



**UNIVERSIDAD TÉCNICA DE COTOPAXI**  
**FACULTAD CIENCIAS DE LA INGENIERÍA Y APLICADAS**  
**CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS**  
**COMPUTACIONALES**

**PROPUESTA TECNOLÓGICA**

**“APLICACIÓN DE SISTEMAS EMBEBIDOS EN LA CONSTRUCCIÓN DE  
ROBOTS MÓVILES”: EL CASO DEL CLUB DE ROBÓTICA DE LA  
UNIVERSIDAD TÉCNICA DE COTOPAXI.**

**AUTOR:**

Sosa Valverde Marco Antonio

**TUTOR:**

PhD. Gustavo Rodríguez Bárcenas

**LATACUNGA-ECUADOR**

**2019 - 2020**

## DECLARACIÓN DE AUTORÍA

Yo, Marco Antonio Sosa Valverde, declaro ser autor del presente proyecto de investigación: **“APLICACIÓN DE SISTEMAS EMBEBIDOS EN LA CONSTRUCCIÓN DE ROBOTS MÓVILES”**; EL CASO DEL CLUB DE ROBÓTICA DE LA UNIVERSIDAD, siendo el **PhD. Gustavo Rodríguez Barcenás** tutor del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.



Marco Antonio Sosa Valverde

1802622728

### **AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN**

En calidad de Tutor del Trabajo de Investigación sobre el título: **“APLICACIÓN DE SISTEMAS EMBEBIDOS EN LA CONSTRUCCIÓN DE ROBOTS MÓVILES”**: EL CASO DEL CLUB DE ROBÓTICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI del señor **MARCO ANTONIO SOSA VALVERDE**, Postulante de la Carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas, de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, 07 de Febrero del 2020

**Tutor de Titulación**  
**PhD. Gustavo Rodriguez**  
CC. 1757001357

## APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la FACULTAD CIENCIAS DE LA INGENIERÍA Y APLICADAS; por cuanto, el postulante Sosa Valverde Marco Antonio con el título de Proyecto de titulación: **"APLICACIÓN DE SISTEMAS EMBEBIDOS EN LA CONSTRUCCIÓN DE ROBOTS MÓVILES"**; EL CASO DEL CLUB DE ROBÓTICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, 07 de febrero del 2020

Para constancia firman:



Lector 1 (Presidente)

Ing. Mg. Oscar Alejandro Guaypatín Pico  
CC: 1802829430



Lector 2

Ing. Mg. Karla Susana Cantuña Flores  
CC: 0502305113



Lector 3

Ing. Mg. Alex Christian Llano Casa  
CC: 0502589864

## AVAL DEL IMPLEMENTACIÓN DEL PROYECTO DE INVESTIGACIÓN

Latacunga, 06 de enero del 2020

Señores

Tribunal de Validación de Proyectos del Consejo Directivo de la Facultad de CIYA

Presente

De mis consideraciones:

Reciban un cordial y atento saludo de quienes laboramos en el Dirección de Investigación de la Universidad Técnica de Cotopaxi.

El presente tiene la finalidad de informar que el estudiante Marco Antonio Sosa Valverde egresado de la Facultad de Ciencias de la Ingeniería y Aplicadas, Carrera de Ingeniería en Informática y Sistemas Computacionales, ha presentado, desarrollado e implementado el tema de tesis "APLICACIÓN DE SISTEMAS EMBEBIDOS EN LA CONSTRUCCIÓN DE ROBOTS MÓVILES": EL CASO DEL CLUB DE ROBÓTICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI, club perteneciente y tutelado por nuestra Dirección, la propuesta tecnológica fue aprobada de manera satisfactoria ya que cumple con los requerimientos metodológicos y constituye un aporte científico técnico para el presente y futuro respaldo en investigaciones similares dentro del ámbito académico de esta institución, quedando constancia el documento físico en la biblioteca de la IES - UTC.

La investigación cumple y acepta las disposiciones establecidas en la Reglamentación de la Dirección de Investigación de la Universidad Técnica de Cotopaxi.

Atentamente,



PhD. Carlos Javier Torres Milla  
C.C. 0502329238

Director de Investigación Universidad Técnica de Cotopaxi

### Dedicatoria

A la Universidad Técnica de Cotopaxi que me abrió sus puertas y ha sido un puntal fundamental para culminar mis estudios de manera correcta, a todos los ingenieros, amigos incondicionales que con sus consejos y sabiduría me encaminaron y apoyaron amigos que han estado presente en las buenas y en las malas, que han sido el aliento para seguir adelante transformando una tristeza en alegría.

### **Agradecimiento**

Al ser divino por bendecirme, para llegar hasta donde he llegado, sé que estuviste presente he hiciste realidad este sueño anhelado. A la Universidad Técnica de Cotopaxi, Carrera de Ingeniería en Sistemas, que con la sabia guía de sus docentes contribuyeron a mi aprendizaje que serán la base que me servirá en mi vida profesional.

A los ingenieros Edwin Vaca, Carolina Villa, Paulina Freire, Secundino Marrero, por sus concejos en la realización de este proyecto.

A mi Tutor de tesis que ha sido una persona íntegra y me apoyado con sus amplios conocimientos para poder llevar a cabo la ejecución de mi proyecto. A todos que directa e indirectamente siempre creyeron en mí y estuvieron en todo momento brindándome su apoyo incondicional.

**UNIVERSIDAD TÉCNICA DE COTOPAXI**  
**CARRERA DE SISTEMAS DE INFORMACIÓN**  
**INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES**

**Título: “Aplicación de sistemas embebidos en la construcción de robots móviles”:** El caso del club de robótica de la Universidad Técnica De Cotopaxi.

**Autor:** Marco Antonio Sosa Valverde

**Tutor:** Gustavo Rodríguez Bárcenas PhD.

**RESUMEN**

El presente proyecto muestra el diseño, construcción y validación de una aplicación micro controlada en robótica móvil denominada “BRICK”, el mismo que tiene como objetivo la programación de un robot a través de un IDE (entorno de desarrollo integrado) con la utilización de software libre y operado por un computador. En la comprobación de esta investigación se expone la importancia de la utilización de electrónica en sistemas embebidos, lo cual evidenció que este tipo de sistemas son flexibles y se pueden adaptar a un propositivo general o específico en base a la utilización de una metodología en proyectos mecatrónicos que son operadas por señales generadas mediante una pantalla táctil o diversos medios como, infrarrojos, bluetooth, comandado desde un celular inteligente, emitiendo órdenes de acuerdo con las necesidades del usuario. Estas actividades accionan el funcionamiento de los actuadores que fueron integrados en la programación en base a algoritmos de comunicación a través de señales inalámbricas.

**Palabras claves:** Micro controlador, electrónica, mecatrónica, bluetooth, robótica, programación, comunicación.

**TECHNICAL UNIVERSITY OF COTOPAXI**  
**CAREER OF INFORMATION SYSTEMS**  
**COMPUTER ENGINEERING AND COMPUTATIONAL SYSTEMS**

**Title: “Application of embedded systems in the construction of mobile robots”:** The case of the robotics club of the Technical University of Cotopaxi.

**Author:** Marco Antonio Sosa Valverde

**Tutor:** Gustavo Rodríguez Bárcenas PhD.

**ABSTRACT**

This project shows the design, construction and validation of a micro-controlled application in mobile robotics called "BRICK". It aims to program a robot through an IDE (integrated development environment) by using free software and operated by a computer. In this research confirmation, the importance of the use of electronics in embedded systems is exposed. It revealed that this type of systems are flexible and can be adapted to a general or specific proposal based on the use of a methodology in mechatronic projects. They are ran by signals controlled in a touch screen or other different methods such as infrared rays or bluetooth, commanded from a Smartphone and giving orders according to the needs of the user. These patterns activate the operation of the devices that were integrated in the robot programming based on algorithms communication through wireless signals.

**Keywords:** Micro controller, electronics, mechatronics, bluetooth, robotics, programming, communication.



## *AVAL DE TRADUCCIÓN*

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal CERTIFICO que: La traducción del resumen ABSTRACT del proyecto de investigación: al Idioma Inglés presentado por el señor Sosa Valverde Marco Antonio, Egresado de la Carrera de SISTEMAS INFORMÁTICOS de la FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS, cuyo título versa "APLICACIÓN DE SISTEMAS EMBEBIDOS EN LA CONSTRUCCIÓN DE ROBOTS MÓVILES": EL CASO DEL CLUB DE ROBÓTICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI, lo realizó bajo mi supervisión y cumple con una correcta estructura gramatical del idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo al peticionario a hacer uso del presente certificado de la manera ética que estimare conveniente.

Latacunga, 18 de Diciembre del 2019

Atentamente,

Cárdenas Guanolaína Olga María de los Ángeles  
DOCENTE CENTRO DE IDIOMAS  
C.C. 1707530141



## ÍNDICE

Portada	
Declaración de autoría.....	ii
Aval del director de tesis.....	iii
Aval de los miembros del tribunal.....	iv
Aval de implementación.....	v
Dedicatoria.....	vi
Agradecimiento.....	vii
Resumen.....	ix
Abstract.....	x
Aval de traducción.....	xi
Índice.....	xii

### ÍNDICE GENERAL

1. INFORMACIÓN BÁSICA.....	1
2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA.....	2
2.1 TÍTULO DE LA PROPUESTA.....	2
2.2. TIPO DE PROPUESTA/ALCANCE.....	2
2.3. ÁREA DE CONOCIMIENTO.....	3
2.4. SINOPSIS DE LA PROPUESTA TECNOLÓGICA.....	3
2.5. OBJETO DE ESTUDIO Y CAMPO DE ACCIÓN.....	3
2.5.1. Objeto de estudio.....	3
2.5.2 Campo de acción.....	3
2.6 Situación problemática y problema.....	3
2.6.1. Situación problemática.....	3
2.6.2. Problema.....	4
2.7. Hipótesis o formulación de preguntas directrices.....	4
2.8. Objetivos.....	4
2.8.1. Objetivo general.....	4
2.8.2. Objetivos específicos.....	4
2.8.2.1. Descripción de las actividades y tareas propuestas con los objetivos establecidos...6	
3. MARCO TEÓRICO.....	8
3.1. Los sistemas embebidos.....	8
3.1.1. La Memoria rom en un sistema embebido.....	10

3.1.2. La Memoria ram en un sistema embebido.....	10
3.1.3. El cpu en un sistema embebido.....	10
3.1.4. Unidades de entrada y salida.....	10
3.1.4.1. Unidades de entrada.....	11
3.1.4.2. Unidades de salida.....	11
3.1.4. La robótica, definición e historia.....	11
3.1.4.1. Historia de la robótica.....	12
3.1.4.2. Tipos y características de los robots.....	17
3.1.4.1. Androides bípedos.....	17
3.1.4.2. Robots móviles.....	17
3.1.4.3. Robot zoomórfico.....	18
3.1.4.4. Robot poliarticulado.....	19
3.1.5. Mirada al entorno de un club de robótica.....	19
3.2. Aplicaciones precedentes de sistemas embebidos.....	20
3.2.1. Robot Daro.....	20
3.2.2. Robot serpiente.....	21
3.3. Sistemas embebidos en robots móviles.....	23
3.5. Software.....	23
3.5.1. El pensamiento computacional.....	23
3.5.2. Software libre para sistemas embebidos.....	24
3.5.2.1. El bootloader.....	24
3.5.2.2. El entorno de desarrollo integrado o IDE.....	25
3.5.3. Flujograma de programación en un robot móvil.....	26
3.5.3.1. Lenguaje de programación en sistemas embebidos.....	26
3.5.3.1.2. Lenguaje de programación C++.....	27
3.6. Hardware.....	28
3.6.1. Definición de robótica.....	28
3.6.2. Los sistemas embebidos en la robótica.....	28
3.6.2.1. Electrónica de una tarjeta con microcontrolador.....	29
3.6.3. El PCB.....	30
3.6.4. Descripción del microcontrolador Atmega32u4.....	30
3.6.5. Conexión inalámbrica de aplicaciones con sistemas embebidos.....	31
3.6.6. Conector USB.....	32

3.7. Estructura mecánica para un prototipo robótico móvil.....	33
3.7.1 Programa de diseño SolidWorks.....	33
3.7.2. La energía en un prototipo de robot móvil.....	34
4. METODOLOGÍA.....	35
4.1. Procedimientos.....	35
4.1.1. Operacionalización de variables.....	35
4.1.1.2. Variable independiente.....	36
4.1.1.3. Variable dependiente.....	37
4.2. Implementación de la metodología.....	38
4.2.1. Introducción.....	38
4.2.2. Enfoque.....	38
4.3. Tipo de investigación.....	39
4.3.1. Investigación bibliográfica.....	39
4.3.2. Investigación de Campo.....	39
4.3.3. Investigación descriptiva.....	39
4.4. Métodos Técnicas Instrumentos.....	40
4.4.1. Población y muestra.....	41
4.4.1.1. Unidad de estudio.....	41
4.4.1.2. Población.....	41
4.5. Método de desarrollo.....	41
5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	44
5.1. Resultados de la encuesta realizada.....	44
5.2. Resultados de las entrevistas a profesionales del ámbito académico.....	49
5.3. Validación de la propuesta.....	50
5.3.1. Etapas de desarrollo planteado.....	51
5.4. Requerimientos y especificaciones.....	51
5.4.1. Diseño por bloques tarjeta electrónica programable.....	52
5.5. Selección de componentes.....	53
5.5.1. Diseño de la Tarjeta.....	57
5.5.2. Diseño de circuitos impresos con KiCad.....	57
5.5.3. Verificación esquemática de reglas eléctricas en la PCB.....	59
5.5.4. Generación de componentes Netlist.....	59
5.5.5. Traspaso del diagrama esquemático a una vista física lógica de una PCB.....	60

5.5.5.1. Transferencia, revelado y atacado con ácido de circuitos impresos.....	63
5.5.5.1.2. Termo transferencia de un circuito impreso.....	63
5.5.5.1.3. Revelado y atacado con ácido en circuitos impresos.....	64
5.5.5.1.4. Aplicación de la máscara antisoldante.....	66
5.5.5.1.5. Soldado de componentes en la placa impresa.....	67
5.6. Diseño e impresión del prototipo de robot móvil.....	67
5.7. Integración tarjeta electrónica con IDE instalado en un computador.....	69
5.7.1. Programador grabador de bootloader por ISP del Atmega32u4.....	69
5.7.2. Programación pantalla táctil Nextion NX3224T028.....	72
5.7.3. Desarrollo de una aplicación de control en un smartphone.....	76
5.7.3.1. ListPicker de listado de dispositivos disponibles.....	77
5.7.3.2. Conexión del ListPicker con el dispositivo escogido.....	77
5.7.3.3. Programación de funciones en los botones de la interfaz realizada.....	78
5.7.3.4. Exportación y grabado del apk en el smartphone.....	79
5.7.4. Software, programación e IDE.....	79
5.7.4.1. Creación de un proyecto para sistemas embebidos.....	80
5.7.4.2. Estructura del programa de aplicación para el Atmega32u4.....	80
5.7.4.3. Programa de verificación blink.....	81
5.7.4.2. Grabado del programa de verificación blink.....	81
5.7.4.2. Aplicación de direccionamiento controlado: Atmega32u4.....	84
6. PRESUPUESTO Y ANÁLISIS DE IMPACTOS.....	85
6.1. Presupuesto.....	85
6.1.1. Costos Directos.....	85
6.1.2. Costos indirectos.....	86
6.1.3. Presupuesto total.....	87
6.2. Análisis de impactos.....	87
6.2.1. Análisis práctico.....	87
6.2.1. Análisis técnico.....	88
6.2.1.1. Comparativa técnica de esta propuesta con propuestas comerciales.....	88
6.2.3. Análisis social.....	89
6.2.4. Análisis económico.....	89
6.2.4.1. Comparativa económica de esta propuesta con propuestas comerciales.....	89
6.2.5. Análisis epistemológico.....	90

7. CONCLUSIONES Y RECOMENDACIONES.....	91
7.1 CONCLUSIONES.....	91
7.2 RECOMENDACIONES.....	92
8. REFERENCIAS.....	92

## ANEXOS

Anexo I: Arduino ISCP

Anexo II: Diagrama esquemático general 1

Anexo III: Diagrama esquemático general 2

Anexo IV: Diagrama esquemático Lm293d

Anexo V: Vistas de impresión cuerpo del robot

Anexo VI: Vistas de impresión cuerpo del robot 2

Anexo VII: Vistas de impresión del robot armado

Anexo VIII: Código de aplicación: “Mis Letras”

Anexo IX: Encuesta

Anexo X: Manual de usuario

Anexo XI: Evidencia de entrevistas realizadas

Anexo XII: Aval de revisión encuestas y entrevistas realizadas

## **CONTENIDO TABLAS**

Tabla 1: Tareas en relación a los objetivos planteados.....	6
Tabla 2: Características atmega32u4.....	31
Tabla 3: Operacionalización de la variable independiente.....	36
Tabla 4: Operacionalización de la variable dependiente.....	37
Tabla 5: Métodos Técnicas Instrumentos.....	40
Tabla 6: Población.....	41
Tabla 7: Componentes electrónicos.....	54
Tabla 8: Costos Directos materia prima.....	85
Tabla 9: Costos indirectos maquinaria y equipo.....	86
Tabla 10: Costos indirectos varios.....	86
Tabla 11: Presupuesto Total.....	87
Tabla 12: Comparativa sistemas BRICK con Lego.....	88
Tabla 13: Comparación costos tarjeta BRICK con tarjeta Lego.....	90

## CONTENIDO DE ILUSTRACIONES

Gráfico 1: Bloques que conforman el interior de un microcontrolador.....	8
Gráfico 2: El microcontrolador.....	9
Gráfico 3: Robot con orugas para exploración.....	9
Gráfico 4: Las leyes de la robótica según Asimov.....	13
Gráfico 5: La robótica 1.....	14
Gráfico 6: La robótica 2.....	16
Gráfico 7: Sophia.....	17
Gráfico 8: Robot articulado.....	18
Gráfico 9: Flujograma de un prototipo de robot móvil.....	22
Gráfico 10: El software libre de un sistema embebido.....	24
Gráfico 11: Flujograma de programación.....	25
Gráfico 12: Tarjeta de control.....	29
Gráfico 13: El PCB.....	29
Gráfico 14: El microcontrolador.....	30
Gráfico 15: El USB.....	32
Gráfico 16: Mecánica de un prototipo.....	33
Gráfico 17: La batería.....	34
Gráfico 18: Diagrama de actividades generales.....	42
Gráfico 19: Resultado pregunta 1.....	45
Gráfico 20: Resultado pregunta 2.....	45
Gráfico 21: Resultado pregunta 3.....	46
Gráfico 22: Resultado pregunta 4.....	47
Gráfico 23: Resultado pregunta 5.....	47
Gráfico 24: Resultado pregunta 6.....	48
Gráfico 25: Resultado pregunta 7.....	48
Gráfico 26: Resultado pregunta 8.....	49
Gráfico 27: Etapas de desarrollo.....	51
Gráfico 28: Diagrama de bloques tarjeta electrónica programable.....	53
Gráfico 29: Interfaz principal KiCad.....	57
Gráfico 30: Interfaz de KiCad, Netlist.....	59
Gráfico 31: Frontal de impresión circuito PCB.....	60
Gráfico 32: Anverso de impresión circuito PCB.....	61

Gráfico 33: Disposición de elementos del circuito impreso.....	61
Gráfico 34: Traspaso térmico de circuito impreso a baquelita.....	62
Gráfico 35: Utilización de ácido de revelado cloruro férrico.....	64
Gráfico 36: Baquelita revelada.....	65
Gráfico 37: Aplicación de mascara anti soldante.....	66
Gráfico 38: Tarjeta programable soldada.....	67
Gráfico 39: Modelo de robot impreso en 3d.....	68
Gráfico 40: Proceso de integración Atmega32u4 con un computador personal.....	69
Gráfico 41: Grabación del bootloader.....	70
Gráfico 42: Programación Atmega32u4.....	70
Gráfico 43: ISCP programador Atmega32u4.....	71
Gráfico 44: Ventanas de carga del bootloader al integrado Atmega32u4 (IDE).....	72
Gráfico 45: Ventanas de programación pantalla táctil.....	72
Gráfico 46: Gráfico menú de programación pantalla táctil.....	74
Gráfico 47: Interfaz control bluetooth.....	75
Gráfico 48: Recurso de programación.....	76
Gráfico 49: ListPicker.....	77
Gráfico 50: Conexión ListPicker.....	78
Gráfico 51: Programación botón interfaz.....	78
Gráfico 52: Exportación de aplicación desde app inventor 2.....	79
Gráfico 53: Pagina de descarga IDE.....	80
Gráfico 54: Pantalla del IDE en blanco para programación.....	81
Gráfico 55: Programa Blink.....	82
Gráfico 56: Configuración del IDE.....	83
Gráfico 57: Programa de aplicación.....	84

## **1. INFORMACIÓN BÁSICA**

### **PROPUESTO POR:**

Sosa Valverde Marco Antonio

**TEMA APROBADO:** “Aplicación de sistemas embebidos en la construcción de robots móviles”: El caso del club de robótica de la Universidad Técnica De Cotopaxi.

### **CARRERA:**

Ingeniería en Informática y Sistemas Computacionales.

### **COORDINADORES DE LA PROPUESTA TECNOLÓGICA:**

Sosa Valverde Marco Antonio

### **EQUIPO DE TRABAJO:**

PhD. Gustavo Rodríguez

**LUGAR DE EJECUCIÓN:** La ejecución tecnológica se realizará para el área del Club de Robótica de C.I.Y.A de la Universidad Técnica de Cotopaxi ubicada en la Provincia de Cotopaxi, Cantón Latacunga, Parroquia Eloy Alfaro.

### **TIEMPO DE DURACIÓN DE LA PROPUESTA:**

La propuesta durara dos periodos

### **FECHA DE ENTREGA:**

2019 - 2020

### **LÍNEA(S) Y SUB-LÍNEAS DE INVESTIGACIÓN A LAS QUE SE ASOCIA LA PROPUESTA:**

#### **LÍNEA DE INVESTIGACIÓN**

**Línea 1:** Tecnologías de la Información y comunicación.

#### **SUB-LÍNEAS DE INVESTIGACIÓN**

**Sub-Línea 1:** Robótica e Inteligencia Artificial.

### **TIPO DE PROPUESTA TECNOLÓGICA:**

Desarrollo de una tecnología de gestión de la información.

## **2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA**

### **2.1 Título de la propuesta**

**“Aplicación de sistemas embebidos en la construcción de robots móviles:”**: El caso del club de robótica de la Universidad Técnica de Cotopaxi.

### **2.2. Tipo de propuesta:**

El desarrollo del presente proyecto busca incentivar el uso y estudio de la tecnología informática utilizando sistemas embebidos en prototipos de robótica como herramienta de propuestas propias, sin tener que consumir elementos comerciales, y cimentar planes propios que abarquen el diseño y construcción del mismo, para aportar en el campo de la tecnología.

El uso de elementos digitales programables disponibles en los sistemas embebidos como medio de experimentación, transmite a futuros investigadores el uso de la tecnología informática en la utilización de hardware, para permitir potencializar a esta tecnología con alcance a la industria con prototipos robóticos programados de un mayor alcance en cuestión de aplicaciones previamente definidas en un programa realizado en un computador, para resolutorias tecnologías a problemas que se plantean cotidianamente, lo cual generará expectativas importantes a desarrollarse en nuevos temas investigativos o estudios futuros..

### **2.3.Área de conocimiento:**

**ÁREA:** Ciencias

**SUB ÁREA:** Informática

### **2.4.Sinopsis de la propuesta tecnológica:**

El presente proyecto muestra el diseño, construcción y validación de una aplicación micro controlada en robótica móvil denominada “BRICK”, el mismo que tiene como objetivo la programación de un robot a través de un IDE (entorno de desarrollo integrado) con la utilización de software libre y operado por un computador.

En la comprobación de esta investigación se expone la importancia de la utilización de electrónica en sistemas embebidos, lo cual evidenció que este tipo de sistemas son flexibles y se pueden adaptar a un propositivo general o específico en base a la utilización de una metodología en proyectos mecatrónicos que son operadas por señales generadas mediante una

pantalla táctil o diversos medios como, infrarrojos, bluetooth, comandado desde un celular inteligente, emitiendo ordenes de acuerdo con las necesidades del usuario. Estas actividades accionan el funcionamiento de los actuadores que fueron integrados en la programación en base a algoritmos de comunicación a través de señales inalámbricas.

**Palabras claves:**

Micro controlador, robótica, programación, software, comunicación.

**2.5. Objeto de estudio y campo de acción**

**2.5.1. Objeto de estudio**

Los Robots móviles del Club de Robótica de la Universidad Técnica de Cotopaxi.

**2.5.2. Campo de acción**

Aplicación de los sistemas embebidos en la construcción de Robots Móviles.

**2.6. Situación problemática y problema**

**2.6.1 Situación problemática**

Dentro de la revolución de la ciencia y la técnica alcanzadas durante los últimos años la sociedad se encuentra dentro de un proceso globalizador que implica cada vez más una adaptación a las nuevas tecnologías que están transformando la vida laboral de los seres humanos, aunque existe una resistencia al cambio, el análisis detalla la sesgada visión del nuevo entorno, que encuentra sobrevalora esta herramienta que tendrá el trabajador en el futuro, ya que las labores que cumplen los robots a través de medios mecánicos están relacionadas principalmente en evitar los excesos laborales que llevan alto riesgo o que son trabajos repetitivos que causan afectación a la salud ocupacional.

Como parte del proceso de transformación digital, las instituciones han establecido adoptar tecnologías inteligentes, dentro del sistema de preparación de los estudiantes con una tendencia más reciente a la automatización de procesos robóticos.

Aunque el uso de la inteligencia artificial ha crecido a pasos agigantados en el sector educativo aún se tiene mucho camino por recorrer, para lo que el diseño y construcción de soluciones en donde intervienen robots en la automatización de procesos se convierte en una necesidad imperiosa.

El Ecuador no puede estar aparte de este proceso, ya que las nuevas tecnologías y la exigencia de los centros educativos a nivel superior necesita tener profesionales preparados en esta área para enfrentar este desafío de acuerdo con las necesidades del entorno local y ser competitivos con el mercado externo. En el país hay una limitada preparación relacionada con la automatización de la industria haciéndose necesario que se realice trabajos de investigación relacionados con el estudio enfocados en los robots móviles como recursos innovadores con uso de la tecnología que mejoren los sistemas de producción.

Esta realidad se refleja en la provincia de Cotopaxi en donde existen proyectos iniciales en algunas Universidades del centro del país, que tratan de incorporar sistemas robóticos para realizar actividades específicas en el ámbito de prototipos los cuales se necesitan realizar una serie de pruebas para su implementación que resuelvan problemas cotidianos.

### **2.6.2 Problema**

¿Qué incidencia tienen los Sistemas Embebidos en la construcción de Robots Móviles en el Club de Robótica de la Universidad Técnica de Cotopaxi?

## **2.6. Hipótesis**

La aplicación eficiente de sistemas embebidos permitirá un adecuado diseño e implementación de prototipos de Robots Móviles como herramienta práctica en el Club de Robótica de la Universidad Técnica de Cotopaxi.

## **2.7. Objetivos**

### **2.7.1. Objetivo general:**

- Implementar un sistema embebido en una tarjeta de control programable mediante la metodología de proyectos mecatrónicos industriales para un prototipo de robot móvil en el club de robótica de la Universidad Técnica de Cotopaxi.

### **2.7.2. Objetivos específicos:**

1. Realizar un análisis de los fundamentos teóricos relacionados a los sistemas embebidos en los robots móviles.
2. Aplicar técnicas en la recolección de datos sobre la utilización y aplicación en sistemas embebidos para el desarrollo de un prototipo de un robot móvil.
3. Utilizar como base un sistema embebido para diseñar una tarjeta de control programable para sistemas de robots móvil.

4. Realizar una valoración económica y tecnológica, que permita la identificación de los principales aportes del proyecto.

### 2.7.2.1 Descripción de las actividades y tareas propuestas con los objetivos establecidos

**Tabla 1:** Tareas en relación a los objetivos planteados

OBJETIVOS	ACTIVIDADES	RESULTADO DE LAS ACTIVIDADES	DESCRIPCIÓN DE LAS ACTIVIDADES
Realizar un análisis de los fundamentos teóricos relacionados a los sistemas embebidos en los robots móviles.	<ul style="list-style-type: none"> <li>• Investigar información refrendada del tema.</li> <li>• Elegir información que tenga objetividad del tema proyectado.</li> </ul>	<ul style="list-style-type: none"> <li>• Bibliografía.</li> <li>• Detalle de aplicación precedente.</li> </ul>	Para el desarrollo se buscó herramienta certificada como son Google Académico, libros de la biblioteca de la Universidad Técnica de Cotopaxi, entre otras fuentes científicas.
Aplicar técnicas en la recolección de datos sobre la utilización y aplicación en sistemas embebidos para el desarrollo de un prototipo de un robot móvil.	<ul style="list-style-type: none"> <li>• Recolectar datos de una encuesta que den pauta en el desarrollo de sistemas embebidos para robots móviles didácticos.</li> <li>• Seleccionar los datos recolectados de la entrevista, con técnicas de investigación adecuadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Análisis de Encuestas.</li> <li>• Análisis de Entrevistas.</li> </ul>	La utilización de la metodología ayudará a desarrollar la aplicación de un sistema embebido en robots móviles con el análisis de los requerimientos en el diseño del prototipo para la realización del mismo.
Utilizar como base un sistema embebido para diseñar una tarjeta de control programable para sistemas de robots móvil.	<ul style="list-style-type: none"> <li>• Desarrollar la tarjeta electrónica con un sistema embebido cargado con un bootloader que sea reconocido por un IDE instalado en una computadora, para su programación y uso en un prototipo robótico.</li> </ul>	<ul style="list-style-type: none"> <li>• Tarjeta de control.</li> <li>• Diseño de circuito impreso.</li> <li>• Esquema electrónico del circuito.</li> </ul>	En el desarrollo del prototipo de una aplicación embebida en robot móviles, se utilizó la carga de un bootloader de software libre que permite mejor el manejo de la comunicación entre la aplicación embebida y el IDE instalado en la computadora, para permitir experimentación con el prototipo robótico.

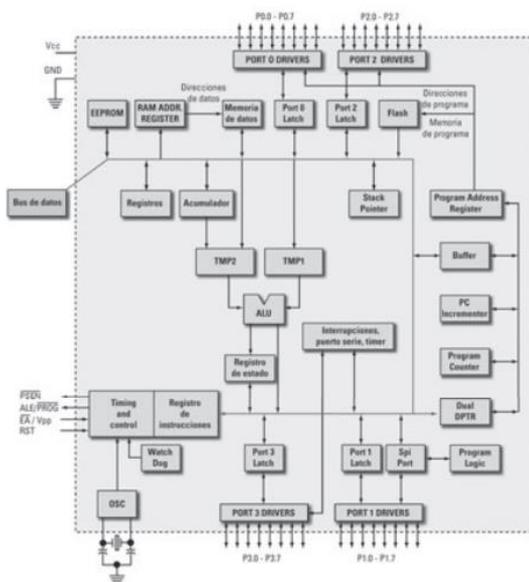
<b>OBJETIVOS</b>	<b>ACTIVIDADES</b>	<b>RESULTADO DE LAS ACTIVIDADES</b>	<b>DESCRIPCIÓN DE LAS ACTIVIDADES</b>
<p>Realizar una valoración económica y tecnológica, que permita la identificación de los principales aportes del proyecto.</p>	<ul style="list-style-type: none"> <li>• Detallar los resultados económicos del desarrollo del prototipo robótico implementado.</li> <li>• Evaluar los beneficios que aporte el desarrollo de esta plataforma, para la didáctica en prototipos robóticos.</li> </ul>	<ul style="list-style-type: none"> <li>• Valoración Tecnológica</li> <li>• Valoración Económica.</li> </ul>	<p>En la realización de la valoración económica se sacó los resultados de los materiales que se ocupó para la realización del prototipo robótico móvil, así como también se da criterios sobre los posibles usos que esta plataforma podría aportar en futuros desarrollos.</p>

### 3. MARCO TEÓRICO

Este capítulo configura la visión general de la presente propuesta, con tres puntos iniciales desglosados del título del proyecto: microcontroladores, robótica, y clubs universitarios, presentando la justificación del mismo, que ha inspirado el desarrollo de esta propuesta que será de amplio alcance, además de generar los retos asociados a la metodología empleada.

#### 3.1. Los sistemas embebidos

Un microcontrolador es un circuito integrado, en cuyo interior posee toda la arquitectura de un computador, esto es CPU, memorias RAM, EEPROM, y circuitos de entrada y salida, [24].

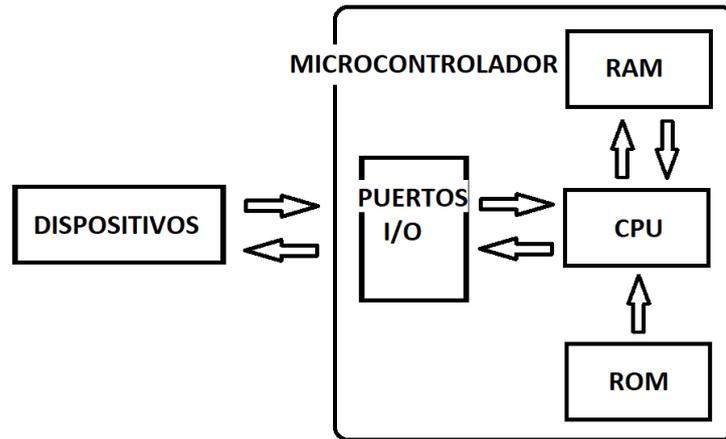


**Gráfico 1:** Bloques que conforman el interior de un microcontrolador

**Fuente:** D. Benchimol, Microcontroladores

La microcomputadora, actualmente conocida como microcontrolador es la que dentro de ella se encuentra una CPU, una memoria de programa, una memoria de datos, el circuito de reset y el circuito oscilador, además de los puertos de entrada/salida, también conocidos como PORTS I/O, [32].

Un sistema embebido o microcontrolador es un circuito integrado que encapsula en su interior una microcomputadora completa interconectada, (CPU, RAM y ROM, puertos de entrada y salida y periféricos), el cual para realizar cualquier función específica requerirá de un programa, derivando su propósito fundamental a leer y ejecutar algoritmos que un usuario escriba.



**Gráfico 2:** El microcontrolador

Actualmente un microcontrolador conjuga la utilización del pensamiento computacional como actividad básica e indispensable en la utilización y diseño de circuitos y sistemas que los incluyan, simplificando, modularizando además de flexibilizar un sistema, ya que un mismo circuito se puede utilizar para diferentes funciones con solo cambiar el programa almacenado en memoria del sistema embebido, limitando funciones solo a la imaginación del usuario, que puede utilizarlo en campos como la robótica, automatismo, industria, entretenimiento, telecomunicaciones, instrumentación, hogar, y hasta en lo automotriz entre otras.



**Gráfico 3:** Robot con orugas para exploración

**Fuente:** R. Tagliaterri, S. Balzarini, Robótica

### **3.1.1. La Memoria rom en un sistema embebido**

ROM (Read Only Memory): memorias de solo lectura programadas por máscara, los datos almacenados se escriben en fábrica y no es posible modificarlos ni borrarlos, [32].

La memoria rom o read only memory, es una memoria de solo lectura, que ocupa el cpu del microprocesador como contenedor del programa de instrucciones máquina que controlan la aplicación escrita por el usuario, este tipo de memoria no permite hacer cambios a la misma, y es grabada de fábrica por la empresa productora.

### **3.1.2. La Memoria ram en un sistema embebido**

RAM significa Memoria de Acceso Aleatorio (Random Access Memory), es decir, podemos acceder a cualquier posición de la memoria para poder leerla o escribirla, [32].

La memoria ram realiza el almacenamiento de información que será utilizada por el cpu para realizar todo tipo de operaciones que se haya programado a realizar por el sistema embebido.

### **3.1.3. El cpu en un sistema embebido**

La CPU es el elemento principal de un microcontrolador, se conecta con los periféricos para conformar la estructura interna de estos, [32].

La unidad de procesamiento central o CPU, actúa como el “cerebro” de un sistema computacional, administrando todas las tareas que este realice y llevando a cabo las operaciones con los datos, [32].

La cpu o unidad central de proceso, es el núcleo del microcontrolador, se encarga de ejecutar las instrucciones almacenadas en la memoria, procesando las tareas del microcontrolador al ejecutar las instrucciones del programa grabado por el usuario, y de esta manera interactuar con el exterior a través de las unidades de entrada y salida.

### **3.1.4. Unidades de entrada y salida**

Para que un procesador pueda comunicarse con el mundo externo necesita de unidades de entrada y de salida que codifiquen los mensajes para interpretarlos, [32].

Un microcontrolador como un circuito integrado sólido que incluye en su interior las tres unidades funcionales de una computadora, cpu, memorias rom y ram, también posee unidades de E/S (entrada/salida).

El funcionamiento de estas unidades de E/S, es organizando de forma interna en el microcontrolador, agrupando de manera multifuncional los pines del integrado con funciones programadas por el usuario para propósito general, en las que un cero programado en el pin significa entrada y un uno programado en el pin significa salida (0 = entrada, 1 = salida).

#### **3.1.4.1. Unidades de entrada**

Se ocupan de codificar los mensajes o señales del exterior para que el procesador pueda interpretarlos, [32].

Conocidos como puerto de entrada o in, es la que permiten leer datos del exterior, los cuales serán interpretados por el cpu del microcontrolador según el programa que el usuario haya grabado en la memoria ram del sistema embebido, para que esté funcionando algorítmicamente con un programa realice una tarea.

#### **3.1.4.2. Unidades de salida**

Permiten observar los resultados arrojados por el procesador de una manera más cómoda que si se presentaran como unos y ceros, [32].

El puerto de salido o out, es el que realiza acciones pre programadas según el algoritmo implementado grabado en la memoria interior del microcontrolador, el cual seguirá entregando su señal según el usuario lo haya dispuesto.

### **3.1.4. La robótica, definición e historia**

La robótica es una rama de la ingeniería que utiliza elementos de la ingeniería electrónica, ciencias de la computación y la ingeniería mecánica que busca la construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano, o que requieren del uso de inteligencia, [39].

La palabra robótica cubre muchos conceptos diferentes, girando en torno a una misma idea, ser la ciencia encaminada a diseñar y construir aparatos y sistemas capaces de realizar tareas propias de un ser humano.

Históricamente la robótica contempla un desarrollo con bases en orígenes y hechos registrados a través de la historia, acotando que anteriormente a esta era, los robots eran conocidos con el nombre de autómatas, y la robótica no era conocida como ciencia, siendo acuñada la palabra robot después del origen de los autómatas.

Desde el principio de los tiempos, el hombre ha deseado crear vida, vida artificial, la robótica ha llenado esa tendencia como la ciencia y tecnología de los robots que combina diversas disciplinas para su realización como la mecánica, electrónica, e informática, en su diseño y manufactura.

El análisis y relaciones de los robots con la humanidad, da como concluyente a esta creación como el elemento principal añadido en la vida actual, que no solo en películas, sino alrededor de todo el mundo conduce acciones, acciones con robots que la humanidad tiene la obsesión de crear, y que ha llevado al hombre a fabricarlos más resistentes, ágiles, y sofisticados; robots que hacen cosas que los seres humanos no puede o simplemente no desea realizar, axioma que la vida artificial realiza a diario, y que ha salido de obras literarias de ficción como RUR (Robos Universales Rossum), escrita por Karel Capek en 1920, [5], y que popularizaron el término robot, además de contar con las pantallas de los cines, que predijeron a esta ficción como una realidad, la cual con muchos intentos de lograr ocupar espacios de trabajo repetitivos o peligrosos, hoy en día hace pintura de autos en las plantas de automóviles, exploración de riesgo caminando en el espacio o dentro de volcanes activos, o manejando trenes en países industrializados, todas estas acciones realizadas con la coordinación de ordenes programadas y ejecutadas con impulsos eléctricos, que hacen que el robot realice los movimientos requeridos por la tarea.

Se ha de hacer hincapié que, para todas las acciones anteriores, sin la informática que provee los algoritmos, traducidos a programas, no serían posible, ya que la coordinación mecánica requerida en los movimientos del robot, la informática lo provee, dando un grado de inteligencia a la máquina, que adaptando autonomía y capacidad interpretativa y correctiva cumple tareas con precisión.

A lo anterior se ha de añadir, que los robots dado la proliferación de su uso, se están volviendo accesibles en costos, además de reducir su tamaño, una tendencia relacionada como la miniaturización de los componentes electrónicos que se utilizan para controlarlos.

#### **3.1.4.1. Historia de la robótica**

La robótica ha creado una mitología de la modernidad. Desde el flautista mecánico del relojero Jacques de Vaucanson, el monstruo de Frankenstein de Mary Shelley y el mito clásico de Pigmalión, pasando por el Golem de Praga, [40].

Por siglos el ser humano ha construido máquinas que imiten las partes del cuerpo humano, Jacques de Vaucansos en el siglo XVIII tomando ese conceto construyó varios músicos de tamaño humano que esencialmente eran robots mecánicos diseñados para la diversión.

Uno de los autómatas de Henri Maillardet fue capaz de crear cuatro dibujos y escribir tres poemas, este es el caso de un ser artificial que podría intervenir o producir independientemente algo con características centrales que contribuye al aura cultural de un robot, [41].

Los avances en las máquinas de hilar: de manuales e individuales, a mecánicas o hidráulicas de producción masiva, con las sucesivas y/o simultáneas mejoras propuestas por Arkwright, Wyatt, Crompton y Cartwright, por mencionar a los principales y más avanzados, [42].

Como cita el párrafo anterior, esta y otras invenciones mecánicas creadas por mentes geniales, anticipaban que estaban dirigidas a la producción, y así nacieron ideas que ejemplarizaban la producción textil con telares automáticos, la hiladora mecánica de Compto 1779, el telar mecánico de Cartwright en 1785, el telar de Jacquard en 1801, entre otros, impulso a la robótica como la necesidad imperiosa de buscar alternativas para facilitar y optimizar herramientas autómatas indispensable en cualquier tarea productiva que desarrolle personal, social y tecnológicamente el medio.

Llamó al robot muchacho, como hacían todos los terrestres, pero el ser mecánico no se dio por aludido, aunque bien mirado no tenía motivo para sentirse molesto, pues sus reacciones se regían por las Leyes de la Robótica, [43].

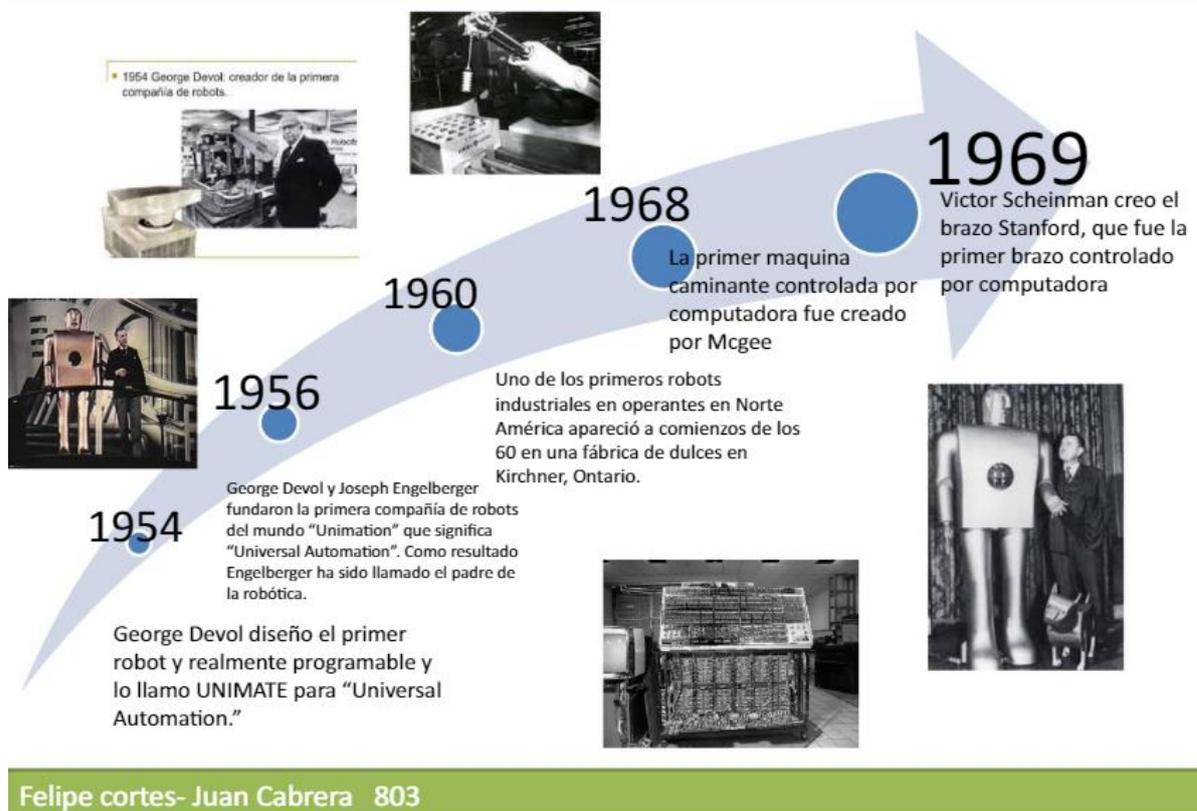


Gráfico 4: Las leyes de la robótica según Asimov

Isaac Asimov es el padre fundador de la robótica en la ciencia ficción moderna. En su relato de 1957, enunció las tres leyes que rigen la conducta de todo robot debería tener estas tres leyes que son:

1. Un robot no puede dañar a un ser humano o, por omisión de la acción, permitir que reciba daño.
2. Un robot debe obedecer las órdenes dadas por los humanos, excepto cuando entren en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia siempre que dicha protección no entre en conflicto con la primera y la segunda ley.

## Línea De Tiempo De La Robótica



**Gráfico 5:** La robótica 1

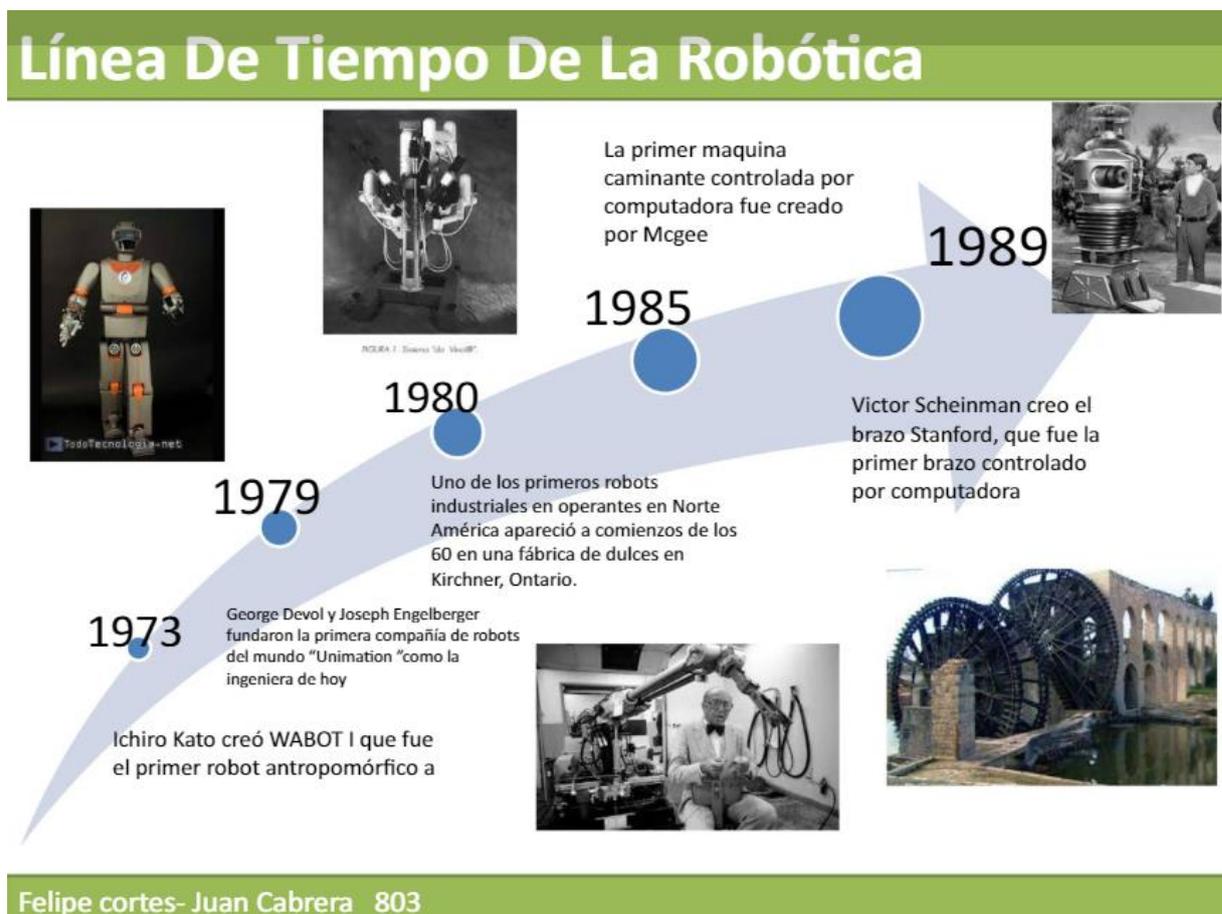
**Fuente:** F. Cortes, J. Cabrera, Línea del tiempo de la robótica

Aunque se esté un poco alejado de la visión de autores tan renombrados con Asimov, se ha de destacar que el desarrollo tecnológico y la reducción de portes en componentes electrónicos, a hecho incluir a computadoras electrónicas miniaturizadas o microcontroladores, llamados técnicamente sistemas embebidos, que combinan control de actuadores, transmisión de

potencia a través de engranes, y la tecnología en sensores, flexibilizando mecanismos autómatas con factores evolutivos que se desarrollan desde la década de los 50's con emulación del procesamiento de información.

En 1954 se construyó en Estados Unidos el primer robot programado construido por George Devol al que llamó Unimate, [44].

El robot unimate es el primer robot digitalmente operativo y programable que ha hecho posible las fundaciones de la industria de robótica moderna.



**Gráfico 6:** La robótica 2

**Fuente:** F. Cortes, J. Cabrera, Línea del tiempo de la robótica

El primer manipulador electro mecánico fue diseñado en 1969 por Víctor Scheinman en la Universidad de Stanford, constituido por eslabones y articulaciones, [45].

Uno de los primeros robots fue el llamado Shakey, desarrollado en el Instituto de Investigaciones de Stanford en 1960 y capaz de tomar bloques de una pila utilizando una cámara de video como sensor visual, y de procesar información en una pequeña computadora, [46].

Este se podría considerar el primer robot avanzado, porque Shakey combinó sensores, con visión por cámara, para una navegación dirigida que buscaba precisión.

Genghis es un pequeño robot autónomo para caminar construido a mediados de 1989 por robótica móvil group de la MIT tiene seis patas que le permite a Genghis caminar con una marcha ondulada a través del terreno irregular y seguir a las personas, [47].

El robot tipo hexápodo denominado Genghis fue desarrollado por la MIT en 1989, utilizando microprocesadores para su control, además de sensores y servo motores.

Honda se propuso hacer el robot más pequeño y más ligero, y en su momento desarrollado P3 (altura de 160 cm) como un robot más similar en sus proporciones a los humanos en 1997, [48].

Honda en su afán de nuevas tecnologías introdujo al robot humanoide P3 que fue presentado en 1998 como parte del humanoide proyecto de la compañía.

El AIBO, es una notable pieza de hardware robótico disponible comercialmente. Un AIBO tiene 15 grados de libertad, en sus patas y cabeza, una cámara CCD a color que puede procesar imágenes a 25-30 cuadros por segundo, un acelerómetro de tres ejes para pose corporal, botones en la parte posterior, cabeza y almohadillas para los pies, indicador LED para depuración visual y una conexión inalámbrica con una tarjeta ethernet para comunicación con una computadora, [49].

En 1999, Sony introdujo a AIBO, un perro robótico capaz de interactuar con humanos que podía caminar, pasear, comunicarse con humanos, y reconocer rostros, entornos, y voces en su entorno, fue un desarrollo para diversión comercial de una empresa de mucha trayectoria originaria de Japón.

Sophia, es un robot humanoide social desarrollado por Hong Kong, empresa Hanson Robotics, en colaboración con la empresa matriz de Google Alphabet y SingularityNET, que proporciona el sistema de reconocimiento de voz y el software de inteligencia artificial, fue activada en abril de 2015, con primera aparición pública en Austin, Texas, en marzo de 2016, [50].

Sophia es un robot tipo humanoide con un referido pronombre femenino que tiene la ciudadanía saudí, implicando que este robot de gran capacidad de inteligencia artificial, puede votar, casarse, dado que puede deliberar, con respuestas a preguntas de un interlocutor hacia ella, hasta el momento se lo considera uno de los robots IA más avanzados.



**Gráfico 7:** Sophia

**Fuente:** Valentín Olavarria [52]

#### **3.1.4.2. Tipos y características de los robots**

Actualmente en la creación de robots se innova para generar sistemas inteligentes, flexibles y fiables para que el ser humano pueda trabajar utilizando estos elementos en diferentes campos de aplicación, algunos con diseños propios del entorno, [33], los cuales los podemos clasificar en:

- Androides bípedos.
- Móviles.
- Zoomórficos.
- Poliarticulados.

##### **3.1.4.1. Androides bípedos**

Son robots que parecen y actúan como los humanos. Tratan de imitar el comportamiento del hombre. Actualmente, su utilidad es solo para experimentación. En este modelo, la principal limitante es la implementación del equilibrio a la hora del desplazamiento, [33].

El robot humanoide bípedo está basado en el movimiento y locomoción sobre dos extremidades inferiores, de forma similar a la de los humanos, con un símil a un cuerpo completo que incluye brazos; centrándose en su capacidad para realizar tareas con definición autónoma en un sentido amplio.



**Gráfico 8:** Robot articulado

**Fuente:** R. Tagliaterra, S. Balzarini, Robótica

#### **3.1.4.2. Robots móviles**

Poseen una gran capacidad de desplazamiento y un alto nivel de inteligencia. El desplazamiento se logra mediante plataformas o carros con sistemas de locomoción rodantes. Tienen gran aplicación en líneas de montajes industriales, donde hay que asegurar el transporte de piezas de un punto a otro. Por lo general, son guiados por telemando o por la información recibida del entorno mediante sensores, [33].

Este tipo de robot tiene la característica de moverse en su entorno y no están fijados en una ubicación física, pudiendo ser autónomos, con capacidad de navegar en un entorno no controlado sin la necesidad de dispositivos de guía predefinida para un espacio relativamente controlado.

#### **3.1.4.3. Robot zoomórfico**

Su sistema de locomoción imita al de los animales. La aplicación de estos robots se da en el estudio de terrenos accidentados, volcanes y exploración de planetas, ya que son hábiles para sortear obstáculos. Pueden subclasificarse en caminadores y no caminadores; estos últimos son muy poco evolucionados, [33].

Constituyen una clase de robot caracterizada principalmente por sus sistemas de locomoción que imitan a los diversos seres vivos del mundo animal, como mascotas, entre otras, sus posibles sistemas agrupan a los robots zoomórficos como capaces de imitar las características en las que se basaron en su construcción.

#### **3.1.4.4. Robot poliarticulado**

Se caracterizan por agrupar robots primordialmente sedentarios, que poseen un número reducido de articulaciones para mover sus elementos terminales. Su utilidad principal es en el área industrial, donde se encargan de desplazar elementos que requieren cuidados especiales, [33].

Son un grupo de robots de muy diversa forma y configuración, cuya característica común es la de ser básicamente estacionarios, estando estructurados para mover sus elementos terminales en un determinado espacio de trabajo según uno o más sistemas de coordenadas, y con un número limitado de grados de libertad.

#### **3.1.5. Mirada al entorno de un club de robótica**

Al otro lado del campus, en una habitación sin ventanas del segundo piso, cuatro estudiantes se amontonan alrededor de un extraño marco de 3 pies de altura construido con tubería de PVC. Lo han equipado con hélices, cámaras, luces, un láser, detectores de profundidad, bombas, un micrófono subacuático y una pinza articulada. En la parte superior se encuentra un maletín negro impermeable que contiene un nido de procesadores pirateados, minúsculos ventiladores y leds. Es un robot subacuático barato, pero sorprendentemente funcional, capaz de grabar pings de sonar y recuperar objetos a 50 pies debajo de la superficie. Los cuatro adolescentes que lo construyeron son inmigrantes mexicanos indocumentados que llegaron a este país a través de túneles o escondidos en los asientos traseros de los automóviles.[3].

La propuesta de un club de robótica, usa la lúdica para propiciar, mediante la programación y construcción de robots, el aprendizaje cognitivo que estimula el desarrollo de la creatividad, integrando los diversos saberes, para compaginar el trabajo en equipo, centrándose en la comunicación grupal, cuyo entorno propicia la interacción y por supuesto la socialización entre los alumnos.

El club de robótica de la Universidad Técnica de Cotopaxi, es una cimiento nueva en el campo académico de la Alma Mater ecuatoriana, que con solo pocos años de creación, ante las décadas de sus antagonistas, nació en constancia a la injusticia hecha a esta noble institución, en la recalificación Universitaria; y aunque inexperto, este club, ha traído logros detallados en medalleros.

Tenemos gente que cree en nosotros, así que ahora tenemos que creer en nosotros mismos. [3].

En un mundo cada vez más globalizado los encargados de aportar innovación son los jóvenes de intuición investigativa que por cuenta propia forman grupos humanos que utilizan la ciencia con miras a ser las nuevas generaciones de investigadores innovadores de talante científico en los países del mundo.

Dado lo anterior se hace explícita la necesidad de aumentar el porcentaje de jóvenes que formen clubes tecnológicos que voluntariamente aporten institucionalmente en las dependencias Universitarias, conformando grupos de investigación destacada a lo largo de su vida académica, y que conceptualicen con la ingeniería el importante papel que desempeña el universitario en la sociedad.

### **3.2. Aplicaciones precedentes de sistemas embebidos**

La robótica expresa la síntesis del desarrollo científico y tecnológico de la humanidad, se inicia con las máquinas simples (estructuras, palancas, ruedas, plano inclinado, engranajes y poleas); continúa con las máquinas complejas motorizadas y posteriormente automatizadas a través de la computadora utilizando software de programación para la automatización y control. [8].

Lo que detalla este entorno, prevé el desarrollo que los robots empiezan a tener al ser parte de la vida de las personas, esta propuesta ha examinado trabajos precedentes de desarrollos tecnológicos y científicos en aplicaciones similares a esta investigación, lo que ha dado énfasis a la mejor utilización de recursos para obtener experiencia y llegar a conclusiones de cómo realizar una mejor aplicación de los sistemas embebidos en robots móviles, dicha revisión de antecedentes fundamenta y explica teóricamente todo lo que se va a realizar.

#### **3.2.1. Robot Daro**

Los sistemas embebidos para robots móviles deberían ser una práctica académica usual, aprovechando la creatividad natural del entorno robótico que proporciona las diversas aplicaciones libres que existen, identificando elementos necesarios para lograr respuestas ideales a la creatividad aplicada en el robot, la mecánica, conjuntamente con la eléctrica y electrónica de un diseño, sistemáticamente harán que la programación de los elementos utilizados, sean convenientes didácticamente en una propuesta, así lo demuestra la propuesta del robot Daro.

El robot DARO como plataforma robótica de aprendizaje, surge como una iniciativa del grupo de trabajo del semillero de investigación Robótica-UPTC de la Universidad Pedagógica y Tecnológica de Colombia, sede Tunja, Colombia, para ofrecer una herramienta de fácil

construcción y aprendizaje de la robótica no solo para los estudiantes de primeros semestres de Ingeniería Electrónica y áreas afines, sino también para instituciones de educación básica secundaria, buscando motivar e incentivar al estudiantado en la exploración y explotación de sus capacidades y habilidades que le permitan un mejor desempeño profesional. [1].

Esta propuesta ha hecho uso de una tarjeta comercial de la marca Arduino, diseñando los complementos a esta tarjeta, que acopla el pequeño tamaño de la misma, la simplicidad adoptada ha hecho que Daro haga de su entorno, una propuesta interesante y asequible; movilidad e interconectividad están presentes en este acople a la tarjeta principal.

Aunque Daro está diseñada en base a la experimentación, no cuenta con un sistema de visor de datos, o inclusión de elementos externos: la carga de software, entre las que se denota un precargado de bootloader, no existen, y se basa en la adquisición comercial de la tarjeta, lo cual ante un fallo de firmware (software), dejara a este entorno sin poder realizar más prácticas, recomendando que didácticamente la cargar firmware libre, como es de Arduino, se documente, para objetivar la realización de una tarjeta diseñada en base a aplicaciones de sistemas embebidos.

### **3.2.2. Robot serpiente**

Otro proyecto presentado dentro del campo de los robots móviles que ocupan sistemas embebidos es el desarrollado en la propuesta Diseño e Implementación del Sistema Electrónico y Comunicación para el Control un Robot Modular Tipo Serpiente.

Las condiciones del diseño electrónico están basadas en los componentes necesarios para generar el movimiento del robot basado en su locomoción, tamaño, diámetro corriente, aplicación y medio ambiente de desempeño. [2]. De esta manera se manipula con programación a un robot serpiente de manera modular; implementando tarjetas electrónicas en una relación maestro-esclavo para el control articular de cada módulo mecánico. El firmware ha sido escrito en MikroC Pro.

Esta propuesta idealiza la utilización de microcontroladores de una familia sin interfaz USB nativa, aunque su diseño, y aún más, su modalidad hace interesante su uso, pero, esta tarjeta solo ha sido diseñada para la propuesta que lleva su título, sin poder adaptar más funciones, utilizando en su programación un IDE pagado, lo que hacen de su uso, un amplio bagage de conocimientos a personas iniciadas en el uso de sistemas embebidos.

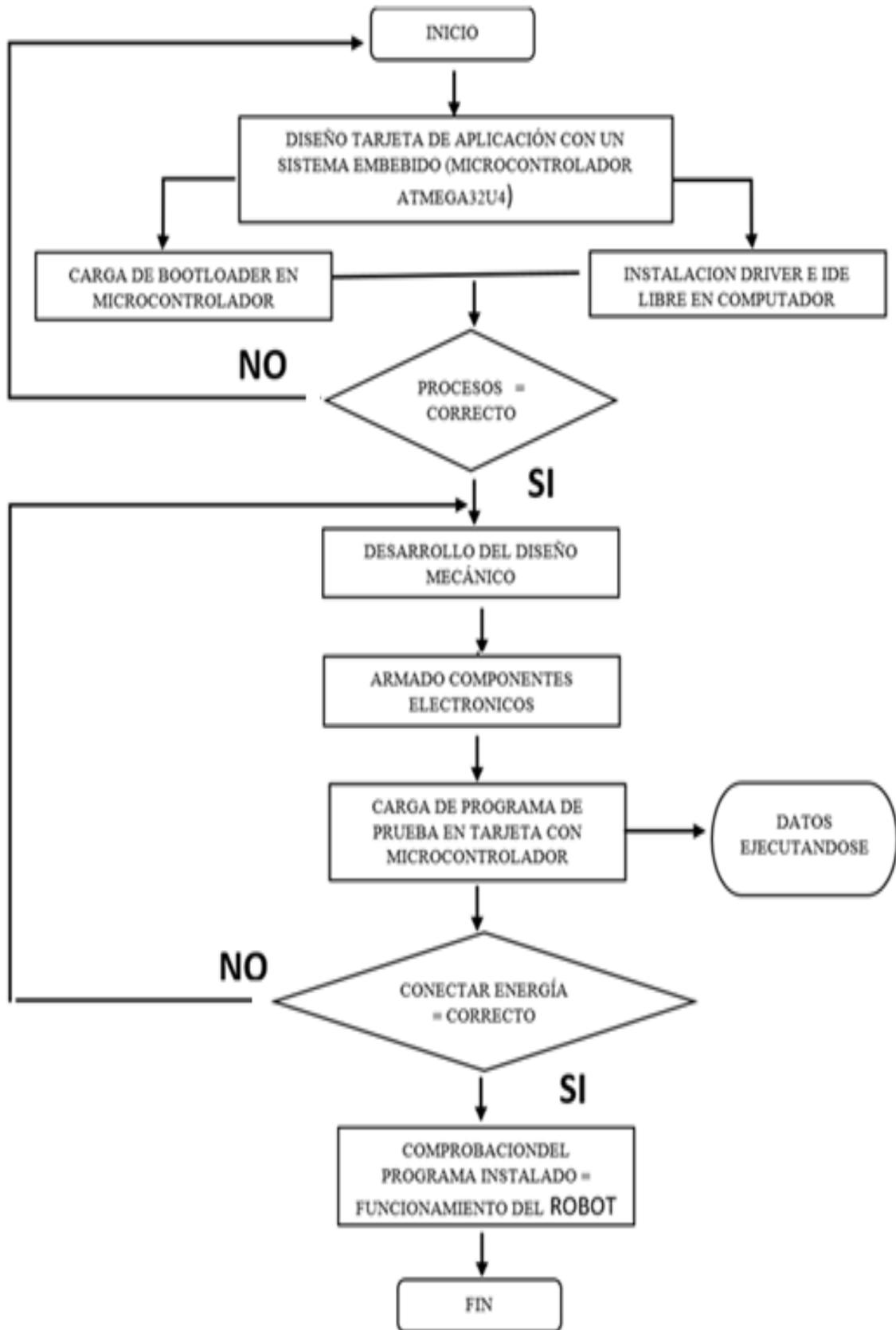


Gráfico 9: Flujograma de un prototipo de robot móvil

### 3.3. Sistemas embebidos en robots móviles

Un robot es un dispositivo con un determinado grado de movilidad, que puede realizar un conjunto de tareas en forma independiente y que se adapta al mundo en el que opera. [5].

Algorítmicamente los sistemas embebidos conceptúan la integridad de un sistema electrónico que combinados en una estructura tanto lógica (programación) como mecánica hacen funcional un prototipo de robot para su construcción. Detallando lo anterior, como se observa en el flujograma planteado, la robótica engloba tres áreas en su fundamentación como son la computación (software), electrónica y mecánica.

### **3.5. Software**

#### **3.5.1. El pensamiento computacional**

Introducir aspectos de programación no sólo apunta a un aprendizaje técnico, sino que permite desarrollar una serie de habilidades, como el pensamiento analítico o de solución de problemas, que son muy requeridos en los trabajos que tienen que ver con tecnología y ciencia, pero que también se pueden aplicar a otras áreas,[5]

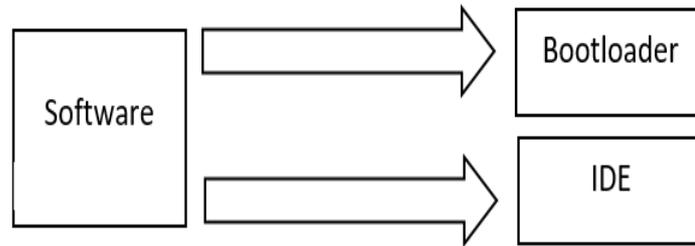
El término acuñado para pensamiento computacional, [4], constata la importancia de resolver problemas dividiendo el mismo en tareas simples, que, miran su entorno emulando y comprendiendo el comportamiento humano, basándose en los conceptos fundamentales de la ciencia de la computación, con diseños de sistemas derivados en el reflejo del espectro inmenso de la informática,

En referencia a lo anterior, y dado el trabajo académico sobre el pensamiento computacional, [4], se denota que haciendo uso de los conceptos fundamentales de la informática, con organización y análisis lógico, se puede abstraer un problema, generando patrones algorítmicos que darán una serie de pasos optimizados, para descomponer factores según la dimensión dada al reto, con la utilización de soluciones más pequeñas cuya composición al unir la abstracción del mismo, lleve a la resolución de lo planteado.

Un programa es un conjunto de instrucciones u órdenes que le indican a la computadora cómo se realiza una tarea. [5].

En los sistemas embebidos para robótica, el pensamiento computacional incorpora el organizar la información con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos, el descomponer un problema cuya combinación lleve a la resolución de un reto para reconocer patrones ayuda a que se generalicen soluciones.

### 3.5.2. Software libre para sistemas embebidos



**Gráfico 10:** El software libre de un sistema embebido

Software libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. [8].

En el campo del software relacionado a la programación de sistemas embebidos de esta propuesta, se optará por software libre, que en esta plataforma, necesita ser instalado en el microcontrolador, el firmware o programa llamado bootloader (gestor de arranque), presta solución a la conexión del prototipo robótico provisto de una tarjeta diseñada con un microcontrolador, con el IDE (entorno de desarrollo integrado) libre instalado en el computador personal, tanto bootloader, como IDE son herramientas de software libre, para programar prototipos de robots móviles.

#### 3.5.2.1. El bootloader

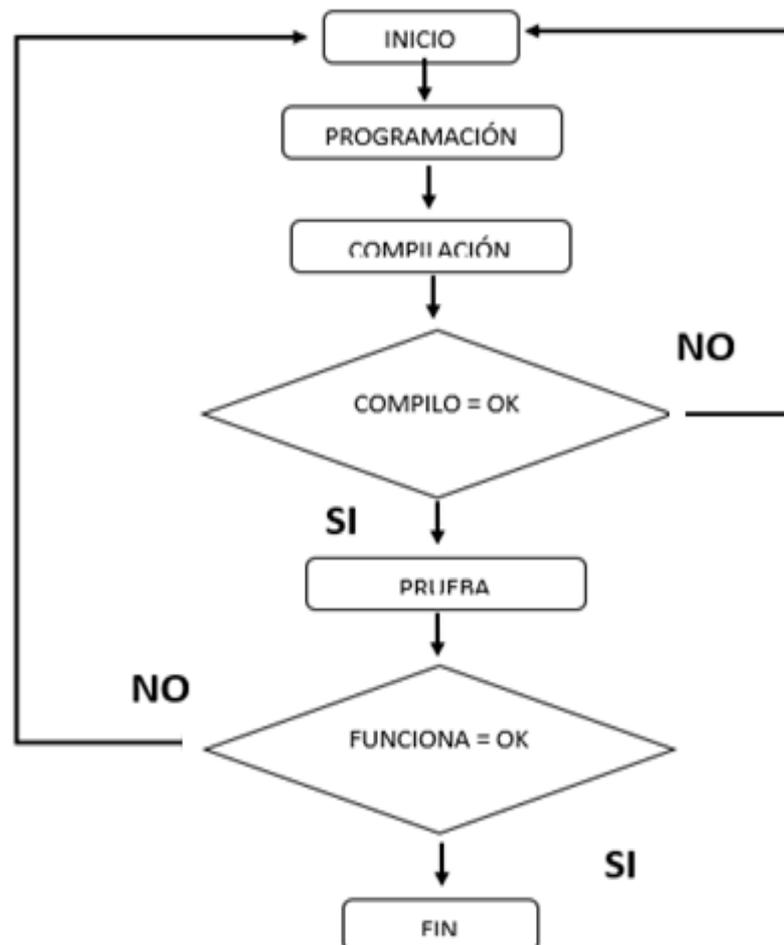
“La función de un bootloader es permitir cargar nuestros programas al microcontrolador conectando la placa a nuestro computador mediante un simple cable USB estándar, pero si ese microcontrolador no tiene grabado ningún bootloader, la escritura de su memoria no se puede realizar de esta forma tan sencilla y debemos utilizar otros métodos, como el ICSP”. [7].

La programación de un sistema embebido utilizando de base un Bootloader o gestor de arranque, se basa en usar un pequeño programa que se ejecuta al iniciar el sistema y que se encarga de gestionar la carga del programa en el que se está trabajando en función de condiciones que se pueda realizar directamente ente el computador personal y la placa que contenga el microcontrolador, sin desmontar electrónica o hardware que ayudara en las pruebas que se está desarrollando.

#### 3.5.2.2. El entorno de desarrollo integrado o IDE

Las siglas IDE vienen de *Integrated Development Environment*, lo que traducido a nuestro idioma significa Entorno de Desarrollo Integrado. Esto es simplemente una forma de llamar al conjunto de herramientas software que permite a los programadores poder desarrollar (es decir, básicamente escribir y probar) sus propios programas con comodidad. [7].

Una plataforma de investigación de prototipos en robótica debe cumplir muchos requisitos, entre los fundamentales esta su accesibilidad, tanto económica como didáctica, un IDE libre, hará que un sistema como el propuesto en este prototipo, resulte útil en el momento del desarrollo de programas aplicable en un prototipo robótico, al reconocer al microcontrolador conectado al computador, y que este cargado en su memoria con un bootloader, como una tarjeta de desarrollo que introducirá en su memoria un programa o firmware de forma simple y sencilla.



**Gráfico 11:** Flujograma de programación

### 3.5.3. Flujograma de programación en un robot móvil

La barra y la consola de mensajes informan en el momento de la compilación de los posibles errores cometidos en la escritura de nuestro sketch (programa), además de indicar el estado en tiempo real de diferentes procesos, como por ejemplo la grabación de ficheros “.ino” al disco duro, la compilación del sketch, la carga al microcontrolador, etc. Es interesante observar también que cada vez que se realice una compilación exitosa, aparecerá en la consola de mensajes el tamaño que ocuparía el sketch dentro de la memoria Flash del microcontrolador. [7].

El flujograma de programación en prototipos robóticos en cualquier máquina construida por un ser humano, son los pasos requeridos para lograr que puede una plataforma presentar autonomía a través de un programa de control realizado en un computador, estos pueden ser para locomoción que le permiten moverse libremente por un espacio definido, o la utilización de sensores, emulando sentidos humanos a través de electrónica que le permiten percibir el entorno, inteligente, para que puede tomar información y responder, según reglas definidas interactuando y creando comunicación.

La palabra Robot viene de Robota, una palabra checoslovaca que significa trabajo.[1].

### **3.5.3.1. Lenguaje de programación en sistemas embebidos**

Por “lenguaje de programación” se entiende cualquier idioma artificial diseñado para expresar instrucciones (siguiendo unas determinadas reglas sintácticas) que pueden ser llevadas a cabo por máquinas. [7].

En un microcontrolador, un lenguaje de programación realiza la función de coordinar acciones a través de software que se graba dentro de un circuito integrado electrónico, el mismo que contiene toda la estructura de un computador, unidad de proceso o CPU, memoria RAM (memoria de acceso aleatorio), memoria ROM (memoria de solo lectura) y circuitos de entrada/salida.

Los bloques condicionales, los bloques repetitivos, las variables, etc, así como también diferentes comandos –asimismo llamados “órdenes” o “funciones” – que nos permiten especificar de una forma coherente y sin errores las instrucciones exactas que queremos programar en el microcontrolador de la placa. [7].

Como todo sistema embebido, este es programable y puede ejecutar un infinito número de tareas y procesos, todos relacionados con los programas que sean aplicados, adaptando código a didácticas que podrá realizar esta plataforma de prototipo robótico.

### **3.5.3.1.2. Lenguaje de programación C++**

El lenguaje C y su “pariente” C++ son dos de los lenguajes más importantes y extendidos del mundo por varias razones: porque son lenguajes muy potentes y a la vez ligeros y flexibles, porque poseen un ecosistema amplísimo de librerías que los dotan de funcionalidad que otros lenguajes no ofrecen, porque los programas escritos y compilados en estos lenguajes son tremendamente eficientes y rápidos, y porque existen compiladores para prácticamente cualquier tipo de hardware (con lo que hoy en día podemos ver multitud de software escrito con estos lenguajes ejecutándose en una gran variedad de máquinas). [7].

La programación de un sistema embebido parte de la comunicación del microcontrolador con el IDE para poder cargar un firmware.

Comercialmente los sistemas embebidos ofrecen opciones diversas para programar un microcontrolador, y dado que se está utilizando un IDE y bootloader libre, esta plataforma utiliza en los programas C++, lenguaje compatible con la herramienta de software escogidos.

El lenguaje de programación C++, fue diseñado a mediados de los años 80 por Bjarne Stroustrup, [23], con muchas variables en su utilidad, como lo es en microcontroladores, en que presenta mecanismos que permiten la abstracción y manipulación de objetos, por esto C++ es un lenguaje de programación multiparadigma (programación estructurada y la programación orientada a objetos).

Es evidente que es imposible escribir un programa directamente en código máquina: por eso existen los compiladores. [7].

Destacando la versatilidad del IDE utilizado en esta propuesta, que es el que compila las instrucciones realizadas en C++, se ha de denotar, que esta interfaz (IDE) cumple varias funciones, entre ellas está la de ser un bloc de notas que recibe el lenguaje que introduce el usuario por teclado, y como segundo parámetro, está el de transformar el lenguaje de alto nivel C++ en instrucciones de ceros y unos o lenguaje máquina, para que el microcontrolador realice las tareas programadas.

## **3.6. Hardware**

### **3.6.1. Definición de robótica**

El origen de la palabra robot es atribuida a una obra checoslovaca publicada en 1917, denominada Rossum's Universal Robots, su autor fue Karel Kapek, quien acuñó la palabra robota, que significa servidumbre o trabajador forzado, [5], dicha obra mimetiza el comportamiento de los seres humanos reproducidos en un ente no biológico.

El futuro depara la coexistencia humano máquina, siendo este artilugio material el complemento que a diario se utilizara como otra herramienta más en el bolsillo del profesional; por lo que, para definirlo, la robótica se estudia lo expectante del mundo, una ciencia multidisciplinaria que explora el desarrollo de factibilidades mecánicas y electrónicas en el desarrollo de tecnologías aplicables en funciones programables en dispositivos llamados robots.

#### **3.6.1.1. Usos de la robótica**

El uso de robots diseñados para cumplir infinitas funciones, crean situaciones complejas como el uso de los robots móviles con características didácticas, dado que la evolución tecnológica en la robótica jugara un papel central en el futuro dentro de las investigaciones científicas, los prototipos móviles que engloben en su caracterización de desarrollo el aprendizaje de la programación de forma simple e intuitiva en ambientes académicos universitarios serán indispensables para la robótica, [6].

#### **3.6.2. Los sistemas embebidos en la robótica**

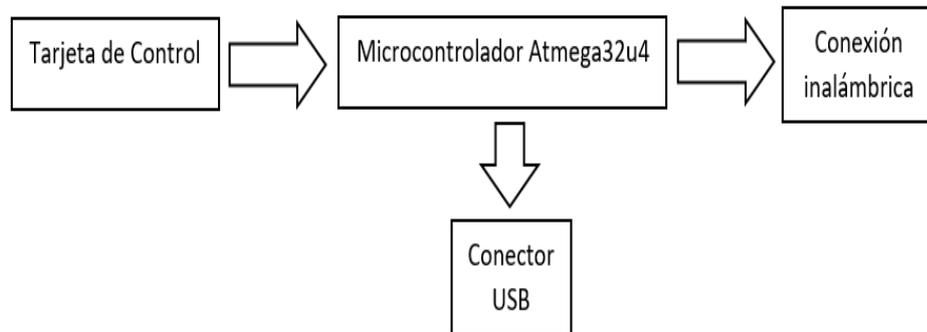
Un microcontrolador es un circuito integrado o “chip” (es decir, un dispositivo electrónico que integra en un solo encapsulado un gran número de componentes) que tiene la característica de ser programable. Es decir, que es capaz de ejecutar de forma autónoma una serie de instrucciones previamente definidas por nosotros. [7].

Los sistemas embebidos o microcontroladores son circuitos integrados electrónicos programables considerados como el cerebro de un sistema como el robótico, el cual al ser algorítmico hace sustentable la concepción de control programable en aplicaciones para el ambiente tecnológico actual, conjugando características en dispositivos altamente eficientes para el uso de memoria de datos y secuencias, demostrados en velocidad de ejecución, que por ejemplo son utilizados desde utensilios como máquinas de cafetería a un computador en el cual se hace todos los procesos que el usuario desempeñe tanto en hobby como en trabajo.

##### **3.6.2.1. Electrónica de una tarjeta con microcontrolador**

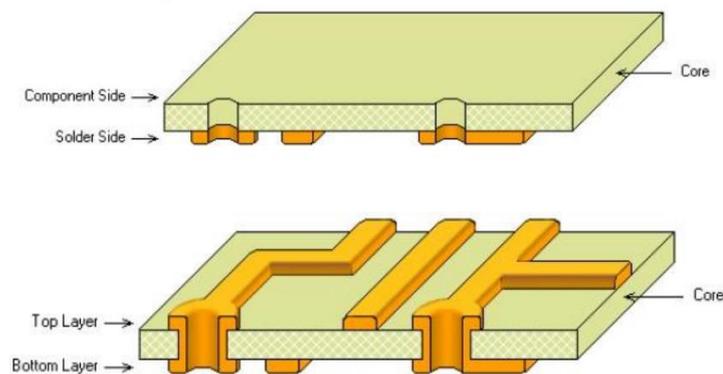
El estudio de las tarjetas de desarrollo busca desarrollar placas que permitan el diseño de sistemas reales, fortaleciendo la programación de tal manera que se integren los conocimientos

de electrónica, electricidad y sistemas con la praxis a través de dispositivos especialmente diseñados para actividades académicas. [9].



**Gráfico 12:** Tarjeta de control

En el diseño y aplicación de dispositivos que generan transmisión y recepción de información; esta plataforma propuesta usa electrónica programable aprovechada de un microcontrolador, para entradas y salidas tangibles en los pórtricos conectados y programados para el funcionamiento del prototipo de robot móvil propuesto, que íntegramente estarán dispuestos en una PCB (placa de circuito impreso), diseñada para esta plataforma de control de un robot móvil.



**Gráfico 13:** El PCB

**Fuente:** Pérez-Cáceres, Metodología para la Simulación y Fabricación de Circuitos Impresos en Radiofrecuencia

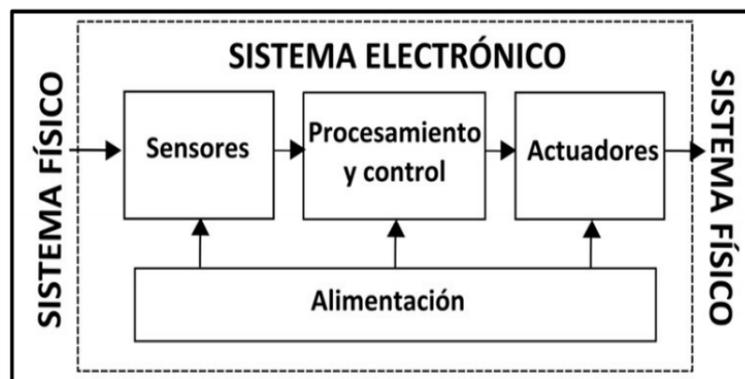
### 3.6.3. El PCB

Actualmente, los circuitos impresos (PCB), constituyen el medio que más se utiliza para la integración de sistemas electrónicos funcionando como interconexión y soporte de los componentes. [10].

Las Placas de Circuito Impreso (PCB) están en todos los dispositivos electrónicos, desde su aparición son utilizadas para la creación de placas base de todo tipo, sobre todo para reducir en gran medida el espacio utilizado y poder hacer así, crear dispositivos electrónicos más pequeños con patrones de interconexión, de diferentes componentes electrónicos de, o a otras capas de la propia PCB, para la utilización de los diferentes puertos de entrada y salida que se precisarán en un dispositivo electrónico.

La fabricación de una PCB está compuesta de un material aislante que es recubierto de un material conductor como el cobre, el cual presenta un parámetro de espesor estándar de 1.6mm de material depositado en forma plana en una superficie plana.

Una PCB une con diferentes líneas planas conductoras diseñadas sobre el lado cobreado del material aislante, las conexiones de todo el circuito que se diseñó.



**Gráfico 14:** El microcontrolador

**Fuente:** Arduino. Curso práctico de formación

#### **3.6.4. Descripción del microcontrolador Atmega32u4**

Un microcontrolador es un circuito integrado o “chip” (es decir, un dispositivo electrónico que integra en un solo encapsulado un gran número de componentes) que tiene la característica de ser programable. Es decir, que es capaz de ejecutar de forma autónoma una serie de instrucciones previamente definidas por nosotros. [7].

El microcontrolador ATmega32U4 presenta un formato SMD (Elemento de montaje superficial), ofrece funcionalidades superiores a otros elementos de su misma rama, incorpora memoria RAM, y comunicación USB directa, sin necesidad de circuitos suplementarios para comunicación por este pÓrtico.

**Tabla 2:** Características atmega32u4

MICROCONTROLADOR	Atmega328	Atmega2560	Atmega328	Atmega32u4	Atmega32u4 +AR9331	Atmel SAM21 +ATA8521
<b>Pines Digitales I/O</b>	14 (6 PWM)	14 (15 PWM)	14 (6 PWM)	20 (7 PWM)		8
<b>Pines AnalÓgicos</b>	6	16	8	12		8
<b>Memoria Flash (KB)</b>	31,5	256	30	32		256
<b>UART</b>	1	4	1		1 (no disponible)	1
<b>SPI</b>	SI					
<b>I2C</b>	SI					

**Fuente:** Dispositivo electrÓnico para el estudio de redes

### 3.6.5. Conexi3n inalÁmbrica de aplicaciones con sistemas embebidos

La tecnologÍa inalÁmbrica Bluetooth, es la primera comunicaci3n abierta que utiliza ondas de radio de corto alcance (2.4 Giga hertzios de frecuencia) con el objetivo de comunicar dentro de un radio de 10 metros dispositivos m3viles o computadores y otros dispositivos mÁs pequeÑos (de pila de bot3n). [12].

La tecnologÍa inalÁmbrica Bluetooth es un Área de aplicaci3n del espectro radioelÉctrico que conecta dos dispositivos bidireccionales configurados segÚn la necesidad del problema planteado, sin la utilizaci3n de contactos fÍsicos para el envi3-recepci3n de datos inmersos en ondas dispersas en el aire a una frecuencia de 2.4 Ghz.

Una de las tecnologÍas mÁs comunes es el Bluetooth, que permite generar una red inalÁmbrica entre dispositivos m3viles, lo cual facilita la interacci3n entre smartphones y robots a travÉS de aplicaciones como Bluetooth terminal. [13].

La utilizaci3n de m3dulos de interconexi3n en prototipos de robots m3viles, ocupa funciones de transmisi3n-recepci3n; la funci3n de emisi3n, se utiliza una App con interfaz grÁfica programada en un smartphone.

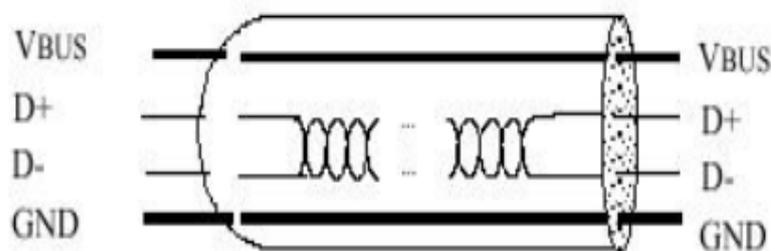
En la parte del receptor que esta conecta al microcontrolador, utilizará un dispositivo Bluetooth, el cual se integrará a la tarjeta de esta propuesta; al dispositivo bluetooth se le interconectará a el sistema embebido, con sus respectivos pines.

### 3.6.6. Conector USB

USB is a likely solution any time you want to use a computer to communicate with an external device. Internal devices, such as fingerprint readers, can use USB as well. The interface is suitable for mass-produced, consumer devices as well as specialized, small-volume products and one-of-a-kind projects. [14].

La comunicación serie del protocolo USB es la base de las transmisiones de datos de los microcontroladores que estén basados en este estándar, estándar que de forma física un hardware definido en un conector mecánico para comunicar y proveer de alimentación eléctrica a periféricos y dispositivos electrónicos,

A nivel eléctrico, el cable USB transfiere la señal y la alimentación sobre 4 hilos. [15].



**Gráfico 15:** El USB

**Fuente:** Ingeniería en Microcontroladores, Protocolo USB [15]

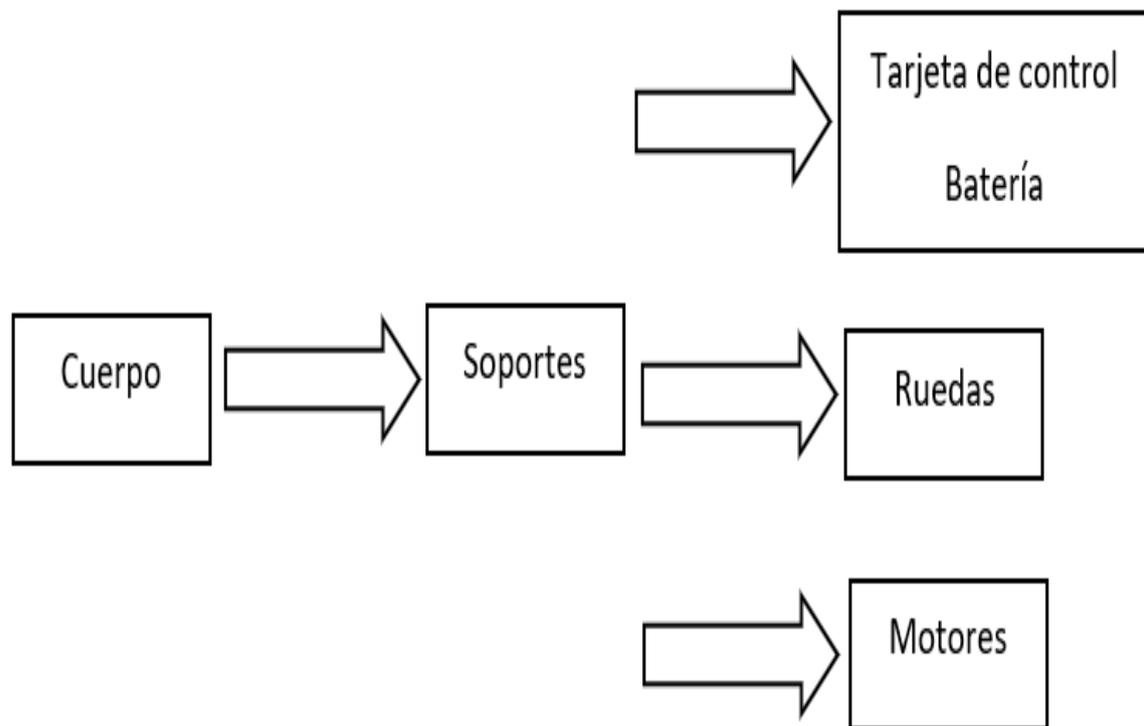
Como indica el grafico superior, un USB ha hecho que la forma que un microcontrolador se comunica con un computador se estandarice, y el protocolo que provee esta comunicación pueda venir internamente como el caso del microprocesador Atmega32u4, facilitando que es el dispositivo que controla los puertos y dispositivos realicen las operaciones de recepción transmisión de datos.

### 3.7. Estructura mecánica para un prototipo robótico móvil

En la mecánica, esta se basará generalmente en el diseño de la estructura que contendrá al robot, donde se depositaran todos los componentes del mismo, actualmente, el ámbito robótico está teniendo un gran aliado con la tecnología de impresión 3d, que hace plausible diseños prácticos

y muy exactos en materiales plásticos fundidos, esta propuesta hará uso de esta tecnología para la realización de esta plataforma.

En el lenguaje de los gráficos en 3D, un modelo es un archivo que contiene la información necesaria para ver o “renderizar” un objeto en tres dimensiones. [16].



**Gráfico 16:** Mecánica de un prototipo

El prototipo del modelo mecánico del robot desplegará un esquema geométrico del prototipo para crear el diseño final, mismo que se puede dividir por bloques.

### **3.7.1 Programa de diseño SolidWorks**

El programa SolidWorks, emplea el diseño asistido por computadora, en la modelación de piezas y conjuntos, para extraer los planos y simular su comportamiento (esfuerzos, transferencia de calor, movimiento, etc.) con el propósito de optimizar y ajustar la estructura del proyecto, [31].

El producto SolidWorks es un software del tipo CAD, gratuito en la versión de uso estudiantil, muy utilizado para diseño mecánico.

Su entorno está basado en gráficos que permite de manera intuitiva la rápida modelación de sólidos en 3d, con opciones de fácil opción para modificar el diseño del modelo proyectado,

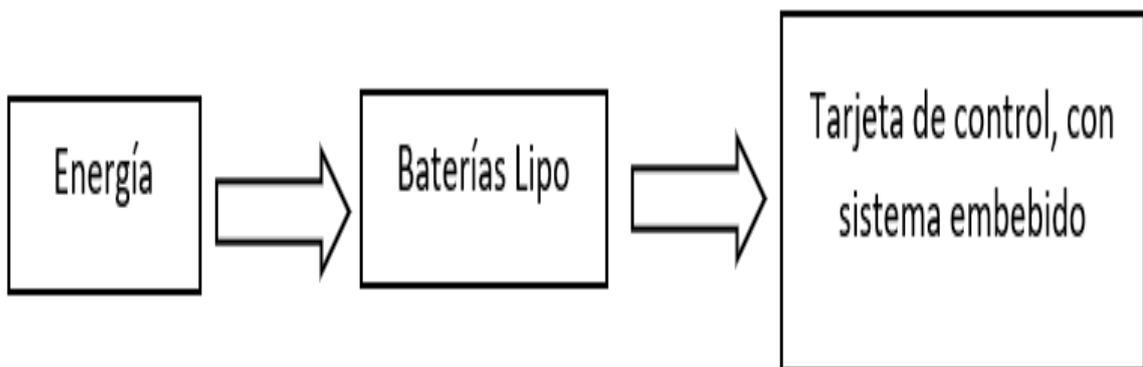
proporciona al usuario un software de diseño de fácil uso con poderosas herramientas que hacen del proceso de crear en 3d, un método validado que comunica la gestión del desarrollo correcto antes de fabricar.

### 3.7.2. La energía en un prototipo de robot móvil

Llamamos fuente de alimentación eléctrica al elemento responsable de generar la diferencia de potencial necesaria para que fluya la corriente eléctrica por un circuito y así puedan funcionar los dispositivos conectados a este. Las fuentes que utilizaremos más a menudo en nuestros proyectos serán de dos tipos: las pilas o baterías y los adaptadores AC/DC. [7].

La energía del robot estará proporcionada por una batería, que es un elemento eléctrico que transforma energía química en energía eléctrica, internamente tiene elementos llamados celdas, que, conectados en serie, dan un voltaje y amperaje; la batería se calcula según lo que el robot necesite.

Las baterías proveen voltaje continuo provisto por dos polos, un positivo y un negativo, el cual puede variar según las necesidades del proyectista que este diseñando el robot, este voltaje es suministrado de fuentes como baterías de alta prestación del tipo LIPO (polímero de litio) que son recargables.



**Gráfico 17:** La batería

## 4. METODOLOGÍA

### 4.1. Procedimientos

Las variables de la investigación fueron precedentemente señaladas, con lo que se implementara un cuadro donde se describen la forma en que se han obtenido la información, para lo cual se expone sus dimensiones, aspectos, indicadores, ítems e instrumentos.

#### **4.1.1. Operacionalización de variables**

Para resumir lo anterior, se ha procedido a un análisis de resultados e interpretación de las consecuencias operativas de las variables (características y atributos) de la propuesta, para el diseño y aplicación investigativa en la que se definirá la investigación y metodología basados en los objetivos e hipótesis.

#### 4.1.1.2. Variable independiente

Sistemas embebidos.

**Tabla 3:** Operacionalización de la variable independiente

<b>Variable Independiente</b>	<b>Categorías</b>	<b>Indicadores</b>	<b>Ítems</b>	<b>Instrumento</b>
<b>Concepto de Sistemas embebidos</b>  Es un circuito integrado electrónico que en un solo encapsulado contiene a una computadora programable. [7]. Este circuito integrado o sistema embebido cumple una labor específica, [23], utilizando para ello toda la arquitectura de un computador, esto es CPU, memorias RAM, ROM, y circuitos de entrada y salida. [24].	CPU (Unidad central de proceso).	Velocidad de funcionamiento del microcontrolador en MHz.	Frecuencia en Mhz.	Medición con frecuencímetro de Mhz.
	Memoria RAM/ROM.	Memoria en kilobytes para grabado de programas en microcontrolador.	Capacidad en Kb.	Despliegue de información en el IDE de la capacidad de memoria.
	Unidades de entrada y salida de señales.	Entrega de un voltaje de 5v en los pines de salida del microcontrolador, y recepción de señales de 5v en los pines de entrada del microcontrolador.	Capacidad entrada salida en voltios.	Verificación con un multímetro de los pines de entrada y salida del microcontrolador del voltaje enviado y recibido.
	Codificación y compilación.	Verificación en pantalla del IDE de los Kb utilizados en memoria del programa guardado.	Información desplegada por el IDE de Kb. utilizados.	Despliegue de información en el IDE de la memoria utilizada en KB. de la compilación y grabado del programa en el sistema embebido.

### 4.1.1.3. Variable dependiente

Construcción de Robots Móviles.

**Tabla 4:** Operacionalización de la variable dependiente

<b>Variable Dependiente</b>	<b>Categorías</b>	<b>Indicadores</b>	<b>Ítems</b>	<b>Instrumento</b>
<p><b>Concepción de construcción de Robots Móviles.</b></p> <p>Los robots móviles se pueden clasificar de acuerdo con el tipo de locomoción utilizado: ruedas, patas y orugas. [25].</p> <p>En robots con ruedas, se condiciona a diámetros de rueda y motores que deben ser definidos. [4].</p> <p>Actualmente los prototipos de robots móviles se proyectan e imprimen en 3d, diseñando un objeto en tres dimensiones. [16].</p>	Construcción de un robot móvil con impresión 3d.	Estimación de tiempo de la impresión del prototipo de robot móvil.	Modelo geométrico impreso en 3d.	Registro del tiempo de impresión del prototipo en 3d.
	Locomoción y libertad de movimiento en 360 grados en un plano bidimensional.	Medición en grados del desplazamiento del prototipo de robot móvil.	Grados de desplazamiento del prototipo.	Graduación en grados de la posición del prototipo robótico por movimiento independiente de las ruedas para cambio de posición.
	Semiautónomo.	Señal eléctrica de 5v DC en sensores del prototipo de robot móvil.	Voltaje entregado por el sistema embebido.	Verificación del voltaje entregado con un multímetro de los pines de los sensores conectados a la tarjeta de control.
	Comunicación remota.	Señal eléctrica de 5v DC en la tarjeta de control del prototipo de robot móvil en respuesta a acciones realizadas en la aplicación de control en un teléfono inteligente.	Voltaje entregado por el sistema embebido.	Verificación del voltaje de los pines de salida del módulo de control remoto bluetooth integrado en la tarjeta de control del prototipo de robot móvil.

## **4.2. Implementación de la metodología**

### **4.2.1. Introducción**

Métodos lógicos. Son aquellos que se fundamenta en el razonamiento en función de procesos inductivos o deductivos. Este tipo de métodos están relacionados con disciplinas del saber. [28].

El proyecto de aplicación de sistemas embebidos en prototipos de robots móviles es interpretativo, desarrolla un análisis crítico y establece la explicación sintetizada y técnica del estudio de la utilización de tarjetas de control programables con sistemas embebidos en un club de robótica, además, la descripción y comprensión de los elementos electrónicos de las mismas.

La investigación por ser sistemática, genera procedimientos, presenta resultados y debe llegar a conclusiones, ya que la sola recopilación de datos o hechos y aun su tabulación no son investigación, solo forman parte importante de ella. La investigación tiene razón de ser por sus procedimientos y resultados obtenidos. [26].

Etimológicamente, un método es un camino para llegar a un fin propuesto, aplicando el juicio y estudio de una forma disciplinaria, para que el pensamiento y a las acciones, obtengan eficientemente un resultado en un proyecto, facilitando los procesos para dinamizar las actividades.

### **4.2.2. Enfoque**

Cuantitativo. En la investigación presente se hace objetiva a través de la utilización de sistemas embebidos para el desarrollo de prototipos robóticos, utilizando una metodología con enfoque cuantitativo.

La información obtenida recolecta datos registrados en forma numérica, para analizar y comprobar la correlación del problema, especificando y delimitando los datos obtenidos en función de las variables.

Por otra parte, este estudio se considera de carácter descriptivo, de modo que las estrategias de investigación utilizadas se presentan como alternativas que parecen apropiadas en una etapa de aproximación inicial al tema de las aplicaciones de sistemas embebidos, particularmente en un contexto específico, como es el de los prototipos robóticos.

### **4.3. Tipo de investigación**

#### **4.3.1. Investigación bibliográfica**

La investigación bibliográfica, consiste en explorar, revisar y analizar libros, revistas científicas, publicaciones y demás textos escritos por la comunidad científica en formato impreso o material en línea. [28].

La utilización de la bibliografía para la realización del proyecto de investigación se estableció como bases teóricas bibliográficas determinadas por las variables independiente y dependiente del proyecto, sistemas embebidos y construcción de robots móviles, en los que se pudo establecer diferentes autores que han realizado un estudio profundo y han llegado a establecer parámetros en la construcción de este tipo de sistemas móviles.

#### **4.3.2. Investigación de Campo**

El objeto de estudio de la investigación de aplicaciones de sistemas embebidos en prototipos de robots móviles elaborados en un club de robótica, considera los lugares donde se desarrollan esas actividades, recopilando experiencia de los que compiten con clubes de similares actividades, permitiendo así, el análisis de elementos utilizados en robótica.

#### **4.3.3. Investigación descriptiva**

Las aplicaciones de sistemas embebidos en tarjetas de control para prototipos robóticos deben ser descritos, tomando en cuenta que cada prototipo robótico que se desarrolla es diferente en su construcción, lo que se determina con particularidades funcionales, es así, que la investigación descriptiva ayuda a investigar mediante la observación dichas funcionalidades.

Además, tomando en cuenta que el estudio de las aplicaciones de sistemas embebidos programables, abre camino en el planteamiento actual de la robótica, se elaboraran apuntes de observación y descripción de acuerdo a funcionalidades en aplicaciones de prototipos robóticos.

#### 4.4. Métodos Técnicas Instrumentos

**Tabla 5:** Métodos Técnicas Instrumentos

No.	MÉTODOS	TÉCNICAS	INSTRUMENTOS
1	<p><b>Inductivo</b> Para realizar este proyecto de aplicaciones de sistemas embebidos, se investigará partiendo de un tema específico, desglosado en subtemas y así llegar al estudio de su utilización en tarjetas de control en prototipos de robots móviles.</p>	<p><b>Fichaje</b> Esta técnica ayuda a recoger bibliográficamente los libros, artículos y revistas, adecuados para la investigación.</p>	<p><b>Fichas bibliográficas</b> Este tipo de instrumento facilita la investigación median la lección de libros apropiado para poder concluirá con el proyecto.</p>
2	<p><b>Deductivo</b> La investigación del proyecto se desarrollará desde un estudio de la utilización de sistemas embebidos en un club de robótica, de esta manera, el método permitirá partir de una síntesis para lograr llegar a un resultado concreto.</p>	<p><b>Observación</b> Es importante este tipo de técnica ya que ayudara a recopilar información mediante la observación directa de los prototipos robóticos, además, del estudio de los elementos que los componen.</p>	<p><b>Apuntes de descripción</b> Fichas de Observación.</p>
3	<p><b>Descriptivos</b> El proyecto tiene como objetivo la descripción de los sistemas embebidos en prototipos de robots móviles, mediante este método se podrá hacer un análisis descriptivo de los elementos que componen estos robots.</p>	<p><b>La entrevista</b> Sera una guía y herramienta, realizada a docentes profesionales de la especialidad involucrados con sistemas embebidos en prototipos robóticos, quienes proporcionaran información pertinente de diferentes puntos de vista, que facilitara la realización del proyecto, (adjunta en Anexos).</p>	<p><b>Encuesta</b> Cuestionario de entrevista.</p>

#### 4.4.1. Población y muestra

##### 4.4.1.1. Unidad de estudio

Aplicadas a las personas de las cuales se va a obtener la información para la diligencia de las técnicas de investigación.

##### 4.4.1.2. Población

En el ámbito de la investigación se ha tomado en cuenta a profesionales del área de robótica, y la población del club de robótica de la Universidad Técnica de Cotopaxi.

**Tabla 6:** Población

INVOLUCRADOS	CANTIDAD	%
Docentes de la especialidad	5	1.3 %
Coordinadores	1	0.26 %
Estudiantes	20	98.44 %
Total	26	100 %

Es importante señalar que, por tener una población pequeña, no se amerita el cálculo de la muestra, dando a conocer que se trabajara con toda la población.

#### 4.5. Método de desarrollo

Métodos empíricos. Los métodos empíricos se acercan al conocimiento de la realidad por medio del estudio directo y aplicación de la experiencia. [28].

Se utiliza la metodología para el desarrollo de máquinas del tipo mecatrónico, es decir máquinas en donde los sistemas mecánicos, electrónicos y computacionales convergen de forma significativa en el diseño del mismo. [30].

La metodología de desarrollo a utilizar viene de la mano en el seguimiento de un plan y adaptación continua a las circunstancias de la evolución del proyecto, con práctica y objetiva construcción de una aplicación de sistemas embebidos para robots móviles, que converja el diseño computacional integrado electrónico programable experimental que ofrece un microcontrolador de la familia Atmega32u4.



La disposición propuesta para las ruedas con tracción es de dos, con tracción independiente ubicadas en la parte posterior.

Las tres ruedas configurarán un triángulo para garantizar estabilidad del prototipo.

2. Diseñar completamente el prototipo mecánico del proyecto.

La utilización del software SolidWorks en la versión educativa, aportara la interfaz en la que se plasmara con medidas, y según la aplicación central de control del sistema embebido, un diseño en 3d.

Las dimensiones del chasis a ser diseñado e imprimir en 3d, serán de 10cm x 10cm.

3. Verificar por técnicas computacionales los resultados del diseño.

Para procesar los archivos de diseño en 3D generados con SolidWorks, se procede a una revisión visual del objeto en la pantalla del programa, ayuda que presta el software con una generación esquemática; el objeto será, desglosado en secciones antes de ser fabricado en la impresora 3D.

El proceso anterior se lo realizara en CURA, un software libre de transformación de código STL (standard triangle language) para transmitir a la máquina 3D las instrucciones de impresión.

4. Construir el prototipo con los mecanismos que han sido probados en simulación.

Se procederá a unir las partes impresas en 3d, con lo cual el prototipo conjugará ensambles y tornillería en la fijación de elementos del mismo.

5. Diseñar el sistema de control que se utilizará.

La unidad central de control es la que contendrá al sistema embebido Atmega32u4, que se ubicara en una PCB, unida a elementos electrónicos que unidos creen el control del prototipo de robot móvil, para lo cual se utilizara el software libre Kicad, en el que se trazaran tanto líneas de cobre, como sócalos y sujetadores de integrados que se soldaran y energizaran para probar el funcionamiento correcto de esta unidad.

En la PCB, ya soldados y ubicados los elementos de la tarjeta de control, se procederá a grabar el bootloader en el integrado Atmega32u4, para lo cual se utilizará el IDE de Arduino como interfaz de carga de firmware en el integrado, se utilizara la herramienta Arduino como quemador de integrado a integrado, que se conectará al computador como programador de la memoria del

integrado, para grabar el bootloader, proceso que dará un gestor de arranque para cada vez que la tarjeta de control se conecte, el computador y el IDE de Arduino, reconozca la misma, posibilitando grabar en memoria programas realizados por el usuario.

6. Integración de prototipo con los componentes electrónicos necesarios para inteligenciar el mismo.

Para completar el prototipo móvil, se unirá la estructura impresa en 3d con la tarjeta de control, conectando componentes externos; previo a ser la unidad central programada y probada en funcionalidades del prototipo diseñado.

7. Realizar las pruebas del funcionamiento del prototipo equipado con sus sensores y actuadores, mediante el monitoreo del mismo utilizando medios computacionales para generar pruebas en tiempo real.

Se procederá a grabar un programa en la tarjeta de control, con lo cual se realizará pruebas de movimiento del prototipo de robot móvil.

## **5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS**

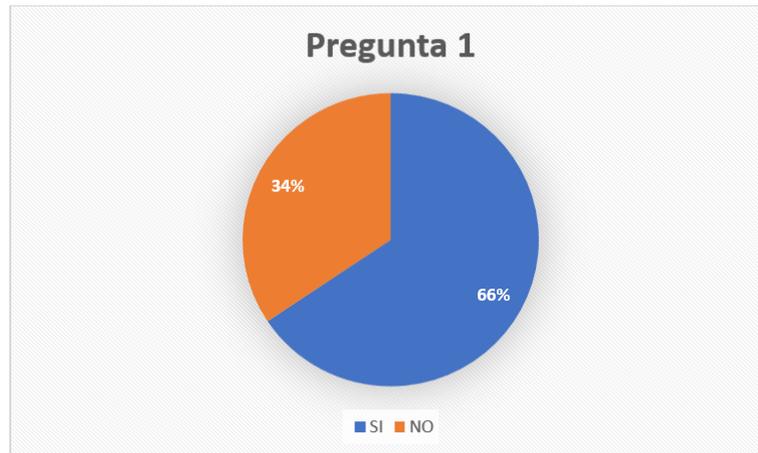
En esta sección de la investigación se contrastará los resultados obtenidos de la utilización de los instrumentos utilizados, encuesta y entrevista, utilizados en la investigación.

### **5.1. Resultados de la encuesta realizada**

Durante el transcurso de la realización de esta tesis, se llevó a cabo una encuesta (ver anexos) para medir conocimiento y aplicabilidad en relación a los sistemas embebidos, las respuestas recabadas han proporcionado una orientación valiosa para establecer un orden de prioridad que es preciso introducir en esta propuesta (anexo 9: encuesta).

El objetivo de esta encuesta es: Identificar las experiencias en uso de aplicaciones de sistemas embebidos a través de una investigación especializado en el tema para un mejor entendimiento de los dispositivos electrónicos, el mismo que se consultó a través de las siguientes preguntas:

**Pregunta 1:** ¿Cree UD. que se cuenta con el conocimiento para el funcionamiento y programación de un sistema embebido en una tarjeta de control en prototipos de robots móvil?

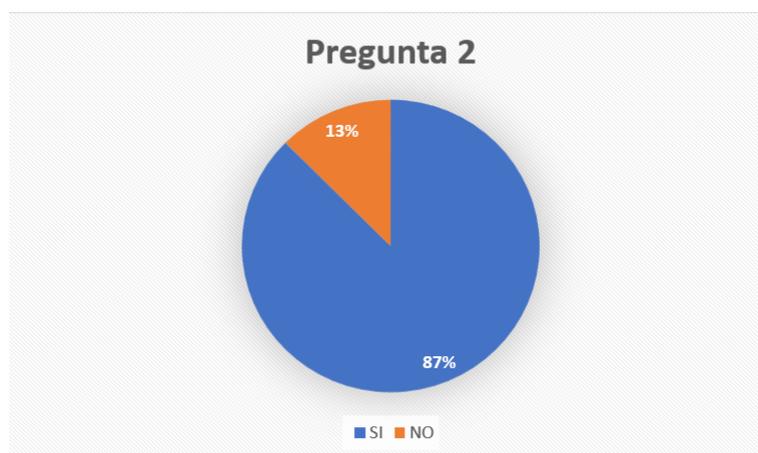


**Gráfico 19:** Resultado pregunta 1

**Interpretación:** La gráfica refleja que el 66% de los usuarios tienen conocimiento sobre sistemas embebidos, mientras el 34% no poseen el conocimiento para poder utilizar esta tecnología.

**Análisis:** De acuerdo con los resultados de esta pregunta, en las encuestas se ha podido determinar que toda la población a la que está dirigida esta propuesta, tiene conocimiento en el funcionamiento de los sistemas embebidos, y cree que se debe de contar con un grado de conocimiento básico en programación, para utilizar tarjetas de control que determinen movimientos en automatismos que cumplan una determinada tarea.

**Pregunta 2:** ¿Conoce UD. que la memoria RAM y ROM de un sistema embebido se utilizan para el grabado y despliegue de programas que se diseñen para control en prototipos?

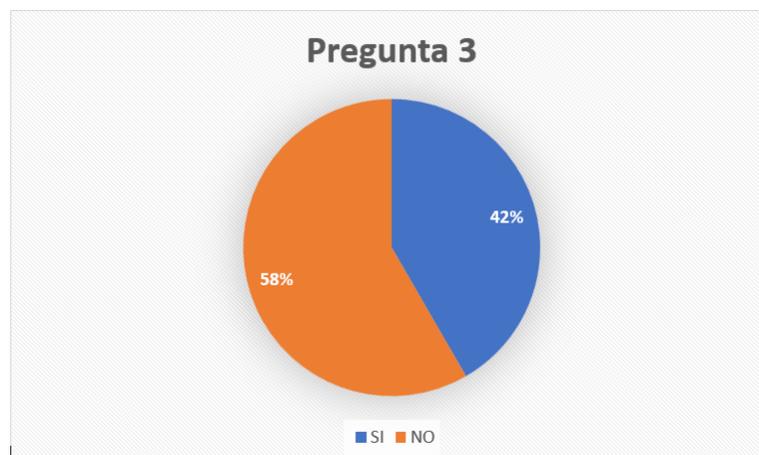


**Gráfico 20:** Resultado pregunta 2

**Interpretación:** Un 87% tiene conocimiento estructural de la tecnología que tiene un sistema embebido para su programación, y un 13% no posee esa información.

**Análisis:** De acuerdo con los resultados de esta pregunta, se conoce que el funcionamiento de un sistema embebido debe de contar con una preparación básica de lo que es programación y electrónica para su diseño, el cual permita determinar cuál debe ser las rutas que debe recorrer el sistema robótico.

**Pregunta 3:** ¿Conoce UD. que los sistemas embebidos trabajan con 5 voltios de corriente continua, en entrada y salida de señales programables que controlan prototipos robóticos?



**Gráfico 21:** Resultado pregunta 3

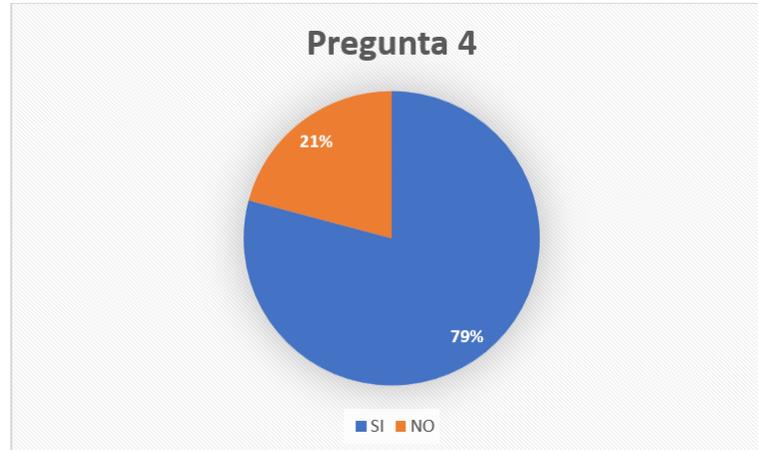
**Interpretación:** En esta pregunta se destaca un 42% de personas encuestadas con conocimiento de la energía para el funcionamiento de un sistema embebido y su operabilidad en aplicaciones robóticas, y un 58% desconoce estos conocimientos.

**Análisis:** De acuerdo con los resultados de esta pregunta, comúnmente los sistemas embebidos trabajan con un sistema de voltaje de 5 voltios de corriente continua, las que emiten señales de salida que previamente deben ser programadas para controlar su funcionamiento.

**Pregunta 4:** ¿Utilizaría Ud. un sistema embebido que sea programable de forma amigable con un IDE y lenguaje fácil de codificar en aplicaciones de prototipos de robots móviles?

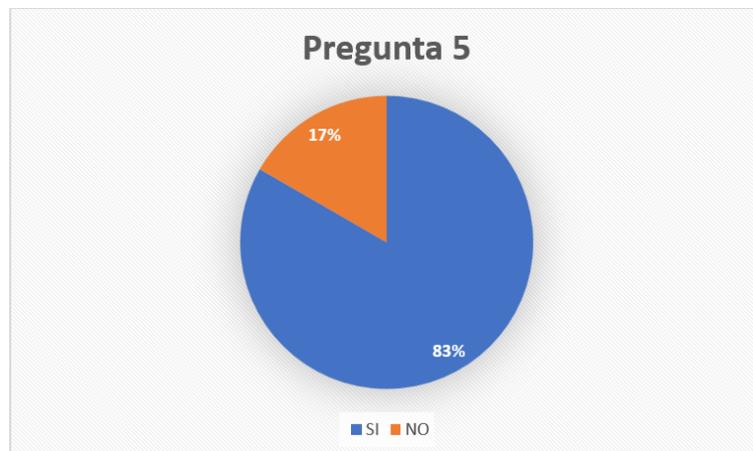
**Interpretación:** Un 79% de los encuestados ocuparían sistemas embebidos programables si cuentan con un software amigable para robótica, y un 21% no utilizarían este tipo de sistemas.

**Análisis:** De acuerdo con los resultados de esta pregunta, el lenguaje que más se sugiere para que funcione la aplicación con sistemas embebidos, denotan que debe ser amigable, para que sea utilizado por una persona que tenga conocimientos básicos en esta área del saber.



**Gráfico 22:** Resultado pregunta 4

**Pregunta 5:** ¿Tiene UD. conocimiento que se pueden construir objetos impresos en 3d relacionados con prototipos robóticos?

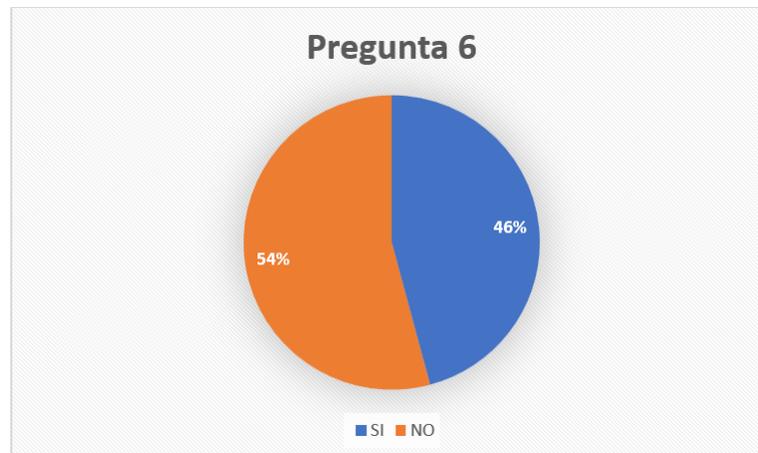


**Gráfico 23:** Resultado pregunta 5

**Interpretación:** Un 83% poseen conocimientos de la tecnología en 3D para imprimir robots, y un 17% no poseen ese conocimiento de la tecnología actual utilizada en impresión de robots.

**Análisis:** De acuerdo con los resultados de esta pregunta, la aplicación debe contar con un diseño que permita ser impreso en las impresoras 3d el cual puede acoplarse a diferentes actividades relacionadas a la creación de prototipos.

**Pregunta 6:** ¿Conoce Ud. que los robots móviles del tipo terrestre tienen grados de libertad en su movimiento?

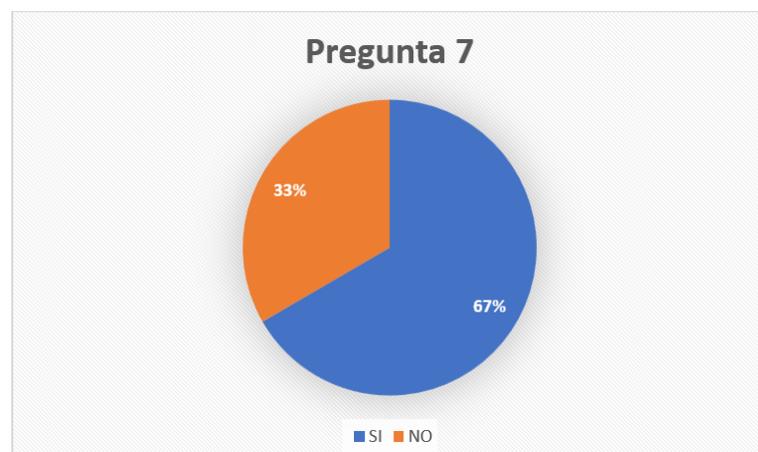


**Gráfico 24:** Resultado pregunta 6

**Interpretación:** Un 54% de los encuestados conocen de la cinemática de movimientos en un robot, y un 48% no poseen este conocimiento de movimiento en un robot.

**Análisis:** De acuerdo con los resultados de esta pregunta, los robots móviles deben contar con una orientación basadas en grados de libertad, como el giro completo en su base de 360 grados para robots móviles terrestres, lo cual ayuda a ser multidireccional y programable según la actividad a desarrollar, de acuerdo con la programación requerida.

**Pregunta 7:** ¿Ha utilizado UD. sensores y actuadores de 5 voltios en sistemas embebidos que ayuden en la experimentación práctica con prototipos de robots móviles?

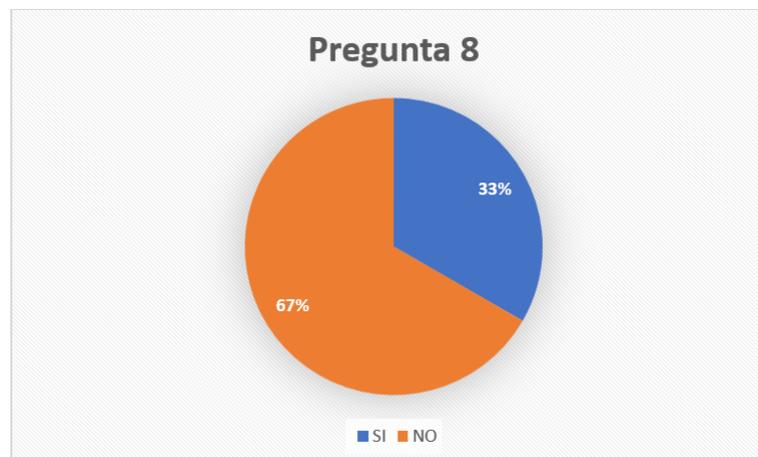


**Gráfico 25:** Resultado pregunta 7

**Interpretación:** Un 67% de los encuestados han utilizado sensores de un voltaje definido en prototipos de robots móviles, y un 33% no han utilizado este tipo de dispositivos.

**Análisis:** De acuerdo con los resultados de esta pregunta, se determina que la mayoría de los encuestados reconocen los sistemas de programación deben contar con sensores y actuadores que indiquen movimiento y reacción a obstáculos o señalización sónica de acuerdo a la actividad del robot que realice.

**Pregunta 8:** ¿Ha utilizado UD. sistemas de control remoto en tarjetas de control con sistemas embebidos para prototipos de robots móviles?



**Gráfico 26:** Resultado pregunta 8

**Interpretación:** Un 67% de los encuestados no han utilizado tecnologías de transmisión remota en el control de prototipos robóticos, y un 33% no tienen conocimientos de esta tecnología.

**Análisis:** De acuerdo con los resultados de esta pregunta, se recomienda que el sistema de control debería contar con un sistema de recepción y emisión de señales a través de un medio, como por ejemplo los módulos bluetooth, los cuales relacionan una aplicación programada en Android a ser instalada en un celular inteligente, en el control de un prototipo a desarrollar.

## 5.2. Resultados de las entrevistas a profesionales del ámbito académico

A continuación, se resume los resultados de las entrevistas realizadas a los señores profesores de la Universidad Técnica de Cotopaxi, cuerpo docente de la institución que tienen relación con las áreas de automatización, electrónica y programación de la Facultad de Ciencias de la Ingeniería y Aplicadas, registros que están añadidos en anexos de este proyecto, y de los cuales se ha sacado los siguientes resultados (Anexo 11: Evidencia de entrevistas realizadas):

Los profesionales encuestados enuncian que están de acuerdo en el uso de sistemas embebidos relacionado a la automatización de procesos que conllevan que los nuevos profesionales tengan conocimientos de cómo desarrollar este tipo de aplicaciones, ya que se observa que el uso de sistemas móviles que cumplen tareas específicas en las diferentes áreas de trabajo, están de acuerdo que los nuevos sistemas electrónicos se han convertido en un factor indispensable, por lo que se convierte en una necesidad que se debe incorporar dentro de la malla curricular para desarrollar las potencialidades en los ámbitos de la programación de software relacionados con la robótica.

Para lo que se debe tomar en cuenta que el desarrollo de proyectos de robótica si bien están relacionados con un sistema de programación necesitan específicamente un diseño relacionado con los movimientos que debe realizar un robot, y que se adapten a condiciones específicas. Por lo que se debe profundizar en realizar proyectos que incorporen sistemas embebidos en la construcción de robots programables, que cumplan actividades enfocadas en recibir órdenes a través de diseños de sistemas electrónicos en tarjetas programables.

Todo conlleva al encamina al desarrollo de la automatización que se debe realizar a través de investigaciones que logren fomentar la producción en serie, previo un estudio donde van a ser ubicados de acuerdo con las necesidades que tienen el país como una medida de abaratar costos y llegar a niveles de competitividad con otros países que han incrementado su uso en las diferentes actividades relacionadas con la industria.

Por lo tanto, es fundamental implementar asignaturas donde los estudiantes tengan mayor posibilidad de realizar aplicaciones para desarrollar su conocimiento y que realicen propuestas en procura de resolver diversas problemáticas que se presentan en el entorno de su profesión.

### **5.3. Validación de la propuesta**

Este capítulo integrara los siguientes planteamientos, mismos que se derivan de forma metódica del desarrollo de esta tesis hasta el momento:

- Validar el carácter práctico de la propuesta mediante el análisis de los componentes utilizados para que un sistema embebido funcione en una tarjeta electrónica proyectada y construida.
- Acrecentar el valor practico de la investigación mediante un ejemplo aplicado en la tarjeta programable de la idea contenida a lo largo de la tesis con el desarrollo de un

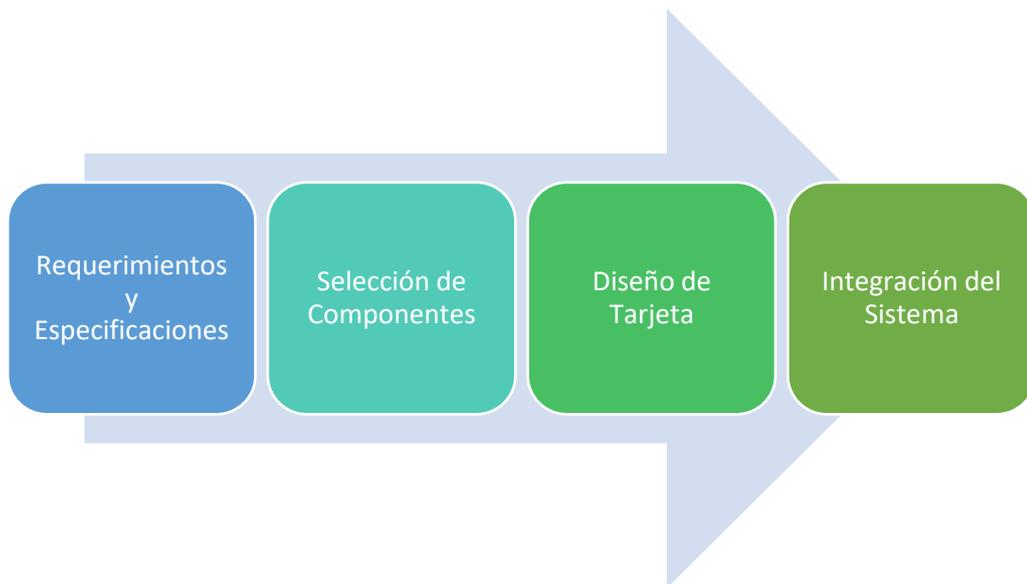
prototipo de robot móvil que utiliza sistemas embebidos, facilitando su entendimiento y estimación de su próximo alcance.

### 5.3.1. Etapas de desarrollo planteado

La disposición de una tarjeta electrónica que contiene un sistema embebido como componente principal, hace uso del enfoque de un hardware acoplado a un sistema de software dedicado, característica particular que da aspectos que difieren este sistema de otros sistemas electrónicos no programables, como:

- Son sistemas dedicados
- Son sistemas reactivos (sensores o medios mecánicos de funcionamiento)
- Prestan fiabilidad en el dimensionamiento del código diseñado

Enfocándose en estas características particulares detalladas anteriormente, es necesario desglosar este proyecto de electrónica programable que utiliza sistemas embebidos, en una evolución en etapas para sus diferentes niveles de abstracción, el cual proyectado en un diagrama de flujos puntualiza los siguientes compendios:



**Gráfico 27:** Etapas de desarrollo

### 5.4. Requerimientos y especificaciones

Los requisitos y especificaciones del sistema, tanto de hardware como de software, toman en cuenta la forma autónoma de la electrónica a utilizar (tarjeta a diseñar), así como del sistema donde se programará sus funcionalidades (IDE), planteamiento que, se ha diseñado en un

diagrama de bloques de la tarjeta proyectada, especificando sus funciones, para lo cual se ha utilizado software libre, conceptuando a continuación a Kicad, Arduino IDE, App Inventor 2, SolidWorks, Cura.

KiCad es una aplicación de software libre para diseño electrónico automatizado (EDA = Electronic Design Automation), es un proyecto inicializado en 1992 por Jean Pierre Charras, [34].

Arduino has endless applications as it has been used extensively for creating projects by hobbyist, amateurs and professional in various fields of engineering, [35].

App Inventor for Android (AIA) is a cloud-based visual development environment for the Android platform, which allows build Android apps using a web browser, [36].

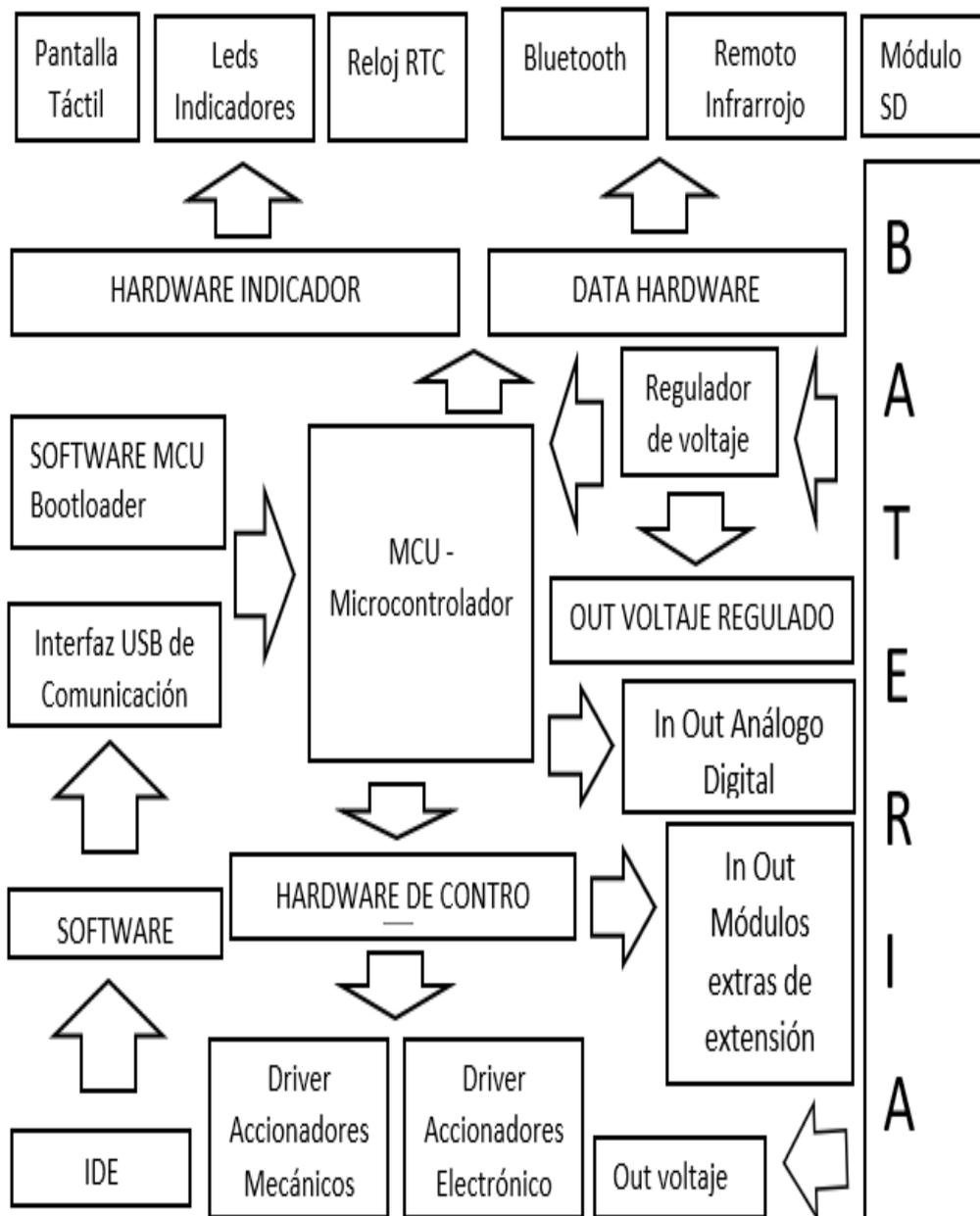
SolidWorks es un software de amplio uso en programas para el desarrollo de diseño de maquinaria. Al utilizar este software para el modelado de robots, se está ampliando el campo de aplicación de esta herramienta computacional sin aumentar el costo de los proyectos relacionados, [37].

Software de aplicación cura: Es un instrumento de software gratuito de código abierto para tecnologías de impresión 3D aplicado a la ingeniería que brinda un ambiente de desarrollo integrado (IDE), prepara el modelo 3D en instrucciones que su impresora 3D requiere para producir un objeto, disponible en todas las plataformas actuales, [38].

#### **5.4.1. Diseño por bloques tarjeta electrónica programable**

En el diseño de una tarjeta electrónica programable con base en un sistema embebido, se toman consideraciones que se abstraen del proyecto, los mismos que serán parte tanto del conjunto que dará sus funcionalidades, como del software necesario para que sea operable, el mismo que a continuación se lo planteara en unidades, a partir de la ampliación de la ilustración 4 del capítulo 1 de esta propuesta que consta de:

- Microcontrolador o sistema embebido.
- Fuente de Alimentación o batería.
- Reguladores de voltaje.
- Hardware indicadores.
- Hardware de adquisición de datos.



**Gráfico 28:** Diagrama de bloques tarjeta electrónica programable

### 5.5. Selección de componentes

En la explicación de un sistema que se esté proyectando, el análisis estructural del mismo converge en un estudio y selección de todos los diferentes materiales que lo conformaran, explicando el por qué y cómo han sido escogidos debido a sus características específicas (capítulo 1), y la funcionalidad que proporcionan, dilucidando la futura propuesta caracterizadas por ser eficaces y sencillas, con un grado de eficiencia que brinde relatividad productiva y económica, unida a un nivel de fiabilidad alta y de múltiple uso.

**Tabla 7:** Componentes electrónicos

ILUSTRACIÓN	NOMBRE	CARACTERÍSTICAS	FUNCIONES
	Microcontrolador Atmega32u4.	<ul style="list-style-type: none"> <li>- Voltaje: 2.7, 5.5 VDC.</li> <li>- Consumo: 27 mah</li> <li>- Encapsulado: TQFP.</li> <li>- Puertos: 14</li> <li>- Frecuencia de operación: 16 MHz.</li> <li>- Memoria Flash: 32 Kbytes.</li> <li>- Fabricante: Microchip.</li> </ul>	Microcontrolador CMOS (semiconductor complementario de óxido metálico) de 8 bits ( $2^8$ o 256 posiciones con un valor de cero o uno) de bajo consumo de potencia basado en la arquitectura RISC (conjunto de instrucciones reducidas), con USB integrado.
	Cristal oscilador de cuarzo.	<ul style="list-style-type: none"> <li>- Frecuencia de 16mhz.</li> <li>- Encapsulado HC-49S.</li> </ul>	Cristal oscilador de 16mhz de resonancia mecánica vibratoria piezoeléctrica, que produce una frecuencia en rangos de tiempo.
	Condensador cerámico de 22 picofaradios.	<ul style="list-style-type: none"> <li>- Tipo: Cerámico.</li> <li>- Capacidad: 22 pf.</li> <li>- Max voltaje: 50 VDC.</li> </ul>	Dispositivo pasivo, utilizado para almacenar energía sustentando un campo eléctrico.
	Batería LIPO.	<ul style="list-style-type: none"> <li>- Capacidad 1600mAh.</li> <li>- Voltaje 11.4 dc.</li> <li>- Numero celdas 3S.</li> </ul>	Baterías recargables ligeras de polímero de litio que almacenan una gran cantidad de energía, están compuestas por varias celdas de voltajes entre 3,7 y 4,2V.

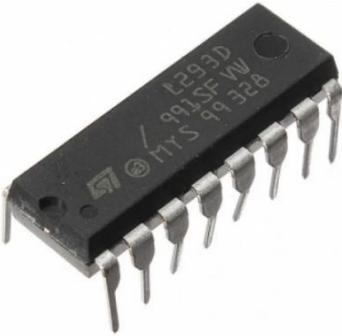
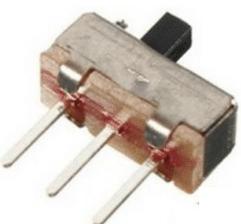
ILUSTRACIÓN	NOMBRE	CARACTERÍSTICAS	FUNCIONES
	Integrado L298n.	<ul style="list-style-type: none"> <li>- Driver Controlador de motores.</li> <li>- Voltajes 5-24 dc.</li> <li>- Corriente max 2 amp.</li> <li>- Pines de control digital o análogo.</li> </ul>	El L298n es un integrado de control de motores que proporciona la corriente necesaria para aplicaciones de robótica.
	Integrado AMS1117.	<ul style="list-style-type: none"> <li>- Regulador de voltaje de 5V desde 6,5VDC 12VDC.</li> <li>- Corriente de salida: 800mA.</li> </ul>	El AMS1117 es un componente electrónico que convierte una tensión de voltaje de ingreso a una tensión de 5 voltios, necesaria para los circuitos que alimentara.
	Led indicador.	<ul style="list-style-type: none"> <li>- Componente electrónico que funciona a un voltaje de 1.5 voltios.</li> </ul>	Los leds indicadores presentan una señal lumínica al recibir una tensión en su entrada.
	Mini Interruptor.	<ul style="list-style-type: none"> <li>- Componente mecánico de interrupción de alimentación de voltaje.</li> </ul>	Los interruptores son componentes eléctricos, que mecánicamente a través de una palanca, prenden o apagan un sistema.
	Puente rectificador de diodos.	<ul style="list-style-type: none"> <li>- Componente electrónico de rectificación hasta 200 voltios 2 amperios.</li> </ul>	Componente con 4 diodos que rectifican la corriente en su polaridad.
	Cloruro de hierro FeCl <sub>3</sub> .	<ul style="list-style-type: none"> <li>- Elemento químico de corrosión de metales.</li> </ul>	Compuesto químico que, disuelto en agua, corroe superficies de metal que no estén protegidas.

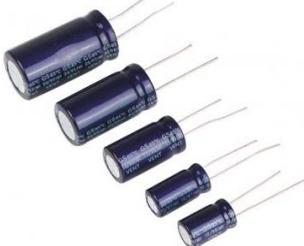
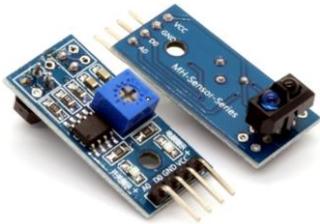
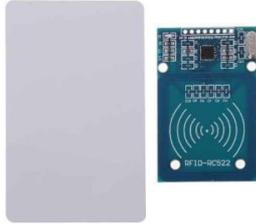
ILUSTRACIÓN	NOMBRE	CARACTERÍSTICAS	FUNCIONES
	Condensadores electrolíticos,	- Componente electrónico de filtración de corriente.	El condensador filtra los rizados de corriente para hacer la tensión DC lo más estable posible.
	Módulo lectura escritura micro sd. wg0009	- Componente electrónico que funciona a 3.3 voltios y 5v.	El módulo wg0009 lee una memoria sd, pudiendo recibir o enviar datos contenidos en la memoria sd.
	Módulo TCRT5000.	- Conjunto de componentes electrónicos infrarrojos que funciona a 3 voltios, provisto de un fotodiodo, y un fototransistor.	El módulo TCRT5000 es un conjunto electrónico emisor receptor de luz infrarroja, que emite una señal a ser leída por un sistema embebido.
	Módulo Rfid Rc522.	- Elemento que funciona a un voltaje de 5 voltios, de lectura.	El rfid es un identificador por radiofrecuencia para leer etiquetas a distancia de forma inalámbrica.
	Motor de engranaje.	Elemento electromecánico.	Un motor eléctrico dc usa el magnetismo producido por una corriente eléctrica para girar un rotor que está conectado a un eje con piñones.
	Rueda de caucho.	- Elemento mecánico.	Elemento de movimiento redondo de fijación con el motor para desplazamiento.

ILUSTRACIÓN	NOMBRE	CARACTERÍSTICAS	FUNCIONES
	Baquelita, o elemento de cobre plástico.	- Elemento conector sólido de componentes electrónicos.	Es un elemento plástico que posee un film de cobre, el cual se lo diseña y corroe pistas, para dibujar y colocar elementos interconectados.
	Rueda de garrucha.	- Elemento mecánico.	Es una rueda que se direcciona según la orientación que la fuerza de empuje le apliqué.

### 5.5.1. Diseño de la Tarjeta

El diseño de una tarjeta electrónica programable con un componente activo o sistema embebido, se lo realizará proyectando en un software electrónico Cad, haciendo de una descripción esquemática un archivo factible para la construcción de esta tarjeta, que cuando se lo fabrique físicamente acoplara en un sistema de sujeción denominado baquelita a los componentes electrónicos que lo conformaran.

### 5.5.2. Diseño de circuitos impresos con KiCad

Un software de diseño electrónico tipo Cad permite el diseño y construcción de una placa de circuito impreso con la ayuda de información de cada componente utilizado, facilitando la adición y edición de componentes que se utilizaran automatizando las tareas de desarrollo.



**Gráfico 29:** Interfaz principal KiCad

El proceso de desarrollo empieza por esquematizar los componentes en KiCad (Anexo 2, 3, 4: Diagrama esquemático general), para ello en el botón menú se clikea new para crear un nuevo proyecto, el cual se guardará en una carpeta con el nombre del nuevo proyecto, después se ejecuta el archivo creado entrando a la dependencia bin, lo cual muestra la pantalla principal de proyectos.

En la dependencia Bin se escoge Eeschema (editor de esquemas), el cual entregará una hoja en blanco, a la cual, a través de las pestañas derechas de esta aplicación, dispondrá al usuario de botones de:

- Agregar un componente
- Agregar un conector de alimentación
- Agregar un cable
- Agregar una unión

Dado que se está creando un nuevo esquema, se empezará por pulsar en agregar un componente buscando según la referencia del mismo en el cuadro de dialogo Name, que acepta una letra inicial para desplegar opciones.

Luego de escoger el componente buscado, se ha de colocar el componente según el criterio de diseño, para moverlo o rotarlo se ha de colocar el indicador del ratón sobre el componente y presionar M para mover y R para rotarlo.

Para cambiar las propiedades del componente escogido se procede con un clic derecho sobre el mismo el cual despliega un menú contextual, donde se selecciona la opción apropiada al diseño, mismo que se guardara en el esquema de acuerdo a un número generado automáticamente, más el valor que tenga este componente.

Todos los componentes que tendrá la placa se han de seleccionar en esta ventana, ya que de aquí se partirá a la sección de trazado de líneas de conexión, los componentes utilizados, incluidos integrado principal(microcontrolador), swichts, etc.

Han de tener una alimentación y el voltaje correspondiente, para lo cual se ocupara el símbolo Flag, mismo que desplegara una ventana, a la cual tras presionar OK, muestra las coincidencias de alimentación guardadas en la biblioteca principal del sistema, registrando y guardando estos datos para futuros cambios según las necesidades del proyectista; para continuar con la alimentación, se ha de proveer al esquema de la conexión a negativo o aterrizado a tierra, el

cual se lo despliega del cuadro de componentes con Vss, conectándolo donde se necesite en el esquema.

Las Flags también se utilizan para conexiones con componentes a los cuales se realizará puentes, dando nombres a las Flags desde el origen hasta el destino.

Para las líneas de conexión, se ha de presionar en el botón de líneas de conexión, haciendo las conexiones necesarias entre los componentes de la PCB que se está construyendo.

Para reafirmar conexiones y referencias de componentes al acabar el diagrama esquemático, se utiliza la barra de herramientas superior, presionando el botón Notación Esquemática, corrigiendo referencias o símbolos de precaución en el que los componentes utilizados no han sido referenciados, para que la sincronización del esquemático con el circuito impreso se realice.

### 5.5.3. Verificación esquemática de reglas eléctricas en la PCB

La verificación de las reglas eléctricas del esquemático se ejecuta presionando el botón ERC, el cual verifica conexiones en el esquemático, indicando puntos sin conexión y puntos de falla de haberlos.

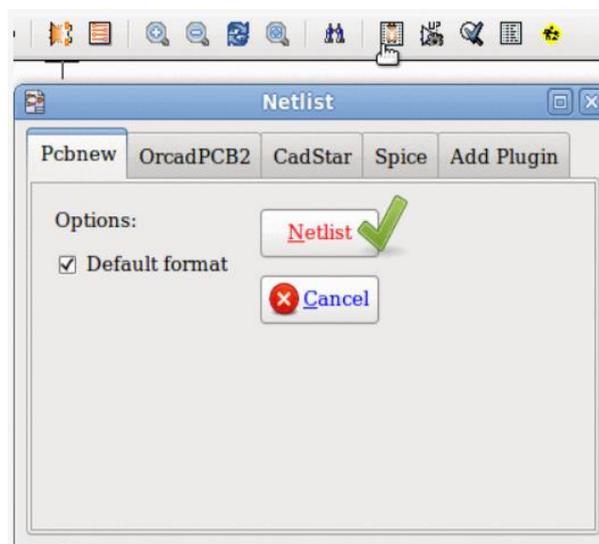
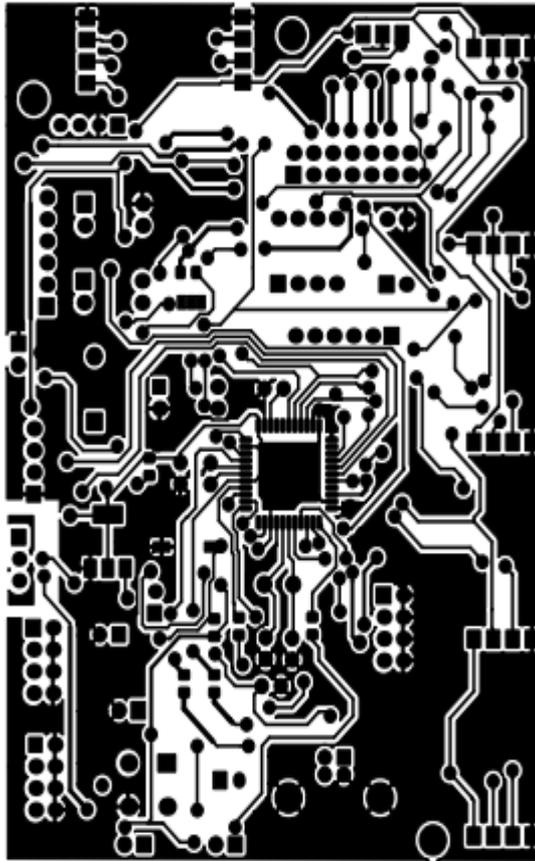


Gráfico 30: Interfaz de KiCad, Netlist

### 5.5.4. Generación de componentes Netlist

Este asistente genera la información de los componentes del esquemático, los que pasan a ser pistas y módulos de la PCB proyectada, esta información se guardara como una lista con el nombre del proyecto, con la extensión punto net.



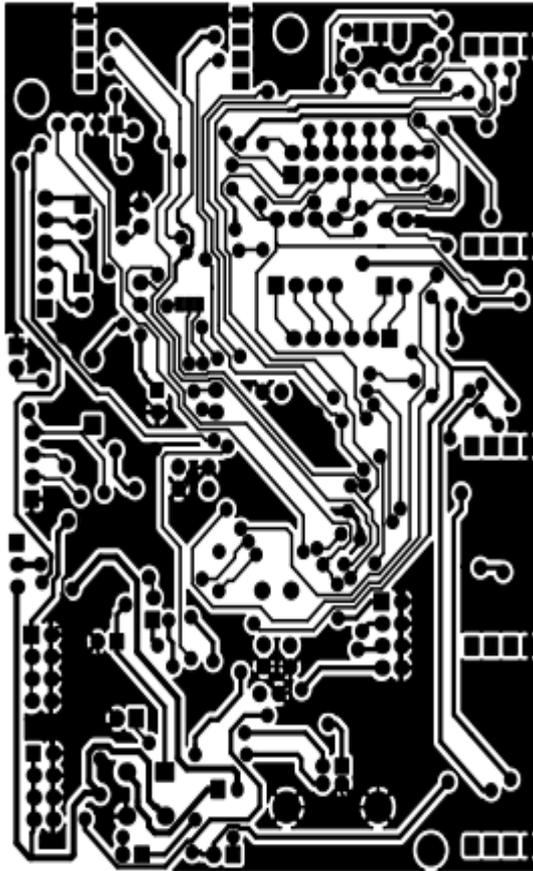
**Gráfico 31:** Frontal de impresión circuito PCB

#### **5.5.5. Traspaso del diagrama esquemático a una vista física lógica de una PCB**

El traspaso del diagrama esquemático a ser proyectado en una virtualización posible del PCB, se realiza con ayuda del programa KiCad, utilizando el módulo herramienta Cvpcb, que en la función de abstracción del proyecto a plasmarse, generara una vista física posible, haciendo que los componentes con diferentes valores, sean definidos con tamaños y disposición, solo basta con un doble clic sobre el componente para definirlo, por ejemplo, el caso del microprocesador principal en el cual está basado este proyecto, que puede ser DIL o de dos líneas de disposición de pines o SMD, para aparentar una disposición de pines tipo sello postal.

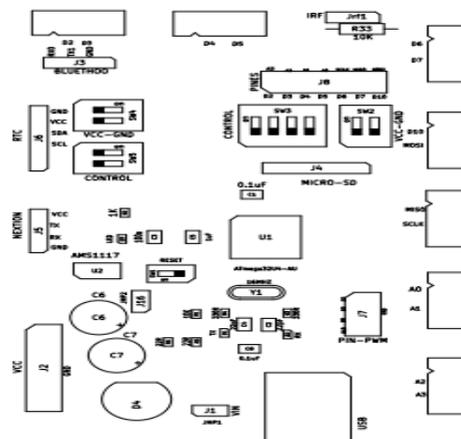
Después de la selección y definición de los componentes con el módulo Cvpcb, se utilizará el siguiente módulo herramienta de KiCad, que está en el panel de herramientas llamado Pcbnew, la cual desplegará una nueva ventana, en la que ofrecerá en su parte superior un conjunto de herramientas para leer la netlist, verificar el diseño, hacer un autorouting, etc.

Se procede a presionar sobre leer la netlist, desplegando el programa Cad todos los componentes apilados, para su despliegue por parte del proyectista.



**Gráfico 32:** Anverso de impresión circuito PCB

Para establecer el trabajo que se dispondrá en la baquelita, se dispondrá de dos posibilidades, una de una cara de cobre, y la otra versión de dos caras de cobre, en el caso de este proyecto, se ha procedido a escoger una PCB de dos caras, dado la cantidad de componentes y conexiones a realizar, el cual se lo define en menú, preferencias, general



**Gráfico 33:** Disposición de elementos del circuito impreso

A continuación, utilizando las facilidades de KiCad que ofrece con una herramienta adicional denominada Autoplacement, se procede a optimizar el diseño al generar automáticamente una posición de los dispositivos, generando adicionalmente las pistas de conexión, para lo que pide al usuario definir el tamaño la placa de circuito, la cual es específica en la pestaña combo box, en el ítem Copper, EdgePcb, después de escogida esta opción, se utiliza la herramienta regla, ubicada en la parte derecha de la pantalla, la cual ayuda a visualizar medidas de la PCB, una vez realizado esta tarea, en la barra superior, escogemos Autoplacement, ubicando sobre cualquier parte de la PCB el señalador del mouse del PC, para con el botón derecho del ratón, elegir la opción Autoplace All modules, ubicando en la parte escogida con medidas, todos los componentes del esquema.

Para la definitiva posición de los componentes, se los moverá y ubicará a conveniencia, observando las referencias del mismo para que se puedan leer correctamente los valores.

En el trazado, se ha de ocupar nuevamente si es necesario el Autorouting, habilitando esta opción desde la barra de herramientas, y utilizando el mouse del equipo, con el clic derecho, para escoger la función Autorute All modules, con lo que el ruteado de cada nodo en la lista de pistas y pines, funcionalmente hará secuencias de conexión en las capas disponibles, para después ir corrigiendo manualmente disposiciones a conveniencia del proyecto.



**Gráfico 34:** Traspaso térmico de circuito impreso a baquelita

### **5.5.5.1. Transferencia, revelado y atacado con ácido de circuitos impresos**

Las técnicas que se emplean en este proyecto son métodos para crear placas de circuitos impresos, los cuales ocupan medios físicos en su realización como métodos de termo transferencia y atacado con ácido percloruro férrico, para una posterior aplicación de máscara anti soldante.

#### **5.5.5.1.2. Termo transferencia de un circuito impreso**

La preparación del material necesario para este método comienza por el corte del material aislante o baquelita, que tendrá las medidas resultantes del diseño realizado en un software Cad electrónico, limando las asperezas que pueden quedar en los bordes de la placa y que son producidos por las herramientas de corte, después se limpiara con lana de acero, flotando suavemente el cobre de la baquelita, para posteriormente ser lavada la placa con un solvente como el alcohol isopropílico para quitar grasa que se adhieren al cobre con la manipulación manual que se hace en esta placa, de modo que el toner a ser aplicado posteriormente se adhiera mejor al cobre.

El paso subsecuente en el traspaso térmico del diseño empieza con la impresión del circuito desarrollado en el programa de diseño electrónico escogido, el cual proveerá un diagrama esquemático de la PCB, mismo que proyecta un circuito tipo Cad, con líneas de conexión entre los diferentes componentes, siendo este circuito el adecuado para ser transferido a un papel fotográfico del tipo brillante, para posteriormente ser impreso con un toner tipo polvo que ocupa las impresoras láser, el papel será cortado de acuerdo con las medidas adecuadas a la placa que se utilizara, para que el Cad del PCB pueda ser alojado sin inconvenientes en ella.

El siguiente paso es recortar el papel para colocarlo sobre la placa en la posición que guarde el centrando el diseño, aplicando cinta adhesiva en los bordes, que ayude a sujetar mejor los elementos.

El elemento para la transferencia térmica en estos procesos es por lo general una plancha con potencia ubicada entre 1000W y 1500W para resultados óptimos.

El tiempo de transferencia térmica por lo general está entre tres a cinco minutos, observando en el papel fotográfico comienza a denotar el toner de las pistas, indicando que se ha realizado correctamente el proceso.

Una vez transferido el toner al cobre de la baquelita, viene la etapa de retiro con agua del papel que contenía el toner del diseño.



**Gráfico 35:** Utilización de ácido de revelado cloruro férrico

El agua, se encargará de facilitar la acción de retirar el papel dejando el toner en el cobre, para lo cual se debe remojar la baquelita en el líquido, y esperar algunos minutos, observando que se comienza a transparentar el papel, apareciendo las líneas del circuito impreso, el tiempo considerado en este proceso es de tres minutos a cinco.

Una vez retirado los restos de papel y el toner está transferido en el cobre con el diseño cad por completo, llega el momento del siguiente paso que es el comenzar a trabajar con el ácido de revelado para PCB's.

#### **5.5.5.1.3. Revelado y atacado con ácido en circuitos impresos**

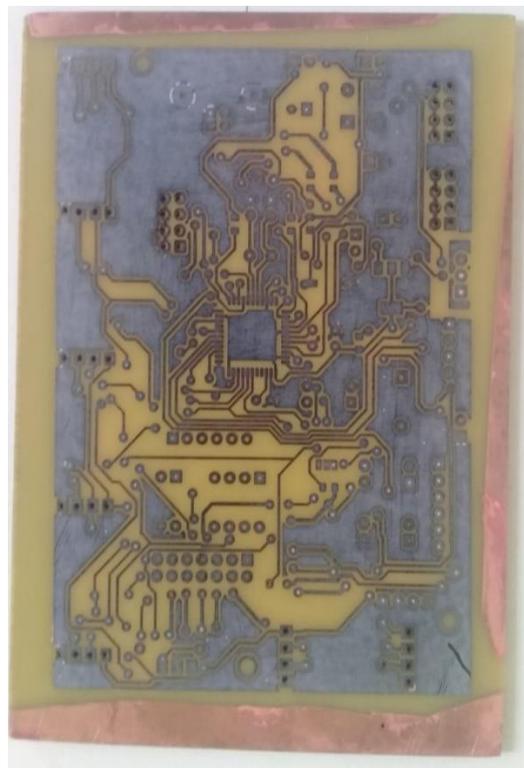
En el revelado de la placa PCB se ocupará percloruro férrico, un químico que ataca al cobre, dejando la parte protegida por el toner intacto.

La preparación de este ácido comienza por diluirlo en agua caliente en una proporción de dos partes de agua por una de ácido, el cual viene en forma de polvo, siendo preciso diluir el ácido lo más fuerte posible para disminuir el tiempo que demorará en completarse el proceso y lograr un ataque al metal de la placa más controlado.

La utilización de este elemento debe ser manejado con mucho cuidado, manipulando el mismo con precaución para evitar accidentes; para la utilización de este revelador, se ha de proveerse de un recipiente de plástico de acuerdo al porte de la baquelita, y el líquido ha de cubrir por entero el elemento; el ácido ataca el metal descubierto de la baquelita.

En el proceso de revelado con el ácido percloruro férrico varia en el tiempo que necesita, el cual varía entre 10 a 20 minutos, siendo necesario oxigenar la placa con continuo movimiento del recipiente, dejando a ratos la placa al descubierto.

Este método exige la observación por parte del proyectista, dado que constantemente se ha de revisar el recipiente en el que esta la placa con el ácido, hasta que la placa de baquelita que se ha tratado, presenta la eliminación total de las zonas de cobre que no fueron tratadas con el toner transferido térmicamente. El proceso inmediato será sacar el PCB, limpiando la placa con cuidado, aplicando agua para dejar sin restos de ácido la placa, para por último aplicar bicarbonato de sodio, que bloquea en forma definitiva la acción corrosiva del ácido sobre el cobre.



**Gráfico 36:** Baquelita revelada

Una vez que la PCB está seca, se procede a retirar el toner de la placa, el cual se lo realiza con un solvente fuerte como el thinner, dejando limpia la placa, para el siguiente proceso que es el de taladrar los agujeros en donde los componentes se colocaran y soldaran para finalizar el trabajo en la PCB.

El proceso de agujereado en la PCB, se lo realizara con un taladro en el que se colocara una broca de 1/32 pulgadas, los agujeros resultantes que se realizarán en los pads de soldadura

diseñados en el software Cad electrónico utilizado, servirán para colocar los terminales de los componentes.

La placa terminada y lista para montar los componentes de nuestro desarrollo, deberá pasar un proceso mas que es la aplicación de la capa anti soldante.



**Gráfico 37:** Aplicación de mascara anti soldante

#### **5.5.5.1.4. Aplicación de la máscara antisoldante**

La obtención de un PCB de buena calidad, en la que premie el aspecto profesional, no sólo dependerá de un buen software de diseño Cad electrónico, sino también de una prolija terminación con la aplicación de una máscara antisoldante (Solder Mask), en las caras de la PCB que poseen líneas de conexión de cobre.

Esta capa de protección y acabado para PCB's hace que la construcción de los PCB sea más confiable, dado que este elemento brinda una protección contra la corrosión, cortocircuitos entre líneas de conexión continuas en el proceso de soldadura donde el estaño no se propaga por todas partes y se limita a los pads de componentes donde la laca no se adhiere del cobre de la PCB.

El proceso de aplicación de la máscara antisoldante comienza por cubrir la PCB con esta pintura tipo laca brillante que viene en varios colores, en los que destaca el verde, que se le aplicado a este proyecto, siendo este paso fundamental, porque su aplicación se la realiza de manera uniforme en el lado de las vías de cobre, luego, se la cubre con celofán transparente del porte de la placa, que ha sido impreso con toner de polvo de una impresora láser con los pads de donde se soldaran los componentes electrónicos, el programa de diseño cad para PCB permite

imprimir sólo los pads y las zonas de soldadura, haciendo calzar donde se cubrirá para que no solidifique esta laca, dejando puntos de soldadura a posterior, una vez realizado este paso, se aplica luz durante al menos 10 minutos curando la pintura que queda expuesta a la radiación, finalmente, se retira el PCB y se lava con alcohol isopropílico que es un solvente suave, para retirar los restos de pintura dejando libres las partes que recibirán las soldaduras con estaño.

#### **5.5.5.1.5. Soldado de componentes en la placa impresa**

La soldadura de un circuito impreso, es el procedimiento para colocar en forma definitiva componentes sobre una placa de circuito impreso, para lo cual se debe limpiar las superficies de los elementos que se van a soldar, para posteriormente proceder con ayuda de la herramienta llamada caudín a fundir el metal denominado estaño que se calienta simultáneamente con la punta del soldador al circuito impreso con los elementos a soldar, depositando una pequeña cantidad de estaño, la cual será suficiente para cubrir las superficies a soldar, se debe esperar a que se enfríe el estaño sin que se muevan las superficies soldadas, procedimiento que presenta un aspecto en los llamados puntos, de brillante y cóncavo, con lo cual visualmente se comprueba que la soldadura está hecha correctamente, para un contacto permanente del circuito.



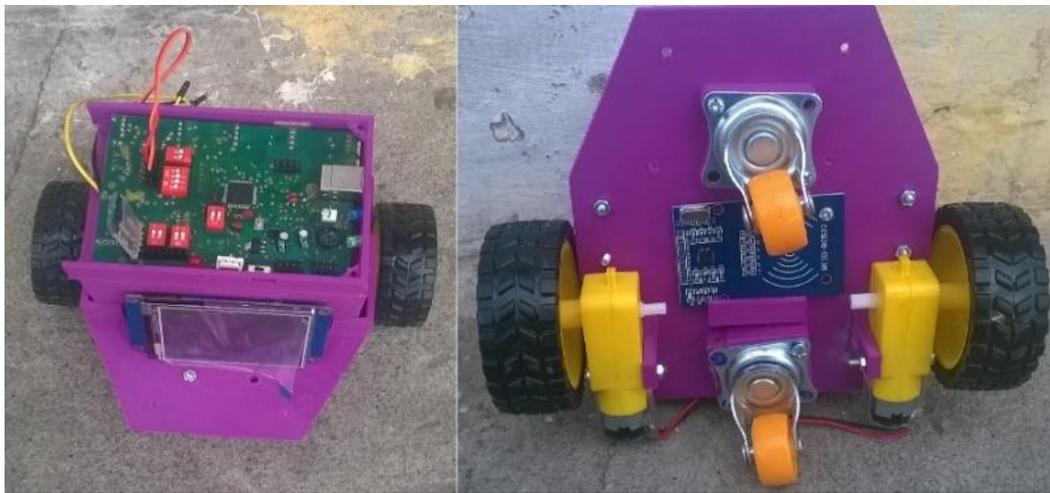
**Gráfico 38:** Tarjeta programable soldada

### **5.6. Diseño e impresión del prototipo de robot móvil**

Para la creación del prototipo que contiene la tarjeta programable con el sistema embebido Atmega32u4 se ha utilizado el software de diseño SolidWorks en la versión estudiantil.

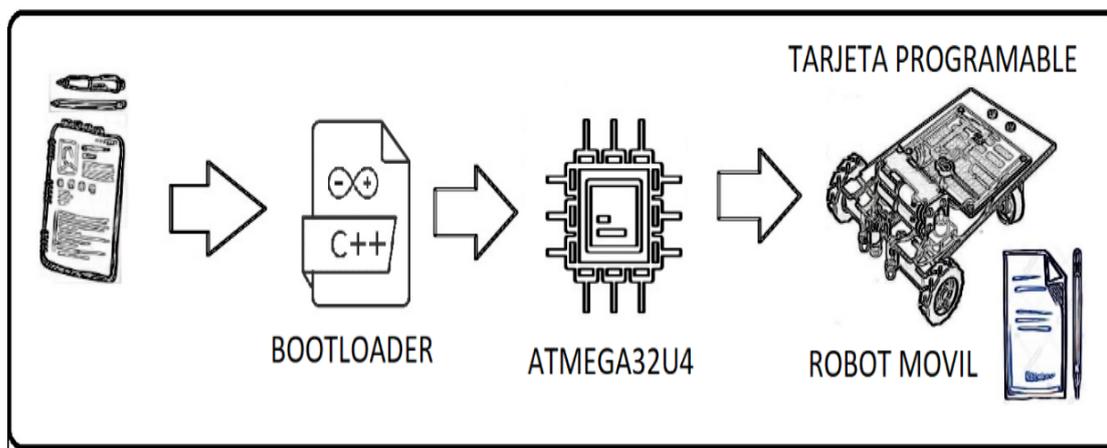
La descripción de este software hace referencia a un programa de diseño mecánico para elementos en 3D enfocado a un producto terminado, presentando en pantalla el diseño mecánico realizado.

El diseño en 3D contempla altura, ancho y profundidad (Anexo 5, 6, 7: Vistas de impresión cuerpo del robot), para crear un espacio en 3D o tres dimensiones, modelando una forma a objetos individuales que luego serán impresos con la ayuda de una impresora de elementos plásticos por fundición, que unidos a los elementos mecánicos, eléctricos y electrónicos harán del prototipo de robot móvil realizado, un conjunto viable para el programa de aplicación creado.



**Gráfico 39:** Modelo de robot impreso en 3d

El software de diseño SolidWorks, al igual que otros programa de diseño para impresión en 3D, hace uso de archivos en extensión stl (standard triangle language), que luego serán importados a el programa de posicionamiento del espacio de la impresora a utilizar denominado cura, software libre que calcula tiempo de impresión según el archivo, además de ir detallando el mismo archivo con el tipo de material a utilizar, que en este caso será pla (polímero biodegradable ácido poli láctico derivado del maíz), en este mismo archivo, se escogerá el tipo de relleno de material que por defecto viene en forma de panal de abeja a un porcentaje de distancia entre celdas del 20 por ciento, tomando en cuenta que la pieza vaya a ser mecanizada con perforaciones, se ha cambiado a un porcentaje de relleno del 50 por ciento incrementando resistencia en este parámetro que dará su estructura, también se definirá la temperatura de fundición del pla que es 105 grados, se añadirá también la temperatura de la base de la impresora, denominada cama caliente que es de 60 grados, para que la maquina impresora por fundición 3d, realice el proceso de fundición del prototipo a fabricar.



**Gráfico 40:** Proceso de integración Atepga32u4 con un computador personal

## 5.7. Integración tarjeta electrónica con IDE instalado en un computador

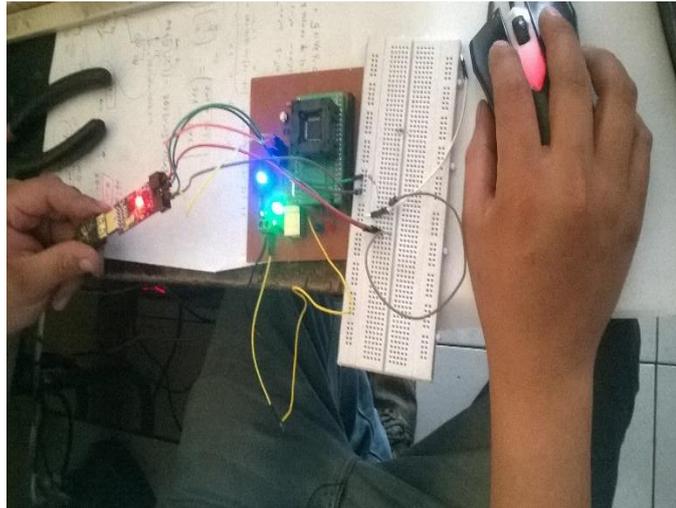
Para que un hardware desarrollado como el presente, pueda ser reconocido en un entorno de un sistema operativo de un computador, este debe tener una integración con el sistema donde operara, teniendo en cuenta que la integración de los componentes activos como el sistema embebido Atmega32u4, y el computador, presentara una validación en forma física cuando se conecten a través del USB provisto en los dos dispositivos.

Para que esta conexión se presente, como se ha explicado anteriormente en los capítulos posteriores, se instalara un bootloader en el integrado Atepga32u4, que provee la herramienta libre de software de Arduino, que lo integrara al entorno del sistema operativo con el IDE, donde el usuario podrá desarrollar aplicaciones.

### 5.7.1. Programador grabador de bootloader por ISP del Atepga32u4

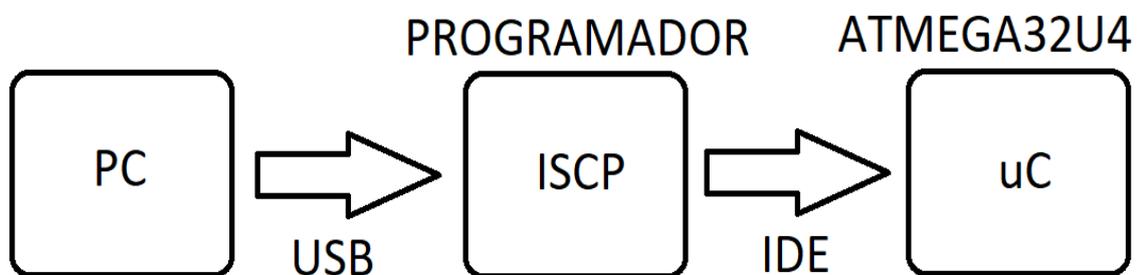
Una de las tareas importantes en que se emplea las herramientas de software Arduino, es la de convertir una placa de cualquier familia Atepga, que esté conectada por el puerto USB e instalado con el IDE del software Arduino, en un grabador de bootloader (Anexo 1: Arduino ISCP).

Detallando al bootloader o gestor de arranque, se ha de explicarlo como básicamente un archivo .hex, firmware que ocupa 0.5 kb de los 32 kb disponibles en memoria, que realiza dos funciones, primero espera en segundos a ver si la computadora está tratando de programarlo, receptando el programa de la computadora y cargándolo en la memoria (en una ubicación específica para no sobrescribir el gestor de arranque), apagando y volviendo a encender el integrado para que el gestor de arranque pueda comenzar a ejecutarse nuevamente con el nuevo programa grabado.



**Gráfico 41:** Grabación del bootloader

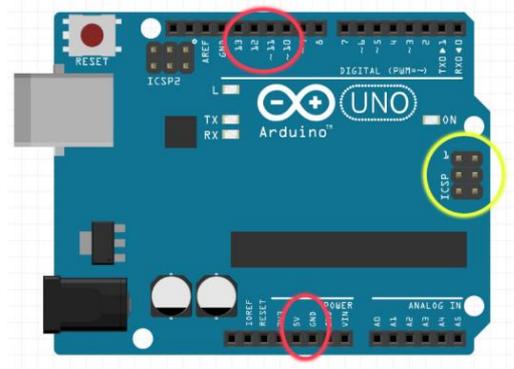
Como segunda función, de no intentar cargar un código, entiende como fuente de energía al USB conectado al computador, ejecutando el código que ya está almacenado en la memoria función que lo realizara mientras la placa tenga energía, de allí la importancia de utilizar un bootloader cargado en un integrado, que, además de lo señalado anteriormente, permite eliminar la necesidad de hardware adicional como un programador externo en cada grabado de un nuevo firmware que el usuario vaya a ejecutar.



**Gráfico 42:** Programación Atemga32u4

Como indica el gráfico anterior, un Arduino uno provisto de conexiones a los pines ICSP del integrado a grabar hará que el programa de inicio o bootloader, puede hacer que un hardware de diseño propio y construido como el presente proyecto, tenga reconocimiento en una computadora personal, para interconectar y ser un grabador en la memoria del microcontrolador proyectos que el usuario desee, para lo que en la preparación de cargado del bootloader se toma en cuenta dos pasos, la conexión física del proceso de grabado en la memoria del integrado, y la configuración del software.

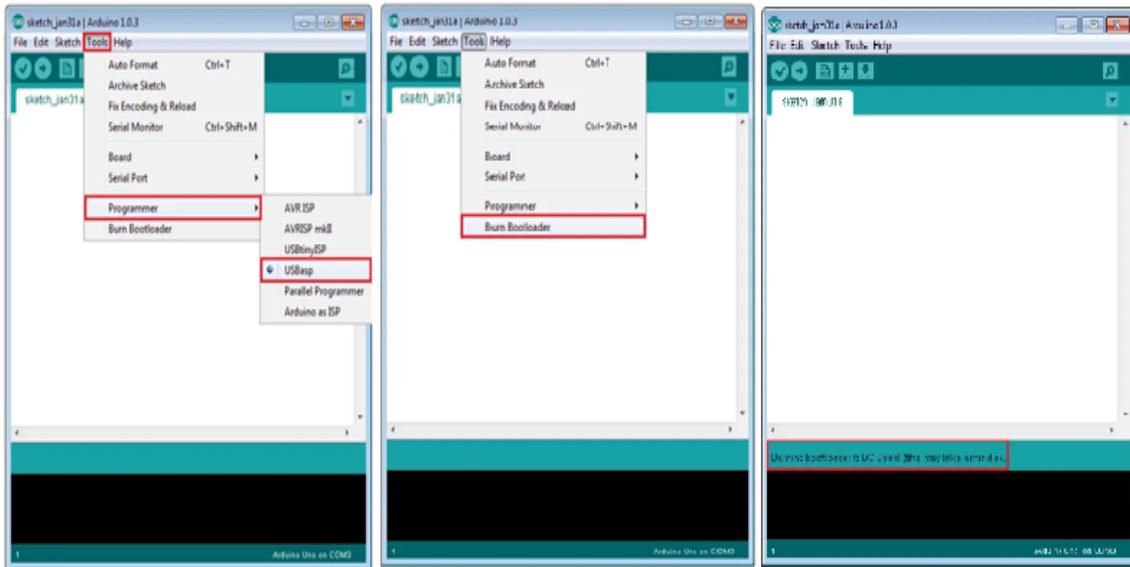
Las interconexiones físicas que se utiliza en el arduino uno son 6, las cuales son: pin VCC o ICSP 2, GND o ICSP 6, MISO pin 12 o ICSP 1, MOSI pin 11 o ICSP 4, SCK pin 13 o ICSP 3, RESET o ICSP 5, que se interconectan entre el micro de programación (arduino uno) y el micro objetivo (BRICK atmega32u4).



**Gráfico 43:** ISCP programador Atmega32u4

Para la configuración del software, se ha de insertar en el pórtico USB el Arduino uno, reconfigurando en la pestaña herramientas el puerto serie que maneja la placa que se está utilizando como programador, ya reconocido el dispositivo, se lo ha de configurar como programador, el proceso empieza en la pestaña de menú, ejemplos, Arduino ISP, Arduino ISP, el IDE al recibir la orden anterior procederá a abrir una nueva ventana con el firmware que convierte al Arduino uno en un programador. posteriormente, se ha de cargar el firmware bootloader en el Arduino uno apretando el botón de subir o upload que graba las tareas y el programa en sí que será trasladado al integrado de la placa que se esté instalando.

Teniendo listo los dos parámetros anteriores se procederá a escoger en herramientas o tools a él integrado de la familia Atmega a utilizar, en este caso es el Atmega32u4, que mediante un archivo específico disponible para este sistema embebido, creará la compatibilidad de este integrado (Atmega32u4) con el software Arduino, para posteriormente poder utilizar el IDE de esta herramienta, procedimiento que es el último paso, y se selecciona desde Tools/Burn Bootloader, para con esta acción, ir grabando el bootloader en el integrado Atmega32u4; en esta instancia, el procedimiento de programación verifica la firma del microcontrolador antes de realizar cualquier acción de escritura, si el procedimiento ha sido correcto, aparecerá un mensaje de confirmación en la parte inferior de la interfaz del software Arduino IDE.

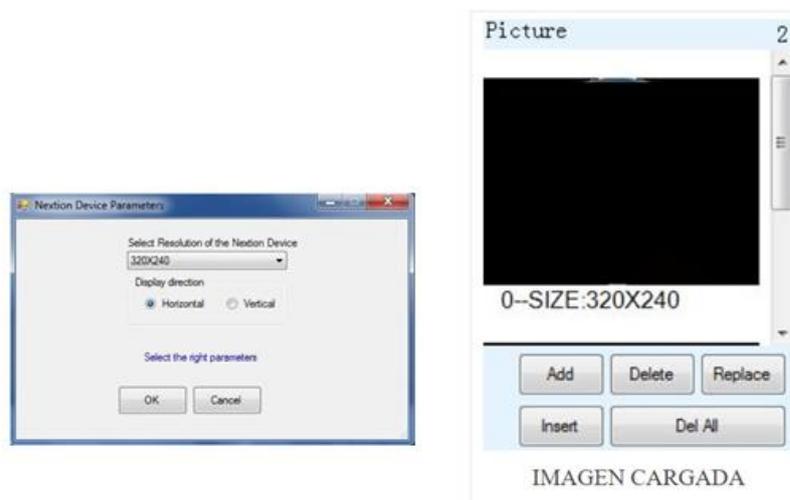


**Gráfico 44:** Ventanas de carga del bootloader al integrado Atmega32u4 (IDE)

### 5.7.2. Programación pantalla táctil Nextion NX3224T028

En este proyecto se ha utilizado una pantalla táctil programable nextion NX3224T028, una interfaz HMI (human machine interface) que proporciona control y visualización en un proceso sistematizado de fácil acceso cuando se utiliza sistemas embebidos como el Atmega32u4.

El paquete de esta interfaz lo conforma el hardware pantalla táctil TFT thin film transistor liquid crystal display o pantalla de cristal líquido transistor de película fina, y el software editor de diseño de interfaz, que da acceso a botones, cuadros de texto, barra de progreso, slider, panel de instrumentos, etc.



**Gráfico 45:** Ventanas de programación pantalla táctil

La pantalla utiliza un puerto serie para hacer comunicación según el usuario necesite, aprovechando protocolos propios de los sistemas embebidos como el protocolo UART, universal asynchronous receiver transmitter o transmisor receptor asíncrono universal.

Para programar este sistema se ha de acceder al programa nextion studio, creando un nuevo proyecto en file/new, para como primer paso configurar el tamaño, utilizando la pantalla de 320 x 240 en este proyecto, el display se lo utilizara en forma horizontal, marcando en el casillero del software esta configuración, como segundo paso, se ha de cargar las imágenes que se utilizaran para que aparezcan en la pantalla, accediendo en el panel de control, en el lado izquierdo inferior de la pantalla a picture, pulsando el botón de add o añadir, abriendo una nueva ventana donde se seleccionara las imágenes que sean necesarias, el tercer paso es la configuración de la pantalla, teniendo como recurso a un simulador que proporciona el programa en su parte central, aquí se puede de acuerdo a la imagen cargada, disponer funciones como botones o acciones complementarias según se necesite.

El cuarto paso es dar funciones, como acciones a los botones que se utilizaran para el direccionamiento, teniendo como base a imagen que se cargó anteriormente, para lo que en el lado izquierdo del programa se selecciona la opción necesaria como el botón que es necesario, apareciendo un recuadro gris, ubicando este recuadro en el área de la imagen en donde se accionará, activándose cuando se presione en el área designada.

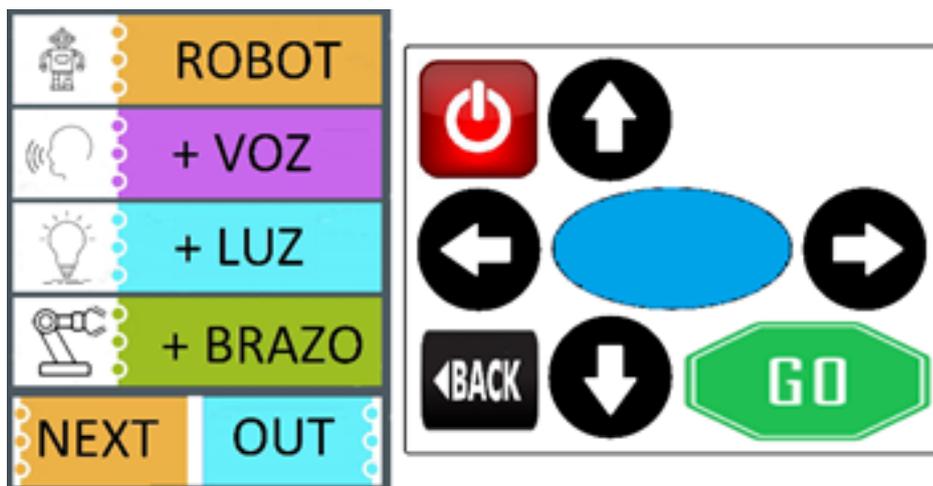
El quinto paso será la configuración de la imagen, para cuando se oprima el botón cambie de página indicando otra imagen, para lo cual se ingresa el valor de 1 en la pestaña picc, que es la imagen escogida actualmente, y que aparecerá como botón, después se selecciona picc2 que indicara una ruta a una nueva página con otra imagen, indicando cuando se presione el botón y que esta encendido, además de llevar la interfaz a otra página.

Sexto paso, se creará una nueva página, para lo cual el software proporciona del lado derecho la instancia page, la cual al presionarla con un clic derecho generará opciones, escogiendo el crear una nueva página, en esta página nuevamente se repite los pasos tres, cuatro y cinco.

Séptimo paso, programación de las funciones de los botones, para lo cual se procede a situarse en la parte central inferior, en donde se escoge la pestaña llamada touch release event, marcando el checklist user code con verificado, escribiendo después en la sección en blanco, que cuando presione el botón, será desde page 1, después se selecciona el botón de la otra página creada y también se escribe a donde se dirigirá, en este caso page 0, en ambos activar se debe activar el

checlist send component id, para que el sistema embebido Atmega32u4 reciba el dato por el puerto serial, y conjuntamente con el programa desarrollado en el IDE de arduino, realice las acciones indicadas.

Esta pantalla en su software ofrece un simulador de cómo se observará la interfaz creada, con los datos que mandaría por el puerto serial, para lo que se utiliza los botones de la barra superior compile, que indicará si hay algún error en las configuraciones y debug para inicio de la simulación.

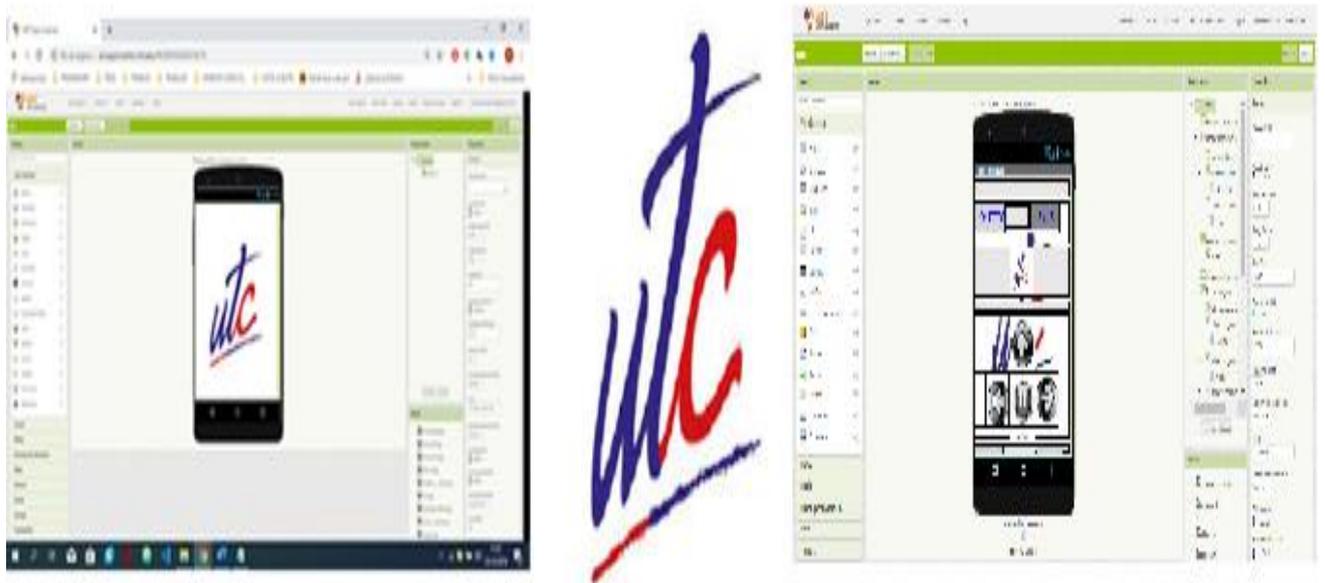


**Gráfico 46:** Gráfico menú de programación pantalla táctil

Octavo paso, grabado del programa en la pantalla táctil, para lo cual se utiliza una tarjeta microSD conectada a la computadora, y en el programa en la pestaña file/Open/build folder, se localizara el archivo .tft, copiando y pegando el archivo a la tarjeta microSD, la cual necesita tener el formato FAT32 (file allocation table), desconectada la pantalla, se inserta la tarjeta microSD para después alimentar con voltaje la pantalla, reconociendo de inmediato la pantalla el hardware conectado, e indicando que el proceso de carga del programa está listo, con el mensaje de update sucsessed, se procederá después del mensaje a desconecta la pantalla y retirar la tarjeta microSD, para por último, alimentar nuevamente la pantalla y ejecutara el programa que se ha creado.

### 5.7.3. Desarrollo de una aplicación de control en un smartphone

Para la aplicación de control remoto de este proyecto, se utilizará el software libre online de la Universidad de Massachussets, app inventor 2, que permite construir diversas aplicaciones de control de un sistema embebido como el Atmega32u4 con una interfaz a medida.



**Gráfico 47:** Interfaz control bluetooth

El programa presenta dos ventanas de programación, una en la que se crea el diseño de la aplicación con elementos de un menú y la posterior, de integración de las funcionalidades de los elementos seleccionados para programar mediante bloques.

Para iniciar en la creación de la aplicación, se establecerá una sesión con una cuenta en la web del software, pulsando en la interfaz de mit app inventor 2, ingresando después a start new project, identificando con un nombre a la aplicación, con lo que se desplegara la subsecuente página, en la que se da acceso a las ventanas anteriormente señaladas, dónde se creara el diseño de la aplicación con botones, colores, menús, imágenes, etc, con la posibilidad de directamente controlar elementos nativos del celular como giroscopios, micrófono, altavoz y módulos Bluetooth, que como una aplicación para celular con android, presenta opciones de previsualización de como quedara la aplicación, con opciones de selección de propiedades de componente con el panel llamado Properties, dónde se pueden cambiar las características de los elementos seleccionados.

En el despliegue de elementos de esta aplicación que se construirá se utilizara el elemento List Picker, que es menú desplegable, que se programara para que muestre los dispositivos Bluetooth cercanos y se conecte a uno, integrándolo a la pantalla interfaz principal de control, cambiando en properties, su identificador desde el campo text con su id dispositivos.

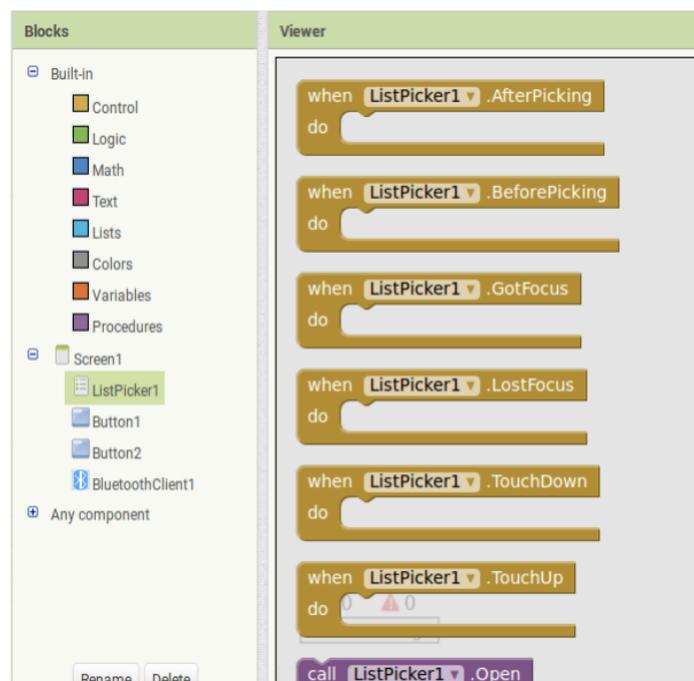
El siguiente paso será utilizar desde el menú dos elementos de tipo botón, para encender y apagar la conectividad entre el smartphome con android y el sistema embebido.

Continuando el proceso, se escogerá del menú un tipo de componente llamado no visible, que sirven para activar funcionalidades como el bluetooth, que conectará el Atmega32u4, con el smartphome, para lo cual el celular, a través de la interfaz de la aplicación desarrollada, acezará la antena bluetooth con el sistema embebido, utilizando en el proceso, al módulo HC-05 como enlace entre los dispositivos.

Para la realización del proceso anterior, se procede en el menú del software a escoger en connectivity al elemento bluetooth client, integrándolo a la pantalla.

Complementando la interfaz que se utilizara en este proyecto, y dado que se ha planteado demostrativamente una situación direccional de posicionamiento acoplado a sensores en una cuadrícula que contiene múltiples metas, estas en sus direcciones de despliegue necesita de botones, los cuales se les denominara arriba, abajo, izquierda y derecha, con los que el prototipo de robot móvil, comandado por el sistema embebido Atemga32u4, recorrerá metas que el usuario planifique, como la de llegar a una letra desde una posición no definida, pudiendo incluir estas órdenes directamente desde esta aplicación diseñada para el celular o desde una pantalla táctil integrada en el conjunto de hardware programable diseñado en esta propuesta.

En la programación de los anteriores elementos, se ha de configurar en las ventanas del software los métodos design con blocks, para que enlacen funcionalidades de la programación del sistema, que es propio del software utilizado, para lo cual se ingresa a la sección blocks.



**Gráfico 48:** Recurso de programación

En blocks, dado que la conexión es la función principal de esta aplicación, se programará esta en dos funcionalidades, la primera que muestre un listado de dispositivos, y la posterior que, escogido el dispositivo, se conecte.

### 5.7.3.1. ListPicker de listado de dispositivos disponibles

El ListPicker o menú desplegable, se ha de escoger del menú, señalando en las opciones que se encuentran a la derecha, abriendo las acciones que, con este objeto, la aplicación desarrollada, señalara la funcionalidad que realizara antes y después de oprimir su representación en la interfaz, por lo que cuando se presione el ListPicker. este inicializará buscando y desplegando una lista de dispositivos bluetooth disponibles en su rango de alcance del módulo hc-05, que es de 5 a 10 metros, para que el dispositivo móvil con comunicación bluetooth activada, y una lista de dispositivos disponibles, pueda conectarse al que el usuario escoja de la lista en pantalla.

En el menú de bloques de ListPicker1 se escogerá al bloque BeforePicking, que se lo asociará con la parte del menú denominado control, seleccionando la condicional if, que

después de seleccionar en el menú al objeto BluetoothClient1, se añadirá el bloque BluetoothClient1 Available, también se sacará de este menú de opciones de este dispositivo, al bloque BluetoothClient1AddressesAndNames a utilizarlo después de que en la pestaña de la condicional if, se incruste el bloque Set ListPicker Elements, creando parte lógica de la búsqueda.

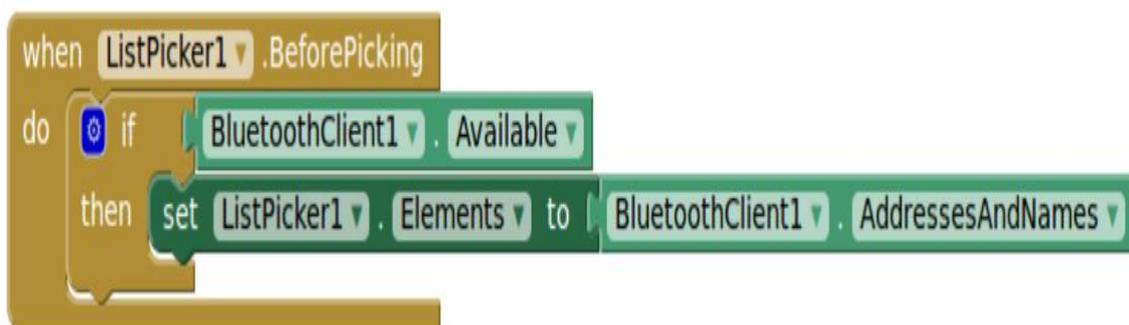
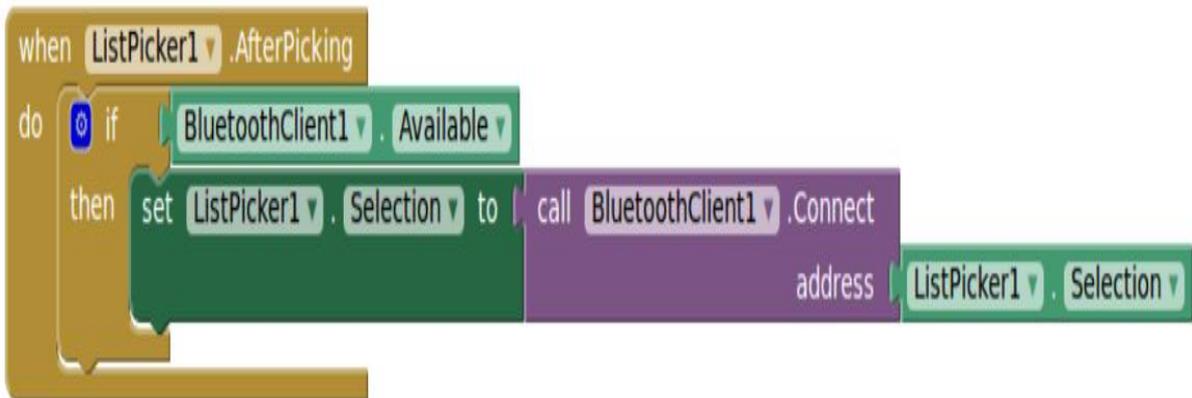


Gráfico 49: ListPicker

### 5.7.3.2. Conexión del ListPicker con el dispositivo escogido

Utilizando nuevamente el menú de bloques de ListPicker1 se escogerá al bloque AfterPicking, asociándolo reiteradamente con la parte del menú denominado control, seleccionando la condicional if, que requerirá de un nuevo bloque del objeto BluetoothClient1, específicamente

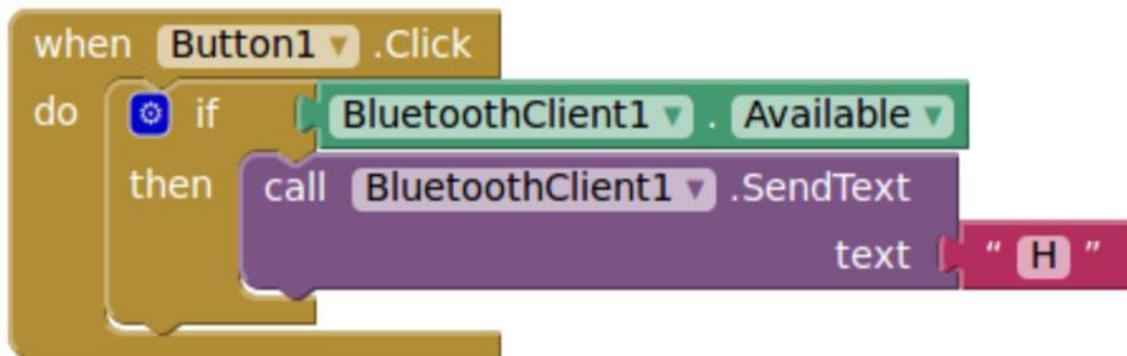
el bloque BluetoothClient1 Available, para la conclusión de esta parte del programa, se requerirá del objeto ListPicker1, el bloque set ListPicker1 Selection to, unido a Call BluetoothClient1 Connect, que también se unirá a un bloque del ListPicker1 Selection como ultima instrucción, con lo cual el Atmega32u4, a través del módulo HC-05 podrá utilizar la conexión bluetooth del smarthphone con el prototipo de robot móvil que se ha desarrollado.



**Gráfico 50:** Conexión ListPicker

### 5.7.3.3. Programación de funciones en los botones de la interfaz realizada

Aunque se realizado la conexión desde la aplicación, también se ha de realizar la programación de los botones que contendrá la interfaz.



**Gráfico 51:** Programación botón interfaz

Para lo cual se escogerá del objeto Button la funcionalidad When do Click, asociándolo con el objeto control condicional if, que nuevamente se lo asociara con los bloques que se sacara del objeto BluetoothClient1 las funcionalidades BluetoothClient1 Available y call BluetoothClient1 SendText, que hará la función de enviar una letra al módulo HC-05 conectado al Atmega32u4, que recibirá esta señal, ejecutando un programa según la letra recibida, por

último se sacara la función text, del menú TEXT, con lo que a este último módulo se le pondrá una letra, que identificara al botón utilizado.

#### 5.7.3.4. Exportación y grabado del apk en el smartphone

El software libre en línea app inventor 2 presenta opciones de exportación del software creado, el cual puede ser exportado en el formato de archivo .apk, para ser instalado en el smartphone que tiene el sistema operativo Android.

Los pasos para realizar este procedimiento comienzan por después de concluido la programación de la aplicación, escoger y presionar la pestaña build, en la que se coloca el puntero del mouse sobre la opción apk, que reconoce Android.



**Gráfico 52:** Exportación de aplicación desde app inventor 2

Después de haber escogido la opción de archivo .apk, se presiona en guardar en el computador, para lo cual el programa empieza una función de compilar y empaquetar el archivo, para descargar y poder a través de un cable, conectar a un smartphone al computador personal y copiar el archivo fuente de esta operación. Como último paso, el archivo copiado al celular se lo escogerá en el sistema del celular, dándole doble clic instalando esta aplicación, dejando listo el sistema para ser utilizado.

#### 5.7.4. Software, programación e IDE

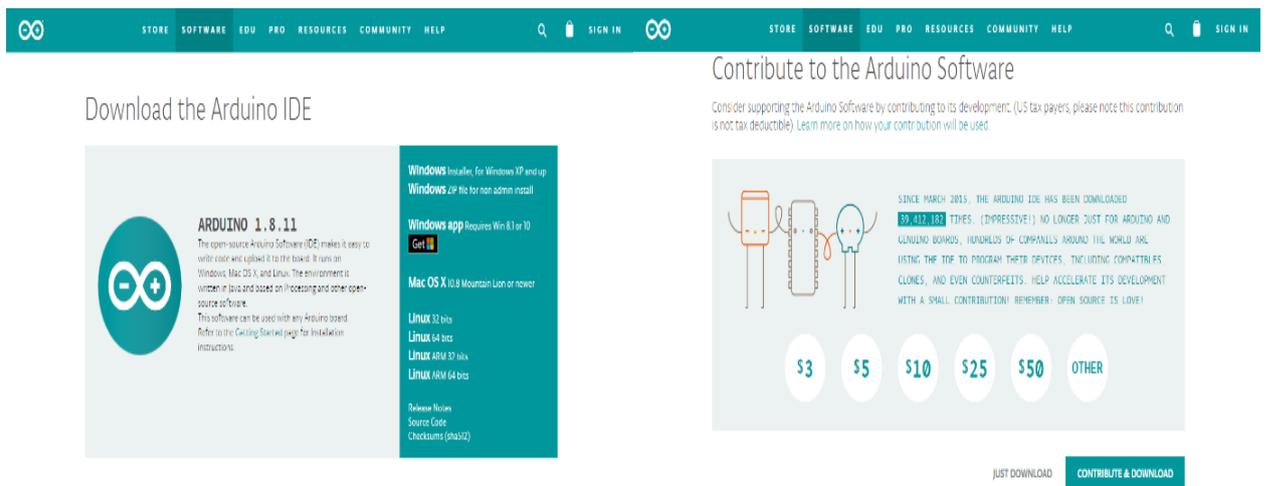
Para programar un sistema embebido, este sistema utiliza aplicaciones como el IDE o entorno de desarrollo integrado, disponible en <https://www.arduino.cc/en/Main/Donate>, el que se

procederá a descargar e instalar en el sistema que se utilizará, este software será manipulado para programar y grabar la tarjeta programable desarrollada en este proyecto de investigación denominado BRICK.

Describiendo el IDE, en concepción se observa a una aplicación multiplataforma desarrollado para la programación, la cual contiene:

- Editor de código.
- Compilador.
- Grabador de programa.
- Bibliotecas de código (colecciones de funciones).

Utilizando los elementos anteriores, el IDE igual que los entornos de programación de sistemas embebidos, a los programas realizados para estas minicomputadoras los denomina firmware o sketches, y están escritos en lenguaje C ++ para microcontroladores, que, para esta propuesta, utilizara software libre y gratuito.



**Gráfico 53:** Pagina de descarga IDE

#### 5.7.4.1. Creación de un proyecto para sistemas embebidos

Una vez que el software ha sido descargado e instalado desde la página <https://www.arduino.cc/en/Main/Donate>, se procede a crear un proyecto dándole clic en la pestaña archivo, y escogiendo la ubicación, en la cual se creará una carpeta, y dentro de la carpeta se hará clic en nuevo, con lo cual, al crear un proyecto, se generan automáticamente dos funciones, setup y loop, estructuras ubicadas en la pantalla de la aplicación a las cual se le programara los algoritmos que realizara el sistema embebido.

### 5.7.4.2. Estructura del programa de aplicación para el Atmega32u4

La estructura básica que utiliza el IDE de programación de Arduino se compone de diferentes partes necesarias, o funciones que encierran en bloques las declaraciones con las instrucciones a realizar, las cuales son:

**Encabezado** // Zona de declaración de librerías, variables a utilizar

**void setup()** { // instrucciones; aquí se realiza la configuración inicial de variables, se ejecuta sólo una vez cada vez que se inicia el programa, definiendo la función de cada pin del sistema embebido (entrada o salida), siendo la parte del código que se quiera ejecutar una sola vez // }

**void loop()** { //instrucciones; aquí se aplica el código que se va a repetir de forma indefinida, el loop es la función principal dentro del sketch que una vez ejecutada la función setup, la función loop se ejecutará y repetirá indefinidamente según el algoritmo// }



**Gráfico 54:** Pantalla del IDE en blanco para programación

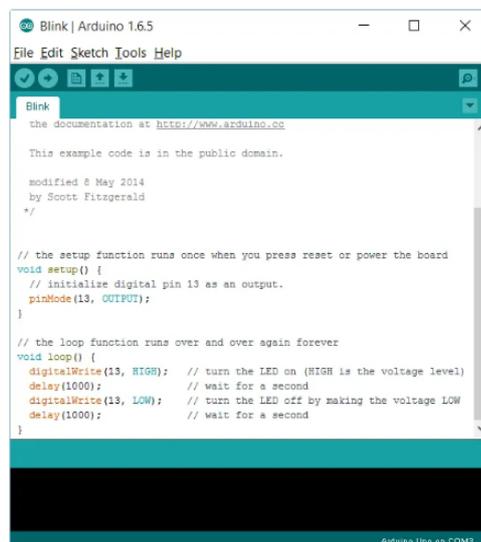
### 5.7.4.3. Programa de verificación blink

Para verificar la tarjeta programable construida, dado que este prototipo conjuga electrónica y programación, se utilizará al igual que con un lenguaje de programación, un despliegue de un mensaje similar al hola mundo, pero con la utilización de la electrónica de la tarjeta construida, que será el equivalente a encender un led de manera intermitente, con lo cual se comprobará la electrónica de su funcionamiento, los pines de salida y sus conexiones.

### 5.7.4.2. Grabado del programa de verificación blink

Para el grabado del programa (sketch), y habiendo realizado el procedimiento anterior de carga de bootloader, físicamente la placa en su diseño se le ha añadido una conexión a través de un cable USB, esta conexión permite subir programas a la placa, así como proporcionar alimentación de voltaje.

Para grabar el sketch, es necesario familiarizarse con el software libre utilizado en esta propuesta, el cual es intuitivo, ubicando todas sus funciones de esta aplicación en una barra de iconos que son visibles en la parte superior izquierda de la interfaz, que en letras contiene las funciones archivo, edición, programa, herramientas, y ayudas; seguido a estas funciones, se encuentran más abajo de estas pestañas cinco botones.

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.6.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for file operations and execution. The main text area shows the following code:

```
the documentation at http://www.arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

The status bar at the bottom indicates "Arduino Uno en COM2".

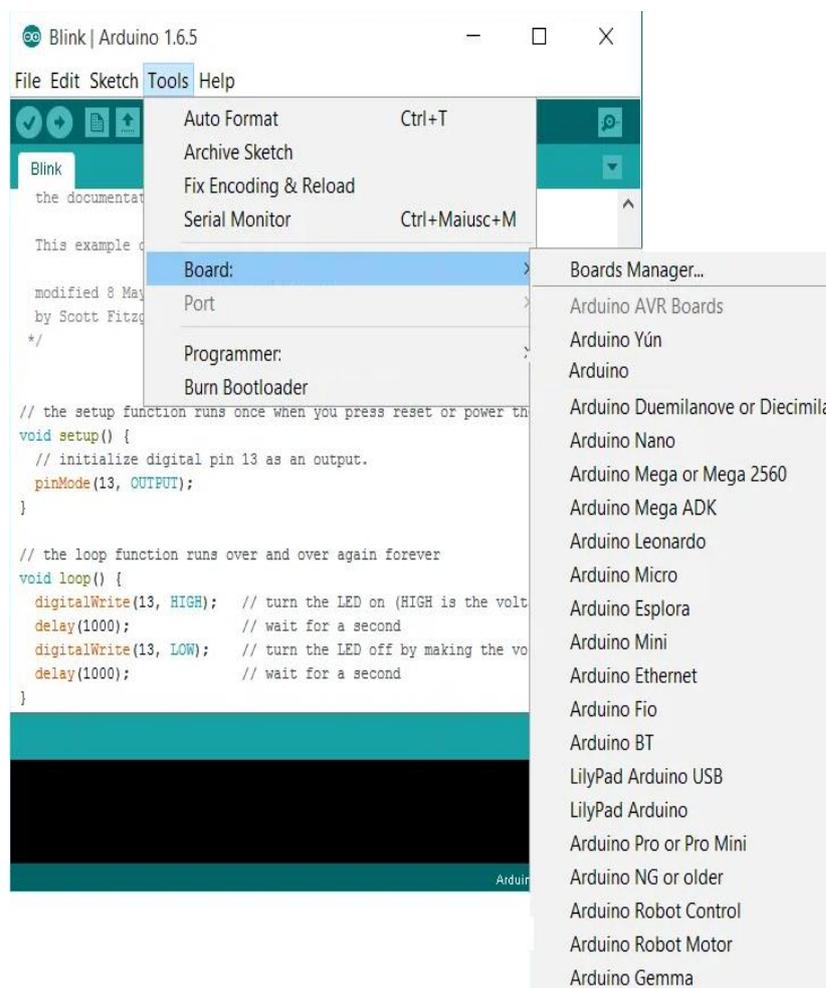
**Gráfico 55:** Programa Blink

El primer botón es para verificar el código, se utiliza para compilar y verificar que no hay errores en él. El segundo botón, subir, se usa para grabar en memoria del sistema embebido el programa en la tarjeta programable, enviando el sketch para su ejecución.

Añadido a estos botones, se tiene tres iconos que se utilizan para acciones rápidas como la de crear un nuevo proyecto o nuevo, abrir o guardar un proyecto, y salvar instrucciones añadidas constantemente al sketch principal.

Para comprobaciones e interacciones con el IDE y la tarjeta, la interfaz del programa ofrece funciones especiales como la del icono en la parte superior derecha, que se utiliza para abrir el monitor serie que ofrece este sistema, con el cual se puede comprobar el código realizado.

Físicamente al sistema embebido Atmega32u4 se le enviara a través del IDE un sketch o programa, para lo cual siendo este equipo un hardware reconocido por un software, este debe ser inicializado, para lo que se debe configurar la tarjeta conectada por el USB, para lo cual en herramientas, se selecciona a la pestaña placa, y en placas a la pestaña leonardo, la cual es la que contiene el bootloader instalado para el integrado Atmega32u4, también se ha de comprobar que la tarjeta a la que se transferirá el proyecto este reconocida a través del puerto COM al que está conectado, para lo cual en herramientas, se escogerá la pestaña puerto y en puerto se elegirá al que el sistema designe como tarjeta conectada, con lo cual a través de esta configuración, la tarjeta será reconocida por el software y la grabación del sketch se realizara correctamente.

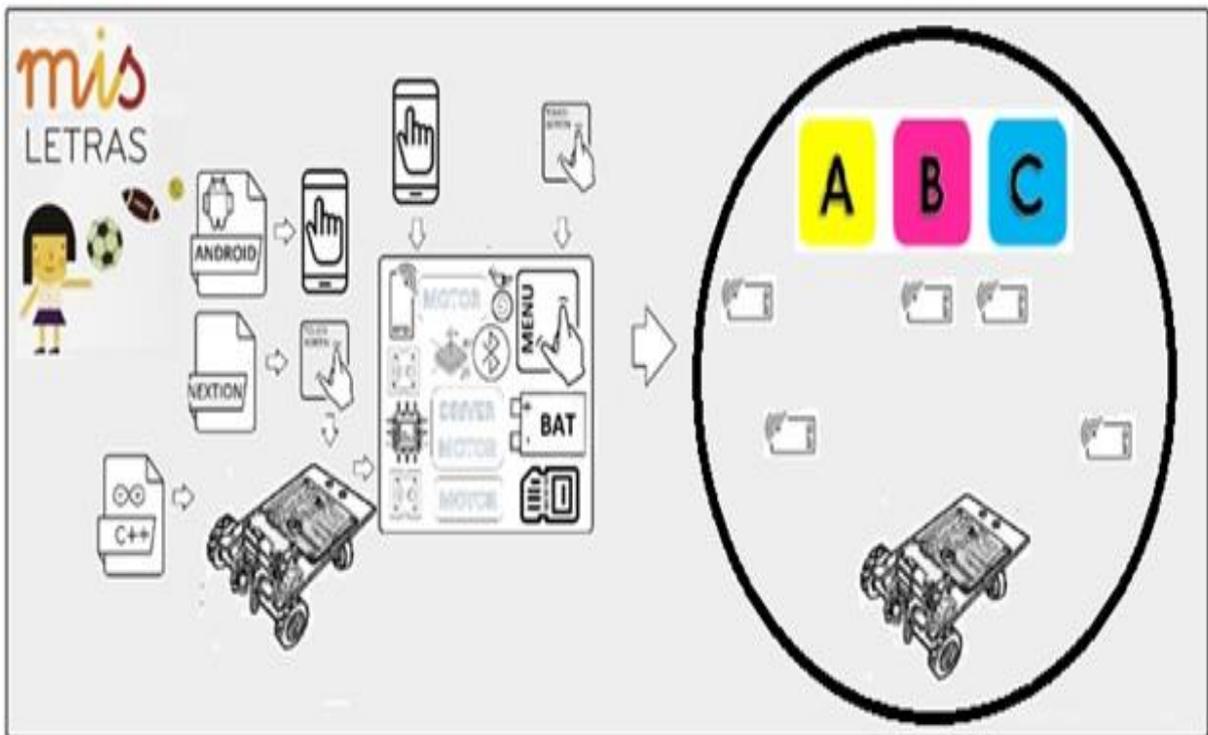


**Gráfico 56:** Configuración del IDE

Como ejemplo de un algoritmo de prueba, se realizó el ejemplo blink (archivo/ejemplos/basics/blink) a uno de los pines específicamente el pin 13, al cual se le programara a usarse como un led de propósito general, este viene incluido en la tarjeta, y al grabar el programa que se detalla a continuación, se encenderá y apagará rítmicamente.

1. /\*Ejemplo Blink \*/
2. void setup() {
3. pinMode(LED\_BUILTIN, OUTPUT);
4. }
5. void loop() {
6. digitalWrite(LED\_BUILTIN, HIGH); // encender el LED
7. delay(1000); // espera por un segundo
8. digitalWrite(LED\_BUILTIN, LOW); // apagar el LED
9. delay(1000); // espera por un segundo

#### 5.7.4.2. Aplicación de direccionamiento controlado: Atmega32u4



**Gráfico 57:** Programa de aplicación

El programa de aplicación del sistema embebido Atmega32u4 se ha denominado mis letras (añadido en la primera parte de anexos), el cual cumple la tarea de direccionamiento y posterior posicionamiento sobre una letra, para lo cual se ha dotado al sistema de pequeños obstáculos que se reconocerán con el módulo rfid Rc522, que detectara una señal magnética en el trayecto hasta llegar a una meta, en caso de pasar sobre uno de los módulos rfid el sistema emitirá un sonido lo cual indicara que deberá iniciar de nuevo el recorrido, haciendo el manejo de este sistema entretenido a quien esté tratando de llegar a la meta con esta aplicación.

Para el control a distancia o in situ, se dispondrá de medios alternativos de manipulación, que el programa desarrollado en el IDE de Arduino en el lenguaje C++, receptara, el primero será por medio de un smartphone con una aplicación de control, el segundo será a través de la pantalla táctil provista en el cuerpo del prototipo, que según pulsaciones comandara orientación, El anexo 8 del código de aplicación: “Mis Letras” transcribe el programa que se cargará en memoria de la tarjeta programable de este proyecto.

## 6. PRESUPUESTO Y ANÁLISIS DE IMPACTOS

### 6.1. Presupuesto

En este capítulo, se describe el análisis económico y financiero del prototipo de robot móvil con sistemas embebidos, para lo cual se presenta para la construcción del prototipo los gastos para un óptimo y viable desarrollo de la propuesta.

#### 6.1.1. Costos Directos

En el análisis de costos directos se toman en cuenta los gastos por materiales, los mismos que se detallan a continuación:

**Tabla 8:** Costos Directos materia prima

Costo de elementos(materia prima directa)			
Elemento	Cantidad	Costo individual	Costo total
Microcontrolador Atmega32u4	1	15,00	15,00
Regulador de Voltaje 5v	1	1,00	1,00
Conector USB	1	0,75	0,75
Leds indicadores	3	0,15	0,45
Condensador cerámico de 33 pF	2	0,15	0,30
Cristal de 16 Mhz	1	0,75	0,75
Resistencia de 330 ohm 1/4 W	3	0,10	0,30
Micro pulsadores mecánicos	5	0,10	0,50
Condensadores de 10µF	2	0,25	0,50
Pcb 20*30cm de fibra de vidrio	1	7,00	7,00
Batería LiPo 1600mah	2	15,00	30,00
Sensor ultrasónico HC-SR04	1	3,00	3,00

<b>Costo de elementos(materia prima directa)</b>			
<b>Elemento</b>	<b>Cantidad</b>	<b>Costo individual</b>	<b>Costo total</b>
Cable 22 AWG	10	0,35	3,50
Conectores Macho – Hembra	20	0,23	4,60
Motor DC	2	14,00	28,00
Servomotor Sg-90	1	4,00	4,00
Ruedas De Goma 32x7mm	2	8,50	17,00
Filamento PLA para impresora 3d	1	35,00	35,00
Pernos M6	8	0,15	1,20
Tuercas M6	8	0,10	0,80
Tornillos M4	21	0,10	2,10
Tuercas M4	21	0,10	2,10
		<b>Total</b>	<b>\$157,55</b>

### 6.1.2. Costos indirectos

Para el análisis de costos indirectos, se detallan los valores que permanecen constantes independientemente de los niveles de producción, para los cuales se toman en cuenta los gastos en mano de obra, maquinaria, transporte utilizado, de oficina, etc., los mismos que se detallan a continuación:

**Tabla 9:** Costos indirectos maquinaria y equipo

<b>Materia prima indirecta</b>			
<b>Maquinaria/Equipo</b>	<b>Costo/Hora</b>	<b>Horas/Equipo</b>	<b>Costo total</b>
Impresión 3D	15,00	10	150,00
Taladro de Banco	5,00	2	10,00
Esmeril	5,00	2	10,00
Otros equipos			50,00
		<b>Total</b>	<b>\$ 220</b>

**Tabla 10:** Costos indirectos varios

Descripción	Valor
Materiales de Oficina	50,00
Transporte	80,00
Internet	20,00
Otros	30,00
<b>Total</b>	<b>\$ 180</b>

### 6.1.3. Presupuesto total

Consiste en precisar los estimados de inversión para la buena culminación de todo un proceso de planeación que comprende todas las áreas que se han elaborado con cálculo, planificación y formulación a un estado económico de la propuesta, sumando los gastos directos y gastos indirectos, con el cual sacar el costo de total neto.

**Tabla 11:** Presupuesto Total

Descripción	Subtotal
<b>Costos Directos</b>	
Tabla 7	157,55
Tabla 8	220
<b>Costos Indirectos</b>	
Tabla 9	180
SUBTOTAL	557,55
Imprevistos (3,5%)	25,53
<b>COSTO TOTAL</b>	<b>\$754,88</b>

### 6.2. Análisis de impactos

El presente resumen incluido en el Informe final del presente estudio, da la caracterización técnica y económica del sistema desarrollado denominado BRICK de esta investigación que ha utilizado el integrado Atmega32u4, objetivando el carácter técnico comparativo entre este proyecto y el equivalente comercial de la compañía Lego, llamado Lego Mindstorn, identificando puntos críticos, facilidades y desarrollos prácticos en torno a éstos.

El carácter de la investigación de esta propuesta basado en el propio desarrollo de una tarjeta programable basa su estrategia en el poco desarrollo regional que existe referido al tema, lo que imprime a esta investigación un valor adicional, como precedente para orientar futuras investigaciones y profundizar en el campo de la programación de sistemas embebidos.

### 6.2.1. Análisis práctico

El desarrollo de los Sistemas Embebidos ocupa toda la gama de la tecnología, esperando a él experimentador con sus propuestas, al día de hoy todos los adelantos tecnológicos que se ocupan a diario en nuestros oficios, o son parte de nuestra vestimenta, están realizados con microcontroladores, o elementos electrónicos programables, que facilitan procesos en respuesta a acciones consistentes para el entorno en el que se apliquen.

### 6.2.1. Análisis técnico

El desarrollo de aplicaciones con Sistemas Embebidos para robots móviles generara aportes científicos, técnicos y metodológicos en prácticas didácticas a el perfil profesional de los futuros ingenieros, al contar con una nueva herramienta que brinda simplicidad, flexibilidad y menores tiempos en el desarrollo de prototipos robóticos a su alcance.

**Tabla 12:** Comparativa sistemas BRICK con Lego

<b>Tarjeta de desarrollo BRICK</b>	<b>Lego Mindstorm NTX</b>
- Controlada por un integrado comercial.	- Controlada por un integrado dedicado.
- Interfaz de desarrollo programable libre.	- Interfaz de desarrollo exclusivo de lego.
- Pantalla táctil programable multicolor.	- Pantalla monocolor solo de información.
- Módulos removibles expansibles.	- Módulos no removibles, con expansiones.
- Modulo bluetooth removible.	- Módulo bluetooth no removible.
- Módulo de reloj integrado removible.	- Módulo de reloj no incluido.
- Diseño de firmware libre y disponible.	- Diseño de firmware no libre disponible.
- Entorno de desarrollo libre IDE, con herramientas, y una comunidad expansible.	- Entorno de desarrollo no libre, comunidad colaborativa, con herramientas pagadas,
- Tecnología libre, en software y hardware.	- Tecnología pagada, no modificable.
- Posibilidad de modificar diseño electrónico	- No es modificable.
- Lenguaje de programación libre, con posibilidad de varios otros productos libres.	- Lenguaje de programación propio de lego y sus productos.

### **6.2.1.1. Comparativa técnica de esta propuesta con propuestas comerciales**

La propuesta de esta investigación denota en esta sección una comparativa de los dos objetos propuestos como tarjetas programables, las cuales, de acuerdo con criterios presentados, mostrando de forma optativa una oferta entre el modelo BRICK y su contraparte comercial de tarjeta programable denominada comercialmente Lego Mindstorm NTX.

### **6.2.3. Análisis social**

Con la Aplicación de Sistemas Embebidos para la construcción de Robots Móviles en el club de robótica de la Universidad Técnica de Cotopaxi, el impacto social que generará, recae sobre los usuarios de este sistema (coordinador, estudiante), ya que al ocupar el sistema, el usuario agilizará la gestión de las actividades sobre esta plataforma, de tal manera de agilizar el tiempo de ejecución de prácticas didácticas para la proyección de prototipos posibles con la utilización de esta aplicación en sistemas embebidos.

### **6.2.4. Análisis económico**

Se ha identificado que mediante la implementación de Sistemas Embebidos para la construcción de Robots Móviles en el club de robótica de la Universidad Técnica de Cotopaxi, representa una inversión mínima a procesos similares que pueden realizar los usuarios de una plataforma para desarrollar prototipos robóticos, que realizan los usuarios (coordinador, estudiante), para la gestión de forma didáctica de prácticas que tendrían que realizar desde cero, con la consecuente inversión e investigación para desarrollar aplicaciones en prototipos, por lo que se considera una inversión aceptable, que minimiza gastos generados en la antigua gestión del desarrollo de prototipos y actividades en robótica.

#### **6.2.4.1. Comparativa económica de esta propuesta con propuestas comerciales**

Económicamente se tomará el costo de los materiales de la propuesta BRICK que da un total de \$ 26,55 dólares, sumando a los mismos insumos varios como consumo de luz de tarifado nacional de la página <http://www.centrosur.gob.ec/calcular-consumo> que da como tarifa el kilovatio hora a un costo de 0,74 centavos de dólar, aplicando la fórmula recomendada por esta web indica que:

$$\text{Consumo} = w * \text{horas} * \text{días} / (\text{KWH}) 1000$$

El watio de consumo se da por los elementos utilizados entre los que están el caufín para soldar con una potencia de 20 watios, una luz de 15 watios, el consumo de la insoladora que es de 50

watios, a un promedio de dos horas diarias de trabajo por una semana (7 días), reemplazando valores es igual a:

$$\text{Consumo} = 85 \text{ w} * 14 \text{ horas} * 7 / 1000$$

$$\text{Consumo} = 8.33 \text{ w} * 0,74$$

$$\text{Consumo} = \$ 6, 16$$

También se incluirá en este cálculo la mano de obra, dato tomado del portal web del gobierno <http://www.ecuadorlegalonline.com/laboral/calculadora-valor-hora-de-trabajo>, el cual se hará en base a el sueldo básico actual que es de \$ 467, dando el valor en jornada normal el costo de hora de \$ 1,95 dólares \* 14 horas da un total de \$ 27,30, sumando estos valores, se compara contra el costo total de la tarjeta Lego Mindstorm en el mercado ecuatoriano del referente de ventas online [www.mercadolibre.com](http://www.mercadolibre.com), que sin aditamentos tiene el costo de \$580 dólares, valor bajo pedido y entrega post venta con días laborables de importación.

**Tabla 13:** Comparación costos tarjeta BRICK con tarjeta Lego

<b>Tarjeta programable BRICK</b>			
Insumos = \$26,55	Consumo = \$ 6,16	Mano de obra = \$ 27,30	Total = \$ 60,01 dólares
<b>Tarjeta comercial Lego Mindstorm NTX</b>			
Costo total de importación = \$580 dólares			
<b>Resumen:</b> Tarjeta Brick = 60,01 dólares – Tarjeta Lego Mindstorm NTX = 580 dólares = 519,99 de diferencia entre las dos propuestas.			

### 6.2.5. Análisis epistemológico

Fundamentar el conocimiento humano y la metodología que ha llevado a la tecnología actual, contrarresta desde un plano inclinado a un ejemplarizador punto de apoyo, con el cual incluso se podría mover al mundo como menciona Arquímedes.

La aplicación de Sistemas Embebidos, apunta al concepto de la generalidad de la utilización del pensamiento computacional, en la que se programa software intangible dirigido a lo físico y tangible, que abarca todo el universo de la nueva tecnología actual, como el de los robots, y que es multi destino según como se lo aplique.

Afirmando la conceptualización sobre los sistemas embebidos, se ha de señalar que el campo del hardware programado es fundamental para el profesional en la ciencia actual, que con diseños y prototipos adapta un entorno con el cual se genera didáctica y experimentación, cualificación de eventos que este proyecto con una tarjeta programable logra a través de herramientas libres.

## **7. CONCLUSIONES Y RECOMENDACIONES**

### **7.1 CONCLUSIONES**

- Al realizar el análisis en base a los fundamentos teóricos de los sistemas embebidos aplicados a robots móviles, se determina que se puede realizar el diseño y construcción de una tarjeta programable para que realice actividades de un programa que emite señales de onda para que reciba ordenes de movimiento, quedando a disposición del club de robótica de la Universidad Técnica de Cotopaxi el presente proyecto para reproducción o proyectos similares.
- Se realiza una valoración de las diferentes herramientas de programación en base a instrumentos de software libre relacionados con sistemas embebidos para el diseño del prototipo del robot móvil que permitió recibir información a través de los diferentes medios de intercomunicación tales como bluetooth, pantallas táctiles entre otras las cuales acoplados a la tarjeta programable diseñada accionaron al robot móvil.
- Se determina las necesidades de los circuitos integrados electrónicos para el diseño y construcción de la tarjeta programable con un microcontrolador Atmega32u4 que cumple con las especificaciones técnicas de intercomunicación con una computadora y periféricos para que efectúe funciones específicas de intercomunicación.
- La valoración económica realizada arrojó un costo total de \$754,88 dólares americanos; ocupando de este total la tarjeta BRICK un costo de \$ 60.01 dólares americanos, considerándose factible en comparación con otras tecnologías como Lego Mindstorm NTX que su costo estaría por los \$ 580.00 dólares americanos aproximadamente; desde el punto de vista tecnológico para la tarjeta BRICK existen ventajas tales como la posibilidad de control por un integrado comercial, que es de fácil adquisición, una interfaz de desarrollo programable libre, pantalla táctil programable multicolor, módulos removibles expansibles que pueden ser diseñados por los usuarios, entre otras.

### **7.2 RECOMENDACIONES**

- Que el club de robótica de la Universidad Técnica de Cotopaxi aplique en sus proyectos la propuesta y resultados de la presente investigación.
- Se recomienda valorar nuevas herramientas de programación orientados a sistemas embebidos que permitan un diseño del prototipo del robot móvil más óptimo y eficientes, de manera que se logre un producto mejor acabado.
- Profundizar en el estudio de nuevos microcontroladores Atmega que permita la concepción de prototipados que sean aplicables en futuros entornos de trabajo para el club de robótica de la UTC.
- Continuar trabajando en optimización, eficiencia y eficacia de los sistemas embebidos en el club de robótica de la UTC de manera que se ahorre factiblemente en el indicador económico y así disminuir los costos de producción de prototipados de este tipo.

## 8. REFERENCIAS

- [1] I. Ruge, F. López, O. Gómez, "Robot Daro: plataforma robótica para educación en ingeniería", *Revista Ingenium*, vol. 18, 35, pp. 58-74, Marzo 2017.
- [2] J. Vera, D. Gómez, H. Rodríguez, "Diseño e Implementación del Sistema Electrónico y Comunicación para el Control un Robot Modular Tipo Serpiente", *Revista de Tecnología*, vol. 16, 2, pp. 114-121, Agosto 2017.
- [3] J. Davis, "La vida robot", *Revista Wired*, vol 13, 04, pp 48-68, Abril 2005.
- [4] E. Silvi, (2016). Programa de Alfabetización Tecnológica: Robótica Educativa [En línea]. Disponible: <https://bit.ly/2vmBMX1>
- [5] G. Zabala, *Robots. O el sueño eterno de las máquinas*, Argentina: Editorial Siglo XXI, 2012
- [6] J. Pomares, C. Jara, F. Candelas-Herías, A. Delgado, A. Alabdo, J. Alepuz, A. Torres, (2016) Coordinación transversal de asignaturas de control y robótica en asignaturas del Máster Universitario en Automática y Robótica [En línea]. Disponible: <https://bit.ly/2Hek11t>
- [7] O. Torrente, *Arduino. Curso práctico de formación*, México: Editorial Alfaomega, 2013.
- [8] L. Arteaga, (2017). ¿Qué es software libre? [En línea]. Disponible: <https://bit.ly/1j1Ai5o>

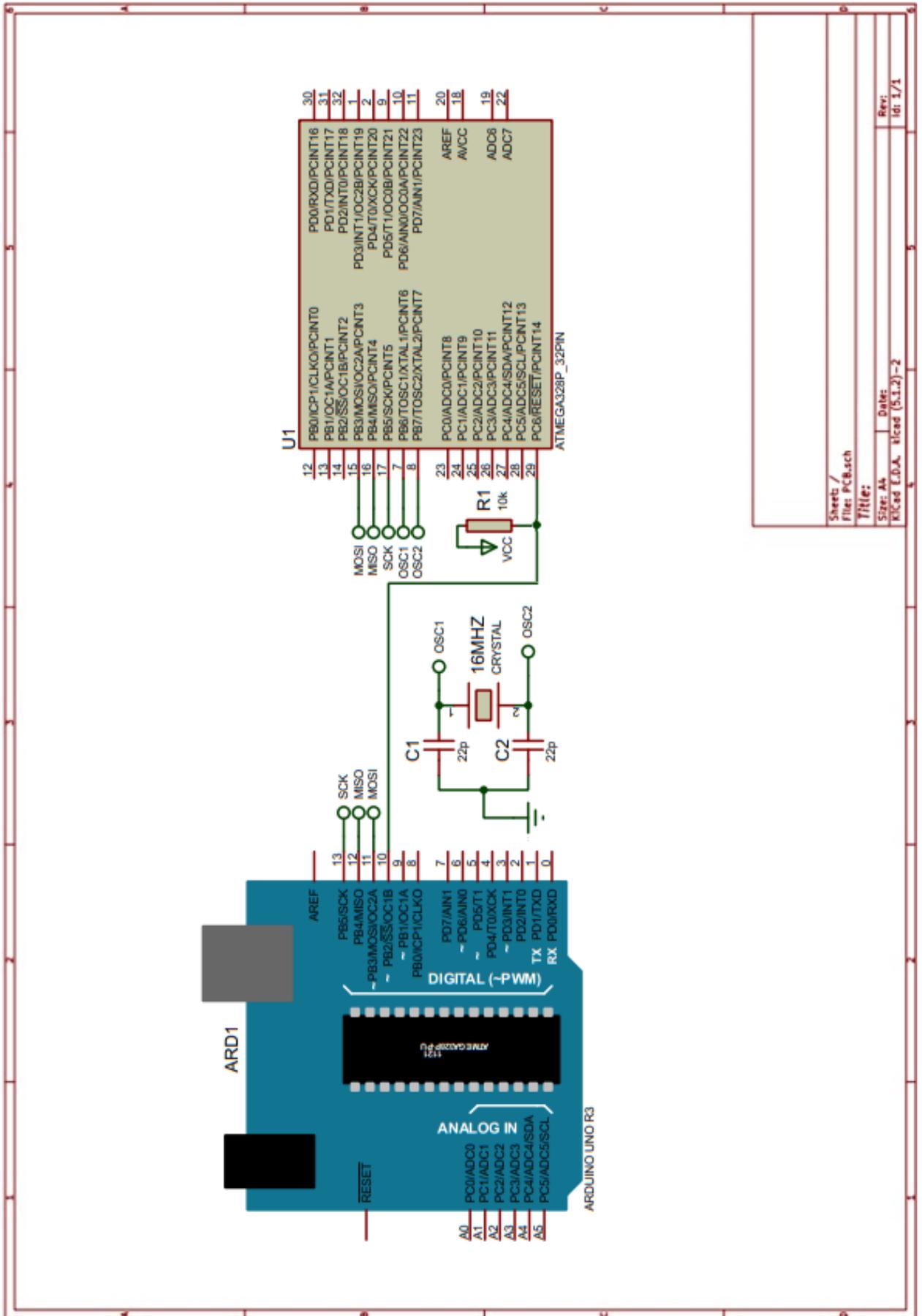
- [9] C. Castro, S. Osorio, "Tarjetas de Desarrollo: Herramientas para el diseño", *Revista Letras con Ciencia Tecno Lógica*, vol. 1, 1, pp. 43-52, Diciembre 2018.
- [10] N. Pérez-Cáceres, "Metodología para la Simulación y Fabricación de Circuitos Impresos en Radiofrecuencia", tesis, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica, 2018.
- [11] P. Martínez, "Dispositivo electrónico para el estudio de cobertura de redes LPWAN", tesis, Universidad Politécnica de Cartagena, Cartagena, Colombia, 2018.
- [12] M. Aguilera, "La revolución tecnológica actual aplicada a los audífonos. ¿Qué hay de nuevo y cuál es su aporte?", *Revista Médica Clínica Las Condes*, vol. 27, 6, pp. 767-775, 2016.
- [13] Calderón, H. A., Mejía, C., & Cobo, L., "Implementación de un robot móvil seguidor de línea y detector de obstáculos con comunicación Bluetooth", *Revista Ontare*, vol. 4, 2, pp. 99-118, Septiembre 2016.
- [14] J. Axelson, *USB complete*, United States of America: Lakeview Research LLC, 2015
- [15] E. López, Ingeniería en Microcontroladores (Protocolo USB), Apuntes de Ingeniería Informática, [En línea]. Disponible: <https://bit.ly/2VyWWAN>
- [16] N. Martínez, J. Olivencia, E. Meneses, "Robótica, modelado 3D y realidad aumentada en educación para el desarrollo de las inteligencias múltiples", *Revista Aula de encuentro*, vol. 18, 2, pp. 158-183, 2016.
- [17] P. Beltrán-Pellicer, "Modelado e impresión 3D como recurso didáctico en el aprendizaje de la probabilidad", *Revista Épsilon*, vol. 34, 95, pp. 99-106, 2017.
- [18] L. Reyes, "Análisis documental de las ventajas de la impresión 3D", *RECI Revista Iberoamericana de las Ciencias Computacionales e Informática*, vol. 6, 11, pp. 1-12, Junio 2017.
- [19] J. Gutiérrez, G. de la Puente, A. Martínez, E. Piña, *Aprendizaje basado en Problemas. Un camino para aprender a aprender*, México: Editorial Universidad Autónoma de México, 2012.

- [20] D. Padilla, A. Martínez, "Experiencia didáctica con Arduino. El aprendizaje basado en proyectos como metodología de trabajo en el aula de secundaria", *Revista Hekademos: revista educativa digital*, vol. 11, 25, pp. 73-82, Diciembre 2018
- [21] C. Saiz, S. Rivas, "Pensamiento crítico y aprendizaje basado en problemas cotidianos", *Revista de Docencia Universitaria*, vol. 10, 3, pp. 325-346, Diciembre 2012.
- [22] A. Saleh, C. Silver, Y. Chen, K. Shanahan, K. Rowe, J. Lester, "Scaffolding peer facilitation in computer-supported problem-based learning environments", *International Society of the Learning Sciences*, 2018.
- [23] G. Galeano, *Programación de Sistemas Embebidos en c*, México: Editorial Alfaomega, 2009.
- [24] C. Reyes, *Microcontroladores PIC Programación en Basic*, Ecuador: Editorial Rispergraf, 2008.
- [25] F. Reys, *Robótica. Control de Robots Manipuladores*, México: Editorial Alfaomega, 2011
- [26] N. López, I. Sandoval, (2016). Métodos y técnicas de investigación cuantitativa y cualitativa [En línea]. Disponible: <https://bit.ly/2VTn2hL>
- [27] J. Vega, L. Cusicanqui, F. Avalos, "El método científico o los métodos de la ciencia, conjetura resuelta", *Revista Pueblo Continente*, vol. 25, 2, pp. 329-334, Noviembre 2016.
- [28] D. Neill, L. Cortez, *Procesos y fundamentos de la investigación científica*, Ecuador: Editorial UTMACH, 2017
- [29] J. Vergara, *Aprendo porque quiero. El aprendizaje basado en proyectos (ABP)*, paso a paso, España: Editorial SM, 2015
- [30] E. Vargas, "Metodología en proyectos mecatrónicos industriales", I Congreso y Taller Internacional de Mecatrónica, Monterrey, 2007.
- [31] C. López, M. Calderón, C. Ávila, "SolidWorks: una alternativa de formación de capital humano y desarrollo endógeno", 21 Encuentro Nacional sobre Desarrollo Regional en México, Yucatán, 2016.
- [32] D. Benchimol, *Microcontroladores*, Argentina: Fox Andina & Dálaga S.A., 2011.

- [33] R. Tagliaterri, S. Balzarini, *Robótica*, Argentina: Fox Andina & Dálaga S.A., 2013.
- [34] A. Medrano, A. Serra, C. Soto, "KiCad, Herramienta de Software Libre de Modelado de Circuitos Impresos para el Desarrollo de Hardware KiCad, free software suite for Electronic Design Automation (EDA) to Hardware Development", *Revista Ciencia e Ingeniería: revista digital*, vol. 38, 2, pp. 177-186, Julio 2017.
- [35] L. Louis, "WORKING PRINCIPLE OF ARDUINO AND USING IT", *Revista International Journal of Control: Revista Automation, Communication and Systems*, vol. 1, 2, pp. 21-29, Abril 2016.
- [36] S. Papadakis, "The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education", *Revista International Journal of Web-Based Learning and Teaching Technologies: revista educativa digital*, vol. 12, 4, pp. 58-77, Diciembre 2017.
- [37] A. Cantillo, "Modelado dinámico del manipulador serial Mitsubishi Movemaster RV-M1 usando SolidWorks", *Revista UIS: revista ingenierias*, vol. 15, 2, pp. 49-62, Junio 2016.
- [38] J. Baca, "Diseño De Una Impresora 3d De Cabezales Intercambiables Para La Escuela De Ingeniería Mecánica-Eléctrica De La Universidad Cesar Vallejo", tesis, Universidad Cesar Vallejo, Trujillo, Perú, 2017.
- [39] R. García, *Asimov: programa de robótica*, Colombia: Tecno parque, 2015.
- [40] J. Mercader, "El impacto de la robótica y el futuro del trabajo", *Revista Facultad de Derecho de México*, vol. 67, 269, pp. 149-174, 2017.
- [41] R. Bojana, "Robotic Art and Cultural Imagination", Conference Digital arts and culture, Copenhagen, 2018.
- [42] A. García de la Cárcova, "Creación y Producción en Diseño y Comunicación", *Revista Centro de Estudios en Diseño y Comunicación*, vol. 15, 82, pp. 39, Julio 2018.
- [43] I. Asimov, "El sol desnudo", España: Ediciones Martínez Roca, 2002.

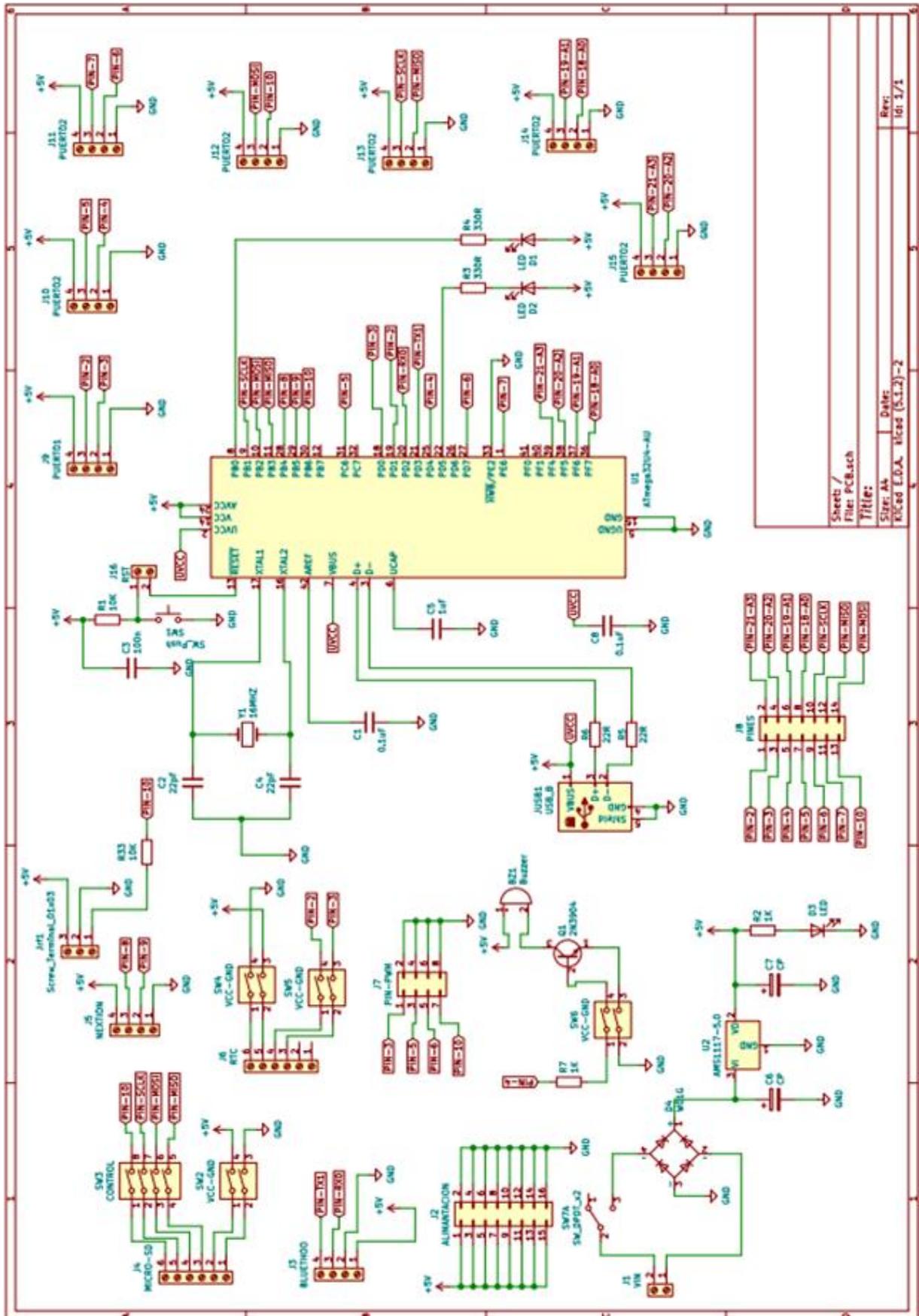
- [44] M. Aguirre, "Tecnología convertida en mito: la obra artística de Eduardo Paolozzi", *Revista Icono 14*, vol. 15, 1, pp. 149-165, Diciembre 2017.
- [45] D. Cuesta, "Diseño y construcción de un brazo robótico de 6 GDL", *Revista Tekhne*, vol. 13, 1, pp. 73-82, 2016.
- [46] G. Bermúdez, "Robots móviles. Teoría, aplicaciones y experiencias", *Revista Tecnura*, vol. 5, 10, pp. 6-17, 2002.
- [47] C. Ferrell, "Global behavior via cooperative local control", *Revista Autonomous Robots*, vol. 2, 2, pp. 105-125, 1995.
- [48] T. Takenara, "The control system for the honda humanoid robot", *Revista Age and ageing*, vol. 35, 2, pp. 124-126, 2006.
- [49] M. Veloso, "CMRoboBits: Creating an intelligent AIBO robot", *Revista AI magazine*, vol. 27, 1, pp. 67-67, 2006.
- [50] U. Pagallo, "Vital, Sophia, and Co. The quest for the legal personhood of robots", *Revista Information*, vol. 9, 9, pp. 230, 2018.
- [51] F. Cortes, J. Cabrera, (2020, Febrero 4). Línea Robótica [En línea]. Disponible: <https://bit.ly/2Saelfx>.
- [52] V. Olavarria, (2020, Febrero 4). Sophia, El Primer Robot Valentín Y Juan [En línea]. Disponible: <https://bit.ly/2H4YUyP>.

# Anexo I: Arduino ISCP



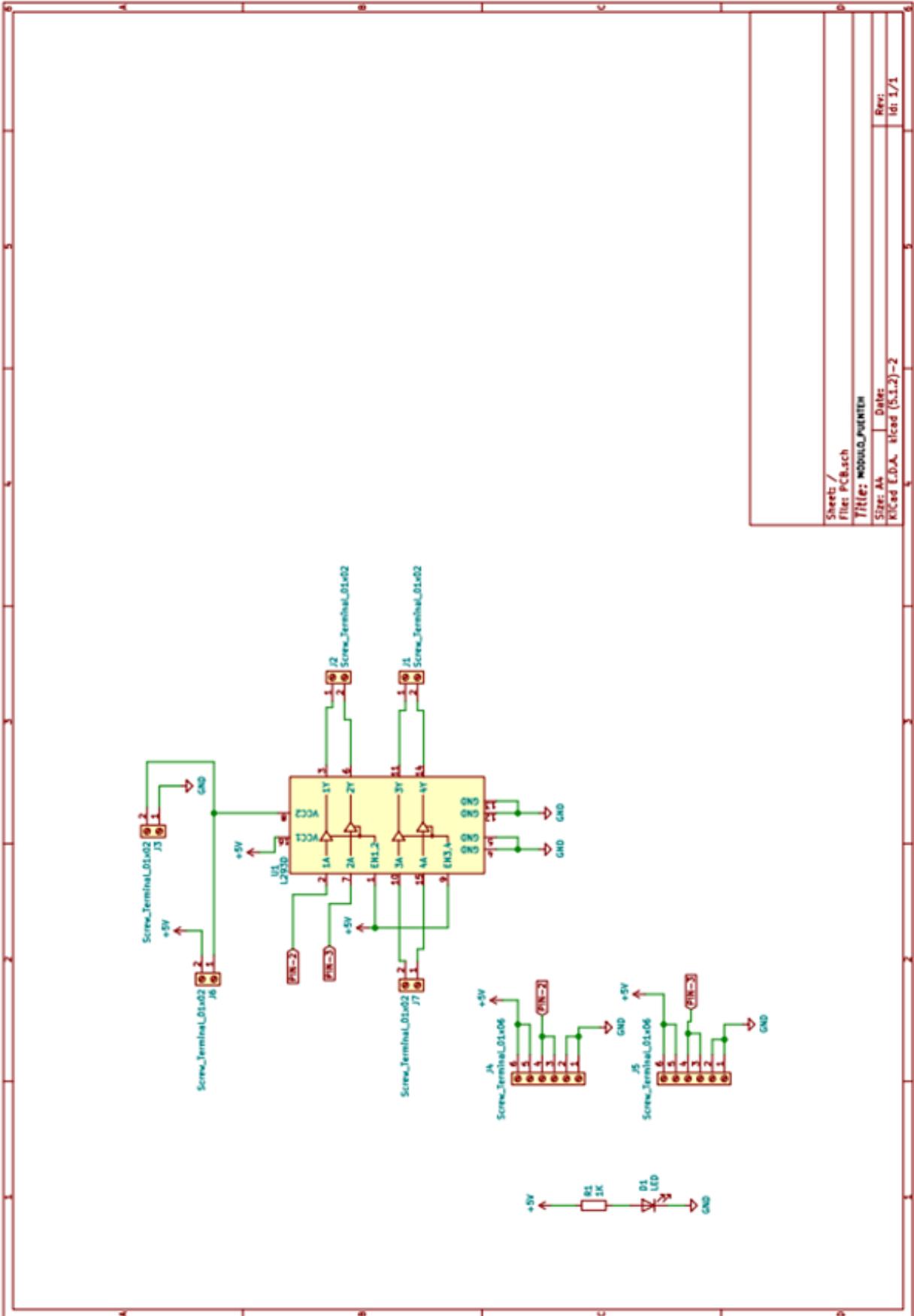


### Anexo III: Diagrama esquemático general 2



Sheet: /	Date:	Rev:
File: PCB.sch		161.1/1
Title:		
Sheet: A4	Date:	
File: E.D.A. Ucad (S.1.2)-2		

Anexo IV: Diagrama esquemático Lm293d

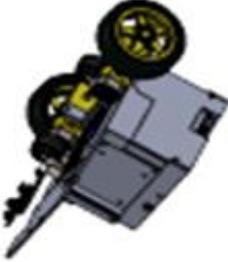
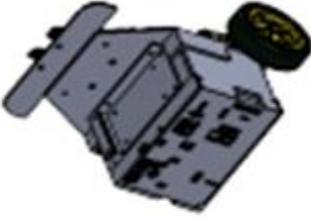


Sheet /	
File: PCB.sch	
Title: MODULO_PUNTERA	
Size: A4	Date:
KiCad 5.1.2-2	Rev: Idi: 1/1





Anexo VII: Vistas de impresión del robot armado

	4	3	2	1	
F	 				F
E					E
D					D
C	 				C
B					B
SI NO SE INDICA LO CONTRARIO, ACABADO: LAS CORTES SE ESPESALAN EN MM ACABADO SUPERFICIAL: TELESCOPICA: LINEAL: ANGULAR				REPERFORAR Y RECAPAR AREAS VUELAS	NO CAMBIE LA ESCALA REVISIÓN
DISEÑO: VERIFICADO: APROBADO: FIRMADO: CADUCADO:	NOMBRE: FECHA: MATERIAL:	FECHA: MATERIAL:	NÚMERO DE DISEÑO: ESCALA: 1:1		
A	ROBOT			A4	A
4	3	2	1	1	

## Anexo VIII: Código de aplicación: “Mis Letras”

1. **#include <SPI.h>**
2. **#include "Nextion.h"**
3. **#include <SoftwareSerial.h>**
4. **#include "RTCLib.h"**
5. **#include <SD.h>**
6. **#include <MFRC522.h>**

7. **int act=0;**

8. **int FaS4=369;**
9. **int Sol4=392; //4**
10. **int Si4=493;**
11. **int La4=440;**

12. **int Do5=523; //5**
13. **int Re5=587; //5**
14. **int Mi5=659; //5**
15. **int Fa5=698; //5**
16. **int FaS5=739; //5**
17. **int Sol5=783; //5**
18. **int La5=880; //5**
19. **int Si5=987; //5**

20. **int Do6=1046; //6**
21. **int Re6=1174; //6**
22. **int Mi6=1318; //6**
23. **int Fa6=1396; //6**
24. **int FaS6=1476; //6**
25. **int Sol6=1567; //6**
26. **int La6=1760; //6**
27. **int Si6=1975; //6**

28. **int Do7=2093; //7**
29. **int Re7= 2349; //7**

30. **int Sol= 783; //5**
31. **int FaS= 1479; //6**
32. **int Si= 1975; //6**

**33. //DEFINIMOS LAS NOTAS**

**34. int redonda=0;**

**35. int redondap=0;**

**36. int blanca= 0;**

**37. int blancap=0;**

**38. int negra=0;**

**39. int negrap=0;**

**40. int corchea=0;**

**41. int corcheap =0;**

**42. int semicorchea=0;**

**43. int semicorcheap=0;**

**44. int seminegra=0;**

**45. int semiblanca=0;**

**46. int blancapuntosemicorchea=0;**

**47. int semicorchea2negra=0;**

**48. //definimos la variable del tiempo**

**49. int bpm= 145;**

**50. SoftwareSerial HMISerial(8, 9);**

**51. RTC\_DS3231 rtc;**

**52. String daysOfTheWeek[7] = { "Dom", "Lun", "Mar", "Mier", "Juev", "Vier",  
"Sab" };**

**53. //String monthsNames[12] = { "Ene", "Feb", "Mar", "Abr", "May", "Jun",  
"Jul", "Ago", "Sep", "Oct", "Nov", "Dic" };**

**54. #define RST\_PIN 10 //Pin 9 para el reset del RC522**

**55. #define SS\_PIN A3 //Pin 10 para el SS (SDA) del RC522**

**56. MFRC522 mfrc522(SS\_PIN, RST\_PIN); //Creamos el objeto para el RC522**

**57. #define MD1 7**

**58. #define MZ1 6**

**59. #define MD2 5**

**60. #define MZ2 A0**

**61. #define buzzer 4**

**62. int estadosen = 0;**  
**63. int estado = 0;**

**64. int adelante = 0;**  
**65. int atras = 0;**  
**66. int derecha = 0;**  
**67. int izquierda = 0;**

**68. int adelantep = 0;**  
**69. int atrasp = 0;**  
**70. int derechap = 0;**  
**71. int izquierdap = 0;**

**72. int ad = 0;**  
**73. int der = 0;**  
**74. int izq = 0;**  
**75. int atr = 0;**

**76. int adp = 0;**  
**77. int derp = 0;**  
**78. int izqp = 0;**  
**79. int atrp = 0;**

**80. int cont = 0;**  
**81. int t5 = 500;**

**82. int hr=0;**  
**83. int mins=0;**  
**84. String dia;**

**85. uint32\_t t0;**  
**86. uint32\_t t1;**

**87. //////////////////////////////////////**  
**88. //-----PAGINA CORRIENTE-----**

**89. NexNumber n1 = NexNumber(0, 2, "ncorriente"); //DATO DE CORRIENTE**

90. NexNumber n2 = NexNumber(0, 3, "ncorriente2"); //DATO DE CORRIENTE

91. NexDSButton bt0 = NexDSButton(1, 3, "bt0"); //INICIAR GRAFICA

92. NexDSButton bt1 = NexDSButton(1, 4, "bt1"); //INICIAR GRAFICA

93. NexDSButton bt2 = NexDSButton(1, 5, "bt2"); //INICIAR GRAFICA

94. NexDSButton bt3 = NexDSButton(1, 6, "bt3"); //INICIAR GRAFICA

95. NexDSButton bt4= NexDSButton(1, 9, "bt4"); //INICIAR GRAFICA

96. //////////////////////////////////////

97. NexTouch \*nex\_listen\_list[] =

98. {

99. /// //-----PAGINA CORRIENTE-----

100. &bt0,

101. &bt1,

102. &bt2,

103. &bt3,

104. &bt4,

105. NULL

106. };

107. void setup() {

108. nexInit();

109. Serial.begin(9600); //Iniciamos la comunicación serial

110. Serial1.begin(9600);

111. rtc.begin();

112. SPI.begin(); //Iniciamos el Bus SPI

113. mfrc522.PCD\_Init(); // Iniciamos el MFRC522

114. pinMode(MD1, OUTPUT);

115. pinMode(MZ1, OUTPUT);

116. pinMode(MD2, OUTPUT);

117. pinMode(MZ2, OUTPUT);

118. pinMode(buzzer, OUTPUT);

119. digitalWrite(MD1, LOW);

120. digitalWrite(MD2, LOW);

121. bt0.attachPop(mv\_adelante, &bt0);

122. bt1.attachPop(mv\_derecha, &bt1);

123. bt2.attachPop(mv\_izquierda, &bt2);

124. bt3.attachPop(mv\_atras, &bt3);

125. bt4.attachPop(mv\_enter, &bt4);

```

126.   negra= 60000/bpm;   //400
127.   negrap=negra*1.5; //600
128.   blanca= negra*2;   //800
129.   blancap=blanca*1.5; //1200
130.   redonda= negra*4;  //1600
131.   redondap= redonda*1.5; //2400
132.   corchea= negra/2;  //200
133.   corcheap=corchea*1.5; //300
134.   semicorchea= negra/4; //100
135.   semicorcheap=semicorchea*1.5; //150

136.   seminegra=negra+ semicorchea; //500
137.   semiblanca=blanca+semicorchea; //900
138.   blancapuntosemicorchea=blancap+semicorchea;
139.   semicorchea2negra=corchea+negra;

140.   dbSerialPrintln("setup done");
141.   rtc.adjust(DateTime(2020,1,14,20,16, 0)); //IGUALAR REJOR
142.   Serial.println("INICIAR");
143.   delay(1000);

144.   t0 = millis();
145.   t1 = millis();

146.   }

147.   void loop() {

148.       nexLoop(nex_listen_list);

149.       if( millis()-t0>=5000 ){
150.           t0 = millis();
151.           DateTime now = rtc.now();
152.           printDate(now);
153.       }

154.       altos();
155.       //////////////////////////////////

```

```

156.     if(Serial1.available()){ // Si llega un dato por el puerto BT se envÃ-a al
        monitor serial
157.         estado = Serial1.read();
158.         Serial.println(estado);
159.         if(estado==49){
160.             adelante++;
161.             zum();
162.             Serial.print("PASOS ADELANTE");

163.         Serial.print(",");
164.         Serial.println(adelante);
165.         }

166.     if(estado==50){
167.         atras++;
168.         zum();
169.         Serial.print("PASOS ATRAZ");
170.         Serial.print(",");
171.         Serial.println(atras);
172.         }
173.     if(estado==53){
174.         derecha++;
175.         zum();
176.         Serial.print("PASOS DERECHA");
177.         Serial.print(",");
178.         Serial.println(derecha);
179.         }
180.     if(estado==52){
181.         izquierda++;
182.         zum();
183.         Serial.print("PASOS IZQUIERDA");
184.         Serial.print(",");
185.         Serial.println(izquierda);
186.         }

187.     if(estado==51){
188.         zum1();
189.         Serial.println("MOVIMIENTOS");
190.         Serial.print("PASOS ADELANTE");
191.         Serial.print(",");
192.         Serial.println(adelante);

```

```

193. Serial.print("PASOS ATRAZ");
194. Serial.print(",");
195. Serial.println(atras);

196. Serial.print("PASOS DERECHA");
197. Serial.print(",");
198. Serial.println(derecha);

199. Serial.print("PASOS IZQUIERDA");
200. Serial.print(",");
201. Serial.println(izquierda);

202. Serial.print("PASOS ATRAS");
203. Serial.print(",");
204. Serial.println(atras);

205. ////////////////
206. delay(t5);
207. if(adelante>0){
208. while(1){

209. Serial.print("MOVIMIENTO ADELANTE");
210. digitalWrite(MD1,HIGH);
211. digitalWrite(MD2,HIGH);
212. delay(150);
213. digitalWrite(MD1,LOW);
214. digitalWrite(MD2,LOW);
215. Serial.println("#####");
216. ad++;
217. Serial.println(ad);
218. Serial.println(adelante);
219. Serial.println("#####");
220. delay(100);
221. altos();
222. Serial.println("LECTURA TARJETA");

223. Serial.println("#####");
a. if(adelante==ad){

```

```

    b. Serial.println("SALIR MOVIMIENTO ADELANTE");
    c. ad=0;
    d. break;
    e. }
224. }
225. }
226. delay(t5);

227. if(derecha>0){
228. while(1){
229. Serial.print("MOVIMIENTO DERECHA");
230. digitalWrite(MZ1,HIGH);
231. digitalWrite(MD2,HIGH);
232. delay(150);
233. digitalWrite(MZ1,LOW);
234. digitalWrite(MD2,LOW);
235. Serial.println("#####");
236. der++;
237. Serial.println(der);
238. Serial.println(adelante);
239. Serial.println("#####");
240. delay(100);
    a. if(derecha==der){

        b. Serial.println("SALIR MOVIMIENTO DERECHA");
        c. der=0;
        d. break;
        e. }
241. }
242. }

243. delay(t5);
244. if(izquierda>0){
245. while(1){
246. Serial.print("MOVIMIENTO IZQUIERDA");
247. digitalWrite(MD1,HIGH);
248. digitalWrite(MZ2,HIGH);
249. delay(100);
250. digitalWrite(MZ2,LOW);
251. digitalWrite(MD1,LOW);
252. Serial.println("#####");
253. izq++;
254. Serial.println(der);

```

```

255.     Serial.println(adelante);
256.     Serial.println("#####");
257.     delay(100);
        a. if(izquierda==izq){

                b. Serial.println("SALIR MOVIMIENTO IZQUIERDA");
                c. izq=0;
                d. break;
                e. }
258.     }
259.     }

260.     delay(t5);
261.     if(atras>0){
262.     while(1){
263.     Serial.print("MOVIMIENTO ATRAS");
264.     digitalWrite(MZ1,HIGH);
265.     digitalWrite(MZ2,HIGH);
266.     delay(100);
267.     digitalWrite(MZ1,LOW);
268.     digitalWrite(MZ2,LOW);
269.     Serial.println("#####");
270.     atr++;
271.     Serial.println(atras);
272.     Serial.println(adelante);
273.     Serial.println("#####");
274.     delay(100);
        a. if(atras==atr){

                b. Serial.println("SALIR MOVIMIENTO atras");
                c. atr=0;
                d. break;
                e. }
275.     }
276.     }

277.     adelante=0;
278.     atras=0;
279.     derecha=0;
280.     izquierda=0;

```

```
281.     }
282.     }

283.     //////////////////////////////////

284.     }

285.     //////////////////////////////////
286.     //////////////////////////////////SSSS
287.     ////////////////////////////////////////////

288.     void mv_adelante()
289.     {
290.     Serial.println("ADELANTE DESDE PANTALLA");
291.     zum();
292.     adelantep++;
293.     Serial.println(adelantep);

294.     }

295.     void mv_derecha()
296.     {
297.     Serial.println("DERECHA DESDE PANTALLA");
298.     zum();
299.     derechap++;
300.     Serial.println(derechap);
301.     }

302.     void mv_izquierda()
303.     {
304.     Serial.println("IZQUIERDA DESDE PANTALLA");
305.     zum();
306.     izquierdap++;
307.     Serial.println(izquierdap);
308.     }

309.     void mv_atras()
310.     {
```

```

311. Serial.println("ATRAS DESDE PANTALLA");
312. zum();
313. atrasp++;
314. Serial.println(atrasp);

315. }

316. void mv_enter()
317. {
318. zum1();
319. Serial.println("ACEPTAR DATOS PANTALLA");
320. Serial.println("MOVIMIENTOS");
321. Serial.print("PASOS ADELANTE");
322. Serial.print(",");
323. Serial.println(adelantep);

324. Serial.print("PASOS DERECHA");
325. Serial.print(",");
326. Serial.println(derechap);

327. Serial.print("PASOS IZQUIERDA");
328. Serial.print(",");
329. Serial.println(izquierdap);

330. Serial.print("PASOS ATRAS");
331. Serial.print(",");
332. Serial.println(atrasp);

333. delay(t5);
334. if(adelantep>0){
335. while(1){
336. Serial.print("MOVIMIENTO ADELANTE");
337. digitalWrite(MD1,HIGH);
338. digitalWrite(MD2,HIGH);
339. delay(500);
340. digitalWrite(MD1,LOW);
341. digitalWrite(MD2,LOW);
342. Serial.println("#####");
343. adp++;
344. Serial.println(adp);

```

```

345.     Serial.println(adelantep);
346.     Serial.println("#####");
347.     delay(200);
      a.  if(adelantep==adp){

          b.  Serial.println("SALIR MOVIMIENTO ADELANTE");
          c.  adp=0;
          d.  break;
          e.  }
348.     }
349.     }
350.     delay(t5);

351.     if(derechap>0){
352.     while(1){
353.     Serial.print("MOVIMIENTO DERECHA");
354.     digitalWrite(MZ1,HIGH);
355.     digitalWrite(MD2,HIGH);
356.     delay(460);
357.     digitalWrite(MZ1,LOW);
358.     digitalWrite(MD2,LOW);
359.     Serial.println("#####");
360.     derp++;
361.     Serial.println(derp);
362.     Serial.println(derechap);
363.     Serial.println("#####");
364.     delay(200);
      a.  if(derechap==derp){

          b.  Serial.println("SALIR MOVIMIENTO DERECHA");
          c.  derp=0;
          d.  break;
          e.  }
365.     }
366.     }

367.     delay(t5);
368.     if(izquierdap>0){
369.     while(1){
370.     Serial.print("MOVIMIENTO IZQUIERDA");
371.     digitalWrite(MD1,HIGH);
372.     digitalWrite(MZ2,HIGH);

```

```

373.     delay(460);
374.     digitalWrite(MZ2,LOW);
375.     digitalWrite(MD1,LOW);
376.     Serial.println("#####");
377.     izqp++;
378.     Serial.println(izqp);
379.     Serial.println(izquierdap);
380.     Serial.println("#####");
381.     delay(200);
    a.  if(izquierdap==izqp){

    b.  Serial.println("SALIR MOVIMIENTO IZQUIERDA");
    c.  izqp=0;
    d.  break;
    e.  }
382.     }
383.     }

```

```

384.     delay(t5);
385.     if(atrasp>0){
386.     while(1){
387.     Serial.print("MOVIMIENTO ATRAS");
388.     digitalWrite(MZ1,HIGH);
389.     digitalWrite(MZ2,HIGH);
390.     delay(1000);
391.     digitalWrite(MZ1,LOW);
392.     digitalWrite(MZ2,LOW);
393.     Serial.println("#####");
394.     atrp++;
395.     Serial.println(atrp);
396.     Serial.println(atrasp);
397.     Serial.println("#####");
398.     delay(200);
    a.  if(atrasp==atrp){
    b.  Serial.println("SALIR MOVIMIENTO ATRAS");
    c.  atrp=0;
    d.  break;
    e.  }
399.     }
400.     }

```

```

401.     adelantep=0;
402.     derechap=0;

```

```

403.     izquierdap=0;
404.     atrasp=0;

405.     }

406.     //////////////////////////////////RTC RELOJ////////////////////////////////
407.     void printDate(DateTime date)
408.     {

409.         // Serial.print(date.year(), DEC);
410.         // Serial.print('/');
411.         // Serial.print(date.month(), DEC);
412.         // Serial.print('/');
413.         // Serial.print(date.day(), DEC);
414.         // Serial.print(" ");
415.         // Serial.print(daysOfTheWeek[date.dayOfTheWeek()]);
416.         // Serial.print(" ");

417.         // Serial.print(date.hour(), DEC);
418.         // Serial.print(':');
419.         // Serial.print(date.minute(), DEC);
420.         // Serial.print(':');
421.         // Serial.print(date.second(), DEC);
422.         // Serial.println();
423.         hr=(date.hour());
424.         mins=(date.minute());
425.         dia=(daysOfTheWeek[date.dayOfTheWeek()]);
426.         n1.setValue(hr);
427.         n2.setValue(mins);

428.     }

429.     //////////////////////////////////LECTURA RFID////////////////////////////////

430.     void altos(){

431.         if ( mfrc522.PICC_IsNewCardPresent())
432.         {

```

```

433.    //Seleccionamos una tarjeta
a.    if ( mfrc522.PICC_ReadCardSerial()
b.    {
c.    // Enviamos serialemente su UID
d.    Serial.print("Card UID:");
e.    Serial.println("MINA DETECTADA");
f.    alarma();
g.    digitalWrite(MD1,LOW);
h.    digitalWrite(MD2,LOW);
i.    digitalWrite(MZ1,LOW);
j.    digitalWrite(MZ2,LOW);

k.    for (byte i = 0; i < mfrc522.uid.size; i++) {
        i.    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        ii.   Serial.print(mfrc522.uid.uidByte[i], HEX);

l.    }
m.    Serial.println();
n.    // Terminamos la lectura de la tarjeta actual
o.    mfrc522.PICC_HaltA();

p.    }
434.    }

435.    }

436.    void zum(){
437.    for (int i = 0; i <= 4; i++) {
438.    digitalWrite(buzzer, HIGH);
439.    delay(20);
440.    digitalWrite(buzzer, LOW);
441.    delay(20);
442.    }
443.    }

444.    void zum1(){
445.    for (int i = 0; i <= 2; i++) {
446.    digitalWrite(buzzer, HIGH);
447.    delay(500);
448.    digitalWrite(buzzer, LOW);

```

```
449.     delay(50);
450.     }
451.     }

452.     void alarma(){
453.     tone(buzzer,Re5,semicorchea);
454.     delay(semicorchea+50);
455.     tone(buzzer,Do5,semicorchea);
456.     delay(semicorchea+50);
457.     tone(buzzer,Re5,semicorchea);
458.     delay(semicorchea+50);
459.     tone(buzzer,Do5,semicorchea);
460.     delay(semicorchea+50);

461.     tone(buzzer,Re5,semicorchea);
462.     delay(semicorchea+50);
463.     tone(buzzer,Do5,semicorchea);
464.     delay(semicorchea+50);
465.     tone(buzzer,Re5,semicorchea);
466.     delay(semicorchea+50);
467.     tone(buzzer,Do5,semicorchea);
468.     delay(semicorchea+50);

469.     tone(buzzer,Re5,semicorchea);
470.     delay(semicorchea+50);
471.     tone(buzzer,Do5,semicorchea);
472.     delay(semicorchea+50);
473.     tone(buzzer,Re5,semicorchea);
474.     delay(semicorchea+50);
475.     tone(buzzer,Do5,semicorchea);
476.     delay(semicorchea+50);

477.     tone(buzzer,Re5,semicorchea);
478.     delay(semicorchea+50);
479.     tone(buzzer,Do5,semicorchea);
480.     delay(semicorchea+50);
481.     tone(buzzer,Re5,semicorchea);
482.     delay(semicorchea+50);
483.     tone(buzzer,Do5,semicorchea);
484.     delay(semicorchea+50);
```

```
485.    tone(buzzer,Re5,semicorchea);  
486.    delay(semicorchea+50);  
487.    tone(buzzer,Do5,semicorchea);  
488.    delay(semicorchea+50);
```

```
489.    ///SILENCIO///
```

## Anexo IX: Encuesta



### ENCUESTA PARA DETERMINAR LOS FACTORES DE ÉXITO DE UNA APLICACIÓN DE SISTEMA EMBEBIDOS PARA ROBOTS MÓVILES

Gracias por dedicar unos minutos de su valioso tiempo a completar esta encuesta, la cual permitirá descubrir los factores de éxito para el desarrollo de una aplicación de sistemas embebidos en robots móviles. Esta encuesta está dirigida a los integrantes del club de robótica de la Universidad Técnica de Cotopaxi sede la matriz, guardando total confidencialidad a la información recabada, que será utilizada únicamente para mejorar la propuesta presente de tesis.

#### OBJETIVO

Identificar las experiencias en uso de aplicaciones de sistemas embebidos a través de una investigación especializado en el tema para un mejor entendimiento de los dispositivos electrónicos

#### INSTRUCCIONES:

Lea cuidadosamente cada una de las preguntas y marque con una x la opción escogida.

1. ¿Cree Ud. que se cuenta con el conocimiento para el funcionamiento y programación de un sistema embebido en una tarjeta de control en prototipos de robots móvil?

SI  NO

2. ¿Conoce UD. que la memoria RAM y ROM de un sistema embebido se utilizan para el grabado y despliegue de programas que se diseñen para control en prototipos?

SI  NO

3. ¿Conoce UD. que los sistemas embebidos trabajan con 5 voltios de corriente continua, en entrada y salida de señales programables que controlan prototipos robóticos?

SI  NO

4. ¿Utilizaría Ud. un sistema embebido que sea programable de forma amigable con un IDE y lenguaje fácil de codificar en aplicaciones de prototipos de robots móviles?

SI  NO

5. ¿Tiene Ud. conocimiento que se pueden construir objetos impresos en 3d relacionados con prototipos robóticos?

SI

NO

6. ¿Conoce Ud. que los robots móviles del tipo terrestre tienen grados de libertad en su movimiento?

SI

NO

7. ¿Ha utilizado Ud. sensores y actuadores de 5 voltios en sistemas embebidos que ayuden en la experimentación práctica con prototipos de robots móviles?

SI

NO

8. ¿Ha utilizado Ud. sistemas de control remoto en tarjetas de control con sistemas embebidos para prototipos de robots móviles?

SI

NO

**Anexo X: Manual de usuario**

**Manual de usuario**

Control por Computador

**Manual de usuario tarjeta programable “Brick”**



Sistemas de  
Información

Marco Antonio Sosa Valverde

Ingeniería en Sistemas



Universidad Técnica de Cotopaxi

## Manual de programación tarjeta electrónica BRICK

En esta manual se describen las características básicas de la tarjeta programable BRICK con sus principales consideraciones para realizar su programación.

### Características generales de la placa

Se trata de una placa open hardware con un diseño de libre distribución y utilización, que ofrece opciones de reproducción (Figura 1 aspecto de la placa).

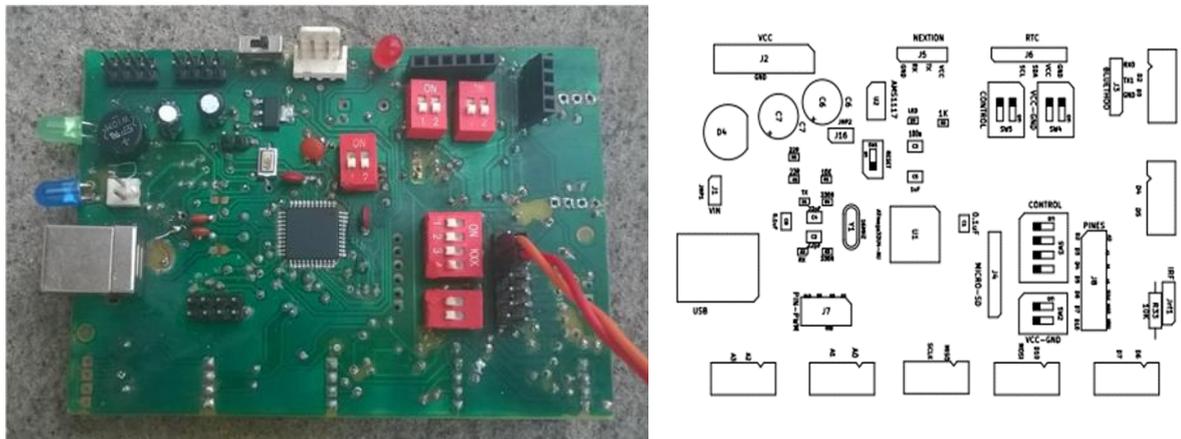


Figura 1. Aspecto de la tarjeta programable BRICK

La tarjeta programable BRICK hace uso en su programación del entorno IDE propio de Arduino, pudiendo la placa ser alimentada por medio del USB en conexión con la computadora, así como también de una fuente externa, después de cargado el programa; de ser preciso utilizar una fuente de alimentación externa, se la utilizará para que la tarjeta funcione en forma autónoma, o de ser el caso, el propio cable USB proporciona alimentación necesaria para la realización de algunos de prácticas, el voltaje de la fuente puede estar entre 6 y 25 Voltios.

### Entorno de desarrollo

Para programar la tarjeta y que tenga conexión con una computadora, BRICK necesita que se descargue el software libre de la página web de Arduino de entorno de desarrollo (IDE), la cual está disponible en <https://www.arduino.cc/en/main/software>, la cual ofrece versiones tanto para

Windows como para MAC y LINUX, que ofrecen la posibilidad de compilar la programación que el usuario realice. En la siguiente figura se muestra el aspecto del entorno de programación.

Para comenzar a trabajar con el entorno de desarrollo y la tarjeta programable BRICK es necesario configurar las comunicaciones entre la placa BRICK y el computador. Para ello se debe abrir en el menú de usuario de la interfaz del programa el menú herramientas en la opción serial port. En esta opción se selecciona el puerto serie al que se ha conectado la placa electrónica, en Windows, si el sistema desconoce el puerto al que está conectado la placa, se puede explorar a través del administrador de dispositivos (puertos COM).



Figura 2. Entorno de desarrollo

Conectado correctamente la tarjeta, y realizado el paso anterior, se comprueba que todo lo que se ha realizado hasta ahora está bien y familiarizarse con la interfaz de desarrollo IDE, para lo cual se procede a abrir uno de los ejemplos propios que el sistema ofrece a los usuarios, recomendando abrir el ejemplo blink, para lo cual se debe acceder a través de archivo, examples, basico, blink.

El ejemplo blink hace que la placa parpadee un LED que está colocado en el pin número 13 de la placa, comprobando tanto conexión con el computador, como operabilidad de la tarjeta, verificando que la tarjeta está trabajando correctamente.

Para subir el programa a la placa BRICK, lo primero que se comprueba es el código fuente y que sea correcto, para ello se pulsa en el menú de la interfaz del software el botón de verificación de código que tiene forma de una flecha, el cual deberá hacer aparecer un mensaje en la parte inferior de la interfaz done compiling, lo cual indica que no hay errores de codificación.

Una vez comprobado el código, se procede a grabarlo en la placa, para ello se pulsa en el menú de usuario de la interfaz el botón de subir el programa el cual comienza la carga del algoritmo realizado.

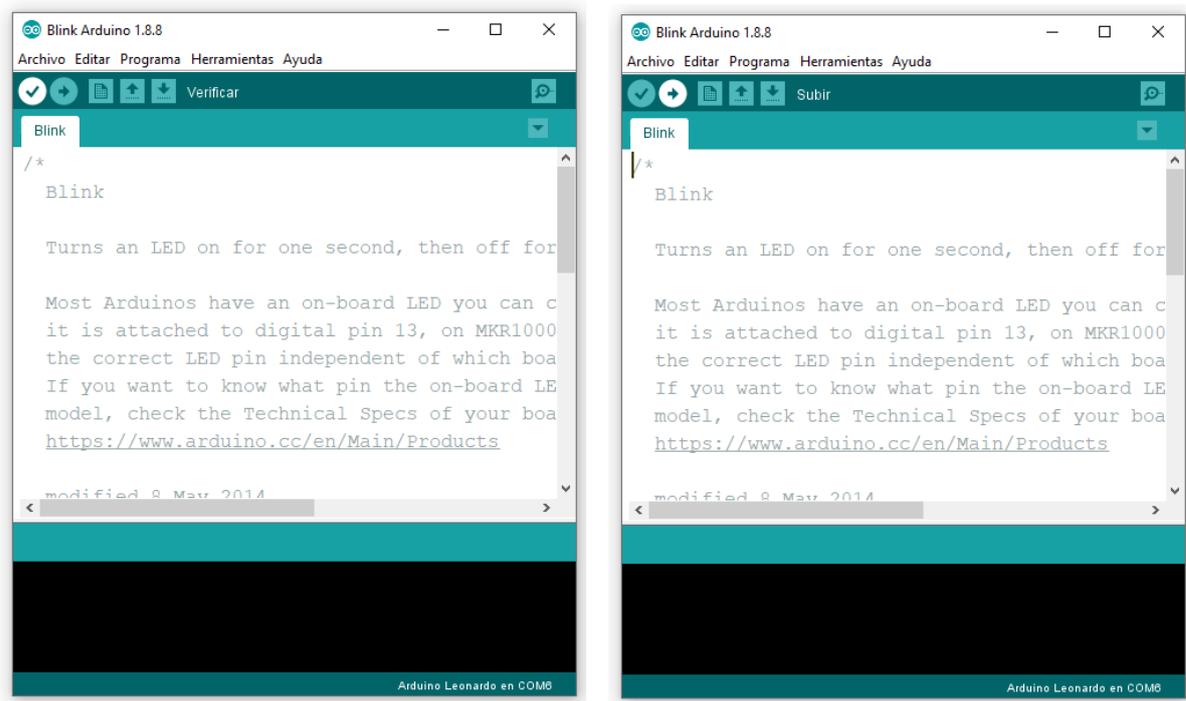


Figura 3. a) compilar programa b) transferencia del programa a la placa

Durante la carga del programa, en la placa conectada por el p rtico USB, este encender n los LED que indican que se est n enviando y recibiendo informaci n por el puerto serie TX/RX, el cual al final desplegar  el mensaje done uploading, para a continuaci n esperar unos 8 segundos aproximadamente para comprobar que todo ha salido bien, si la instrucci n enviada al led colocado en el pin 13 de la placa se enciende y se apaga cada segundo de forma r tmica, entonces el procedimiento con la tarjeta es correcto, teniendo todo listo para empezar a trabajar con la placa.

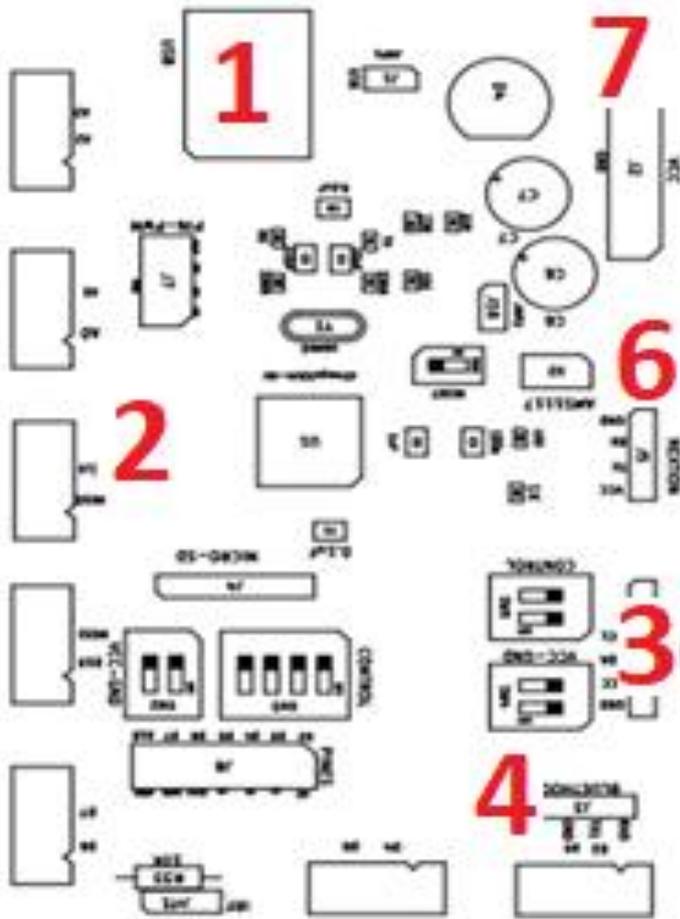
### Descripci n de BRICK

BRICK es un cerebro para un robot, el cual es un sistema embebido en una tarjeta programable micro controlada que proporciona, mediante la toma de decisiones, un comportamiento algorítmico e inteligente.

### **Partes de BRICK e interconexión**

Entre las partes de BRICK se encuentran conectores y pódico de conexión a la computadora, la cual se describe a continuación.

1. Puerto USB. Mediante la conexión a este puerto, se pueden grabar los programas desde el computador personal para actualizar algoritmos y datos desde el PC al robot.



2. Puertos de Salida. El BRICK tiene puertos de salida ubicados al extremo de la tarjeta, la cual posee en su conector a sus dos extremos alimentación positiva y negativa, en la parte central están los pines bidireccionales del microcontrolador Atmega32u4 para utilización con diferentes necesidades que el usuario disponga como lámparas o motores.

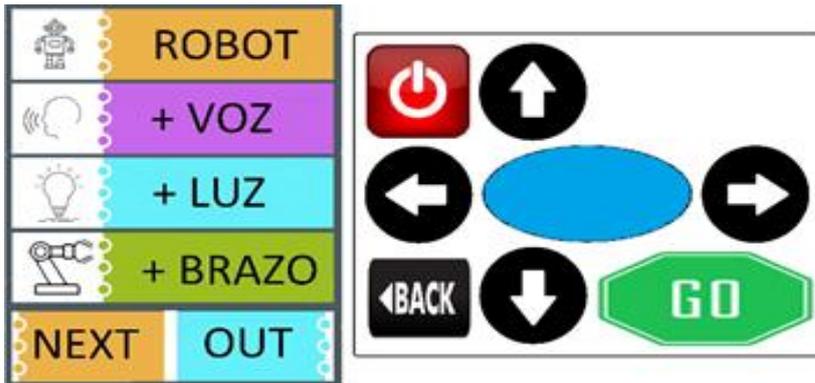
3. Módulo Bluetooth, el cual esta anexado a la tarjeta pudiendo ser removido si no se lo desea utilizar, este módulo da conexión inalámbrica a la tarjeta con dispositivos como un smartphone.

4. Módulo RTC para indicar fecha y hora de ser necesario por el usuario en sus proyectos.

5. Pantalla táctil, la cual obedece a las pulsaciones que el usuario realice, pudiendo ser programada para diferentes funciones, cambiando su interfaz a el gusto de quien la esté usando.

6. Swicht de encendido de la tarjeta programable BRICK.

7. Pines de alimentación de 5 voltios y 12 voltios. A los extremos de la tarjeta se encuentran salidas de voltaje para que el usuario pueda alimentar independientemente de la tarjeta con voltajes más altos al proyecto que esté desarrollando.

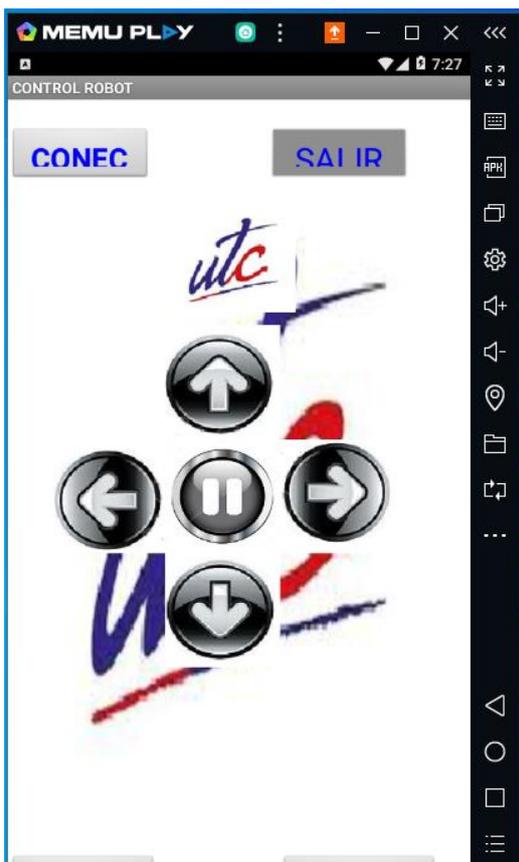


### Control táctil por pantalla

El sistema se ha provisto de un control intuitivo por pantalla el cual obedece a una programación visual de los movimientos, ubicando el robot en dirección de

arriba, abajo, derecha, izquierda, según el usuario comande en ordenes al robot, para posteriormente con un botón central que resume la ordenes, ejecute los movimientos en fin de un objetivo trazado.

El control de la pantalla táctil enviará órdenes directas a la tarjeta programable sin intermedio de ningún dispositivo externo.



### Aplicación en smartphone de control remoto vía bluetooth

La tarjeta inteligente además del control directo a través de la pantalla táctil, cuenta con un control remoto programable exterior, el cual como la pantalla táctil obedece a comandos básicos de dirección, mismos que al aplastar el botón central de entrada de comandos, hará que el robot vía bluetooth obedezca a la dirección programada por el usuario para los movimientos del robot a la meta. Para conectar el dispositivo, se ha de pulsar en conectar en la interfaz diseñada para el smartphone, mismo que previamente debe ser cargado en el celular, cuando se aplaste el botón conectar se ha de escoger la opción del bluetooth hc05 que corresponde al módulo integrado en la tarjeta programable de este

proyecto.

### Pasos rápidos para configurar la placa BRICK

1. En primer lugar, se instala el software IDE descargando desde la página oficial <https://www.arduino.cc/en/main/software>.
2. Instalar el software en su PC. Una vez descargado el software, se ejecuta el archivo .exe IDE que tiene la siguiente distribución: Crear el programa usando el editor de código y cargarlo en la placa. Para ello, es necesario después de instalar el IDE, conectar la placa BRICK al ordenador mediante un cable USB.
3. En el IDE, hay que seleccionar el tipo de placa que se está utilizando configurando desde Herramientas, menú de placas.
4. Por último, se comprueba el código con un ejemplo como Blink, exportando el ejemplo desde ejemplos básicos, Blink, para después con un clic en el icono de comprobación en la parte superior de la ventana del IDE, clic en el botón de compilar y por último grabación del código en la placa, poder trabajar con esta tarjeta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles": El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

<b>Nombres y apellidos:</b>	Ing. Mg. Hernán Rafael Yáñez Ávila.
<b>Título:</b>	Ing. Civil.
<b>Área académica de trabajo:</b>	Docente Universidad Técnica de Cotopaxi áreas de cálculo integral diferencial, álgebra lineal.
<b>Facultad:</b>	CIYA, Carrera de Eléctrica.
<b>Fecha de entrevista:</b>	13 de Junio del 2019.

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

**Firma:**



**Número de Cédula:** C.I. 0501300131

\* **NOTA:** Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles": El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

<b>Nombres y apellidos:</b>	Ing. Mg. Jefferson Alberto Porras Reyes.
<b>Título:</b>	Ing. Eléctrico.
<b>Área académica de trabajo:</b>	Docente Universidad Técnica de Cotopaxi, áreas de electricidad.
<b>Facultad:</b>	CIYA, Carrera de Electromecánica.
<b>Fecha de entrevista:</b>	12 de Junio del 2019.

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

**Firma:**



**Número de Cédula:** C.I. 0704400449

\* NOTA: Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

Tema de tesis: "Aplicación de sistemas embebidos en la construcción de robots móviles": El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

Nombres y apellidos:	Ing. Mg. Edwín Homero Moreano Martínez.
Título:	Ing. Electrónica e Instrumentación.
Área académica de trabajo:	Docente Universidad Técnica de Cotopaxi, área de electrónica de potencia.
Facultad:	CIYA, Carrera de Electromecánica.
Fecha de entrevista:	12 de Junio del 2019.

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

Firma:



Número de Cédula: C.I. 0502607500

\* NOTA: Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles": El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

<b>Nombres y apellidos:</b>	Ing. Mg. Luis Miguel Navarrete López
<b>Título:</b>	Ing. Mecánico.
<b>Área académica de trabajo:</b>	Docente Universidad Técnica de Cotopaxi, áreas de mecánica industrial.
<b>Facultad:</b>	CIYA, Carrera de Electromecánica.
<b>Fecha de entrevista:</b>	12 de Junio del 2019.

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

**Firma:**



**Número de Cédula:** C.I. 1803747284

\* NOTA: Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

**Perfil Coordinador**

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles"; El caso del club de robótica de la Universidad Técnica De Cotopaxi.

**Datos Personales**

<b>Nombres y apellidos:</b>	Ing. Mg. Jessica Nataly Castillo Fiallos
<b>Título:</b>	Ing. Electrónica.
<b>Área académica de trabajo:</b>	Docente Universidad Técnica de Cotopaxi, áreas de electrónica.
<b>Facultad:</b>	CIYA, Carrera de Electromecánica.
<b>Fecha de entrevista:</b>	13 de Junio del 2019.

**Preguntas**

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

<b>Firma:</b>	
---------------	--

<b>Número de Cédula:</b>	C.I. 0604590216
--------------------------	-----------------

\* NOTA: Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles": El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

<b>Nombres y apellidos:</b>	Ing. Mg. Luigi Orlando Freire Martínez
<b>Título:</b>	Ing. Electromecánica.
<b>Área académica de trabajo:</b>	Docente Universidad Técnica de Cotopaxi, áreas de automatización y electrónica.
<b>Facultad:</b>	CIYA, Carrera de Electromecánica.
<b>Fecha de entrevista:</b>	12 de Junio del 2019

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

**Firma:**



**Número de Cédula:** C.I. 0502529589

\* NOTA: Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles"; El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

<b>Nombres y apellidos:</b>	Ing. PhD. Secundino Marrero Ramírez
<b>Título:</b>	Ing. Electricidad
<b>Área académica de trabajo:</b>	Director de la Carrera de Eléctrica, Universidad Técnica de Cotopaxi, áreas de control y automatismos.
<b>Facultad:</b>	CIYA, Carrera de Eléctrica.
<b>Fecha de entrevista:</b>	13 de Junio del 2019

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

**Firma:**



**Número de Cédula:** C.I. 1757107907

\* NOTA: Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Entrevista dirigida al personal docente técnico de la facultad de CIYA de la Universidad Técnica de Cotopaxi.

**Agradecimiento:** El personal de esta propuesta reconoce el tiempo que usted apreciado docente brinda para la realización de esta entrevista. También se desea mencionar que los comentarios e información que proporcione serán muy valiosos para el proyecto de tesis a realizar.

#### Perfil Coordinador

**Tema de tesis:** "Aplicación de sistemas embebidos en la construcción de robots móviles": El caso del club de robótica de la Universidad Técnica De Cotopaxi.

#### Datos Personales

<b>Nombres y apellidos:</b>	Ing. Mg. Edwin Edison Quinatón Arequipa
<b>Título:</b>	Ing. Sistemas Computacionales
<b>Área académica de trabajo:</b>	Docente Universidad Técnica de Cotopaxi áreas de ingeniería de software, programación estructurada, análisis de programación orientada a objetos.
<b>Facultad:</b>	CIYA, Carrera de Sistemas.
<b>Fecha de entrevista:</b>	12 de Junio del 2019.

#### Preguntas

1. De acuerdo con su labor tanto preparatoria como profesional académica, ¿A tenido usted experiencia en el manejo de sistemas embebidos en su área de trabajo?
2. De acuerdo con su experiencia con los sistemas embebidos, sistemas programables utilizados para desarrollo y trabajos académicos afines, ¿Cuál sería su recomendación para su estudio, utilización, y mejor manera de funcionamiento?
3. ¿Conoce usted sobre el lenguaje C++ para sistemas embebidos y cuáles son las dificultades que presenta este lenguaje dirigido a microcontroladores?
4. De acuerdo con su conocimiento con sistemas embebidos, que al ser sistemas programables necesitan de esa área de conocimiento, ¿Cómo usted recomendaría que se oriente el usuario para programar estos sistemas, y en el caso de esta propuesta, para el funcionamiento de un robot?
5. De acuerdo con su conocimiento con sistemas embebidos programables, y que para su funcionamiento en sistemas como los de un robot, utiliza sensores y afines, ¿Cómo recomendaría la programación de sistemas embebidos en base a la codificación por sensores?

**Firma:**



**Número de Cédula:** C.I. 0562863372

\* **NOTA:** Los resultados de la entrevista realizada se tabulará en una síntesis a ser añadida en la presente propuesta.

Anexo XIV: Aval de revisión encuestas y entrevistas realizadas

**FACULTAD CIENCIAS DE LA INGENIERÍA Y APLICADAS**  
**CARRERA INGENIERÍA EN SISTEMAS DE INFORMACIÓN**  
**AVAL DE REVISION ENCUESTAS Y ENTREVISTAS**

En mi calidad de tutor del trabajo de investigación sobre el tema: “Aplicación de sistemas embebidos en la construcción de robots móviles”: el caso del club de robótica de la Universidad Técnica de Cotopaxi del señor Marco Antonio Sosa Valverde, estudiante egresado de la Carrera de Ingeniería en Sistemas de Información de la Facultad de Ciencias de la Ingeniería y Aplicadas, de la Universidad Técnica de Cotopaxi. CERTIFICO; Que he monitoreado el proceso de ejecución, revisión y aplicación de los datos obtenidos de las encuestas y entrevistas realizadas para la presente investigación, el mismo que reúne los requisitos suficientes de aplicabilidad con la consiguiente aprobación para su impresión dentro del esquema del documento. Por lo tanto, autorizo la presentación de estos datos tanto en su análisis dentro de la metodología empleada sustentada en el capítulo 3, como en su sustento físico a colocarlo en el capítulo mencionado y anexos de esta tesis.

**Tutor de Titulación**

**PhD. Gustavo Rodríguez**

**CC. 1757001357**