy útil para realizar tests baratos.

**Modelo de Prototipos de Alta-fidelidad:** El prototipo se implementa de la forma más cercana posible al diseño real en términos de aspecto, impresiones, interacción y tiempo.

### 2.2.11. Modelos para la construcción de hardware

Las metodologías de diseño de hardware denominadas Top-Down, basadas en la utilización de lenguajes de descripción de hardware, han posibilitado la reducción de los costos en la fabricación de circuitos integrados. Esta reducción se debe a la posibilidad de describir y verificar el funcionamiento de un circuito mediante la simulación del mismo, sin necesidad de implementar un prototipo físicamente [19].

En un principio las herramientas CAD, brindaban únicamente la posibilidad de trazar los dibujos referentes al diseño. El diseñador realizaba la descripción sobre un papel utilizando componentes básicos y trasladaba el diseño a la computadora para obtener una representación más ordenada.

Se sigue los pasos necesarios para la construcción de hardware

- La verificación de la documentación necesaria y requerida.
- El diseño del prototipo
- La simulación del prototipo
- El ensamblaje del prototipo
- La verificación del prototipo creado y arreglos en el caso presentarse [19].

### **2.3. Propuesta de solución**

El diseño y desarrollo de un prototipo que facilite el aprendizaje de la lógica de programación en niños ampliará la manera en la cual ellos adquieren conocimiento informático, además que será una manera actual e interactiva de impartir conocimiento técnico.

## **CAPITULO 3**

### **METODOLOGÍA**

#### **3.1. Modalidad de la investigación**

##### **Modalidad Bibliográfica**

La presente investigación es basada en información que se puede hallar en libros técnicos, informes, artículos, los cuales proporcionarán información relevante para llevar a cabo la presente.

##### **Modalidad Aplicada**

Al ser aplicados conocimientos adquiridos a lo largo del estudio académico en lo referente a los módulos de desarrollo de software, lógica matemática, robótica, hardware y redes .El presente trabajo de investigación es proyecto de Investigación Aplicada (I) porque se aplicará definiciones y metodologías ya investigadas no se tendrá que desarrollar metodologías nuevas para la consecución del proyecto.

##### **Modalidad Campo**

Para la obtención de información necesaria se acudirá a la Unidad Educativa Luis A. Martínez.

#### **3.2. Población y muestra**

La población total tomada en cuenta para el presente proyecto son niños de 4 a 10 años de la ciudad de Ambato y la muestra serán estudiantes de la Unidad Educativa Luis A. Martínez.

### **3.3. Recolección de la información**

Para recolectar la información necesaria para el desarrollo de la investigación se utilizará la técnica Encuesta a través de la hoja de encuesta, además se aplicará la técnica de investigación documental de tipo informativa para de esta manera obtener información relevante de fuentes confiables.

### **3.4. Procesamiento y análisis de datos**

Para el procesamiento y análisis de datos se ejecutarán las siguientes actividades:

- Recolección de información utilizando investigación documental.
- Análisis y tabulación de la información obtenida.
- Interpretación mediante gráficos y cuadros informativos.

### **3.5. Desarrollo del Proyecto**

Se presentarán a continuación las actividades a realizar para el cumplimiento de los objetivos específicos, para así poder completar a cabalidad el objetivo general propuesto.

#### **1. Análisis del funcionamiento de prototipos programables con fines educativos en el área de informática dentro del país.**

- Investigación para encontrar diversos prototipos programables en las principales ciudades de Ecuador.
- Documentación de características físicas y la lógica utilizadas por estos prototipos para su funcionamiento.

#### **2. Selección de framework para desarrollo del prototipo y la aplicación.**

- Elaboración y aplicación de encuestas.
- Selección de un framework para el desarrollo de la aplicación móvil.
- Selección de hardware para el desarrollo de la propuesta.
- Selección de una metodología para el desarrollo de la propuesta.



**3. Desarrollo de la propuesta (aplicación y prototipo).**

- Desarrollo de la aplicación móvil para el control del prototipo utilizando la metodología seleccionada en el punto anterior.
- Ensamblaje y programación del prototipo.
- Fusión del proyecto para comprobar funcionamiento en conjunto.
- Análisis del entorno donde funciona el prototipo.

**4. Implementación del prototipo que facilita la enseñanza de programación lógica en niños.**

- Evaluación a niños previa a capacitación en lógica de programación.
- Resultados de evaluación previa a capacitación con Rubus.
- Capacitación a niños en lógica de programación utilizando a Rubus como herramienta de apoyo.
- Evaluación posterior a capacitación con Rubus.
- Resultados de evaluación posterior a capacitación con Rubus.

## **CAPITULO 4**

### **DESARROLLO DE LA PROPUESTA**

#### **4.1. Análisis del funcionamiento de prototipos programables con fines educativos en el área de informática dentro del país**

Para el desarrollo de la propuesta se revisó varios prototipos diseñados y construidos en el país, los cuales han proporcionado una ayuda en el proceso enseñanza aprendizaje de niños entre 4 y 10 años de edad, a través del desarrollo de sus habilidades lógicas y analíticas.

##### **4.1.1. Investigación para encontrar diversos prototipos programables en las principales ciudades del Ecuador.**

Los escasos avances tecnológicos de los últimos años en Ecuador han impedido el desarrollo de este tipo de prototipos, este es el principal inconveniente a más de los recursos económicos limitados que se destinan a este campo, por ello no existe en el ambiente educativo una rama enfocada en la utilización de prototipos electrónicos con fines de enseñanza.

El único robot documentado existente con estas características es Teebot, diseñado conjuntamente por las empresas: Clearmind y EGM, las cuales desarrollaron el software en Java y hardware respectivo, aunque a pesar de ello el prototipo fue publicado y distribuido por la Empresa Pública Yachay. Permite a niños de cinco años en adelante tener conocimiento sobre programación, robótica y electrónica a través de una interfaz en el ordenador por cable a todos sus usuarios, desarrollando habilidades y destrezas enfocadas en informática básica; en sus inicios Teebot tenía un costo de \$500 y con el pasar del tiempo llegó a tener un precio

de \$200. En la actualidad ya no se fabrican más prototipos de este robot, por lo que puede considerarse como un proyecto culminado sin mayor impacto.

Este robot, nació a partir del programa “Retos Yachay 2013”, tuvo como razón principal difundir retos y obtener ofertas de desarrollo por parte del talento de los jóvenes amantes de la tecnología en el Campus Party 2013, feria que se organiza año a año en el país. El sistema de Teebot permite a niños entre 4 y 12 años enviar órdenes a un robot a través de un software específico [20].

#### **4.1.2. Documentación de características físicas y lógicas utilizadas por estos prototipos para su funcionamiento.**

Este producto tiene una interfaz de software desarrollado por ecuatorianos y su parte electrónica tiene un 90% de componentes con fabricación nacional, es por eso que, Santiago Mosquera, gerente general de Clear Minds Consultores, señaló que Teeboot, es más que un robot “es un juguete que va a cambiar la forma en que las niñas y niños de Ecuador y Latinoamérica aprendan con la tecnología” y desarrollen sus habilidades explotando así sus conocimientos en el área [21].

Los niños pueden desarrollar capacidades como: creatividad, adaptabilidad, comunicación, trabajo en equipo y liderazgo. El prototipo presenta un hardware que contiene una estructura adecuada para 4 motores con sus respectivas ruedas que permiten controlar su movilidad y además un microcontrolador el cual es el encargado de controlar todas las acciones de Teebot [20].

Este prototipo adquiere los diferentes comandos desde un ordenador que son comunicados a través de un medio guiado para controlar los diferentes sensores y actuadores del prototipo (motores, leds).

“Para los niños es un juguete, para nosotros constituye la posibilidad de que ellos se familiaricen con la robótica y la programación y visualicen el hecho de que pueden escoger carreras en sistemas. Carreras que les permitan seguir robótica, ingeniería en sistemas o que

les permitan utilizar ingeniería electrónica también” [21]. En la Figura 6 podemos apreciar a Teebot.



**Figura 6.** Robot Teebot 100% Ecuatoriano  
Fuente: [21]

Teebot se convierte en un juguete para niños, se considera que: “Es un producto totalmente escalable. Los niños en la versión inicial van a poder imprimir los accesorios en impresoras 3D e ir agregándolos a Teebot para darle una forma acorde con su creatividad”, de esta manera puede también interactuar con otro tipo de software como el de una impresora 3D [21].

Al ser un modelo inicial, con el pasar del tiempo, la forma y diseño del robot variará y se hará más compleja, explicó Santiago Mosquera. “Finalmente, sobre proyectos futuros, se están buscando vías para que la tecnología permita mantener este vínculo social y que no solo sean consumidores de energía. Con este tipo de herramienta podemos ser creadores de soluciones fomentando la creatividad y el trabajo en equipo” [21].

Teebot aplica una lógica intuitiva y atractiva para niños, utiliza un software creado por Clear Minds desarrollado en java; “Lo que nosotros hacemos en el software es ir poniendo sentencias en el computador, mediante las cuales decidimos qué queremos que haga el robot. Por ejemplo, que vaya hacia adelante, que pite, o encienda las luces. Estas órdenes las da el niño simplemente arrastrando unos íconos de colores, luego las transfiere al robot e inmediatamente podrán ver ejecutar por el robot lo que ellos ordenaron en el computador” [22].

EGM se encargó de la creación del hardware “EGM se ha tardado dos años en madurar la placa en la cual el hardware está totalmente sólido y te permite hacer varias cosas como, por

ejemplo, poner sensores, dispositivos bluetooth, luces y sonidos, de manera que sea mucho más atractivo para los niños” [22].

## **4.2. Selección del framework para el desarrollo del prototipo y la aplicación**

Para la selección del framework de desarrollo primero se aplica una encuesta a padres de familia para así determinar en qué entorno funcionara la aplicación que se desea crear para controlar al prototipo.

### **4.2.1. Elaboración y aplicación de encuestas**

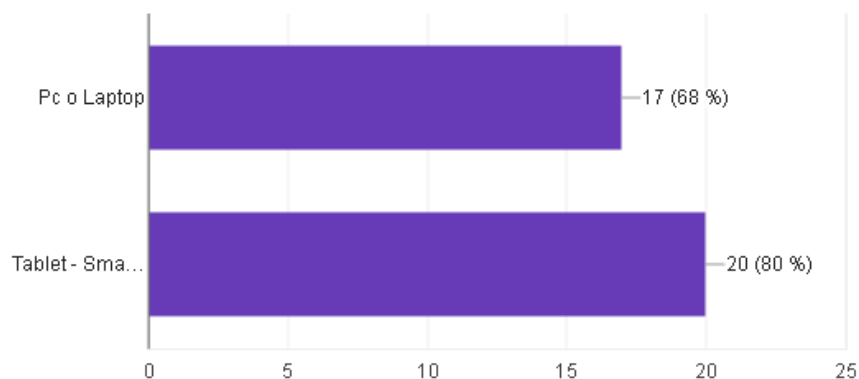
La encuesta fue realizada a 25 padres de familia del curso asignado para la investigación, el cual fue el sexto año de educación básica paralelo “B” de la Unidad Educativa Luis A. Martínez, principalmente por la edad que tiene los niños la cual se encuentra dentro del rango de 9 a 10 años y recibieron capacitación previa acerca del objetivo del proyecto.

### **Procesamiento y análisis de las encuestas**

El resultado de la encuesta aplicada a los representantes de los estudiantes de la Unidad Educativa Luis A. Martínez se representa en las siguientes figuras y se muestra un análisis e interpretación de cada una de ellas.

## Pregunta 1

¿En qué dispositivo le gustaría que su hijo aprendiera a programar?



**Figura 7.** Tipo de dispositivos en donde les gustaría que aprendan a programar  
Elaborado por: Christian Caicedo

## Análisis

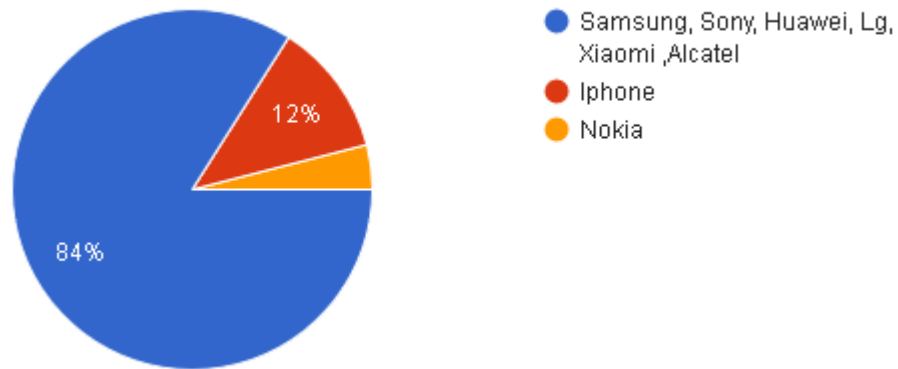
El instrumento de evaluación de prototipo Rubus aplicado a los padres de familia de los alumnos del sexto año de la Unidad Educativa Luis A. Martínez, muestra que el 68% de la población encuestada opina que el dispositivo en que los niños pueden aprender a programar, mas el 80% considera que el dispositivo puede ser una tablet o un Smartphone, los porcentajes de este resultado no son excluyentes entre sí, sino que la pregunta es abierta a una selección conjunta entre las alternativas.

## Interpretación

El uso de tecnología tiene una fuerte tendencia dirigida hacia el uso de tablets y smartphones, a pesar de que el uso del computador no se ve totalmente excluido, la enseñanza de lógica de programación no es indiferente ante esto y por ello los resultados con mayor porcentaje de aceptabilidad se ven reflejados en el correspondiente a tablets y teléfonos inteligentes.

## Pregunta 2

¿El dispositivo móvil que usted posee usted?



**Figura 8.** Tipo de dispositivo que poseen los padres de familia  
Elaborado por: Christian Caicedo

## Análisis

Según el instrumento de evaluación del prototipo Rubus aplicado a los padres de familia de los alumnos del sexto año de la Unidad Educativa Luis A. Martínez, el 84% de padres de familia utiliza dispositivos móviles de marca: Samsung, Sony, Huawei, Lg, Xiaomi o Alcatel, tan solo el 12% de la población encuestada utiliza dispositivos iPhone, y un 4% utilizan Nokia.

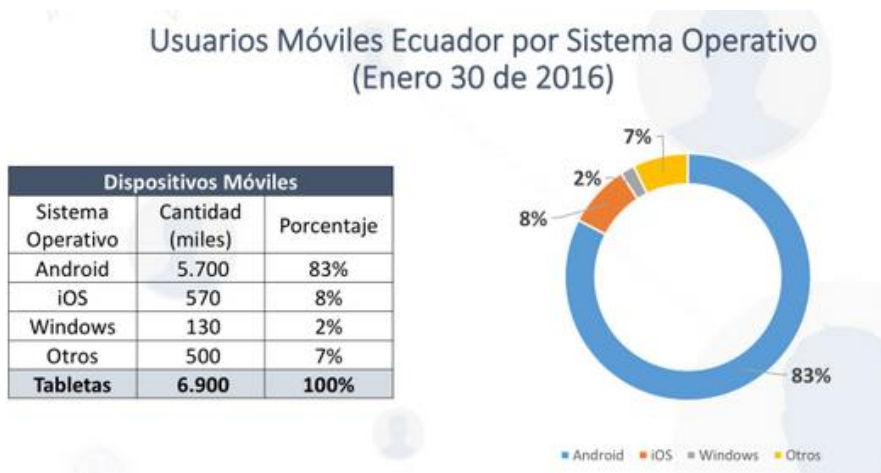
## Interpretación

La encuesta refleja que todos los padres de familia cuentan con un dispositivo móvil, la tendencia general se encamina hacia dispositivos con sistema operativo Android, debido al costo relativamente menor de ellos, no muchos padres de familia tienen dispositivos iPhone ya sea por su precio o por la dificultad de manejarlos, y un porcentaje mínimo utiliza teléfonos con sistema operativo Windows ya que de no tienen mayor acogida en el mercado.

### Determinación del uso de Android en la propuesta

En Ecuador, la población utiliza dispositivos móviles con diferentes fines, los cuales pueden ser: comerciales, académicos, comunicativos o recreativos. Debido a esto nuestro país se sumerge cada vez más en la tecnología.

Facebook presenta estadísticas acerca del sistema operativo que utiliza cada usuario al ingresar a esta plataforma mediante su dispositivo móvil, lo cual se visualiza en la Figura 9 mostrada a continuación:



**Figura 9.** Sistema operativo en dispositivos móviles en Ecuador  
Elaborado por: Christian Caicedo

Son cifras que demuestran la realidad del país, el 83% de la población utiliza dispositivos móviles con sistema operativo Android, a ello se ajustan las marcas: Samsung, Huawei, Lg, entre otros, sin embargo, otros sistemas operativos como iOS dominan el 8% del mercado.

El sistema operativo de Android fue diseñado principalmente para dispositivos móviles con pantalla táctil, como: teléfonos inteligentes y tablets, siendo los dispositivos más utilizados en Ecuador por su bajo costo de adquisición y por sus modelos llamativos.

Utilizando los datos arrojados por las estadísticas ya mostradas se decide el realizar una aplicación Android para la propuesta de desarrollo y así poder controlar el prototipo.



#### **4.2.2. Selección de un framework para el desarrollo de una aplicación móvil**

En desarrollo de software utilizamos, conocemos o, como mínimo, tropezamos con el concepto de framework, al cual se lo puede determinar como “marco de trabajo” o estructura de soporte en el cual programar puede ser organizado con mayor simplicidad. Un buen framework es aquel que resuelve los problemas técnicos más complejos, como, por ejemplo: el manejo de transacciones, permitiendo a los desarrolladores concentrarse en el diseño y desarrollo de la aplicación sin llegar a ser intrusivos para el programador.

Los frameworks suelen incluir: soporte de programas, bibliotecas, lenguaje de scripting, software para desarrollar y unir diferentes componentes de un proyecto de desarrollo de programas [23].

Así es como la utilización de un framework da ciertas ventajas al programador siendo estas entre las más destacadas:

- El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto o una guía que se debe cumplir.
- Facilita la colaboración.
- Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.

Hoy en día con el avance de la tecnología se tiene muchas opciones al momento de escoger un framework para desarrollar software, esto varía en dependencia de los diferentes lenguajes de programación, requerimientos técnicos, plataformas sobre la cual se ejecuta entre otros aspectos.

A continuación, se evalúa arbitrariamente 5 características fundamentales para tomar en cuenta al momento de realizar la selección del framework como: licenciamiento gratuito, fecha de publicación, plataformas de desarrollo, requerimientos mínimos y lenguaje de programación, posterior a ello se selecciona un framework para proceder con el desarrollo de software del prototipo.

**Tabla 1.** Evaluación de características de framework Android Studio  
 Elaborado por: Christian Caicedo  
 Fuente : [24]

**Frameworks de elaboración de la app**

<i>Framework</i>	<b>Características</b>	<b>Especificaciones o requerimientos</b>	<b>Valor de 1 a 2 arbitrariamente</b>
<i>Android Studio</i>	Licenciamiento gratuito	Licencia Apache 2.0.	2
	Fecha de publicación	2014	2
	Varias plataformas de desarrollo	Microsoft Windows, macOS y GNU/Linux	2
	Requerimientos mínimos para desarrolladores	OS: Windows 10/8/7 (32- o 64-bit) CPU: Por lo menos 1 GB para Android SDK, imágenes del sistema de emulador y cachés Resolución de pantalla: 1280x800 mínimo 1440x900 recomendado RAM: 3 GB RAM mínimo, 16 GB RAM recomendado	1
	Lenguaje de programación	Java y XML.	2
		<b>Total</b>	<b>9</b>

**Tabla 2.** Evaluación de características de framework eclipse  
 Elaborado por: Christian Caicedo  
 Fuente : [25].

**Frameworks de construcción de app**

<i>Framework</i>	<b>Características</b>	<b>Especificaciones o requerimientos</b>	<b>Valor de 1 a 2 arbitrariamente</b>
<i>Eclipse</i>	Licenciamiento gratuito	Licencia Publica de Eclipse	1
	Fecha de publicación	2001	1
	Varias plataformas de desarrollo	Microsoft Windows, macOS y GNU/Linux	2
	Requerimientos mínimos para desarrolladores	OS: Windows 10/8/7 (32- o 64-bit) CPU: 150 MB de espacio disponible Resolución de pantalla: 1280x800 mínimo 1440x900 recomendado RAM: 4 GB de RAM.	1
	Lenguaje de programación	IDE, Java SDK, C/C++	2
	<b>Total</b>		<b>7</b>

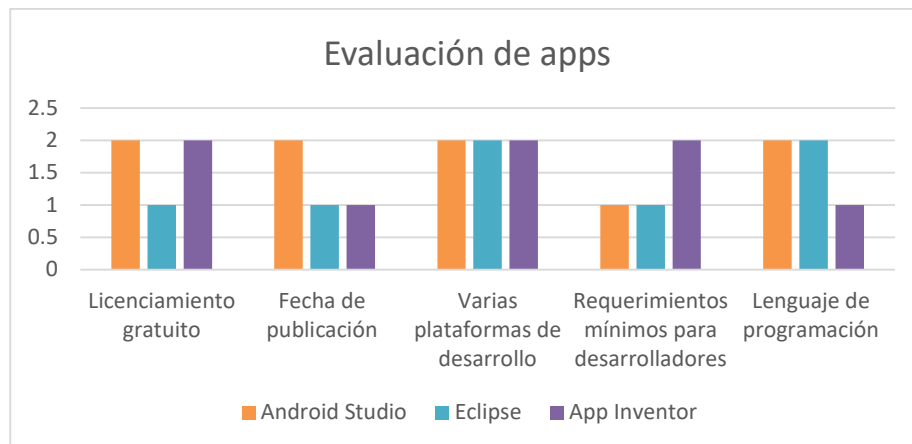
**Tabla 3.** Evaluación de características de framework App Inventor  
 Elaborado por: Christian Caicedo  
 Fuente : [26].

**Frameworks de construcción de app**

<i>Framework</i>	<b>Características</b>	<b>Especificaciones o requerimientos</b>	<b>Valor de 1 a 2</b>
<i>App Inventor</i>	Licenciamiento gratuito	Freeware (Explorer)	2
	Fecha de publicación	2012	1
		Mac OS X 10.5, 10.6	
	Varias plataformas de desarrollo	De Windows: Windows XP, Windows Vista, Windows 7 GNU / Linux: Ubuntu 8 +, Debian 5 +	2
	Requerimientos mínimos para desarrolladores	OS: Windows 10/8/7 (32- o 64-bit) RAM: 4 GB de RAM.	2
	Lenguaje de programación	Kawa	1
		<b>Total</b>	<b>8</b>

## Resultados de la evaluación de selección del framework de app.

En la figura 10 se muestra gráficamente la tabulación realizada para la selección de framework.



**Figura 10.** Evaluación de características de framework  
Elaborado por: Christian Caicedo

**Licenciamiento gratuito.** - Android studio y App inventor alcanzan un valor 2 porque tienen una licencia abierta hacia los desarrolladores, a su vez el framework Eclipse recibe una ponderación de 1 debido a que su licencia gratuita tiene condiciones de regalías.

**Fecha de publicación.** - Eclipse y App inventor reciben el valor 1 porque su software fue creado en años anteriores, sin embargo, Android studio es el framework más actualizado, por ello recibe una ponderación de 2.

**Varias plataformas de desarrollo.** – Cada framework recibe una valoración de 2 ya que no presentan dificultades de funcionamiento en ningún sistema operativo.

**Requerimientos mínimos de desarrolladores.** - Eclipse y Android Studio reciben el valor de 1 debido a que son frameworks que requieren una instalación previa en un computador para que cada desarrollador pueda acceder a ellos, mas App Inventor recibe una valoración de 2 a causa de que se puede acceder a este software a través de internet.

**Lenguaje de programación.** -Eclipse y Android Studio reciben una ponderación de 2 a causa de que son frameworks cuyo lenguaje de programación está enfocado a desarrolladores medios y expertos, en cambio, App Inventor tiene un lenguaje de programación visual centrado en desarrollado para programadores principiantes, por lo que recibe una valoración de 1.

### Decisión de selección del framework

Una vez especificadas las características de algunos frameworks se selecciona *Android Studio* para el desarrollo de la propuesta, debido a que la suma de sus ponderaciones da como resultado un valor de 9, superando a los demás frameworks por su versatilidad y facilidad que ofrece al momento de desarrollar software, además de las varias herramientas que posee.

### 4.2.3. Selección de hardware para el desarrollo de la propuesta

Es necesario seleccionar el hardware que mejor se ajuste a las características del prototipo que se desea construir, con base en las ventajas que presenta cada una de las alternativas de microcontroladores a utilizarse tenemos:

#### Microcontrolador PIC

Es un microcontrolador fabricado por Microchip Technology, hace un par de años era el microcontrolador más vendido en el mundo, sin embargo, los nuevos microcontroladores ARM Cortex han ido restándole popularidad [27].

#### Características

- Núcleos de CPU de 8/16 bits con Arquitectura Harvard modificada
- Memoria Flash y ROM disponible desde 256 bytes a 256 kilobytes
- Puertos de entrada/salida (típicamente 0 a 5.5 voltios)
- Temporizadores de 8/16/32 bits
- Periféricos serie síncronos y asíncronos: USART, AUSART, EUSART
- Conversores analógicos/digital de 8-10-12 bits

**Tabla 4.** Evaluación de características de framework para hardware Microcontrolador PIC  
Elaborado por: Christian Caicedo  
Fuente: [27].

	Característica	Valor de 1 a 2
costo	Bajo	2
aplicabilidad	proyectos	1
software	SDCC código abierto	2
lenguaje de programación	Ensamblador, Basic, C	2
<b>TOTAL</b>		<b>7</b>

## Microcontrolador Arduino

Es un microcontrolador que utiliza una plataforma de código abierto para construir dispositivos digitales y dispositivos interactivos que puedan sensor variables físicas y mediante su tratamiento a través de programación poder controlar prototipos finales [28].

## Características

**Tabla 5.** Características de Arduino  
Elaborado por: Christian Caicedo  
Fuente : [28].

Microcontrolador:	ATmega2560
Alimentación:	5V
Entrada:	7-12V
Límites (max):	5.5-16V
Pines digitales:	54 (14 con PWM)
Pines analógicos:	16
Corriente por pin:	40 mA
Memoria Flash (programa):	256 KB (8 KB usados para el bootloader)
SRAM:	8 KB
EEPROM:	4 KB
Reloj:	16 MHz

**Tabla 6.** Evaluación de características de framework para hardware Arduino  
Elaborado por: Christian Caicedo  
Fuente : [28].

	Característica	Valor de 1 a 2
costo	Varía según el tipo	2
aplicabilidad	proyectos	1
software	Arduino	2
lenguaje de programación	C++	2
TOTAL		7

## Microcontrolador Pololu Baby Orangután

Es un componente electrónico que incluye un microcontrolador AVR ATmega328P interno de alto rendimiento, y dos canales de unidad motriz en el mismo formato de 24 pines, es útil para desarrollar prototipos electrónicos debido al compacto tamaño que posee [29].

## Características

**Tabla 7.** Características de Pololu Baby Orangután  
Elaborado por: Christian Caicedo  
Fuente : [29]

Procesador:	ATmega328P @ 20 MHz
Tamaño de RAM:	2048 bytes
Tamaño de la memoria del programa:	32 Kbytes
Controlador del motor:	TB6612FNG
Canales de motor:	2
Líneas de E / S del usuario:	182
Corriente máxima en una sola E / S:	40 mA
Voltaje de funcionamiento mínimo:	5 V
Voltaje máximo de operación:	13.5 V
Corriente de salida continua por canal:	1 A
Corriente de salida máxima por canal:	3 A

La placa se basa en un potente microcontrolador AVR de Atmel, la baby orangután ofrece un rendimiento significativamente mayor que otras tarjetas similares.

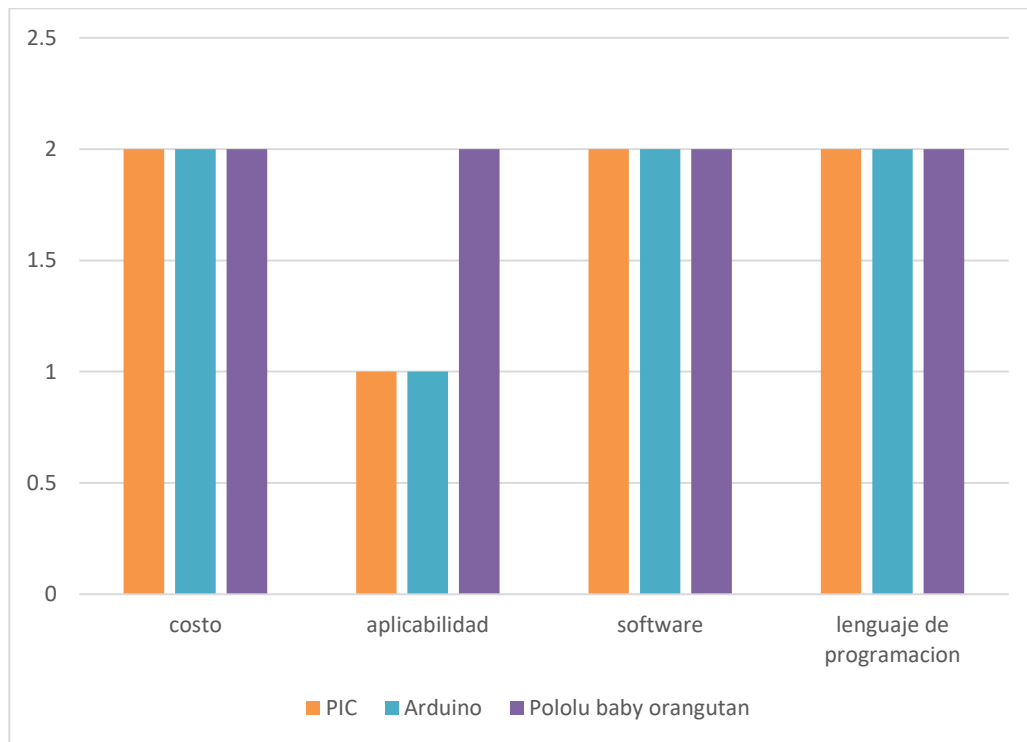


**Tabla 8.** Evaluación de características de framework para hardware Pololu Baby Orangután  
 Elaborado por: Christian Caicedo  
 Fuente : [29]

	Característica	Valor de 1 a 2
costo	bajo	2
aplicabilidad	proyectos, área robótica	2
software	Atmel Studio IDE	2
lenguaje de programación	WinAVR GCC C / C ++	2
TOTAL		8

**Resultados de la evaluación de selección de hardware para el desarrollo de la propuesta**

En la figura 11 se muestra gráficamente la tabulación realizada para la selección de framework.



**Figura 11.** Evaluación de selección de hardware para el desarrollo de la propuesta  
 Elaborado por: Christian Caicedo

**Costo.** –Los tres tipos de microcontroladores reciben una ponderación de 2 ya que sus costos en la actualidad son accesibles al público en general.

**Aplicabilidad.** – El microcontrolador pic está enfocado en la producción en masa por lo que recibe una ponderación de 1 debido a que en proyecto se realizara únicamente un prototipo, no una producción mayor; Arduino también recibe una ponderación de 1 a causa del tamaño que ofrece en la mayoría de sus productos y a que este microcontrolador no posee un driver interno para el control de motores. Estos dos microcontroladores tienen puntos débiles para la implementación del proyecto y la solución a ello se encuentra en la utilización del microcontrolador Pololu baby orangután ya que tiene dimensiones reducidas, está diseñado para creación de prototipos y posee un driver interno para el control de motores por lo cual recibe una ponderación de 2.

**Software.** –La licencia de software es gratuita para los 3 microcontroladores por lo que reciben una ponderación de 2.

**Lenguaje de programación.** – Los microcontroladores pueden ser codificados en varios lenguajes de programación de alto nivel los cuales son generalmente C y C++ por ello reciben un valor de 2.

#### **Decisión del framework y selección de hardware para la construcción del prototipo**

Una vez especificado las características de los microcontroladores tomados en cuenta para este proyecto se selecciona el microcontrolador *Pololu baby orangután*, el cual recibe una valoración total del 100% a su favor.

#### **4.2.4. Metodología utilizada en el desarrollo de la propuesta**

Para el desarrollo de la propuesta se utilizan dos metodologías, la primera es para el desarrollo de software utilizando una metodología por prototipos y la segunda para la construcción del robot (desarrollo de hardware) hace referencia a la metodología bottom-up la cual divide en bloques funcionales la estructura electrónica total.

## Metodología utilizada en el desarrollo de la aplicación móvil

Esta metodología muestra una serie de probables interfaces para una aplicación final diseñando su aspecto, y funcionamiento general. Para muchas interfaces de usuario, este modelo puede resultar tan simple como unos dibujos con lápiz y papel o tan complejo como el propio código operativo final.

### Desarrollo por prototipos

Conocida también como modelo de desarrollo evolutivo, inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria una mayor definición. Este modelo se utiliza para dar al usuario una vista preliminar de parte del software, es decir se desarrollan algunas ideas del prototipo hasta llegar al diseño ideal [16].

Esta metodología propone el seguimiento de una serie de pasos para la consecución del producto final, se presenta un ciclo que se realiza siguiendo la estructura propuesta por la metodología como se muestra en la Figura 12.



**Figura 12.** Metodología desarrollo por prototipos  
Fuente: [16]

El orden lógico de esta metodología cumple con las etapas siguientes:

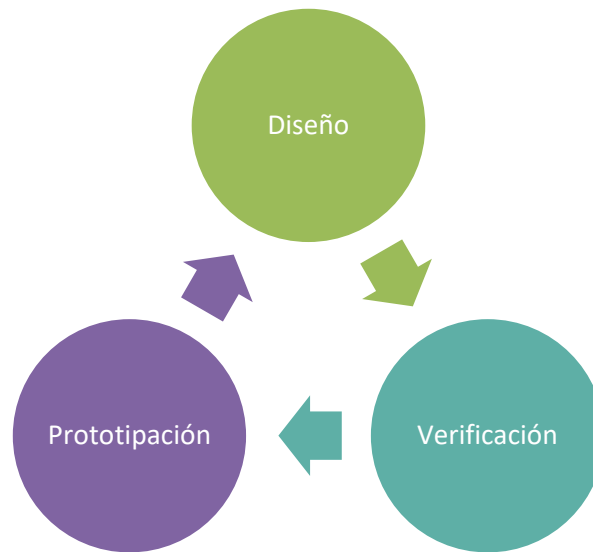
- Recolección y refinamiento de requisitos
- Modelado, diseño rápido
- Construcción del Prototipo
- Desarrollo, evaluación del prototipo por el cliente
- Refinamiento del prototipo
- Producto de Ingeniería

Se deben seguir los pasos mencionados para desarrollar la interfaz ideal, se puede tener un número N de posibles diseños, esto se determina acorde a las necesidades del cliente y del programador estableciendo lo siguiente: Se busca el producto ideal desde el primer diseño, para tener una mejor aceptación con el usuario final, además se debe ser sencillo, interactivo y llamativo para los niños; teniendo en cuenta que el prototipo que se está desarrollando es para niños de 4 a 10 años de edad.

El desarrollo por prototipos se puede utilizar como un modelo del proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquiera de los modelos del proceso expuesto. Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollo de la aplicación y ayuda también al usuario a entender de mejor manera cuál será el resultado final cuando los requisitos están satisfechos.

### **Metodología utilizada en el desarrollo del prototipo**

Para el diseño de hardware, tomando en cuenta que se requiere precisión en los movimientos del robot, se utiliza la metodología bottom-up, la cual divide los módulos funcionales de la parte electrónica del robot en subsistemas independientes completamente funcionales, los cuales en este caso son: alimentación, comunicación, movimiento y sensorización. Todos los subsistemas que emplea el robot se sintetizan en 3 etapas propuestas por la metodología bottom-up en sí, las cuales se estructuran en base a tres etapas direccionadas hacia la calidad total del sistema, estas son: diseño, prototipación y finalmente verificación como se muestra en la Figura 13.



**Figura 13.** Metodología desarrollo de hardware  
Fuente [19]

La etapa de diseño propuesta por esta metodología permite conocer el resultado estético del prototipo, esta etapa puede ser diseñada en los siguientes programas: CAD (Diseño Asistido por Computador), CAD CAM, Solid Works, Inventor, entre otros; Rubus tiene un diseño realizado en el software Solid Works.

La etapa de prototipación se realiza acorde a los requerimientos y características estipuladas por el diseñador para el funcionamiento del robot, en donde se tendrán en cuenta las diferentes particularidades en cuanto a: sensores, actuadores y componentes adicionales para el prototipo. Existen diferentes herramientas que pueden utilizarse durante el diseño e implementación del prototipo, las cuales son:

**Descripción mediante esquemas:** consiste en la realización del circuito mediante un esquema que represente la estructura del sistema.

**Grafos y diagramas de flujo:** Consiste en la realización de diagramas de flujo, para conocer cómo se comportará el sistema desde un paso inicial hasta un paso final

**Lenguajes de descripción:** Consiste en lenguajes de computadora especializados los cuales permiten tener un circuito digital, que presenta el funcionamiento global del prototipo a desarrollarse y da a conocer los componentes por los que está conformado.

## **Diseño Bottom-Up**

Este tipo de diseño comprende la operatividad total del circuito final que requiere el prototipo agrupando las partes funcionales en diferentes módulos que representan el sistema completo del mismo, este tipo de diseño no implica una estructuración jerárquica de sus elementos, simplemente reúne todos sus componentes de bajo nivel para formar el diseño global, los cuales en el presente prototipo corresponden a: alimentación, comunicación, movimiento y sensorización; cada módulo debe ser completamente funcional, puesto que en la fase final de estructuración cada uno de los módulos se integran para lograr un sistema electrónico eficiente.

### **4.3. Desarrollo de la propuesta de la (aplicación y prototipo).**

La creación de todo prototipo controlado mediante una aplicación móvil comprende una fase inicial de software y una etapa posterior relacionada con la parte electrónica que le dota de motricidad y funcionalidad al prototipo.

#### **4.3.1. Desarrollo de la aplicación móvil.**

Utilizando el framework seleccionado para el desarrollo y aplicando la metodología desarrollo por prototipos, se tiene a continuación el desarrollo de la aplicación móvil.



**Figura 14.** Icono de Android Studio

Fuente: [24]

Android Studio es un IDE oficial de Google en colaboración con JetBrains (compañía de desarrollo software especializada en diseño de IDE's) y se ha convertido con el pasar del tiempo en el entorno recomendado para el desarrollo de aplicaciones móviles por su gran versatilidad [24].

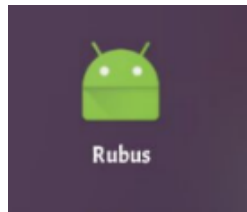
Con la simple descarga de Android Studio se disponen de todas las herramientas necesarias para el desarrollo de aplicaciones para la plataforma Android, la misma que proporciona una serie de ventajas de un entorno Java, tales como:

- Ágil distribución de código, y por lo tanto el trabajo en equipo.
- Reutilización de código y recursos.
- Compilación desde línea de comandos, para aquellas situaciones en las que no esté disponible un entorno de desarrollo.
- Diseño de las interfaces acordes a la propuesta y llamativas para los usuarios [24].

Por lo tanto, el framework para el desarrollo de la aplicación que controlará el prototipo es Android Studio, a continuación, se detallan los prototipos de software desarrollados para llegar al diseño final de Rubus.

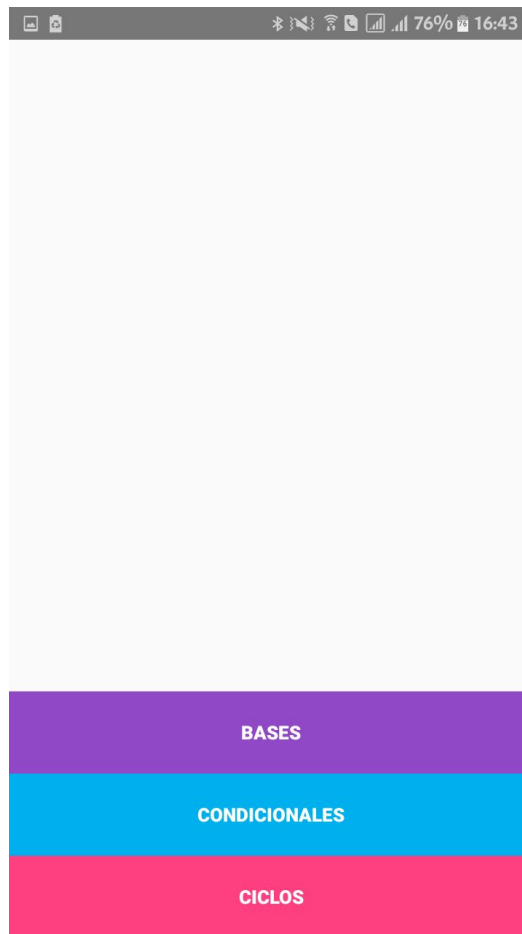
### **Diseño del primer prototipo**

Rubus es el nombre de la aplicación y del robot, este nombre proviene de una agrupación de plantas rosáceas (familia de las moras *Rubus Ulmifolius*), por ello el color preferente de este robot es el morado. El ícono de la figura 15 es el ícono por defecto que ofrece Android Studio, en etapas siguientes se mostrará modificado.



**Figura 15.** Icono de Rubus en el celular  
Elaborado por: Christian Caicedo

El diseño del primer prototipo muestra tres botones los cuales son: bases, condicionales y ciclos, estos pueden ser visualizados en la figura 16, cabe recalcar que ellos aún no presentan funcionalidad alguna, únicamente corresponden a diseños de interfaz.

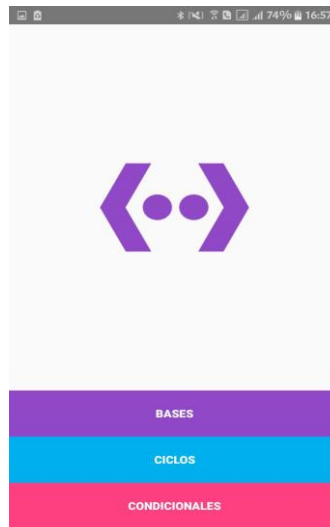


**Figura 16.** Pantalla del programador  
Elaborado por: Christian Caicedo

### **Diseño del segundo prototipo**

A partir del primer prototipo se realizan mejoras en los botones de la interfaz, asignando funcionalidades a cada uno para poder continuar hacia un nuevo entorno en donde se encuentra una interfaz de programación para los usuarios finales que son los niños, cada botón activa o desactiva funcionalidades específicas en la nueva interfaz, en esta pantalla a la que se accede luego de presionar uno de los tres botones se encuentran íconos correspondientes a los movimientos o acciones que cada niño desea que realice el robot Rubus como se indica en la figura 17.





**Figura 17.** Pantalla del programador 2do Prototipo  
Elaborado por: Christian Caicedo

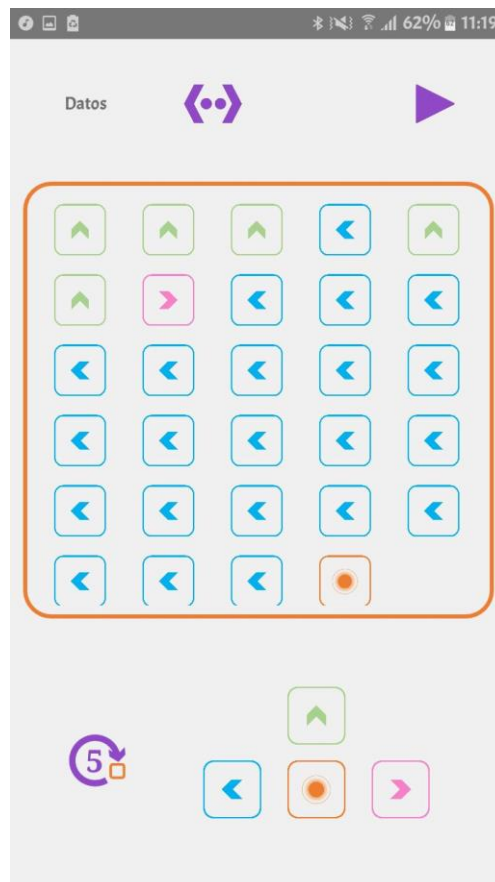
En la figura 18 se muestra la interfaz del segundo entorno, en donde el contenedor de actividades que tiene un borde de color naranja ocupa el 65% de la pantalla, aquí se almacenarán los movimientos realizados en forma arbitraria por el niño que se encuentre manipulando el robot.



**Figura 18.** Botón Bases  
Elaborado por: Christian Caicedo

## Diseño del tercer prototipo

En este diseño se implementa un botón adicional que tiene asociación con el número de ciclos o veces en que se realizarán los movimientos del contenedor de actividades, cabe recalcar que Rubus realiza 9 ciclos como máximo de la serie de actividades almacenadas.

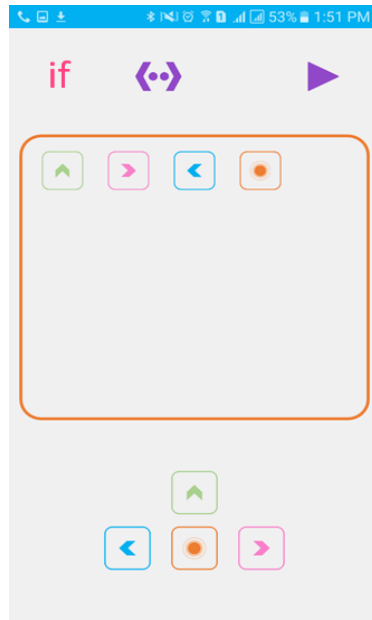


**Figura 19.** Botón ciclos  
Elaborado por: Christian Caicedo

La figura 19 muestra el botón de ciclos que tiene una coloración morada, este botón se inicializa en uno, lo cual quiere decir que el robot realizará la secuencia de acciones del contenedor de actividades una sola vez, y siempre que vuelva a presionarse la visualización del número de ciclos de este botón se incrementará en uno llegando como máximo al valor nueve, lo cual representa como se detalló anteriormente, el número de veces en que se repetirán las acciones del contenedor de actividades.

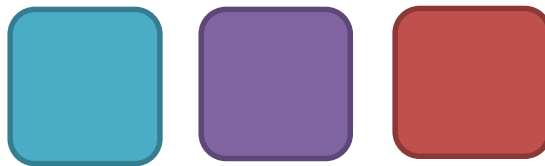
## Diseño del Cuarto prototipo

Para el cuarto prototipo se implementa una etiqueta condicional, la cual agrega la característica de sensorización del color rojo, esto se visualiza con la etiqueta “if” en la pantalla, tal como muestra la figura 20.



**Figura 20.** Etiqueta If Pantalla del programador  
Elaborado por: Christian Caicedo

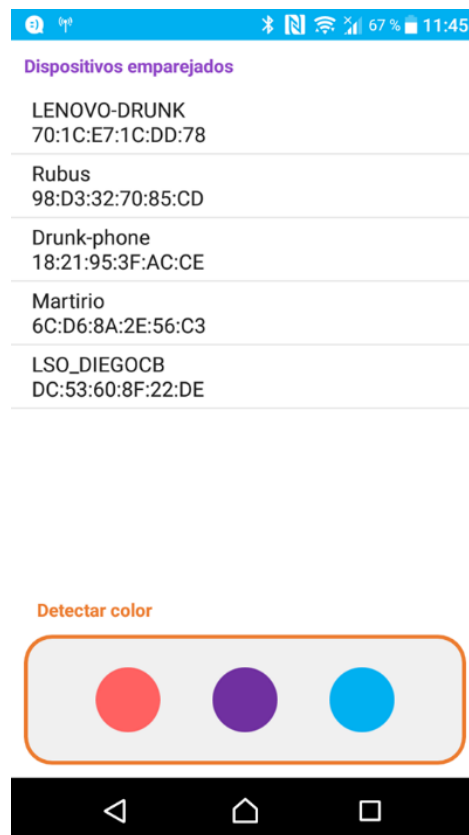
a característica de sensorización permite al robot reconocer el color rojo, con ello se aplica el concepto de condicionalidad, por lo que: cuando el sensor de color está activo y se detecta el color rojo, el robot pierde su energía, por lo que se inhabilita cualquier movimiento, el sensor de color de Rubus está configurado para sensar 3 colores mostrados en la figura 21, en todo momento el robot se encuentra censando el celeste para evitar que Rubus salga de la pista, el color morado es el objetivo y rojo representa un error.



**Figura 21.** Colores que sensa Rubus.  
Elaborado por: Christian Caicedo

## Diseño del Quinto prototipo

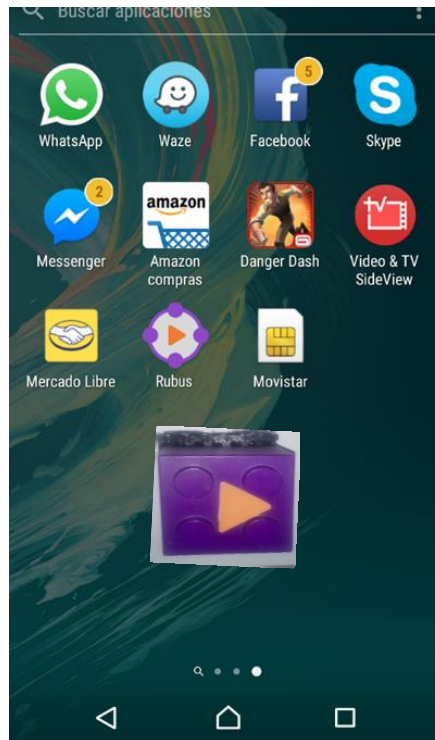
Para el quinto prototipo se añade la parte de conexión entre dispositivos para lograr una comunicación bluetooth entre la aplicación y el prototipo, los dispositivos que se visualizan en la figura 22 son aquellos que ya se encuentran vinculados con el teléfono que contiene la aplicación; además, se añaden tres botones de sensorización de color haciendo especial referencia a los colores: celeste, morado y rojo, para crear una configuración previa de color acorde al entorno en donde se vaya a encontrar Rubus, como se visualiza en la figura 22, esto debido a que cada ambiente tiene diferencias marcadas en cuanto a iluminación, puesto que los valores que genera el sensor al detectar un color difieren en un margen considerable al ser evaluados en el día en comparación a que si fuesen sensados en la noche o en un ambiente cerrado, lo que puede generar errores en el procesamiento de información del robot.



**Figura 22.** Pantalla Inicial del prototipo 5  
Elaborado por: Christian Caicedo

## Diseño del Sexto prototipo (Prototipo final)

En el sexto diseño del prototipo el ícono de la aplicación cambia de manera radical que sea representativo del prototipo que ya está ensamblado como indica la figura 23 para, y a la vez sea acogido para los usuarios de Rubus, quienes son los niños.



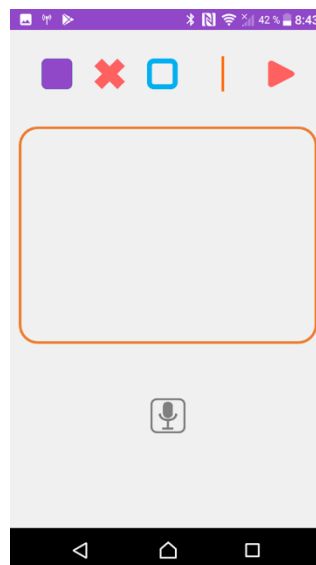
**Figura 23.** Icono de la aplicación del prototipo final  
Elaborado por: Christian Caicedo

Una vez iniciada la aplicación como se muestra en la figura 24 se aprecia el aspecto final de la aplicación, la cual presenta un ícono gris para la conexión entre dispositivos, este cambia de color cuando la comunicación sea exitosa, además incorpora una nueva modalidad de programación por medio de reconocimiento de comandos de voz, para acceder a esta funcionalidad basta con pulsar el botón de color naranja de la pantalla inicial desplegada al abrir la aplicación, además, los tres botones mencionados en el prototipo uno se mantienen con ligeros cambios dentro de la interfaz del programador



**Figura 24.** Pantalla Inicial del prototipo 6  
Elaborado por: Christian Caicedo

Al presionar el ícono de reconocimiento por voz se despliega una interfaz mostrada en la figura 25, en donde a más del contenedor de acciones se visualizan íconos que sirven para enviar la información hacia el prototipo físico y notificar la presencia de colores. El ícono inferior sirve para ingresar órdenes en el contenedor y enviarlas al robot de manera inmediata.

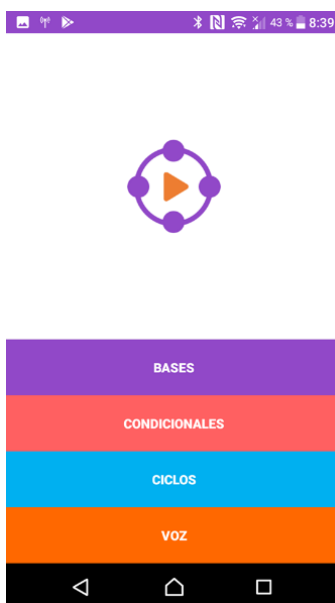


**Figura 25.** Pantalla Botón voz  
Elaborado por: Christian Caicedo

Debido a que el prototipo seis corresponde a la interfaz final de Rubus, se detalla a continuación la lógica de su funcionamiento.

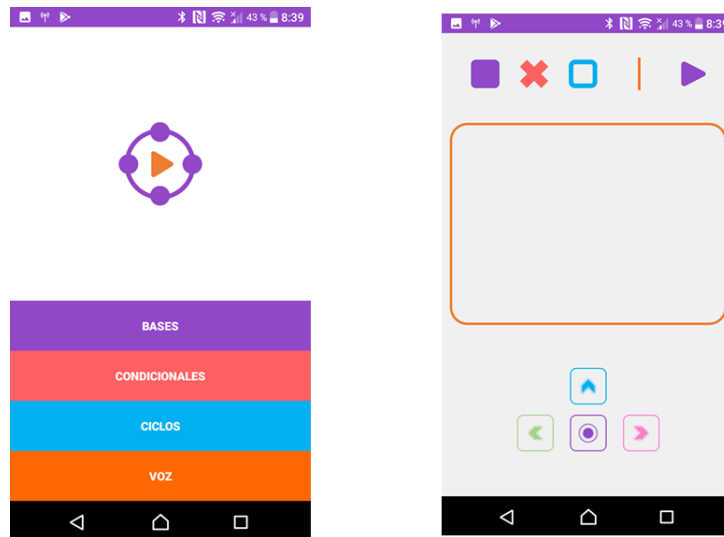
### **Lógica de funcionamiento de la interfaz de Rubus.**

Al abrir la aplicación se visualiza la interfaz de la figura 26, en donde cada uno de los cuatro botones habilitan o deshabilitan funciones en la pantalla de programación siguiente, estos son: botón base, botón condiciones, botón ciclos y el botón correspondiente al reconocimiento por voz; asimismo, en la interfaz del programador, si Rubus detecta el color morado en el cuadrado de la misma tonalidad se realiza una animación de escala, y el celular vibra por una corta cantidad de tiempo, lo que sucede también en el caso de los colores rojo y celeste.



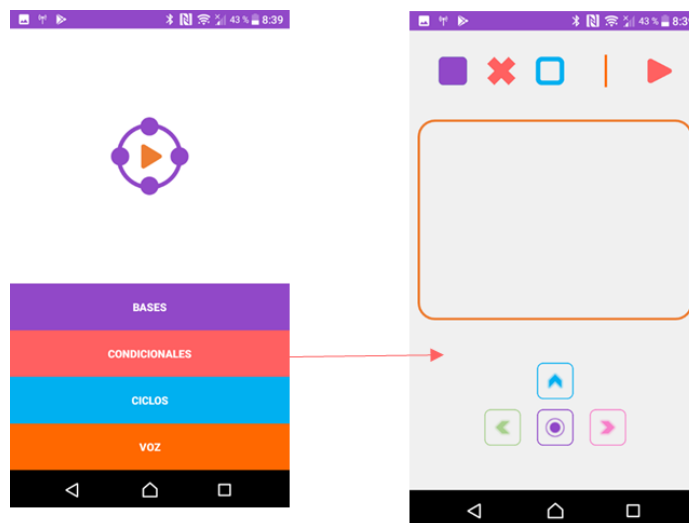
**Figura 26.** Pantalla Inicial del prototipo final  
Elaborado por: Christian Caicedo

En la figura 27 se observa la interfaz de programador que se despliega al momento de presionar el botón de bases, aquí pueden ingresarse instrucciones en el contenedor de actividades al pulsar sobre cualquiera de los íconos de movimiento, y serán enviadas al robot al presionar la flecha de color morado, al mismo tiempo, el robot se encontrará sensando siempre el color celeste para evitar que el robot salga de su zona de desplazamiento.



**Figura 27.** Pantalla del botón bases  
Elaborado por: Christian Caicedo

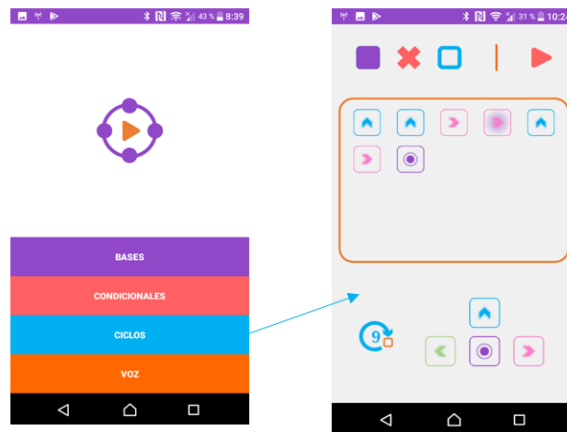
Al presionar el botón de condicionales se despliega una interfaz en la cual desaparece la etiqueta If que se encontraba previamente y en su remplazo al detectarse el color rojo el indicador en forma de X presenta una animación y la flecha de envío se torna de color rosa como se muestra en la figura 28 en tanto que Rubus detiene su funcionamiento puesto que este color representa un obstáculo que el robot debe evitar a cabalidad.



**Figura 28.** Pantalla del botón condicionales  
Elaborado por: Christian Caicedo

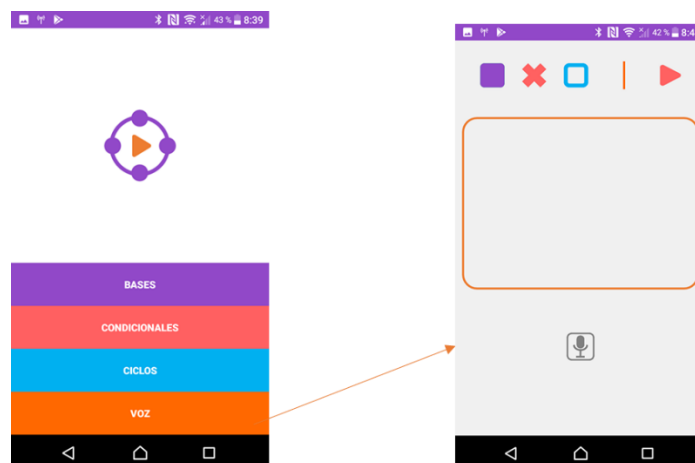


Cuando se presiona el botón de ciclos, se extiende una interfaz en donde a más de los íconos representativos de sensorización, envío de información y programación de acciones se puede visualizar un ícono que muestra el número de veces que se van a realizar las acciones del contenedor de movimientos, como muestra la figura 29, asimismo este ícono es también un botón, el cual al ser presionado incrementa en uno al número de ciclos que las acciones serán realizadas, con un máximo de 9 veces .



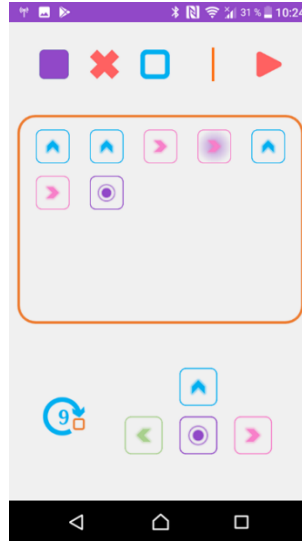
**Figura 29.** Pantalla del botón ciclos  
Elaborado por: Christian Caicedo

Finalmente, las características del botón que activa los comandos de voz propuestos en el prototipo número 6 no se vieron alteradas y su interfaz se observa en la figura 30.



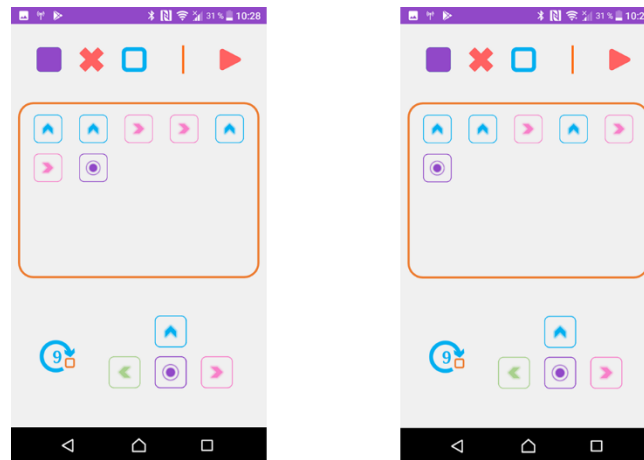
**Figura 30.** Pantalla del botón voz  
Elaborado por: Christian Caicedo

La figura 31 hace referencia al entorno interactivo de la aplicación, en donde una vez enviadas al robot las acciones que se encuentran dentro del contenedor de actividades, cada movimiento adquiere una sombra de color morado en el instante exacto en que es ejecutado, para indicar así qué movimiento de la secuencia enviada está siendo realizado por Rubus.



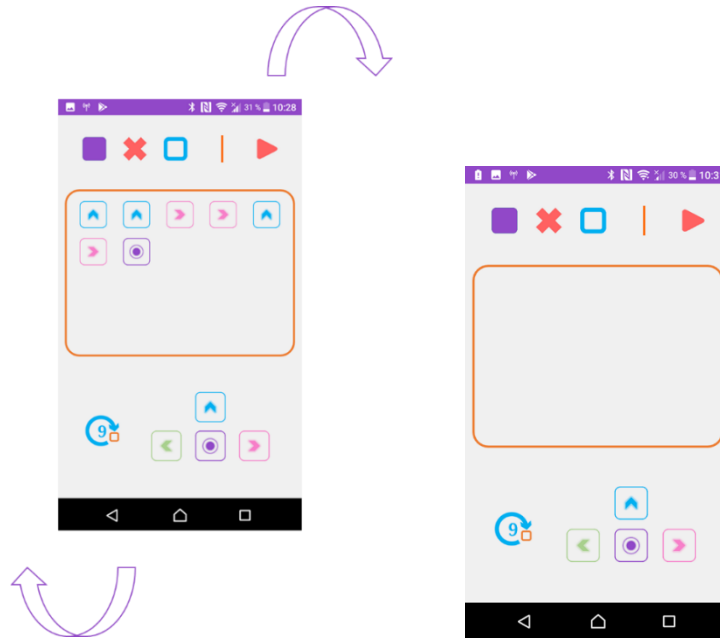
**Figura 31.** Interfaz de trabajo  
Elaborado por: Christian Caicedo

En caso de que se desee eliminar una orden que ya se encuentra dentro del contenedor de actividades, lo único que se debe hacer es dar un click en ella y de esta manera se elimina dicho movimiento de forma inmediata como lo muestra la figura 32.



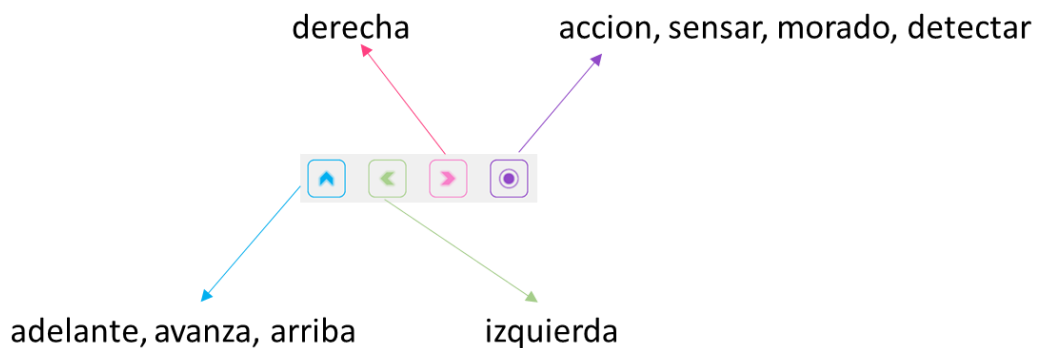
**Figura 32.** Interfaz de trabajo donde se elimina ordenes no deseadas  
Elaborado por: Christian Caicedo

Para que la interfaz de trabajo quede totalmente limpia lo único que se debe realizar es agitar el dispositivo móvil, y todas las órdenes serán eliminadas como se observa en la figura 33.



**Figura 33.** Interfaz de trabajo donde se elimina todas las órdenes  
Elaborado por: Christian Caicedo

Una de las funciones que tiene Rubus es el ingresar las órdenes en forma manual, lo cual se realiza por medio de los botones programados bajo la lógica que muestra la figura 34, estas funcionalidades pueden ser programadas por comandos de voz a través de las palabras asociadas a cada botón.



**Figura 34.** Botones que envían órdenes a Rubus  
Elaborado por: Christian Caicedo

Cada interfaz tiene su programación individual, cuyas instrucciones más relevantes serán detalladas a continuación:

El código mostrado en la figura 35 es un ejemplo claro de cómo el botón bases llama a una siguiente interfaz, esto sucede también con los botones de condicionales, ciclos y voz.

```
void clickBases(){
    btnBases.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent( getApplicationContext(), Programador.class);
            startActivity(i);
            Programador.condicionales =false;
            Programador.ciclo =false;
        }
    });
}
```

**Figura 35.** Programación botón Bases  
Elaborado por: Christian Caicedo

La programación detallada en la figura 36 explica la lógica de los métodos que utiliza la interfaz del programador para el funcionamiento del prototipo.

```
/*
Realiza conexión bluetooth con el prototipo
*/
longClickBuscarRubus(); //busca dispositivos bluetooth y guarda el seleccionado
clickConectarRubus(); //realiza la conexión con el dispositivo almacenado

/*
Añade acciones al contenedor de pasos
*/
clickArriba(); //añade arriba al contenedor
clickDerecha(); //añade derecha al contenedor
clickIzquierda(); //añade izquierda al contenedor
clickAccion(); //añade accion al contenedor

/*
El prototipo repite todas las acciones en el contenedor el numero de
veces en el botón ciclos
*/
clickCiclos(); //suma 1 al contenido del botón ciclos

/*
Comunicación con el prototipo
*/
clickEjecutar(); //envía acciones al prototipo para que las ejecute
```

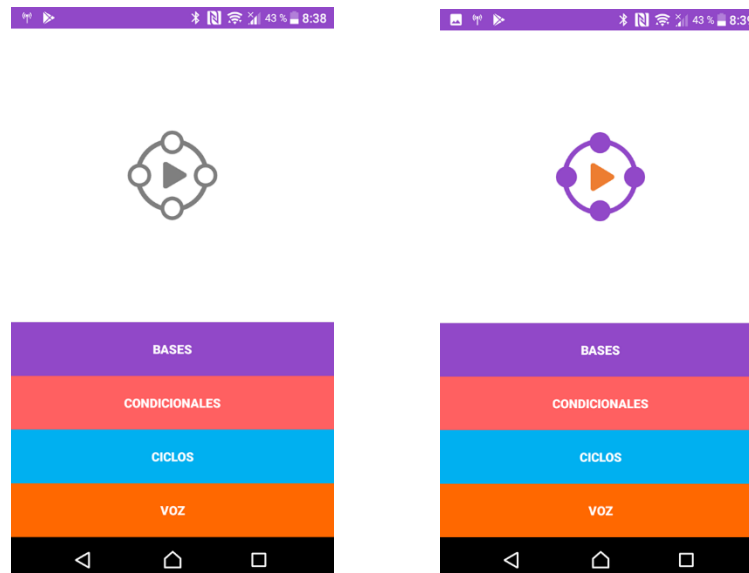
**Figura 36.** Programación de las ordenes de Rubus  
Elaborado por: Christian Caicedo

La figura 37 muestra la codificación representativa para lograr una comunicación Bluetooth llamando a una nueva interfaz en donde se presentan los dispositivos vinculados

```
private void longClickBuscarRubus() {  
    btnBlue.setOnLongClickListener(new View.OnLongClickListener() {  
        @Override  
        public boolean onLongClick(View view) {  
  
            Intent i = new Intent(getApplicationContext(), DispositivosEmparejados.class);  
            startActivity(i);  
  
            return false;  
        }  
    });  
}
```

**Figura 37.** Continuación Programación de las ordenes de Rubus  
Elaborado por: Christian Caicedo

Una vez seleccionado uno de los dispositivos vinculados, inicia el proceso de conexión con el prototipo, y el icono general de la aplicación cambia su tonalidad gris a una tonalidad más colorida, este proceso indica una conexión exitosa como se observa en la figura 38.



**Figura 38.** Emparejamiento de dispositivos  
Elaborado por: Christian Caicedo

### 4.3.2. Ensamblaje y programación del prototipo

La creación de un prototipo electrónico tiene como núcleo un microcontrolador programado para realizar las acciones específicas que se requieren para dotar al sistema de motricidad y funcionalidad a través del control de sensores y actuadores.

#### Programación del Microcontrolador

Para programar el microcontrolador Pololu Baby Orangután se utiliza el framework Arduino IDE, el cual permite realizar una programación estructurada para controlar el prototipo Rubus, las instrucciones son realizadas mediante lenguaje de programación C y C++ y el IDE lo traduce a un código hexadecimal entendible para el microcontrolador. Para el desarrollo de las líneas de programación se requieren librerías especiales para trabajar así de manera idónea, es por ello que se utilizan las librerías siguientes:

La librería SoftwareSerial permite establecer una comunicación bluetooth entre el prototipo y la aplicación de control mientras la librería OrangutanMotors activa los drivers necesarios para controlar motores dc, la sintaxis para su utilización se muestra en la figura 39.

```
//Libreria que realiza la comunicacion bluetooth
#include <SoftwareSerial.h>
//Librerias para controlar motores
#include <OrangutanMotors.h>
```

**Figura 39.** Librerías utilizadas en Arduino  
Elaborado por: Christian Caicedo

Una vez definidas las librerías se asignan pines de conexión, los mismos que deben conectarse a cada uno de los elementos del prototipo.

La figura 40 muestra los motores Pololu 30:1 que requiere Rubus para tener un desplazamiento adecuado, estos pines de conexión en el microcontrolador son asignados por default.

```

OrangutanMotors motors;
//Conectar motores a M2 y M1

```

Motores pololu 30:1



**Figura 40.** Sintaxis objeto motors  
Elaborado por: Christian Caicedo

La figura 41 muestra que el pin #7 del microcontrolador corresponde al pin de transmisión para la comunicación bluetooth , mientras que el pin #4 corresponde al pin de recepción.

```

SoftwareSerial BT (PD7, PD4);
//bluetooth TX a 7
//bluetooth RX a 4

```

Bluetooth hc-06



**Figura 41.** Pines de comunicación  
Elaborado por: Christian Caicedo

Los pines detallados en la figura 42 corresponden a los asociados para la sensorización de color.

```

//      S0 gnd
//      S1 VCC
#define S2 PD1
#define S3 PD0
#define sensorOut PD2

```

Sensor de color  
TCS3200



**Figura 42.** Pines de sensorización  
Elaborado por: Christian Caicedo

Los 2 métodos principales para el desarrollo de la programación del prototipo son el metodo setup y loop los cuales son encargados de la configuración y el funcionamiento total del sistema, su sintaxis se puede apreciar en la figura 43.

```

// Se ejecuta una vez al iniciar
void setup() {
  confRubus();
}

// Se ejecuta una y otra vez
void loop() {
  leerDato();
}

```

**Figura 43.** Sintaxis principal  
Elaborado por: Christian Caicedo

El método de configuración especifica los pines de entrada y salida del microcontrolador los cuales serán utilizados para sensorización, además se establece la velocidad de la comunicación bluetooth como se muestra en la figura 44.

```

void confRubus() {
  //Asigna como salidas a los pines del sensor de color
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  //Asigna como entrada el pin que determina color
  pinMode(sensorOut, INPUT);

  //Realiza la comunicación Bluetooth a 9600 Baudios
  BT.begin(9600);
}

```

**Figura 44.** Configuración de entradas y salidas  
Elaborado por: Christian Caicedo

El método leer dato es el más importante debido a que es el encargado de establecer una comunicación con tramas seriales entre el prototipo y la aplicación como se muestra en la figura 45.



```

//Lee los datos enviados a travez del bluetooth
void leerDato() {
  if (BT.available() > 0) {
    lectura = BT.readStringUntil(';'); //Obtenemos la cadena enviada por la aplicacion
    BT.flush();                       //Limpiamos el buffer del bluetooth
    determinar(lectura);              //Se determina que hacer con la cadena
  }
}

```

**Figura 45.** Metodo de comunicación serial  
Elaborado por: Christian Caicedo

Una vez establecida la comunicación entre la aplicación y el prototipo, el microcontrolador Pololu Baby Orangutan decodifica la información recibida y la procesa para realizar acciones varias, como el código de la figura 46.

```

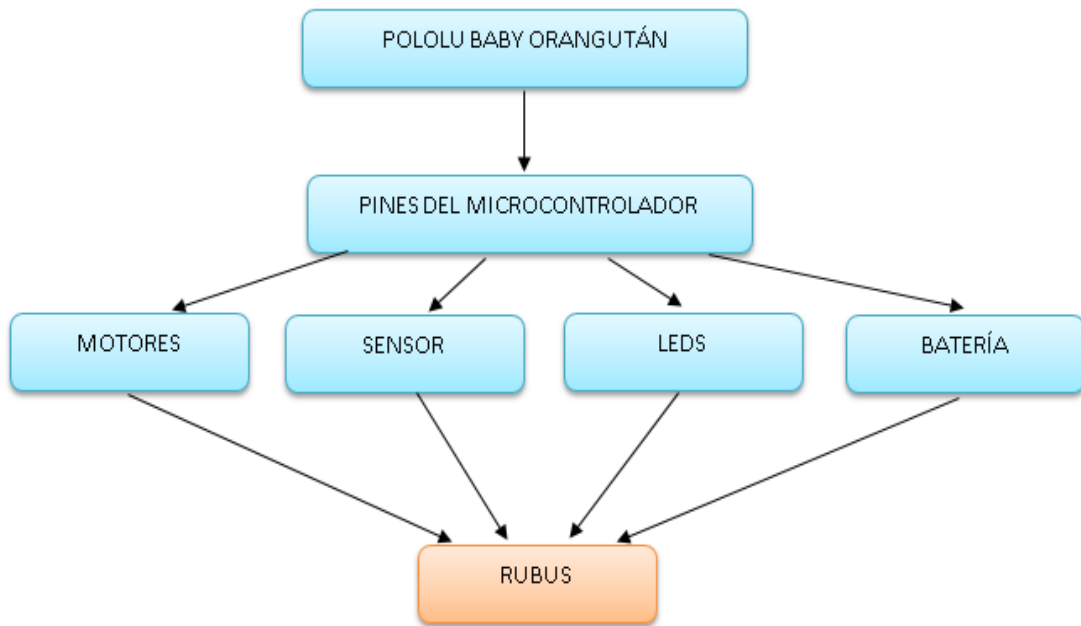
void determinar(String valor) {
  valor.trim();
  if (valor.length() > 0) { //Verifica que el valor enviado no sea ruido
    //Bases y Condicionales
    if (valor.charAt(0) == 'b' || valor.charAt(0) == 'i') {
      basesCondicionales(valor);
    } else {
      //Ciclos
      if (valor.charAt(0) == 'c') {
        ciclos(valor);
      }
    }
  }
}

```

**Figura 46.** Procesamiento de datos  
Elaborado por: Christian Caicedo

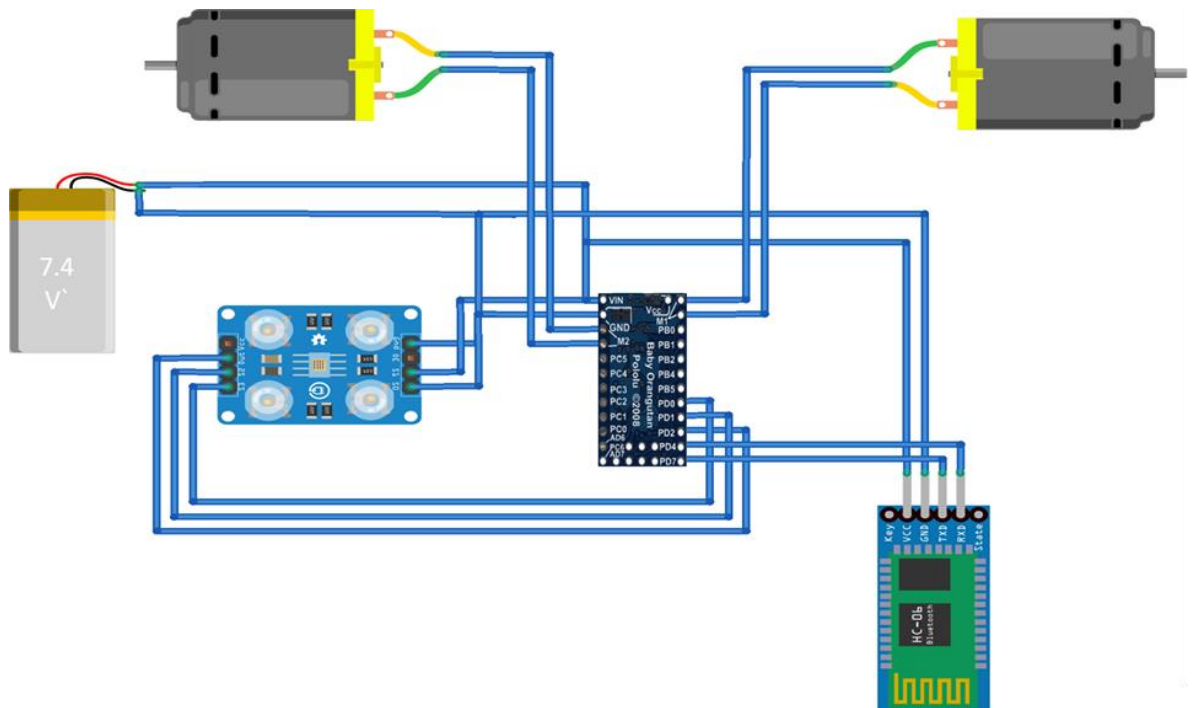
### Diseño electrónico del prototipo

El diseño de la parte electrónica está realizado utilizando la metodología bottom-up, por lo que partiendo desde el núcleo del sistema que es el microcontrolador se dividen los subsistemas de alimentación, comunicación, motricidad y sensorización, los cuales tienen pines específicos asignados a las entradas y salidas como indica el esquema mostrado en la figura 47.



**Figura 47.** Diseño Bottom-Up aplicado en Rubus  
Elaborado por: Christian Caicedo

La figura 48 muestra el esquema electrónico de funcionamiento de Rubus en donde se identifican todos los componentes del mismo, conjuntamente con las respectivas conexiones a los pines de entrada y salida del microcontrolador, las cuales corresponden a las conexiones del módulo bluetooth, motores, batería y motores; posterior a esta simulación estructural y a la respectiva programación del microcontrolador se realiza el ensamble completo del robot para probar su funcionamiento y poder corregir errores en caso de que llegaran a presentarse.



**Figura 48.** Diseño Rubus mediante circuitos  
Elaborado por: Christian Caicedo

### Ensamble del prototipo

A partir del diseño del prototipo utilizando el software Solid Works se tienen varios archivos con extensión .STL contenedores de las partes del case de Rubus, los cuales son impresos utilizando tecnología de impresión 3D con material PLA para tener un modelo exacto, posterior a esto y la verificación de funcionamiento de la parte electrónica se procede con el ensamble del mismo para la realización de pruebas.

### Materiales Electrónicos

A continuación, la tabla 9 describe los elementos electrónicos utilizados para la construcción del prototipo, la mayoría de ellos son de marca Pololu para crear una compatibilidad completa entre sensores, actuadores y microcontrolador:

**Tabla 9.** Materiales Electrónicos  
Elaborado por: Christian Caicedo

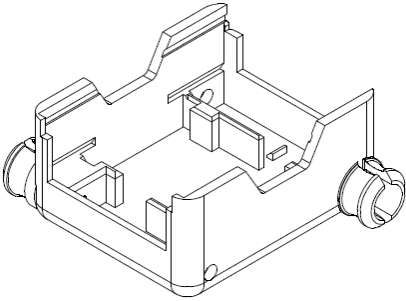

<b>Materiales electrónicos</b>		
<b>Material</b>	<b>Imagen</b>	<b>Características</b>
<b>Baby Orangutan B-328 Robot Controller</b>		-Es el cerebro de Rubus, controla todos los elementos como los motores, bluetooth, sensor.
<b>Motores pololu 30:1</b>		-Son los encargados de mover a Rubus a su destino, puede avanzar hacia adelante una cierta cantidad de tiempo, girar sobre su propio terreno hacia la izquierda o derecha.
<b>Bluetooth hc-06</b>		-Es el intermediario entre Rubus y la App, este recibe las ordenes de la App y las envía hacia Rubus, también envía mensajes hacia la app desde Rubus.
<b>Sensor de color TCS3200</b>		-Es quien censa el área por donde Rubus se está desplazando.

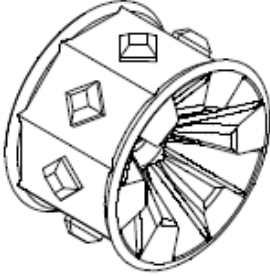
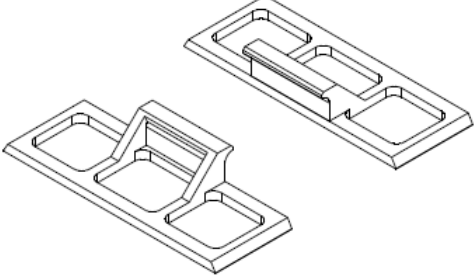


<b>Batería lipo 7.4v</b>		- Alimenta a Rubus y a todos los elementos que son parte del mismo.
--------------------------	---	---

### Partes Case

Para el Case se procede con la impresión del diseño dividido en las siguientes partes:

**Tabla 10.** Partes Case Rubus  
Elaborado por: Christian Caicedo

<b>Case</b>	
<b>Parte del case</b>	<b>Imagen</b>
<b>Chasis</b>	
<b>Ruedas Maestras</b>	

<p><b>Ruedas esclavas</b></p>	
<p><b>Sujetadores de motor</b></p>	
<p><b>Uniones para llanta</b></p>	
<p><b>Enlaces para llanta en forma de cadena</b></p>	

Los costos que el prototipo tendrá son descritos a continuación, la tabla 11 especifica todos los materiales que se utilizaron en la elaboración del prototipo Rubus.

**Tabla 11.** Costos en la elaboración de Rubus.  
Elaborado por: Christian Caicedo

Nombre	Cantidad	Precio unitario	Subtotal
Baby Orangutan B-328 Robot Controller	1	24.50\$	24.50\$
Motores pololu 30:1	2	20\$	40\$
Bluetooth hc-06	1	10\$	10\$
Sensor de color TCS3200	1	8\$	8\$
Batería lipo 7.4v	1	18\$	18\$
Case	1	70\$	70\$
		Total	170.50

#### 4.3.3. Fusión del proyecto

En este punto se une la aplicación con el prototipo para probar su funcionalidad. Rubus, cumple órdenes enviadas por el usuario final, en este caso, un niño, mediante el uso de la aplicación Android desarrollada, atendiendo de manera inmediata a una comunicación bluetooth que a su vez corresponde a una comunicación entre tramas seriales de una conexión maestro-esclavo, en donde el maestro del sistema es el dispositivo móvil que se utilice, y el esclavo corresponde al prototipo electrónico que se encontrará dentro de una alfombra diseñada exclusivamente para el desplazamiento interno de Rubus. Cada orden que se

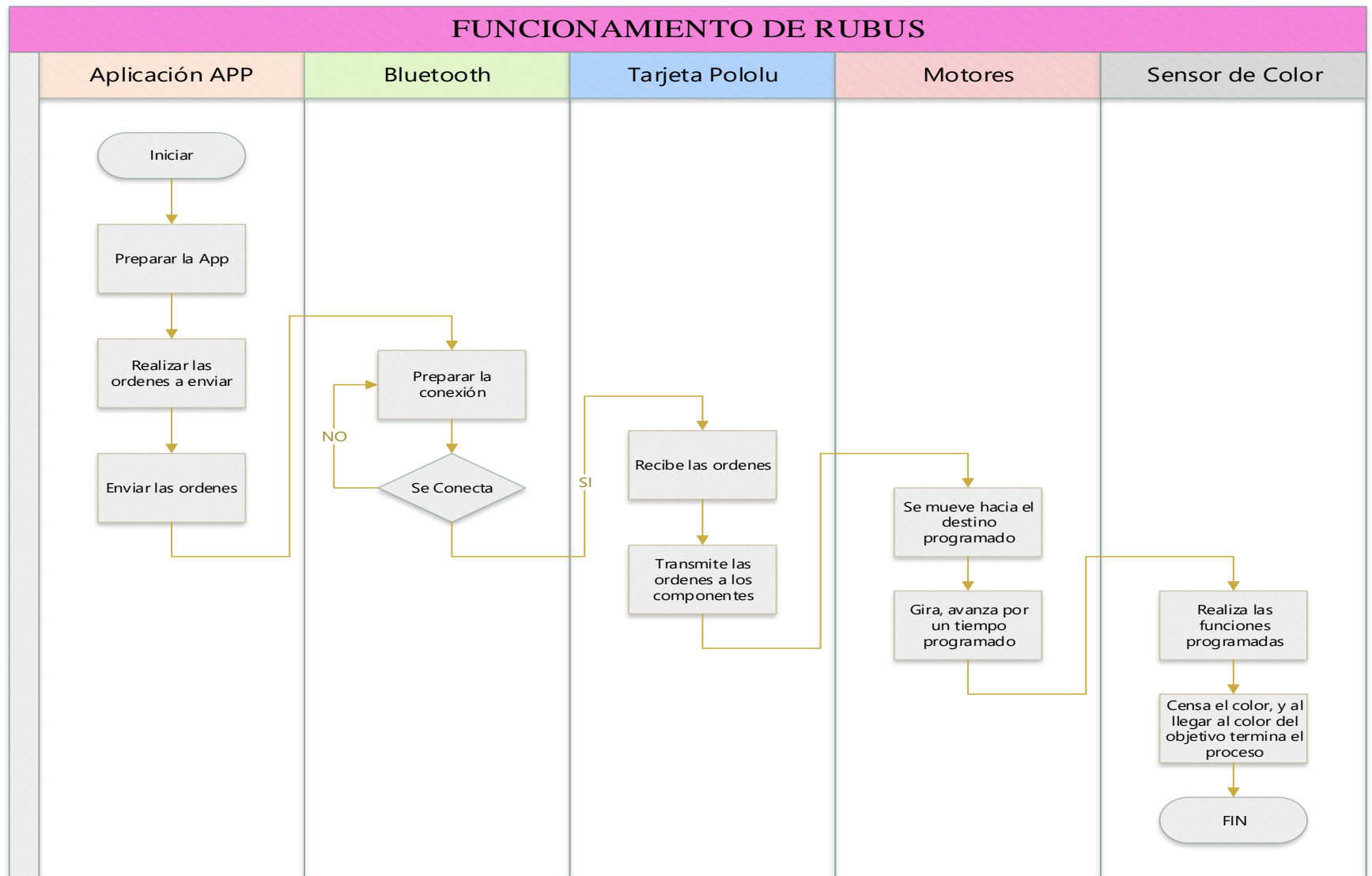
encuentre dentro del contenedor de acciones de la interfaz que presenta la aplicación es enviada hacia el robot mediante el botón de envío, si esto ocurre el robot realizará dichas acciones de forma inmediata en base al número de ciclos programados con anterioridad en la aplicación.

El objetivo de Rubus es llegar a la cuadrícula de color morado que se encuentra en la alfombra de desplazamiento escogiendo un camino óptimo para el robot mediante la programación manual o por medio de comandos de voz de movimientos de desplazamiento hacia: adelante, izquierda y derecha; a su vez dicha alfombra cuenta con obstáculos de color rojo, si Rubus tropieza con ellos todo el sistema se detiene y el robot deja de funcionar, lo mismo que sucede en caso de que el robot detecte el color celeste de los bordes de la alfombra, esto permite que el robot en ningún momento salga de la alfombra de desplazamiento.

Rubus atiende a un proceso de trabajo claramente diferenciado, el cual se puede apreciar en la tabla 12 mostrada a continuación, la cual contiene el algoritmo de funcionamiento de Rubus.

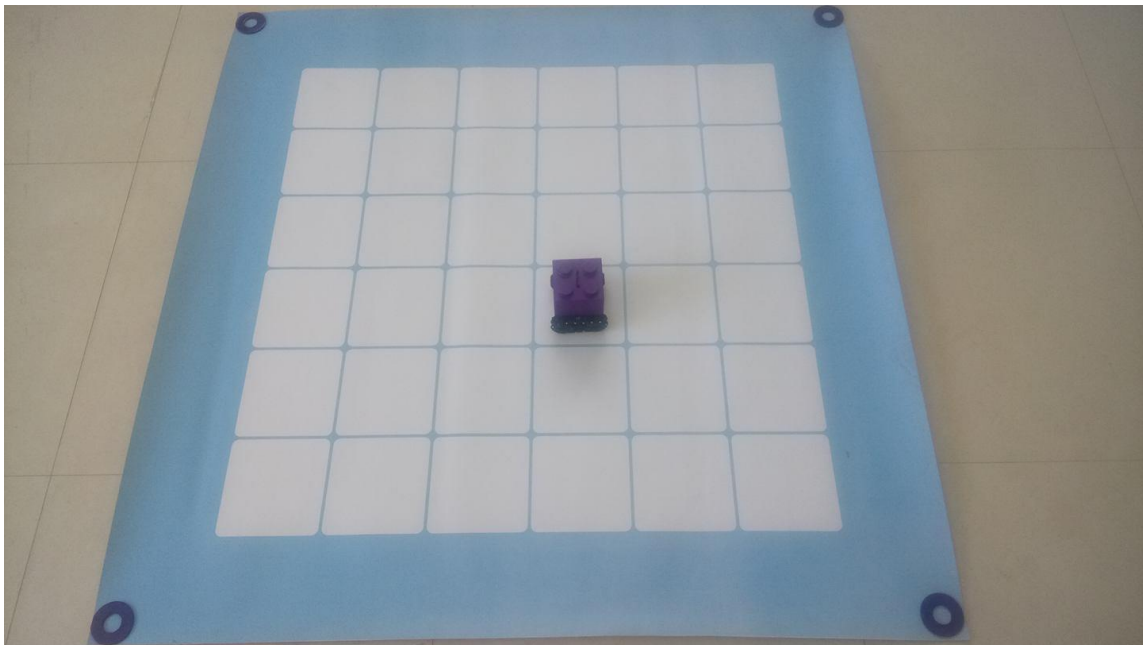


**Tabla 12.** Procedimiento Trabajo Rubus  
Elaborado por: Christian Caicedo



#### 4.3.4. Análisis del entorno donde funciona el prototipo

Rubus desempeña las órdenes que recibe por medio de la conexión bluetooth entre la app y el prototipo, el cual se desplaza en una alfombra especialmente diseñada para ello, debido a que sus movimientos no pueden ser realizados en cualquier superficie a causa de las características propias del case y las ruedas del prototipo, las cuales al estar impresas en material PLA no poseen la tracción suficiente para poder desplazarse en una superficie lisa, por lo que se requiere de una alfombra de desplazamiento cuya superficie tenga un cierto porcentaje de rugosidad pero a la vez se muestre idónea para realizar una impresión en ella, es por esta razón que se realizaron pruebas en distintas superficies tales como papel, plástico, cuerina y cartón, por medio de las cuales se constata que el material óptimo para la impresión de la base de desplazamiento de Rubus es cuerina. Por lo tanto, la alfombra se imprime en este material y tiene unas dimensiones de 123x123cm, internamente se encuentra subdividido en 36 cuadros de 15x15cm en donde desempeña las órdenes de desplazamiento del prototipo Rubus como se muestra a continuación en la figura 49.



**Figura 49.** Superficie de trabajo Rubus  
Elaborado por: Christian Caicedo

#### **4.4. Implementación del prototipo que facilita el aprendizaje de lógica de programación en niños de 4 a 10 años de edad**

Para determinar que el prototipo facilita el aprendizaje de lógica de programación en niños se debe tomar en cuenta 2 escenarios, el conocimiento de lógica de programación que tienen los niños antes y después de una capacitación utilizando Rubus como herramienta de apoyo.

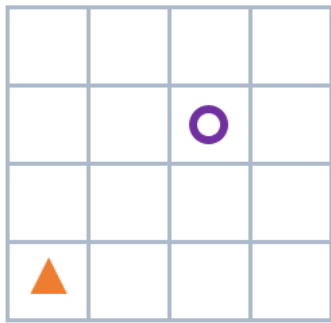
Lógica de programación se define como la capacidad del ser humano para generar algoritmos ya sean computacionales o informales, teniendo en cuenta esto, se realiza un test para evaluar el conocimiento que tienen los niños acerca de elaboración de algoritmos en la Unidad Educativa Luis A. Martínez en la ciudad de Ambato, dichas pruebas fueron ejecutadas con 29 niños del sexto año de educación básica paralelo “A” de la jornada matutina, estos estudiantes recibieron instrucciones previas a la prueba sobre el tema a tratar [10].

##### **4.4.1. Evaluación sobre conocimientos de lógica de programación a niños**

Esta evaluación busca determinar el conocimiento que tienen los niños sobre secuencias, condicionales y ciclos, aspectos fundamentales en la generación de algoritmos, así como también la capacidad que tienen los mismos para generar algoritmos informales. Se realizaron 5 preguntas cada pregunta con una valoración de 2 puntos para así dar un total de 10.

A continuación, se detallan los resultados obtenidos mediante el test aplicado los niños previa a la capacitación e interacción con Rubus.

1) Seleccione la secuencia de pasos (algoritmo) para que triángulo llegue a círculo.



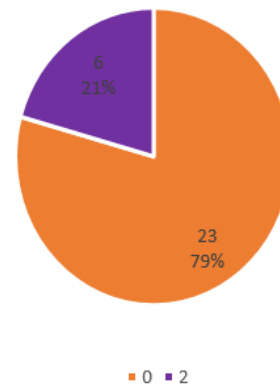
**Figura 50.** Gráfico pregunta 1  
Elaborado por: Christian Caicedo

- a) Adelante, adelante, girar derecha, adelante, adelante
- b) Adelante, girar derecha, girar izquierda, adelante
- c) Girar derecha, girar derecha, girar izquierda, girar izquierda

Por las características de esta pregunta únicamente puede tener 2 posibles resultados, 2 puntos de ser contestada correctamente y 0 de no.

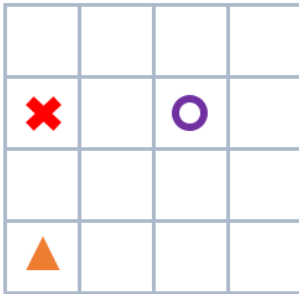
23 niños del total de 29 no pudieron seleccionar el algoritmo correcto de movimientos lo que nos indica desconocimiento en secuencias

Resultados pregunta 1



**Figura 51.** Resultados pregunta 1 previo Rubus  
Elaborado por: Christian Caicedo

2) Escriba la secuencia de pasos que triángulo debe realizar para llegar a círculo, evite que triángulo pase por X.



Secuencia:

.....

.....

.....

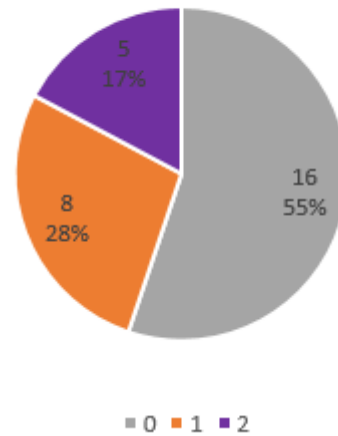
.....

**Figura 52.** Gráfico pregunta 2  
Elaborado por: Christian Caicedo

Al dar opción a que los niños generen su propio algoritmo se debe dar crédito a aquellos que parcialmente cumplen el objetivo así que esta pregunta tiene 3 valoraciones.

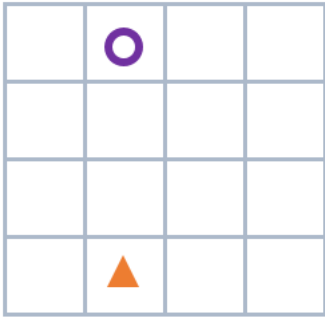
5 niños obtuvieron una valoración de 2, 8 una valoración de 1 y 16 un resultado negativo, El objetivo de esta pregunta era determinar el conocimiento de los niños en el ámbito condicionales

### Resultados pregunta 2



**Figura 53.** Resultados pregunta 2 previo Rubus  
Elaborado por: Christian Caicedo

3) Escriba V (verdadero) o F (falso).



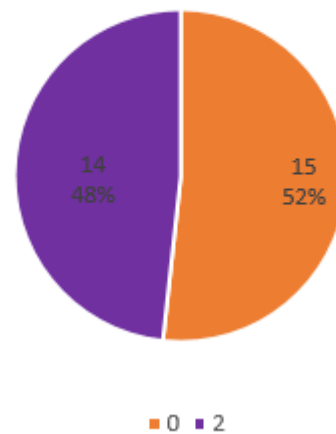
Triángulo debe avanzar 2 pasos hacia adelante para llegar a círculo ( ).

**Figura 54.** Gráfico pregunta 3  
Elaborado por: Christian Caicedo

La naturalidad de esta pregunta permite una valoración de entre 0 y 2 únicamente.

El objetivo de esta pregunta era determinar el conocimiento de ciclos, 14 de los 29 niños respondieron correctamente lo que indica un conocimiento parcial en el ámbito.

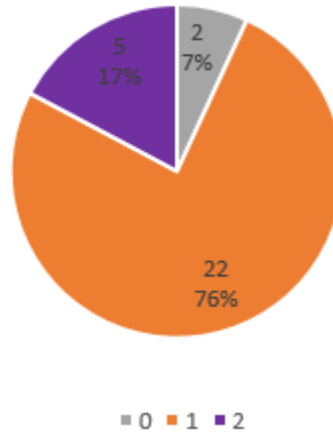
Resultados pregunta 3



**Figura 55.** Resultados pregunta 3 previo Rubus  
Elaborado por: Christian Caicedo

- 4) Describa la secuencia de pasos que usted realiza para llegar a la escuela desde su casa por las mañanas.

#### Resultados pregunta 4



**Figura 56.** Resultados pregunta 4 previo Rubus  
Elaborado por: Christian Caicedo

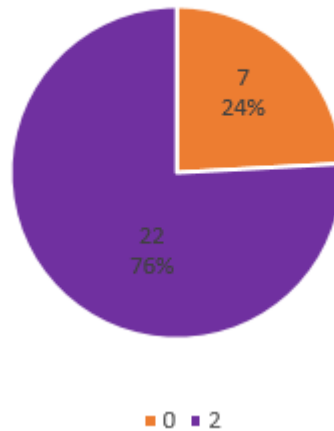
La naturalidad de esta pregunta permite asignar 3 valoraciones 0 de no cumplir el objetivo, 1 de cumplir parcialmente el objetivo y 2 de cumplirlo totalmente.

El objetivo de esta pregunta era determinar el conocimiento de los niños sobre algoritmos informales (algoritmo ejecutado por seres humanos), 17% de estudiantes lo lograron con éxito, 76% lo lograron parcialmente y únicamente el 2% no lo lograron, esto indica que los estudiantes inconscientemente se encuentran capaces de generar algoritmos informales, la mayoría de estudiantes no detallaron a profundidad los pasos que realizan para llegar a la escuela de allí el porqué de su nota.

5) Ordene los pasos para realizar una llamada telefónica.

- a) Levantar el teléfono.
- b) Colgar el teléfono.
- c) Hablar con la otra persona.
- d) Esperar el tono de marcado.
- e) Marcar el número.
- f) Esperar que la otra persona conteste.

### Resultados pregunta 5



**Figura 57.** Resultados pregunta 5 previo Rubus  
Elaborado por: Christian Caicedo

La naturalidad de esta pregunta permite una valoración de entre 0 y 2 únicamente.

El objetivo de esta pregunta era determinar la capacidad que tienen los niños para realizar tareas en un orden lógico. El 76% de los niños respondieron con certeza esta pregunta lo cual nos indica que los niños son capaces de realizar tareas de una manera ordenada.



#### **4.4.2. Resultado de evaluación previa a capacitación con Rubus**

Previa a la capacitación de los niños en lógica de programación utilizando a Rubus como herramienta de apoyo se constata que ellos no se centran en los procesos básicos de un algoritmo cotidiano si no mas bien en cumplir el objetivo del mismo. Los estudiantes obtienen un promedio de 4.62/10 lo que indica que los niños parcialmente tienen conocimiento sobre generación de algoritmos, lo único que haría falta es profundizar en conceptos y así su lógica de programación será potencializada, para ello se utiliza a Rubus como herramienta para facilitar su aprendizaje.

#### **4.4.3. Capacitación a niños en lógica de programación con Rubus**

Los niños del sexto “A” de la Unidad Educativa Luis A. Martínez realizaron un taller de 3 horas en el cual se profundizaron conceptos de algoritmos tanto computacionales utilizando a Rubus y además informales, como punto adicional se habló de secuencias, condicionales y ciclos todo ello utilizando a Rubus como ejemplo y cubriendo así aspectos utilizados en Lógica de programación. Esta actividad se ve reflejada en la figura 58.



**Figura 58.** Capacitación previa a los niños  
Elaborado por: Christian Caicedo

Los niños conocieron a Rubus junto con su funcionamiento, además interactuaron con la aplicación móvil, se aclararon dudas presentes en ellos, esto se realizó en pequeños grupos con la finalidad de asegurar una comprensión total de la operatividad del robot y conocer el nivel de aceptabilidad del mismo, tal como se muestra en la figura 59.



**Figura 59.** Presentación por grupos la App de Rubus  
Elaborado por: Christian Caicedo

Los niños realizaron en su libreta de tareas algoritmos de movimiento para Rubus, en base a la explicación del funcionamiento previamente realizado, Aquel estudiante que lo realizaba con éxito podía programar dicha secuencia en la aplicación y verla reflejada en el robot. En la figura 60 un estudiante se encuentra utilizando el prototipo.



**Figura 60.** Niños utilizando a Rubus  
Elaborado por: Christian Caicedo

Rubus fue aceptado por la mayor parte de niños debido a su interfaz interactiva y al juguete en sí, ya que muy pocas veces un niño tiene la posibilidad de interactuar con un robot en su etapa de pruebas.

En la figura 61 se muestra la etapa final en donde se puede analizar que las pruebas realizadas con los niños de la Unidad Educativa Luis A. Martínez reflejan que los estudiantes se encontraron deseosos de participar en la capacitación utilizando a Rubus como herramienta de apoyo.



**Figura 61.** Resultados de la interacción entre los niños y el prototipo  
Elaborado por: Christian Caicedo

El nombre del prototipo causó agrado en los niños, quienes realizaron preguntas sobre la razón de su nombre, esta inquietud fue resuelta mediante la explicación de la relación que existe entre la palabra Rubus y el color morado el cual es color preferente del robot.

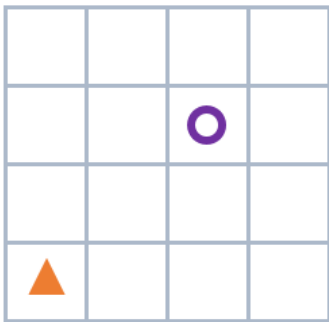
Otra inquietud que se presentó fue el costo del mismo, ante lo cual se explicó que al ser un prototipo tiene un costo elevado ya que es único, pero al llevarlo a un nivel de producción masiva incrementa su accesibilidad al público.

#### 4.4.4. Evaluación sobre conocimientos de lógica de programación a niños posterior a la capacitación utilizando a Rubus como herramienta de apoyo

Posterior a la capacitación, los niños fueron evaluados nuevamente con el mismo instrumento de evaluación realizada previa a su formación en conceptos de lógica de programación.

A continuación, se detallan los resultados obtenidos mediante el test aplicado los niños previa a la capacitación e interacción con Rubus.

- 1) Seleccione la secuencia de pasos (algoritmo) para que triángulo llegue a círculo.



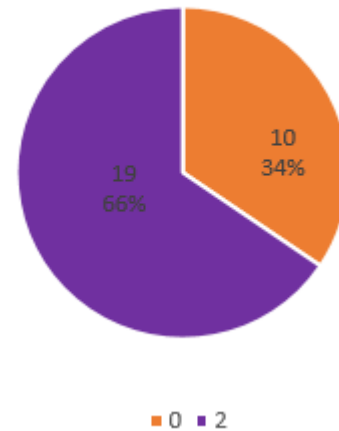
**Figura 62.** Gráfico pregunta 1  
Elaborado por: Christian Caicedo

- a) Adelante, adelante, girar derecha, adelante, adelante
- b) Adelante, girar derecha, girar izquierda, adelante
- c) Girar derecha, girar derecha, girar izquierda, girar izquierda

Por las características de esta pregunta únicamente puede tener 2 posibles resultados, 2 puntos de ser contestada correctamente y 0 de no.

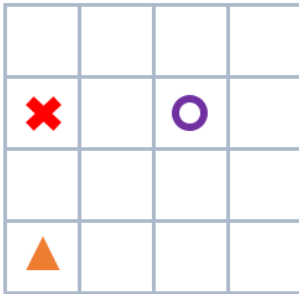
En esta ocasión 19 de los 29 niños seleccionaron el algoritmo de movimientos correcto para llegar al objetivo. Esto indica que su conocimiento en secuencias se incrementó.

**Resultados pregunta 1**



**Figura 63.** Resultados pregunta 1 posterior a Rubus  
Elaborado por: Christian Caicedo

2) Escriba la secuencia de pasos que triángulo debe realizar para llegar a círculo, evite que triángulo pase por X.



Secuencia:

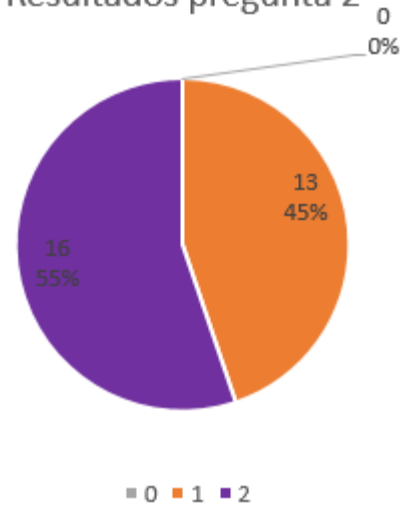
.....  
 .....  
 .....  
 .....

**Figura 64.** Gráfico pregunta 2  
 Elaborado por: Christian Caicedo

Al dar opción a que los niños generen su propio algoritmo se debe dar crédito a aquellos que parcialmente cumplen el objetivo así que esta pregunta tiene 3 valoraciones.

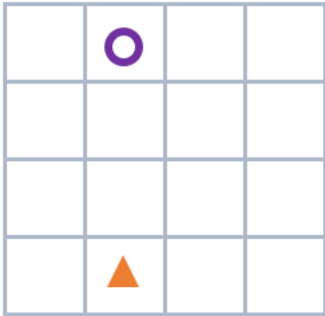
En esta ocasión ningún niño obtiene una valoración de 0, todos los niños realizaron parcial o totalmente el algoritmo de movimiento para llegar a su objetivo. Su conocimiento en el ámbito condicionales fue incrementado utilizando a Rubus en su capacitación.

Resultados pregunta 2



**Figura 65.** Resultados pregunta 2 posterior a Rubus  
 Elaborado por: Christian Caicedo

3) Escriba V (verdadero) o F (falso).



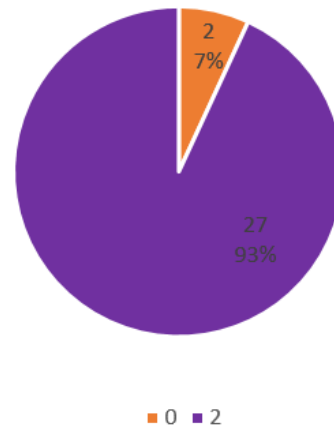
Triángulo debe avanzar 2 pasos hacia adelante para llegar a círculo ( ).

**Figura 66.** Gráfico pregunta 3  
Elaborado por: Christian Caicedo

La naturalidad de esta pregunta permite una valoración de entre 0 y 2 únicamente.

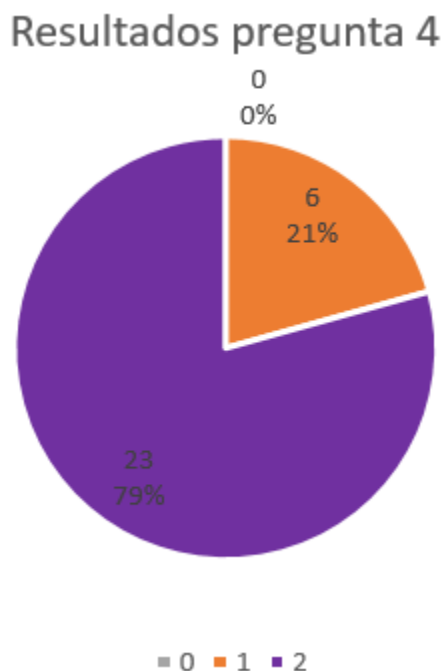
Esta pregunta tuvo un incremento del 45% en estudiantes en el tema ciclos.

Resultados pregunta 3



**Figura 67.** Resultados pregunta 3 posterior a Rubus  
Elaborado por: Christian Caicedo

- 4) Describa la secuencia de pasos que usted realiza para llegar a la escuela desde su casa por las mañanas.



**Figura 68.** Resultados pregunta 4 posterior a Rubus  
Elaborado por: Christian Caicedo

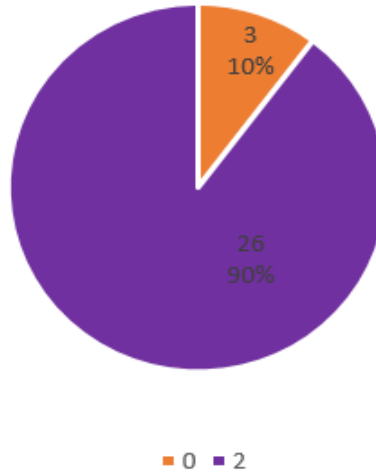
La naturalidad de esta pregunta permite asignar 3 valoraciones 0 de no cumplir el objetivo, 1 de cumplir parcialmente el objetivo y 2 de cumplirlo totalmente.

Nuevamente no se encuentra un resultado negativo lo cual demuestra el interés de los niños en la capacitación. El 79% de los niños fueron más detallistas al momento de describir las actividades que realizan para llegar a la escuela partiendo desde sus hogares mientras que el 21 % restante cumplieron con el objetivo determinado parcialmente ya que detallaron actividades que no eran necesarias para el cumplimiento del objetivo.

5) Ordene los pasos para realizar una llamada telefónica.

- g) Levantar el teléfono.
- h) Colgar el teléfono.
- i) Hablar con la otra persona.
- j) Esperar el tono de marcado.
- k) Marcar el número.
- l) Esperar que la otra persona conteste.

### Resultados pregunta 5



**Figura 69.** Resultados pregunta 5 posterior a Rubus  
Elaborado por: Christian Caicedo

La naturalidad de esta pregunta permite una valoración de entre 0 y 2 únicamente.

La cantidad de niños que pueden realizar actividades de una manera ordenada se ve incrementada un 14% con respecto a la vez anterior. Este indicador nos muestra que con una profundización en el tema a tratar se ven resultados mucho más óptimos.



#### **4.4.5. Resultado de evaluación posterior a capacitación con Rubus**

Posterior a la capacitación en lógica de programación con Rubus como herramienta de apoyo utilizando la pregunta # 4 como referencia más relevante del test aplicado debido a que en esta pregunta los niños generan su propio algoritmo informal detallándolo paso a paso para cumplir con el objetivo dado el cual fue describir el proceso que realizan los niños para llegar a la escuela desde su hogar. La capacitación junto con Rubus permitió incrementar el número de niños que lograron desarrollar el algoritmo de manera detallada y lógica de un porcentaje inicial del 17% al 79% además la nota promedio general de la evaluación posterior a la capacitación tuvo un incremento de 3.8 puntos (de 4.61/10 a 8.41/10) lo cual ratifica que la interacción Rubus y los niños se ve influenciado en el aprendizaje de lógica de programación.

## **CAPITULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- Ecuador únicamente se cuenta con un prototipo cuya finalidad está orientada a enseñar electrónica, robótica y programación a niños, este prototipo se denomina Teebot, en la actualidad ya no se fabrican más prototipos de este robot, por lo que puede considerarse como un proyecto culminado sin mayor impacto. La mayor ventaja que presenta Rubus frente a Teebot es que el Segundo requiere una conexión a través de un medio guiado para su programación, mientras que Rubus utiliza un medio inalámbrico para su comunicación y codificación.
  
- La aplicación móvil muestra una interfaz adecuada para los usuarios finales, que son los niños, utilizando un giroscopio, reconocimiento por voz, vibración y animaciones con colores llamativos, para lo cual utiliza una metodología de desarrollo por prototipos, la cual permite seleccionar las mejores características para la interfaz final, creando así un control interactivo entre la aplicación y el prototipo.
  
- El prototipo electrónico es escalable, ya que al utilizar una metodología bottom-up se divide el funcionamiento del prototipo en subsistemas funcionales dentro de los cuales se encuentran los siguientes módulos: alimentación, comunicación, movimiento y sensorización, incluso es posible incrementar a futuro las funcionalidades del robot adicionando sensores y actuadores.

- Las pruebas realizadas con los niños de la Unidad Educativa Luis A. Martínez muestran un incremento de 3.8 puntos en la nota promedio del curso (de 4.61/10 a 8.41/10) lo cual ratifica que la interacción Rubus y los niños se ve influenciado en el aprendizaje de lógica de programación.

### **Recomendaciones**

- Es preciso realizar una sensorización inicial de color para dotarle al sistema de una estabilidad completa, puesto que cada ambiente presenta condiciones diferentes de iluminación.
- Rubus atiende a un giro de 90° en el sentido especificado; estas acciones tienen un porcentaje de precisión del 90% debido a factores externos, lo cual puede mejorarse incrementando sensores en el subsistema de movimiento.
- La investigación puede ser ampliada para implementar nuevas funcionalidades tanto en la aplicación como en el prototipo, por ejemplo, es factible agregar un encoder para lograr precisión del 99% en los movimientos de Rubus.

## Bibliografía

- [1] E. Rodriguez, «Trastornos del aprendizaje. Estilos de aprendizaje y el diagnóstico psicopedagógico,» *Portal de la red de Salud de CUba*, vol. I, 2004.
- [2] S. Figueroa, «Los niños que sí sabían programar,» 16 Abril 2013.
- [3] J. Velasco, «Niños programadores: para qué sirve la enseñanza de programación en las escuelas,» 19 Agosto 2014.
- [4] J. Cervera, «Adquisición y desarrollo del lenguaje en Preescolar y Ciclo Inicial,» 10 Junio 2011. [En línea]. Available: [http://www.cervantesvirtual.com/obra-visor/adquisicin-y-desarrollo-del-lenguaje-en-preescolar-y-ciclo-inicial-0/html/ffbcaf2e-82b1-11df-acc7-002185ce6064\\_1.html](http://www.cervantesvirtual.com/obra-visor/adquisicin-y-desarrollo-del-lenguaje-en-preescolar-y-ciclo-inicial-0/html/ffbcaf2e-82b1-11df-acc7-002185ce6064_1.html). [Último acceso: 17 Julio 2017].
- [5] W. Merino, «Yachay lanzó Teebot, el primer robot 100% hecho en Ecuador,» *Vistazo*, 02 Abril 2015.
- [6] N. Velasquez, D. Olsen, M. Taylor y Y. Kafay, «Novice Programmers Talking about Projects: What Automated Text Analysis Reveals about Online Scratch Users' Comments,» *IeeeXplore*, vol. I, nº 1, pp. 14-25, 2014.
- [7] J. Diaz, S. Martin y J. Bogado, «Manual de programacion con robots para la escuela,» Soporte Liuen, Madrid, 2012.
- [8] R. Hussein, «Videojuego educativo en 3d para dispositivos móviles android, enfocado al aprendizaje de la logica de programacion para usuarios entre los 5 a 18 años de edad,» Universidad Tecnica de Ambato, Ambato, 2017.
- [9] H. C., «Essentials of Logic Programming,» Oxford, 1990.
- [10] O. Trejos, «La escencia de la logica de programacion,» Madrid, 2000, pp. 45-88.
- [11] J. Garcia, «Programacion estructurada,» *Pearson Prentice Hall*, vol. I, pp. 12-25, 2008.
- [12] G. Unellezes, «Lenguajes de Programación,» Bogota, 2010, pp. 46-67.
- [13] M. Tahaual, «Lenguajes de Programación,» 2014.
- [14] S. Augarten, «The Most Widely Used Computer on a Chip: The TMS 1000.,» *Ticknor & Fields*, pp. 12-23, 2012.

- [15] H. Lopez, «Porque son útiles para comunicar, discutir y definir ideas entre los diseñadores y las partes responsables.,» *Social Media Strategial*, vol. I, pp. 45-55, 2015.
- [16] I. Marcano y R. Manuel, «Método de análisis de problemas algorítmicos (PyR),» *Espacios*, vol. 34, n° 3, pp. 20-32, 2013.
- [17] G. Chin, «Agile Project Management: How to Succeed in the Face of Changing Project Requirements,» *Amacon*, vol. II, pp. 40-55, 2004.
- [18] C. Moises, «Prezi,» 24 Septiembre 2014. [En línea]. Available: <https://prezi.com/flbgs4spwusj/modelos-cascada-espiral-y-prototipos/>. [Último acceso: 29 Noviembre 2017].
- [19] C. Schweers, «Metodologías de diseño de hardware,» de *Descripción en VHDL de arquitecturas*, Lima, 2002, pp. 14-19.
- [20] I. Alarcón, «Presentan un robot educativo hecho en el Ecuador,» 2015 Abril 2015.
- [21] A. Mora, Instituto Ecuatoriano de la Propiedad Intelectual, 6 Febrero 2015. [En línea]. Available: <https://www.propiedadintelectual.gob.ec/teebot-el-robot-que-ensena-programacion-y-robotica-a-ninos/>. [Último acceso: 10 Diciembre 2017].
- [22] P. El Universo, «Teebot, un robot hecho en Ecuador para aprender,» 13 Abril 2015.
- [23] M. Rouse, «TechTarget,» Diciembre 2016. [En línea]. Available: <http://searchdatacenter.techtarget.com/es/definicion/Framework>. [Último acceso: 20 Diciembre 2017].
- [24] A. Studio, «Android Studio realease notes,» 3 Octubre 2017. [En línea]. Available: <https://developer.android.com/studio/releases/index.html>. [Último acceso: 27 Diciembre 2017].
- [25] D. Gallardo, «IBM,» 16 Noviembre 2012. [En línea]. Available: <https://www.ibm.com/developerworks/ssa/library/os-ecov/index.html>. [Último acceso: 27 Diciembre 2017].
- [26] R. Cardenas, «Guia de Iniciacion de APP Inventor,» Cali, 2015.
- [27] Linux, «Linux Droids Blog,» 5 Marzo 2015. [En línea]. Available: <https://linuxdroids.com/2015/03/05/tutorial-de-programacion-de-pics-en-ensamblador/>. [Último acceso: 27 Diciembre 2017].

- [28] Arduino, «BricoGeek,» 12 Agosto 2012. [En línea]. Available: <http://tienda.bricogeek.com/arduino/394-arduino-mega-adk.html>. [Último acceso: 27 Diciembre 2017].
- [29] Pololu, 2014. [En línea]. Available: <https://www.pololu.com/file/0J328/OrangutanSVPGuiaDeUsuario.pdf>. [Último acceso: 28 Diciembre 2017].

# **Anexos**

**Anexo 1 Encuesta dirigida a los padres de familia**





## Encuesta Juguete Rubus

La presente encuesta tiene como objetivo determinar en qué sistema operativo se realizará la aplicación que controle al juguete interactivo Rubus dirigida a padres de familia de los estudiantes del sexto “B” de la Unidad Educativa Luis A. Martínez

Encuesta elaborada por: Christian Caicedo

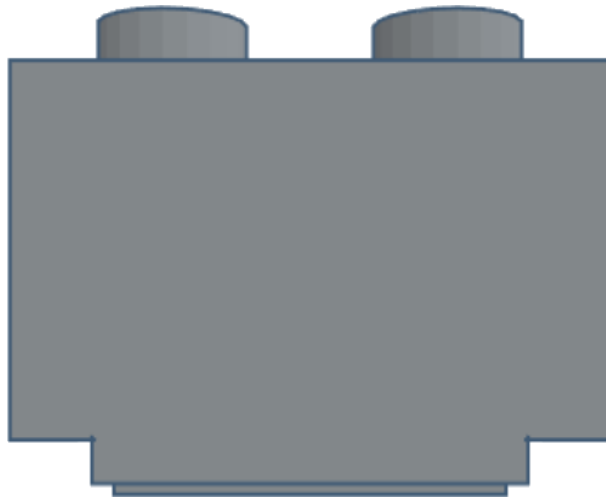
**1. ¿En qué dispositivo le gustaría que su hijo aprendiera a programar? Marque una o varias opciones.**

- Pc o Laptop
- Tablet - Smartphone

**2. ¿El dispositivo móvil que usted posee es? Marque solo una opción.**

- Samsung, Sony, Huawei, Lg, Xiaomi
- iPhone
- Otros: .....

## **Anexo 2 Planos del Case Rubus**



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

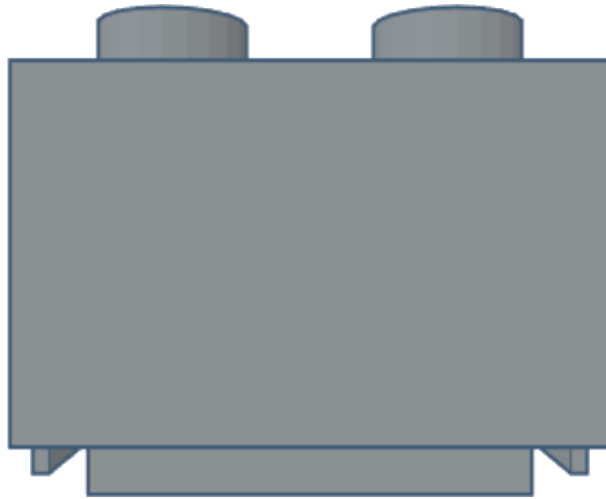
**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista frontal Tapa (Rubus)

**LAMINA:**

1/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

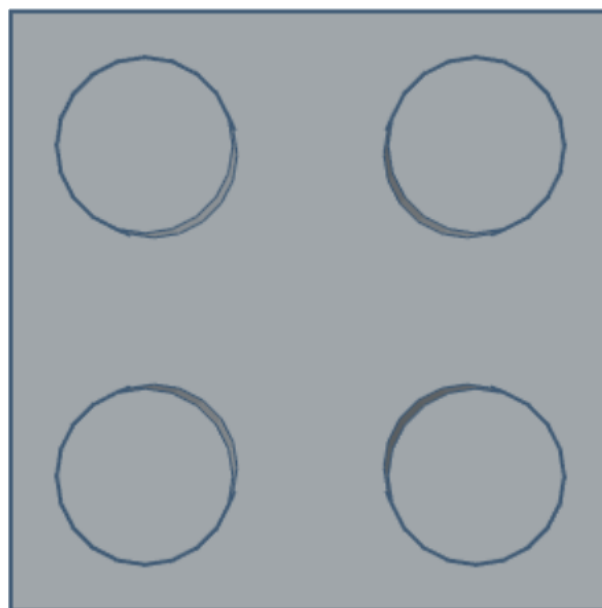
**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Lateral Tapa (Rubus)

**LAMINA:**

2/15

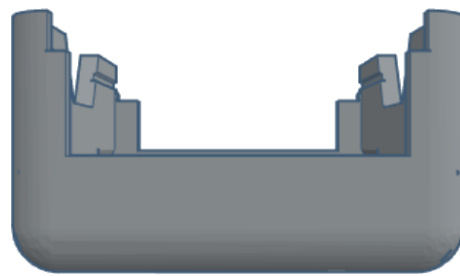


**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

<b>ESTUDIANTE:</b> Christian Patricio Caicedo Barreno	<b>TUTOR:</b> Ing. Edison Álvarez	<b>ESCALA:</b> 1:1
		<b>FECHA:</b> Enero / 2018
		<b>DIRECCIÓN:</b> AMBATO
<b>CONTENIDO:</b> Vista Superior Tapa (Rubus)		<b>LAMINA:</b> 3/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Frontal Base (Rubus)

**LAMINA:**

4/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

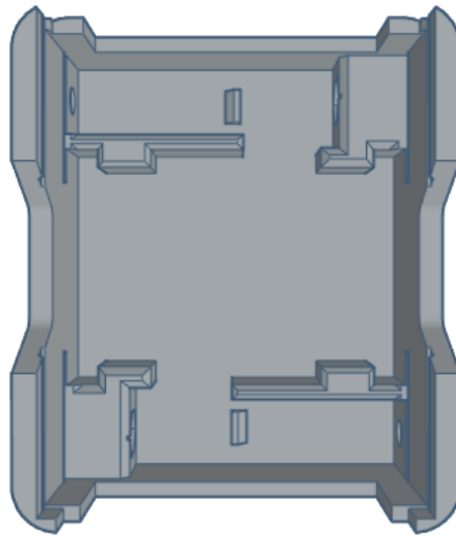
**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Lateral Base (Rubus)

**LAMINA:**

5/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

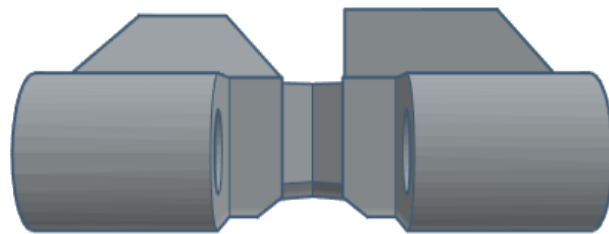
**CONTENIDO:**

Vista Superior Base(Rubus)

**LAMINA:**

6/15



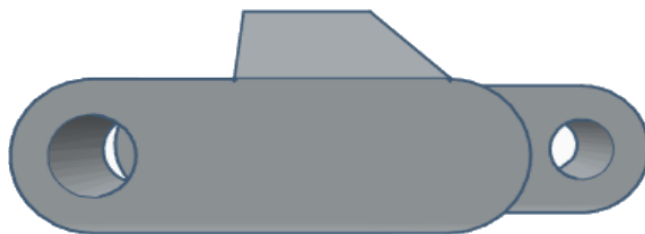


**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

<b>ESTUDIANTE:</b> Christian Patricio Caicedo Barreno	<b>TUTOR:</b> Ing. Edison Álvarez	<b>ESCALA:</b> 5:1
		<b>FECHA:</b> Enero / 2018
		<b>DIRECCIÓN:</b> AMBATO
<b>CONTENIDO:</b> Vista Frontal Enlace (Rubus)		<b>LAMINA:</b> 7/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 5:1

**FECHA:** Enero / 2018

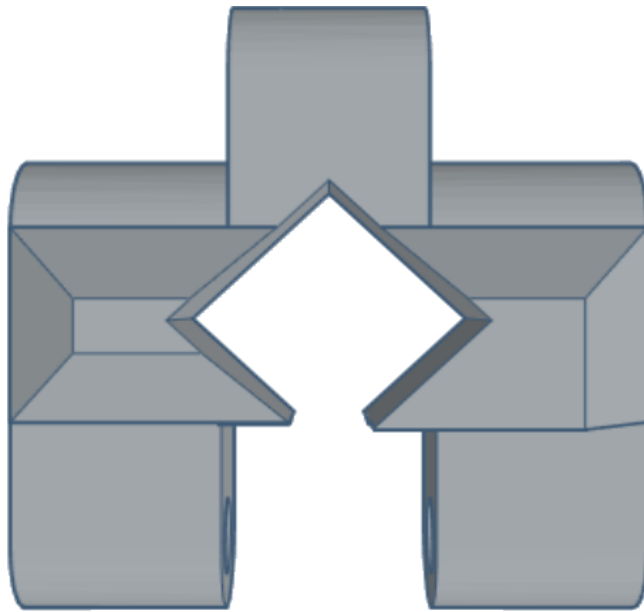
**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Lateral Enlace (Rubus)

**LAMINA:**

8/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 5:1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Superior Enlace(Rubus)

**LAMINA:**

9/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

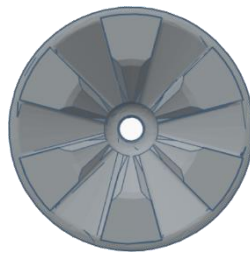
**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Frontal Llanta Esclava (Rubus)

**LAMINA:**

10/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Lateral Llanta Esclava (Rubus)

**LAMINA:**

11/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1 :1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Superior Llanta Esclava (Rubus)

**LAMINA:**

12/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

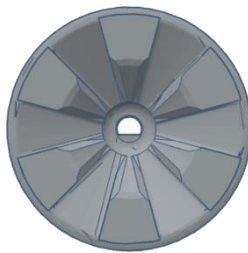
**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Frontal Llanta Maestra (Rubus)

**LAMINA:**

13/15



**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Lateral Llanta Maestra (Rubus)

**LAMINA:**

14/15





**UNIVERSIDAD TÉCNICA DE AMBATO**

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**PROYECTO DE TITULACIÓN**

**ESTUDIANTE:**

Christian Patricio Caicedo Barreno

**TUTOR:**

Ing. Edison Álvarez

**ESCALA:** 1:1

**FECHA:** Enero / 2018

**DIRECCIÓN:** AMBATO

**CONTENIDO:**

Vista Superior Llanta Maestra (Rubus)

**LAMINA:**

15/15

**Anexo 3 fotografías de las pruebas realizadas con Rubus en la Unidad Educativa Luis  
A. Martínez de la ciudad de Ambato**

**Tabla 13.** Fotografías de presentación Rubus con los clientes Capacitación Previa

**CAPACITACIÓN PREVIA**



**Tabla 14.** Fotografías de presentación Rubus con los clientes demostración y ejercicios



**Tabla 15.** Fotografías de presentación Rubus con los clientes culminación de la presentación

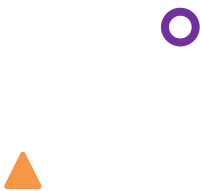
**CULMINACIÓN DE LA PRESENTACIÓN**




**Anexo 4 Test para determinar conocimiento de lógica de programación aplicada a niños**

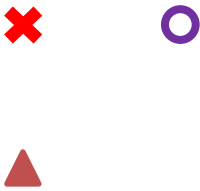
Test para comprobar conocimiento lógico de programación

1) Seleccione la secuencia de pasos (algoritmo) para que triángulo llegue a círculo.



- a) Adelante, adelante, girar derecha, adelante, adelante
- b) Adelante, girar derecha, girar izquierda, adelante
- c) Girar derecha, girar derecha, girar izquierda, girar izquierda

2) Escriba la secuencia de pasos que triángulo debe realizar para llegar a círculo, evite que triángulo pase por 



Secuencia:

.....  
.....  
.....  
.....

3) Escriba **V** (verdadero) o **F** (falso).



Triángulo debe avanzar 2 pasos hacia adelante para llegar a círculo ( ).



- 4) Describa la secuencia de pasos que usted realiza para llegar a la escuela desde su casa por las mañanas.

Secuencia:

.....  
.....  
.....  
.....  
.....

- 5) Ordene los pasos para realizar una llamada telefónica.

- m) Levantar el teléfono.
- n) Colgar el teléfono.
- o) Hablar con la otra persona.
- p) Esperar el tono de marcado.
- q) Marcar el número.
- r) Esperar que la otra persona conteste.



## **Anexo 5 programación de Rubus**

## **Programación del prototipo**

[https://drive.google.com/open?id=1V\\_1a-gbO7pHtkxDWQRUA\\_12i2s4zFjO4](https://drive.google.com/open?id=1V_1a-gbO7pHtkxDWQRUA_12i2s4zFjO4)

## **Programación de la app en Android Studio**

[https://drive.google.com/open?id=1B9AoElwL\\_CtVST1NKADpo0Sxg6AlssmG](https://drive.google.com/open?id=1B9AoElwL_CtVST1NKADpo0Sxg6AlssmG)