



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE GUAYAQUIL**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT INSECTO CON  
ARTICULACIONES THEO JANSEN CON FUNCIONAMIENTO AUTÓNOMO  
Y RADIO CONTROLADO”**

Trabajo de titulación previo a la obtención del  
Título de Ingeniera Electrónico

AUTOR: FÁTIMA ANDREA CHELA CRIOLLO

AUTOR: KARLA NICOLE MANCHENO CEVALLOS

TUTOR: ING. RAFAEL PÉREZ, MSC.

GUAYAQUIL – ECUADOR

2022

## CERTIFICADO DE RESPONSABILIDAD Y AUDITORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Fátima Andrea Chela Criollo con documento de identificación N° 0956083075 y Karla Nicole Mancheno Cevallos con documento de identificación N° 1805470695; manifestamos que:

Somos los autores y responsables del presente trabajo; y autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 01 de agosto de 2022

Atentamente,



---

Fátima Andrea Chela Criollo  
0956083075



---

Karla Nicole Mancheno Cevallos  
1805470695

## **CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Fátima Andrea Chela Criollo con documento de identificación No. 0956083075 y Karla Nicole Mancheno Cevallos con documento de identificación No. 1805470695, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Diseño E Implementación De Un Robot Insecto Con Articulaciones Theo Jansen Con Funcionamiento Autónomo Y Radio Controlado”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 09 de agosto del año 2022

Atentamente,



---

Fátima Andrea Chela Criollo  
0956083075



---

Karla Nicole Mancheno Cevallos  
1805470695

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Rafael Enrique Pérez Ordoñez con documento de identificación No. 0916275076, docente de la Universidad Politécnica Salesiana Sede Guayaquil, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN ROBOT INSECTO CON ARTICULACIONES THEO JANSEN CON FUNCIONAMIENTO AUTÓNOMO Y RADIO CONTROLADO, realizado por Fátima Andrea Chela Criollo con documento de identificación N° 0956083075 y por Karla Nicole Mancheno Cevallos con documento de identificación N° 1805470695, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 09 de agosto del año 2022

Atentamente,



---

Ing. Rafael Enrique Pérez Ordoñez, MSC

0916275076

## **DEDICATORIA**

Este proyecto de Titulación va dedicado en primer lugar a Dios, ya que gracias a él he logrado concluir mi carrera, a mis padres quienes han sido un pilar fundamental para poder culminar mi etapa profesional.

Agradezco a mi padre el Sr. Alberto Chela por ser mi fuente de motivación para superarme día a día, por sus consejos y esfuerzos para lograr este objetivo también a mi madre la Sra. Meri Criollo por brindarme su apoyo incondicional y siempre guiarme para hacer de mí una mejor persona.

**Fátima Andrea Chela Criollo**

## **DEDICATORIA**

El presente trabajo de titulación va dedicado a Dios por haberme dado una familia maravillosa y haber puesto en mi camino personas que me han ayudado a crecer en mi vida personal, estudiantil y laboral.

A mi madre por incentivarme todos los días a lograrlo y enseñarme el valor de la humildad, sacrificio y superación, también a mi padre que me ha permitido contar siempre con su apoyo incondicional y ha fomentado en mi la constancia y el trabajo duro para alcanzar las metas propuestas, la responsabilidad y la ética.

Por último y no menos importante a la Universidad Politécnica Salesiana sede Guayaquil por hacer posible este momento y contar con docentes que me brindaron su conocimiento, y en especial al tutor de este trabajo por su tiempo, paciencia y dedicación que hicieron posible que este se realice con éxito.

**Karla Nicole Mancheno Cevallos**

## **AGRADECIMIENTO**

Son muchas las personas que han contribuido al proceso de este trabajo. En primer lugar, quiero agradecer a mis padres y hermana las personas más importantes de mi vida y las que me dieron fuerzas, palabras de aliento para salir adelante y en ella, gracias a todas las personas que fueron participes de este gran proceso.

Particularmente agradezco al Master Rafael Pérez, docente y tutor, por su paciencia y enseñanzas, compañeros del aula de clases por la amistad y el apoyo durante estos 5 años.

**Fátima Andrea Chela Criollo**

## **AGRADECIMIENTO**

Agradezco a mi madre y a mi padre por su motivación a lo largo de este camino, por sus palabras de aliento y por brindarme en cada momento su comprensión, cariño, y sabiduría por haber creído siempre en mí, y fomentado el deseo de superación que han contribuido a la consecución de este logro.

A la Universidad Politécnica Salesiana sede Guayaquil por permitirme convertir en un profesional con criterio, conocimiento y ética, a mis docentes y compañeros durante estos cinco años que me brindaron su apoyo, amistad, y paciencia, en especial al tutor de este trabajo de titulación quien fue uno de mis primeros docentes que me brindo su conocimiento y me enseñó la importancia de practicar lo aprendido y saber aplicarlo para crecer como persona y profesional.

**Karla Nicole Mancheno Cevallos**



## RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO	TITULO A OBTENER	TEMA DEL PROYECTO
2022	CHELA CRIOLLO FÁTIMA ANDREA  MANCHENO CEVALLOS KARLA NICOLE	ING. RAFAEL PÉREZ, MSC	“INGENIERO ELECTRÓNICO”	DISEÑO E IMPLEMENTACIÓN DE UN ROBOT INSECTO CON ARTICULACIONES THEO JANSEN CON FUNCIONAMIENTO AUTÓNOMO Y RADIO CONTROLADO

Este proyecto de titulación está dirigido a la Universidad Politécnica Salesiana para la carrera de electrónica y automatización el cual consta de un robot insecto con articulaciones Theo Jansen con funcionamiento autónomo y radio controlado. Este robot tiene un mecanismo de desplazamiento mediante articulaciones Theo Jansen con 8 puntos de apoyo por cada lado lo cual le permite su desplazamiento y evita que este pueda volcarse o caerse por alguna circunstancia. Para el movimiento de las articulaciones se utiliza micromotores de 50:1 los que hacen girar el cigüeñal y sincronizan el movimiento de sus extremidades. Además, se utilizó plástico en los puntos de apoyo con el suelo para evitar deslizamiento al momento del desplazamiento del robot.

Este robot dispone de dos tipos de funcionamientos el autónomo y el radio controlado. Para el funcionamiento autónomo se utiliza dos sensores infrarrojos incorporados en la parte frontal y lateral derecha del robot, estos permiten medir la distancia desde el robot hacia los obstáculos existentes en el ambiente con

esta medida de distancia se incorpora un controlador PD el cual mantiene a una distancia de 25 del obstáculo que se encuentre a su derecha y de esta manera se desplaza por ambientes desconocidos evitando el chocarse. Para el funcionamiento en radio controlado se incorporo dos ESPCAM las cuales se encuentra instaladas en la parte frontal y lateral derecha del robot estas cámaras transmiten el video en tiempo real a un computador permitiendo ver los objetos que existen a su alrededor, para escoger la dirección de movimiento se tiene una aplicación móvil en la cual al presionar los pulsadores se mueve hacia adelante, atrás, izquierda y derecha según el usuario.

La aplicación utilizada para el manejo del robot está diseñada en APP Inventor, esta aplicación permite detectar el pulsador presionado por el usuario y enviar un comando hacia FireBase la cual almacena el valor, ya con el valor guardado la ESP32 del robot se conecta a FireBase y según el comando que haya recibido ejecuta una acción. Esta acción puede ser el cambio de modo de funcionamiento entre autónomo y radio controlado.

## ABSTRACT

YEAR	STUDENTS	DEGREE PROJECT TUTOR	DEGREE	TECHNICAL DEGREE PROJECT
2022	CHELA CRIOLLO FÁTIMA ANDREA  MANCHENO CEVALLOS KARLA NICOLE	ING. RAFAEL PÉREZ, MSC	“INGENIERO ELECTRÓNICO”	DESIGN AND IMPLEMENTATION OF A THEO JANSEN INSECT ROBOT WITH JOINTS WITH AUTONOMOUS OPERATION AND RADIO CONTROLLED

This degree project is aimed at the Salesian Polytechnic University for the electronics and automation career, which consists of an insect robot with Theo Jansen joints with autonomous and radio-controlled operation. This robot has a movement mechanism using Theo Jansen joints with 8 support points on each side which allows it to move and prevents it from tipping over or falling for any reason. For the movement of the joints, 50:1 micromotors are used, which rotate the crankshaft and synchronize the movement of its extremities. In addition, plastic was used at the support points with the ground to prevent slipping when the robot moved.

This robot has two types of operations: autonomous and radio controlled. For autonomous operation, two infrared sensors incorporated in the front and right side of the robot are used, these allow the distance from the robot to the existing obstacles in the environment to be measured. With this distance measurement, a PD controller is incorporated, which maintains a distance of 25 from the obstacle that is on his right and in this way he moves through unknown environments avoiding crashing. For radio-

controlled operation, two ESPCAMs were incorporated, which are installed on the front and right side of the robot. These cameras transmit the video in real time to a computer, allowing the objects that exist around it to be seen, to choose the direction of movement. movement there is a mobile application in which pressing the buttons moves forward, backward, left and right according to the user.

The application used to manage the robot is designed in APP Inventor, this application allows detecting the button pressed by the user and sending a command to FireBase which stores the value, once the value is saved, the ESP32 of the robot connects to FireBase and according to the command it has received, it executes an action. This action can be the change of operating mode between autonomous and radio controlled.

## ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUDITORÍA DEL TRABAJO DE TITULACIÓN .....	ii
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA.....	iii
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN .....	iv
DEDICATORIA .....	v
DEDICATORIA .....	vi
AGRADECIMIENTO .....	vii
AGRADECIMIENTO .....	viii
RESUMEN.....	ix
ABSTRACT.....	xi
ÍNDICE GENERAL .....	xiii
ÍNDICE DE FIGURAS.....	xvi
ÍNDICE DE TABLAS.....	xix
1. INTRODUCCIÓN.....	1
1.1. DESCRIPCIÓN DEL PROBLEMA.....	1
1.2. ANTECEDENTES .....	2
1.3. IMPORTANCIA Y ALCANCE .....	2
1.4. DELIMITACIÓN .....	2
1.4.1. DELIMITACIÓN TEMPORAL .....	2
1.4.2. DELIMITACIÓN ESPACIAL .....	2
1.4.3. DELIMITACIÓN ACADÉMICA.....	3
1.5. OBJETIVOS .....	3
1.5.1. OBJETIVO GENERAL.....	3
1.5.2. OBJETIVOS ESPECIFICOS .....	3

1.5.3.	METODOLOGÍA.....	4
1.5.4.	BENEFICIARIOS DE LA PROPUESTA E INTERVENCIÓN.....	4
2.	ESTADO DEL ARTE .....	5
2.1.	MECANISMO THEO JANSEN .....	5
2.2.	NUMEROS SAGRADOS.....	6
2.3.	ESP32 .....	7
2.4.	MOTOR REDUCTOR.....	7
2.5.	DRIVER-TB6612FNG.....	8
2.6.	ESP-CAM.....	9
2.7.	SENSOR SHARP .....	10
2.8.	APP INVENTOR.....	11
2.9.	PROTEUS .....	11
2.9.1.	ETAPAS-CONSTRUCCIÓN EQUIPO ELECTRÓNICO .....	12
2.10.	FIRE BASE.....	13
3.	DISEÑO E IMPLEMENTACIÓN DEL ROBOT.....	15
3.1	DISEÑO MECÁNICO.....	15
3.1.1	DISEÑO DE LA ESTRUCTURA THEO JANSEN.....	15
3.2	DISEÑO ELECTRÓNICO .....	22
3.2.1	DISEÑO PARA EL CONTROL DE MOTORES .....	22
3.2.2	SENSORES DE DISTANCIA .....	25
3.2.3	DISEÑO DEL SISTEMA DE CONEXIÓN DEL ESP32 CAM.....	26
3.2.4	ARMADO DEL SISTEMA ELECTRÓNICO .....	28
3.3	DESARROLLO DE LA APLICACIÓN .....	31
3.3.1	CREACIÓN DE LA BASE DE DATOS EN TIEMPO REAL .....	31
3.3.2	DISEÑO DEL INTERFAZ EN APP INVENTOR.....	35
3.3.3	PROGRAMACIÓN DE MODO MANUAL.....	41
3.3.4	PROGRAMACIÓN DEL MODO AUTOMÁTICO.....	47

4.	RESULTADOS .....	50
4.1	TIEMPO DE DESPLAZAMIENTO .....	50
4.2	ELABORACIÓN DE LA PROGRAMACIÓN EN ARDUINO .....	53
4.3	ELABORACIÓN DE LA PROGRAMACIÓN EN APP INVENTOR .....	54
4.4	CONFIGURACIÓN DE FIRE BASE.....	55
5.	CONCLUSIONES .....	56
6.	RECOMENDACIONES.....	57
7.	BIBLIOGRAFÍA.....	58
8.	ANEXOS.....	60
8.1	ANEXO A. PIEZAS DEL ROBOT .....	60
8.2	ANEXO B. PROGRAMA DE ARDUINO .....	64
8.3	ANEXO C CONFIGURACIÓN DE FIRE BASE.....	79
8.4	ANEXO D DISEÑO DE INTERFAZ APP INVENTOR.....	80
8.5	ANEXO E PROGRAMA DE APP INVENTOR .....	81

## ÍNDICE DE FIGURAS

Figura 1 THEO JANSEN – STRANBEEEST. (Volkskrant, 2015) .....	5
Figura 2 DIMENSIONES PARA PATAS THEO JANSEN.....	6
Figura 3 ESP32. (Guerra Carmenate, 2021).....	7
Figura 4 MOTOR REDUCTOR. (Electrostore, 2021) .....	8
Figura 5 TB6612FNG. (Moviltronics, 2018).....	9
Figura 6 ESP32-CAM. (Cdtecnologia, 2019).....	10
Figura 7 SENSOR SHARP. (Naylamp, 2021) .....	10
Figura 8 APP INVENTOR (Cprmerida, 2017).....	11
Figura 9 SOFTWARE PROTEUS (Hubor, 2020).....	12
Figura 10 5 ESTAPAS PARA CONSTRUCCIÓN DE UN EQUIPO ELECTRÓNICO-PROTEUS. (Hubor, 2020).....	12
Figura 11 UTILIZACIÓN DE PROTEUS EN LA CONSTRUCCIÓN DE EQUIPOS ELECTRÓNICOS. (Hubor, 2020).....	13
Figura 12 FIRE BASE. (Muradas, 2021) .....	14
Figura 13 PATAS THEO JANSEN. ....	15
Figura 14 PIE THEO JANSEN EN SOLID WORK.....	16
Figura 15 MUSLO THEO JANSEN EN SOLID WORK.....	17
Figura 16 LIGAMENTO THEO JANSEN EN SOLID WORK .....	17
Figura 17 PATAS THEO JANSEN IMPRESAS .....	18
Figura 18 CIGÜEÑAL THEO JANSEN.....	18
Figura 19 CIGÜEÑAL THEO JANSEN VISTA FRONTAL.....	19
Figura 20 LIGAMENTO INFERIOR .....	19
Figura 21 LIGAMENTO SUPERIOR .....	20
Figura 22 PATAS Y LIGAMENTOS ENSAMBLADOS .....	20
Figura 23 SOPORTE.....	21
Figura 24 ESTRUCTURA THEO JANSEN ENSAMBLADA .....	21
Figura 25 ESTRUCTURA THEO JANSEN COMPLETA .....	22
Figura 26 ESP32. (Taloselectronics, 2021).....	23
Figura 27 DRIVER TB6612FNG (Llamas , 2016).....	23
Figura 28 CONEXIÓN DEL DRIVER Y BATERÍA.....	24
Figura 29 CONEXIÓN DE MOTORES .....	25



Figura 30 CONEXIÓN DE SENSORES .....	26
Figura 31 CONEXIÓN DE ESP32 CAM .....	26
Figura 32 DIAGRAMA IMPLEMENTADO EN PROTEUS. ....	27
Figura 33 VISTA FRONTAL DE LA PLACA. ....	27
Figura 34 VISTA POSTERIOR DE LA PLACA.....	28
Figura 35 PLACA PCB .....	28
Figura 36 CAJA DE ACRÍLICO .....	29
Figura 37 INSTALACIÓN DE CÁMARA FRONTAL .....	29
Figura 38 INSTALACIÓN DE CÁMARA LATERAL .....	30
Figura 39 VISTA FRONTAL .....	30
Figura 40 VISTA POSTERIOR.....	31
Figura 41 PÁGINA OFICIAL FIRE BASE .....	32
Figura 42 INGRESO DE NOMBRE DEL PROYECTO.....	32
Figura 43 CREACIÓN DE BASE DE DATOS EN TIEMPO REAL.....	33
Figura 44 CONFIGURACIÓN DE LAS REGLAS DE LA BASE DE DATOS ....	33
Figura 45 AUTORIZACIÓN DE LA BASE DE DATOS .....	34
Figura 46 HOST DE LA BASE DE DATOS .....	34
Figura 47 INTERFAZ APP INVENTOR .....	35
Figura 48 LOGOTIPO Y NOMBRES DE INTEGRANTES.....	36
Figura 49 PULSADOR DE ACTIVACIÓN/DESACTIVACIÓN DEL CONTROL AUTOMÁTICO .....	36
Figura 50 CONEXIÓN DE ESP32 CAM .....	36
Figura 51 AÑADIR FIREBASE A APP INVENTOR .....	37
Figura 52 CONFIGURACIÓN DE LA AUTORIZACIÓN Y EL HOST.....	37
Figura 53 SENTENCIA CLIC DEL PULSADOR ARRIBA.....	38
Figura 54 PROGRAMACIÓN DEL PULSADOR ARRIBA .....	38
Figura 55 PROGRAMACIÓN DE LOS PULSADORES AL PRESIONAR .....	39
Figura 56 PROGRAMACIÓN DE LOS PULSADORES AL SOLTAR .....	40
Figura 57 PROGRAMACIÓN DEL PULSADOR DE MODO AUTOMÁTICO ....	40
Figura 58 CONFIGURACIÓN DE LIBRERÍAS Y CLAVES DE ACCESO.....	41
Figura 59 CONFIGURACIÓN DE PINES DEL ESP32.....	42
Figura 60 CONFIGURACIÓN DE LA PWM.....	42
Figura 61 CONFIGURACIÓN DE PUERTOS DE SALIDA.....	43
Figura 62 CONFIGURACIÓN DE CANALES DE SALIDA DE PWM.....	43

Figura 63 ALGORITMO PARA INGRESAR A LA RED WIFI .....	44
Figura 64 ALGORITMO PARA INGRESAR A FIRE BASE .....	44
Figura 65 ALGORITMO PARA IR AL FRENTE .....	45
Figura 66 ALGORITMO PARA IR ATRÁS .....	45
Figura 67 ALGORITMO PARA IR A LA IZQUIERDA .....	46
Figura 68 ALGORITMO PARA IR A LA DERECHA .....	46
Figura 69 ALGORITMO PARA FRENAR .....	47
Figura 70 ALGORITMO PARA MEDIR DISTANCIA CON SENSORES.....	48
Figura 71 ALGORITMO PD.....	48
Figura 72 ALGORITMO DE GIRO.....	49
Figura 73 TIEMPO PROMEDIO EN CADA PRUEBA .....	52
Figura 74 VELOCIDAD PROMEDIO EN CADA PRUEBA .....	52
Figura 75 ROBOT ENSAMBLADO.....	53
Figura 76 ARDUINO IDE.....	53
Figura 77 INTERFAZ CON EL USUARIO .....	54
Figura 78 VARIABLES EN FIRE BASE.....	55
Figura 79 PIE THEO JANSEN EN SOLID WORK.....	60
Figura 80 MUSLO THEO JANSEN EN SOLID WORK.....	60
Figura 81 LIGAMENTO THEO JANSEN EN SOLID WORK .....	61
Figura 82 CIGÜEÑAL THEO JANSEN.....	61
Figura 83 LIGAMENTO INFERIOR .....	62
Figura 84 LIGAMENTO SUPERIOR .....	62
Figura 85 SOPORTE.....	63

## ÍNDICE DE TABLAS

Tabla 1. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT .....	50
Tabla 2. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT .....	50
Tabla 3. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT .....	51
Tabla 4. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT .....	51
Tabla 5. TIEMPOS Y VELOCIDADES PROMEDIO DE LAS PRUEBAS .....	51

## **1. INTRODUCCIÓN**

Este proyecto tiene como principal objetivo el diseño e implementación de un robot insecto con articulaciones Theo Jansen con funcionamiento autónomo y radio controlado mediante el uso de una tarjeta ESP32. De esta manera se crea un robot que pueda ser utilizado para las diferentes asignaturas de la carrera, además de poder ser reprogramado para cumplir con otras funciones.

Este proyecto está basado en dos modos de funcionamiento. El primer modo es el automático, en el cual el robot mediante sus sensores infrarrojos mide la distancia hacia una pared y con esta información usa un control PD para desplazarse dentro de un ambiente desconocido para intentar salir del mismo, para esto se usa el algoritmo de la mano derecha. El segundo modo de funcionamiento es el modo manual, en este el robot recibe la información de la aplicación móvil y realiza movimientos hacia la izquierda, derecha, frente o atrás dependiendo del usuario. Además, tiene dos cámaras incorporadas la que le permite ver su entorno en tiempo real.

### **1.1. DESCRIPCIÓN DEL PROBLEMA**

En la última década el desarrollo tecnológico-científico ha ido evolucionando con el transcurso del tiempo, surgiendo nuevas tecnologías que despuntan en la robótica con un fuerte impacto en la sociedad hacia sistemas autónomos y controlados que parcialmente van cambiando y a su vez van generando impacto en el ámbito estudiantil.

Actualmente los estudiantes de la Carrera de Electrónica y Automatización no cuentan con robots con el mecanismo de Theo Jansen para lo cual la implementación de este tipo de robots les será de mucha utilidad en sus prácticas con controladores PD, comunicación WiFi, diseño de aplicaciones móviles para robótica, algoritmos de desplazamiento y manejo de sensores.

## **1.2. ANTECEDENTES**

El estudio de la robótica en nuestro país no se ha desarrollado con rapidez, esto se debe a la falta de conocimientos, tecnología o personal capacitado. Es por esto que la robótica sigue siendo un tema complejo de tratar dentro del ámbito estudiantil y profesional ya que el desconocimiento del tema provoca que muchas personas piensen que es imposible que se desarrollen estos prototipos en el País.

## **1.3. IMPORTANCIA Y ALCANCE**

El presente proyecto tiene como finalidad llamar la atención y curiosidad hacia los estudiantes con el cual les permita tener conocimientos sólidos de sistemas autónomos, radios controlados que también conlleva el uso de controladores PD actualmente muy utilizados en la carrera de ingeniería y en el diario vivir.

A su vez busca facilitar a la Carrera de Ingeniería Electrónica y Automatización interactuar y diseñar nuevas funcionalidades, permitiendo expandir conocimientos, el cambio en la tecnología seguirá constante, motivo por el cual se deberá fomentar e impulsar la innovación de robots entre los estudiantes de la Universidad Politécnica Salesiana.

## **1.4. DELIMITACIÓN**

### **1.4.1. DELIMITACIÓN TEMPORAL**

El proyecto técnico se diseñó y desarrolló en el período noviembre 2021 hasta junio 2022.

### **1.4.2. DELIMITACIÓN ESPACIAL**

Proyecto técnico fue desarrollado para la Carrera de Electrónica y Automatización de la Universidad Politécnica Salesiana sede Guayaquil.

### **1.4.3. DELIMITACIÓN ACADÉMICA**

A través del diseño y desarrollo de este proyecto se aplicarán varios conocimientos adquiridos a lo largo de la Carrera de Ingeniería Electrónica y Automatización orientado a las materias de informática industrial, teoría de control y programación.

## **1.5. OBJETIVOS**

### **1.5.1. OBJETIVO GENERAL**

Diseñar e implementar un robot insecto con articulaciones Theo Jansen con funcionamiento autónomo y radio controlado mediante el uso de una tarjeta ESP32 para la exploración de lugares desconocidos.

### **1.5.2. OBJETIVOS ESPECIFICOS**

- Diseñar y construir una estructura y soporte de las articulaciones del robot.
- Evaluar el comportamiento de las articulaciones utilizando la ESP32.
- Diseñar e implementar una tarjeta electrónica para el control de entradas y salidas del robot insecto.
- Programar el robot insecto mediante el uso de Arduino IDE y APP inventor para el funcionamiento en modo autónomo y radio controlado.
- Verificar el funcionamiento autónomo y radio controlado del robot insecto en un ambiente preparado.

### **1.5.3. METODOLOGÍA**

Se usa la metodología experimental ya que se implementará un nuevo prototipo de robot insecto con articulaciones Theo Jansen mediante el uso de ESP32. Los estudiantes podrán manipular, configurar, diseñar y comparar redes definidas por software de manera virtual obteniendo resultados positivos y acertados. Estas pruebas fueron desarrolladas en diferentes ambientes con el fin de demostrar el funcionamiento del algoritmo PD y el enlace de comunicación entre el robot y el dispositivo móvil consiguiendo resultados de las prácticas llevando a cabo los conocimientos técnicos y teóricos aprendidos a lo largo de la vida universitaria.

### **1.5.4. BENEFICIARIOS DE LA PROPUESTA E INTERVENCIÓN**

El presente trabajo va a beneficiar a todos los alumnos de la carrera de ingeniería electrónica ellos son los principales beneficiarios del diseño y construcción de este proyecto.

## 2. ESTADO DEL ARTE

### 2.1. MECANISMO THEO JANSEN

El mecanismo denominado Theo Jansen, llamado así por su creador y diseñador, es la parte mecánica de su obra cinética realizada en Holanda denominada “Strandbeest” Significa animal de playa, y se trata de una estructura capaz de acumular la energía del viento para andar. (Onieva & Martinez , 2019)

El artista holandés comenzó a crear programas de simulación algorítmica de vida artificial, su interés por diseñar organismos vivos y autónomos a través de software le llevo a iniciar su serie de esculturas cinéticas, el proyecto que le ha proporcionado un reconocimiento a nivel internacional. Entre otros galardones, Jansen ha recibido el premio especial del jurado en Ars Electrónica 2005. (Onieva & Martinez , 2019)

Sus primeras creaciones fueron “UFO”, una aeronave con forma de platillo volante; y su “máquina de pintar”, un robot que traza grafitis sobre una pared; los cuales mostraron su habilidad para aplicar los conocimientos de ingeniería diferentes proyectos artísticos. (Onieva & Martinez , 2019)



**Figura 1**  
**THEO JANSEN – STRANBEEEST. (Volkskrant, 2015)**



## 2.2. NUMEROS SAGRADOS

Theo Jansen cálculo hace 25 años con un ordenador Atari los 13 números sagrados, el propósito de esto era calcular los 100 mejores tipos de dinámica de marcha para ordenador. (Onieva & Martinez , 2019)

El objetivo de los números sagrados es asemejarse al caminar biológico, este mecanismo consiste en una variedad de diferentes varillas con longitudes distintas, más la longitud de la manivela con el punto fijo a una distancia  $m$  de acuerdo con la tabla de los números sagrados. (Onieva & Martinez , 2019)

Jansen's Linkage es el sistema de trabajo de articulaciones utilizada por Strandbeest. (Shunsuke , Mohan, & Masami , 2013)

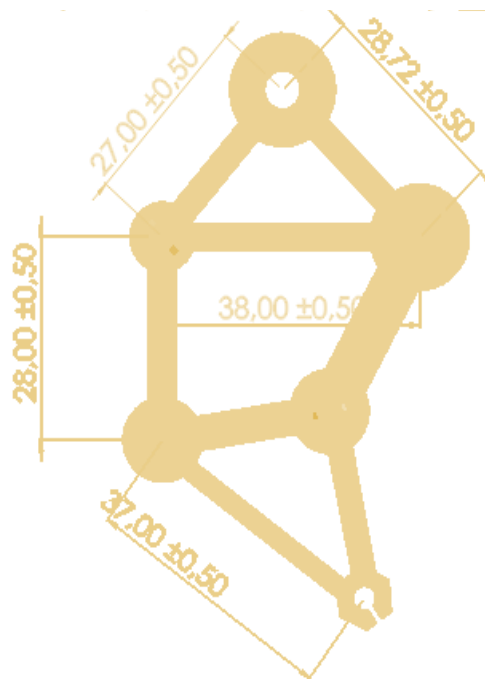


Figura 2  
DIMENSIONES PARA PATAS THEO JANSEN.

### 2.3. ESP32

ESP32 es una placa de desarrollo la cual tiene un chip superior, en capacidades, ya que supera a un ARDUINO UNO y a un ESP8266. Este tiene conectividad Wifi siendo compatible con 802.11 b/g/n En la banda de los 2.4 Hz alcanzando velocidades de hasta 150 Mbits/s. también incluye comunicación Bluetooth compatible con Bluetooth versión 4.2 y Bluetooth Low Energy (BLE). (Guerra Carmenate, 2021)

El bloque de radio está estrechamente ligado a los módulos de comunicación inalámbricos. De hecho, este es el que realmente transmite y recibe la información. (Guerra Carmenate, 2021)

Es decir, toma los datos digitales provenientes de los módulos Wifi y Bluetooth, y los convierte en señales electromagnéticas que viajan por el aire para comunicarse con un teléfono móvil o un router. (Guerra Carmenate, 2021)



**Figura 3**  
**ESP32. (Guerra Carmenate, 2021)**

### 2.4. MOTOR REDUCTOR

Este motor reductor de corriente continua es un motor cepillado de 6 V de alta potencia en miniatura con una caja de engranajes de metal, el cual

tiene una sección transversal de 10 x 12mm, y el eje de salida de la caja de cambios en forma D tiene 9mm de largo y 3mm de diámetro. (Pololu, 2017)

Estos pequeños motorreductores de corriente continua están disponibles en una amplia gama de relaciones de engranajes, desde 5:1 hasta 1000:1 los cuales tienen diferentes motores. (Pololu, 2017)

Los motores HPCB de 6V y 12 V ofrecen el mismo rendimiento en sus respectivos voltajes nominales, solo el motor de 12V consume la mitad de la corriente que el motor de 6V. (Pololu, 2017)



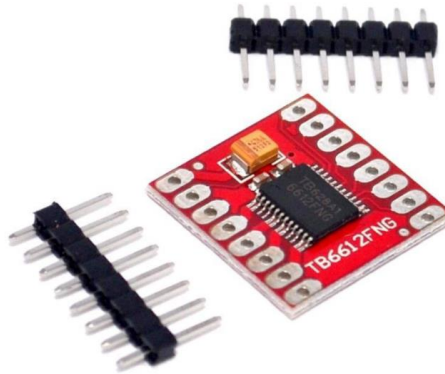
**Figura 4**  
**MOTOR REDUCTOR. (Electrostore, 2021)**

## **2.5. DRIVER-TB6612FNG**

El controlador del motor TB6612FNG puede controlar hasta dos motores de CC a una corriente constante de 1.2 A (pico de 3.2 A). Se pueden usar dos señales de entrada (IN1 e IN2) para controlar el motor en 1 de los cuatro nuevos de función: CW, CCW, freno corto y parada. Las dos salidas del motor (A y B) se pueden controlar por separado, la velocidad de cada motor se controla a través de una señal de entrada PWM con una frecuencia de hasta 100 kHz. El pin STBY debe colocarse alto para sacar el motor de modo de espera. (Moviltronics, 2018)

El módulo también permite controlar un motor paso a paso unipolar o bipolar, el módulo permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de microcontroladores y tarjetas de

desarrollo como Arduino, Raspberry Pi y Launchpads de Texas Instruments. (Moviltronics, 2018)



**Figura 5**  
**TB6612FNG. (Moviltronics, 2018)**

## **2.6. ESP-CAM**

Toda la potencia de la plataforma ESP32 junto con sus capacidades inalámbricas de Wifi, sumando a la Cámara OV2640, le permiten hacer streaming de video e imágenes y servirlos a la red creando un servidor local en el mismo chip. (Cdtecnologia, 2019)

Entre las principales aplicaciones del ESP32-CAM se tiene: Cámara IP de videovigilancia, controlador con Cámara para transmitir imágenes de tu robot móvil, o como sensor para un sistema de visión por computadora básico. (Cdtecnologia, 2019)

Su formato de DIP permite su fácil y rápida integración en cualquier aplicación y montaje en protoboard. Es importante mencionar que a mayor resolución tiene menor cantidad de cuadros por segundo transmitido (FPS). (Cdtecnologia, 2019)

La plataforma ESP32 permite el desarrollo de aplicaciones en diferentes lenguajes de programación, frameworks librerías y recursos diversos. Los más comunes a elegir son: Arduino (en lenguaje C++), dentro del entorno Arduino se podrá utilizar un lenguaje de programación conocido y hacer

uso de un IDE sencillo de utilizar, además de toda la información sobre proyectos y librerías disponibles en internet. (Cdtecnologia, 2019)

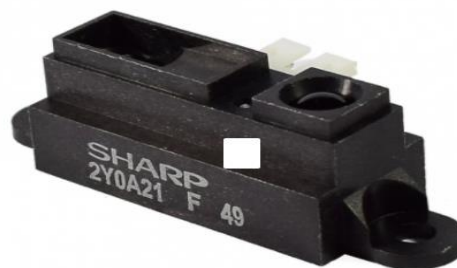


**Figura 6**  
**ESP32-CAM. (Cdtecnologia, 2019)**

## **2.7. SENSOR SHARP**

El sensor Sharp es un sensor óptico capaz de medir la distancia entre él y un objeto (cercano o aproximado), para esto el sensor con la ayuda de un emisor infrarrojo y un receptor miden la distancia usando triangulación. (Naylamp, 2021)

El método de triangulación consiste en medir uno de los ángulos que forma el triángulo emisor-objeto-receptor, el Receptor es un PSD (Position Sensitive Detector) que detecta el punto de incidencia el cual depende del ángulo y a su vez de la distancia del objeto. (Naylamp, 2021)



**Figura 7**  
**SENSOR SHARP. (Naylamp, 2021)**

## 2.8. APP INVENTOR

App Inventor Es un entorno de desarrollo de software creado por Google para la elaboración de aplicaciones destinadas al sistema operativo de Android. El lenguaje es gratuito y se puede acceder fácilmente a la web. (EducationPlus, 2018)

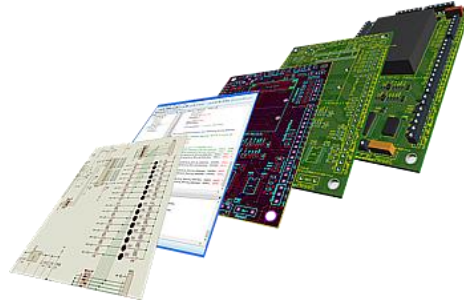
Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil. (EducationPlus, 2018)



Figura 8  
APP INVENTOR (Cprmerida, 2017)

## 2.9. PROTEUS

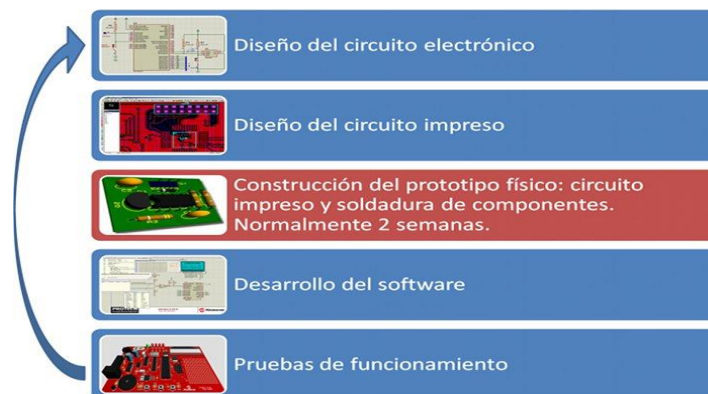
Proteus es una aplicación para la ejecución de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño del esquema electrónico, programación del software, construcción de la placa del circuito impreso, simulación de todo el conjunto, depuración de errores, documentación y construcción. (Hubor, 2020)



**Figura 9**  
SOFTWARE PROTEUS (Hubor, 2020)

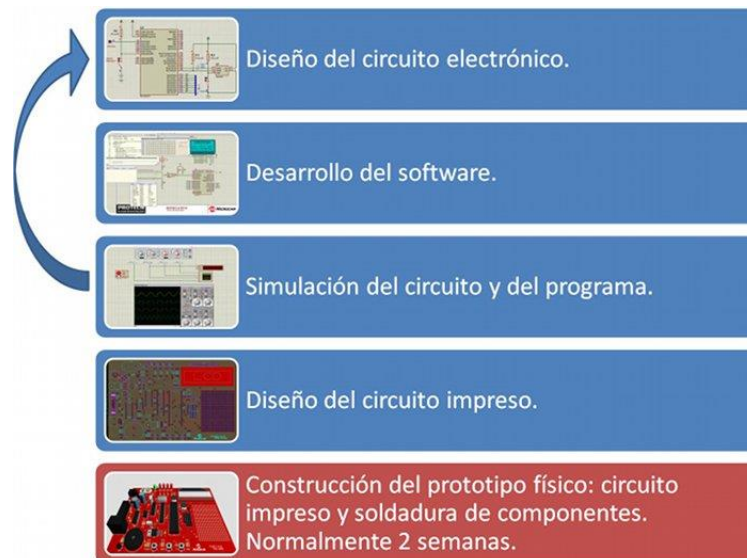
### 2.9.1. ETAPAS-CONSTRUCCIÓN EQUIPO ELECTRÓNICO

El proceso para construir un equipo electrónico basado en un microprocesador se compone de cinco etapas. Sólo al final del proceso somos capaces de detectar los errores y cualquier problema exige volver a ejecutar el ciclo completo: (Hubor, 2020)



**Figura 10**  
5 ESTAPAS PARA CONSTRUCCIÓN DE UN EQUIPO ELECTRÓNICO-PROTEUS. (Hubor, 2020)

El depurado de errores puede convertirse en una labor ardua en tiempo y recursos, lo que conlleva un alto coste económico. Sin embargo, con la herramienta Proteus el proceso queda definido de la siguiente manera:



**Figura 11**  
**UTILIZACIÓN DE PROTEUS EN LA CONSTRUCCIÓN DE EQUIPOS ELECTRÓNICOS. (Hubor, 2020)**

Los diferentes módulos que componen Proteus se pueden adquirir de forma independiente añadiendo nuevas funcionalidades a medida que aumentan nuestras necesidades de desarrollo y producción. Además, la capacidad de simular cada una de las familias de microprocesadores también es objeto de adquisición por separado. De esta manera se puede empezar adquiriendo unas funcionalidades básicas e ir adquiriendo progresivamente nuevas características aprovechando al máximo nuestras inversiones en la herramienta y asegurar al máximo los costes de inversión en el software. (Hubor, 2020)

## 2.10. FIRE BASE

Firestore se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero. (Pérez Cardona, 2021)

La plataforma está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web. Contiene diversas funciones para



que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades. (Pérez Cardona, 2021)

Firebase ayuda a compilar y ejecutar apps exitosas, el cual también cuenta con la confianza de los videojuegos y las apps más grandes del mundo.



**Figura 12**  
**FIRE BASE. (Muradas, 2021)**

### 3. DISEÑO E IMPLEMENTACIÓN DEL ROBOT

#### 3.1 DISEÑO MECÁNICO

El diseño mecánico en la construcción de un robot es una de las partes fundamentales, este permitirá que el robot pueda desplazarse y realizar movimientos. Existen algunos tipos de estructuras mecánicas según la aplicación de cada robot, terreno donde vaya a desplazarse o la tarea que realiza.

En el presente proyecto se plantea la utilización de una estructura mecánica llamada Theo Jansen, la que permite asemejarse al modo de desplazamiento de algunos insectos. Este tipo de estructuras se las utiliza por su gran ventaja para desplazarse en terrenos arenosos o abruptos, lo que para los robots con ruedas es de gran dificultad su desplazamiento.

##### 3.1.1 DISEÑO DE LA ESTRUCTURA THEO JANSEN

Para el diseño de las patas de la estructura Theo Jansen se consideró como medidas iniciales los números sagrados y las dimensiones planteadas en la siguiente ilustración como una referencia para el diseño de las diferentes piezas.

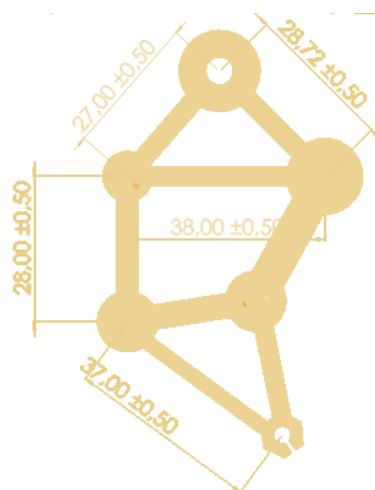
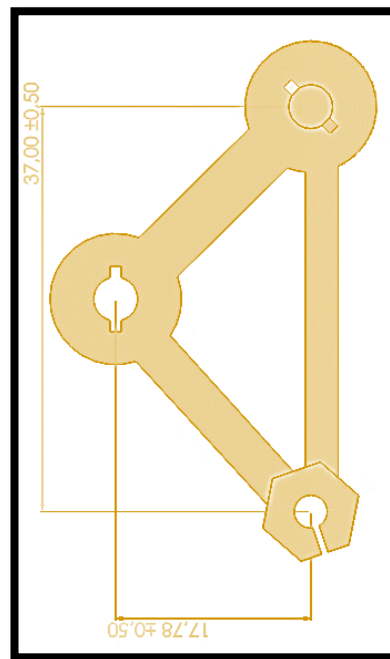


Figura 13  
PATAS THEO JANSEN.

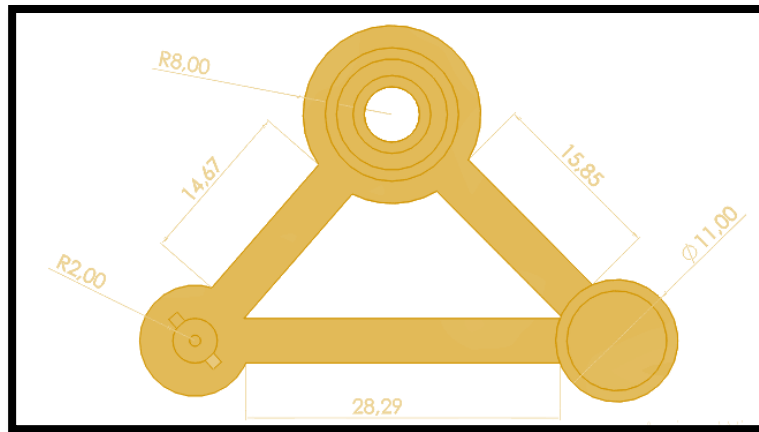
Con las dimensiones y la investigación realizada sobre el funcionamiento del sistema Theo Jansen, se creó el diseño de cada una de las piezas que conforman las patas en el software SolidWork.

Como primera pieza se realizó el diseño del pie, esta es la pieza que está en contacto con el suelo, por eso en uno de sus extremos se tiene un surco en el cual se pegará un elemento de plástico blando para evitar que se resbale al contacto con el piso. En los otros dos extremos se unirán ligamentos que permitirán la trasmisión del movimiento desde el muslo.



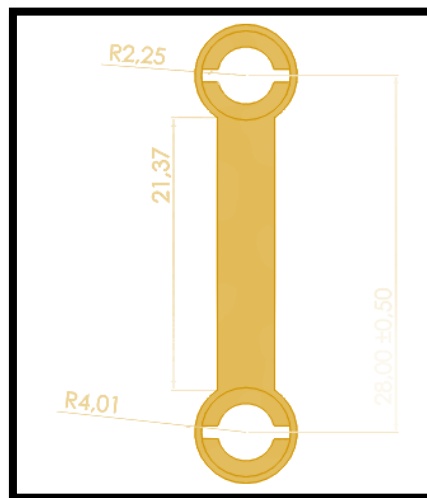
**Figura 14**  
**PIE THEO JANSEN EN SOLID WORK**

Después se realizó el diseño del muslo esta es la pieza que transmite el movimiento desde el eje principal llamado cigüeñal hacia el pie mediante ligamentos. En la siguiente figura se muestra el muslo del robot.



**Figura 15**  
**MUSLO THEO JANSEN EN SOLID WORK**

Al tener el pie y el muslo se realizó el diseño de los ligamentos, estos servirán para unir el muslo al pie. Este permitirá el movimiento hacia arriba y abajo del pie. En la siguiente figura se muestra el diseño del ligamento.



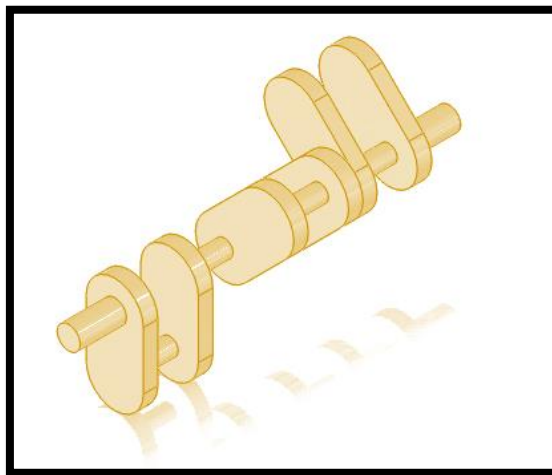
**Figura 16**  
**LIGAMENTO THEO JANSEN EN SOLID WORK**

Ya diseñado el muslo, el pie y el ligamento en 3D se procede a imprimir en plástico PLA para su posterior ensamblaje como se muestra a continuación.



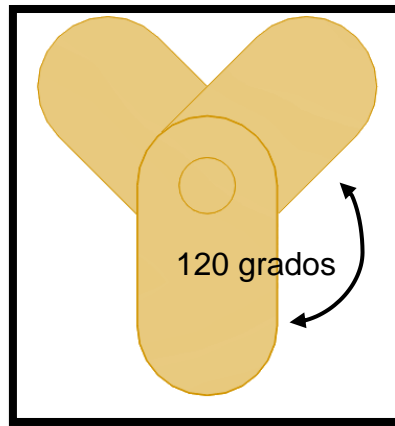
**Figura 17**  
**PATAS THEO JANSEN IMPRESAS**

Como siguiente paso se realizó el diseño del cigüeñal, este elemento está encargado de transmitir el movimiento de su eje hacia los cuatro pares de patas. Además, permite la sincronización de las patas para mantenerse estables mientras se desplaza el robot. A continuación, se muestra el cigüeñal implementado en el software SolidWork.



**Figura 18**  
**CIGÜEÑAL THEO JANSEN**

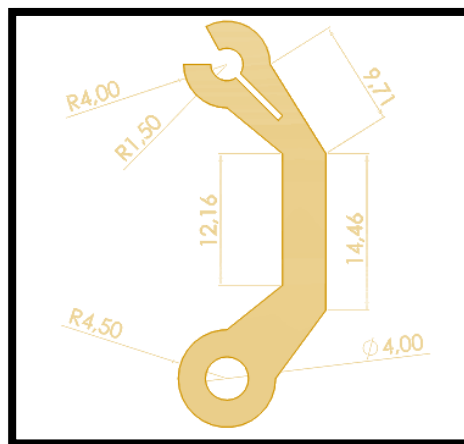
En la siguiente figura se muestra la vista frontal del cigüeñal el cual permite ver el desfase de 120 grados existentes entre cada una de las patas, esto permitirá la sincronización al momento de moverse.



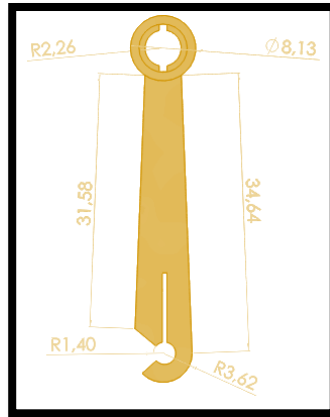
**Figura 19**  
**CIGÜEÑAL THEO JANSEN VISTA FRONTAL**

Después se diseña el ligamento superior e inferior que unirá a la pata con el cigüeñal, estos elementos transmitirán de forma directa el movimiento del cigüeñal sincronizando cada una de las patas para su desplazamiento.

A continuación, se muestra el ligamento superior e inferior en SolidWork.

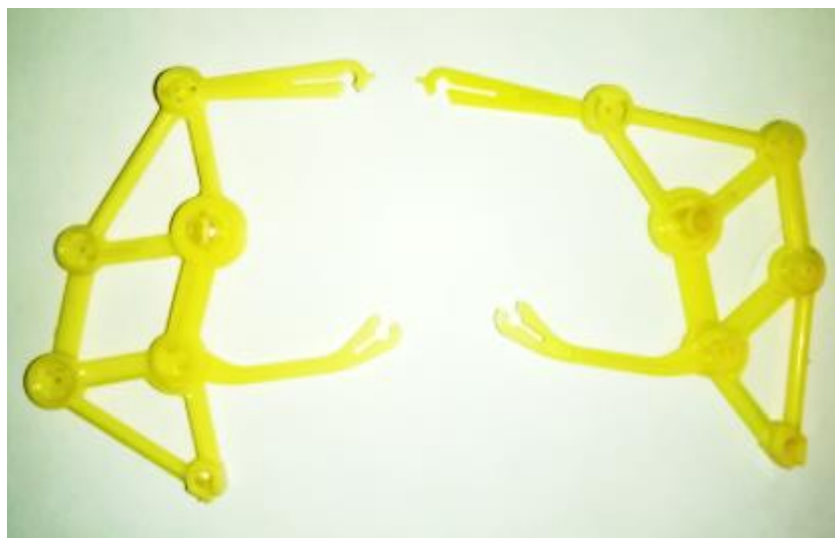


**Figura 20**  
**LIGAMENTO INFERIOR**



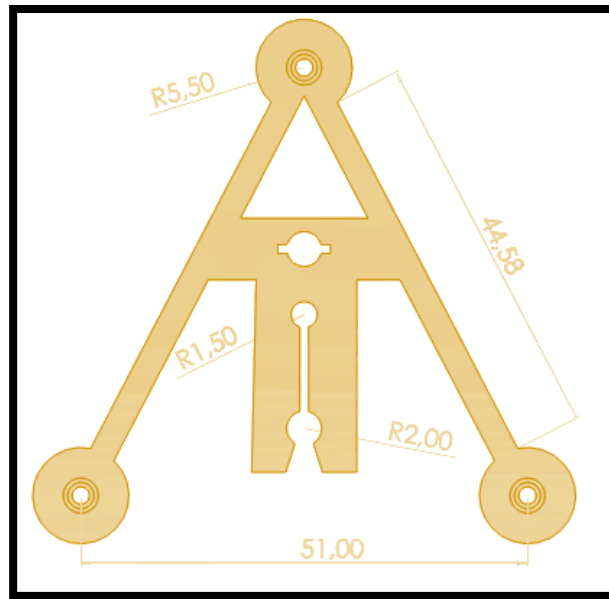
**Figura 21**  
**LIGAMENTO SUPERIOR**

Con los diseños ya creados se imprimió en 3D y realizó el armado de dos patas y sus ligamentos para la unión al cigüeñal. A continuación, se muestra una imagen de su armado.



**Figura 22**  
**PATAS Y LIGAMENTOS ENSAMBLADOS**

Para la unión de otro par de patas y mantener el cigüeñal en forma horizontal se diseña un soporte triangular. Este soporte tendrá 3 ejes metálicos que atraviesan para dar mayor estabilidad a las patas y el cigüeñal. En la siguiente figura se muestra el diseño del soporte en el cual se puede observar 3 perforaciones en cada extremo para los ejes metálicos y una perforación de un diámetro de 2mm para la unión con el cigüeñal.



**Figura 23**  
**SOPORTE**

Con todas las piezas implementadas se realiza el armado de los pies, patas, muslos, ligamentos, soporte y cigüeñal para obtener la estructura Theo Jansen que servirá como mecanismo de movimiento para el robot insecto. A continuación, se muestra la figura de la estructura ensamblada.

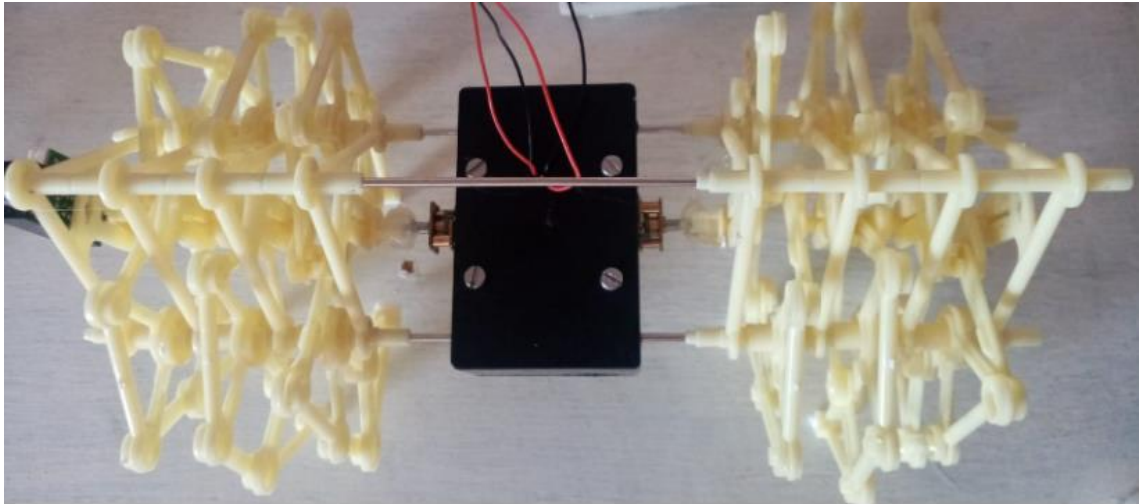


**Figura 24**  
**ESTRUCTURA THEO JANSEN ENSAMBLADA**

Finalmente, con la estructura Theo Jansen ensamblada, se procede a colocar la caja de soporte de motores la cual permitirá la transición de



movimiento desde los motores DC hacia el cigüeñal de la estructura Theo Jansen. En la siguiente figura se muestra la estructura final armada.



**Figura 25**  
**ESTRUCTURA THEO JANSEN COMPLETA**

### **3.2 DISEÑO ELECTRÓNICO**

El diseño electrónico de un robot permite dar movimiento a la estructura mecánica mediante la activación y control de motores de corriente continua. Además, el sistema electrónico tiene las conexiones necesarias para el funcionamiento de los sensores y cámaras lo que permite que el robot pueda interactuar con su entorno.

Dentro del diseño electrónico se tiene el diseño de la placa de control de los motores, sistema de sensores y la conexión de la ESP32 CAM.

#### **3.2.1 DISEÑO PARA EL CONTROL DE MOTORES**

Dentro del diseño de sistema electrónico se considera el uso de una ESP32 como microcontrolador principal el cual maneja los sensores y actuadores incorporados al robot. En la siguiente figura se muestra la ESP32 con sus pines.

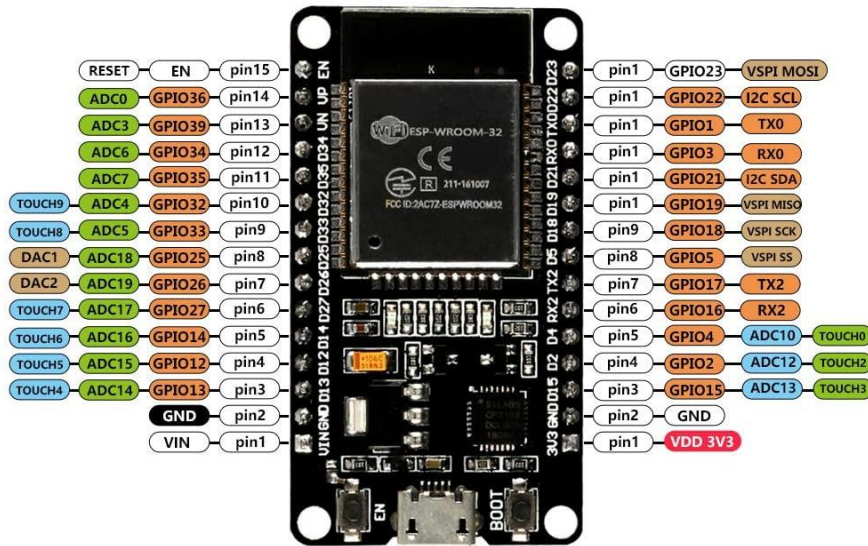
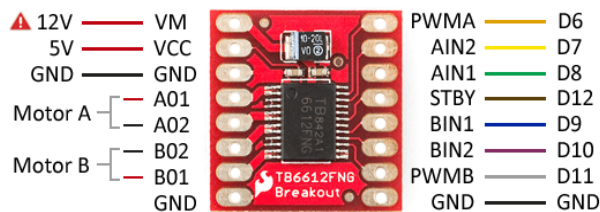


Figura 26  
ESP32. (Taloselectronics, 2021)

El driver TB6612FNG es el encargado de transmitir las señales de control desde la ESP32 al motor. Este driver controla la velocidad del motor según el robot requiera. A continuación, se muestra el driver TB6612FNG y sus pines de conexión.



⚠ Vin entre 5 y 15V. Al usar alimentación externa SIEMPRE poner con GND común.

Figura 27  
DRIVER TB6612FNG (Llamas , 2016)

Utilizando la imagen anterior se detallará las conexiones realizadas del driver hacia la ESP32.

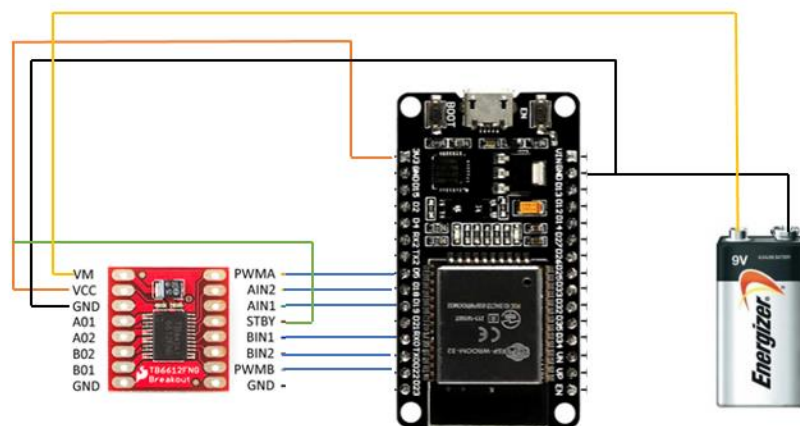
En el primer pin llamado VM se conecta los 9V que provienen desde la batería ya que este voltaje encenderá los motores de corriente continua. En el Pin VCC se debe conectar los 5V lógicos este voltaje será tomado desde el pin VIN de la ESP32. En el TB6612FNG existen 3 pines de GND

estos van a ir conectados al GND de la batería y de la ESP32. Otro de los pines necesarios para el funcionamiento es el llamado STBY (standby) el cual al tener un voltaje de 5V este activa a toda la placa para entrar en funcionamiento, es por esto que este pin se conecta a 5V de forma permanente para que el driver se encuentre en funcionamiento siempre.

Para la conexión de un motor se tiene tres pines el PWMA, AIN1 y AIN2 estos pines permiten el control de velocidad de los motores. El pin PWMA se conecta al pin 5 de la ESP32 el cual genera una PWM a una frecuencia de 5000Hz, este pin permite el control de velocidad del motor. Los pines restantes AIN1 se conecta al pin 19 y el AIN2 al pin 18 estos pines tienen la función de seleccionar el sentido de giro del motor, lo que permitirá avanzar o retroceder al robot.

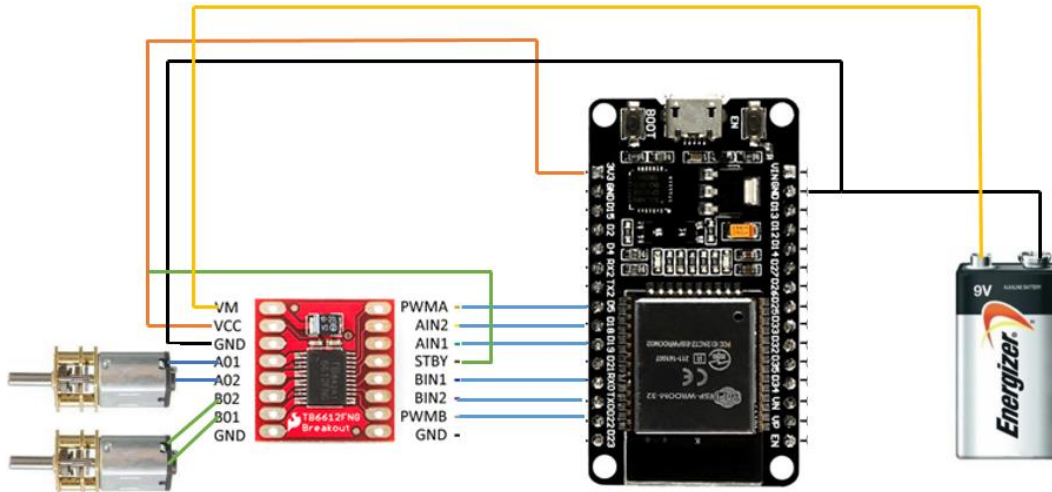
Para el segundo motor el PWMB se conecta al pin 17, BIN1 al pin 16 y BIN2 al pin 4.

A continuación, se presenta el diagrama de conexiones de la ESP32 y el driver.



**Figura 28**  
**CONEXIÓN DEL DRIVER Y BATERÍA**

Finalmente se realiza la conexión del primer motor a los pines A01 y A02. El segundo motor se conecta a los pines B01 y B02. En la siguiente figura se presenta la conexión de los motores hacia el driver.



**Figura 29**  
**CONEXIÓN DE MOTORES**

### 3.2.2 sensores de distancia

Dentro del sistema sensorial del robot se utilizará un sensor Sharp el cual sirve para medir la distancia del robot hacia los obstáculos, permitiéndole detectar la distancia a la que se encuentra del obstáculo y con estos datos se implementa un algoritmo para su desplazamiento de forma autónoma.

El sensor Sharp emite una luz infrarroja para la medición de distancia hacia el objeto. Dentro del robot se incorporan 2 sensores de distancia uno que se encuentra en su lado derecho el cual mide la distancia hacia la pared u objeto. El otro que se encuentra en la parte frontal el cual permite la medición de distancia hacia objeto que se encuentre en el frente para evitar chocarse.

Para la conexión de estos sensores se tiene tres pines VCC, GND y A0. Para la conexión de estos pines se conecta a VCC a 5V, GND de los sensores a GND del robot y A0 a una entrada analógica del ESP32 en este caso se usarán el pin 12 y 14 del ESP32.

En la siguiente figura se muestra la conexión de los sensores.

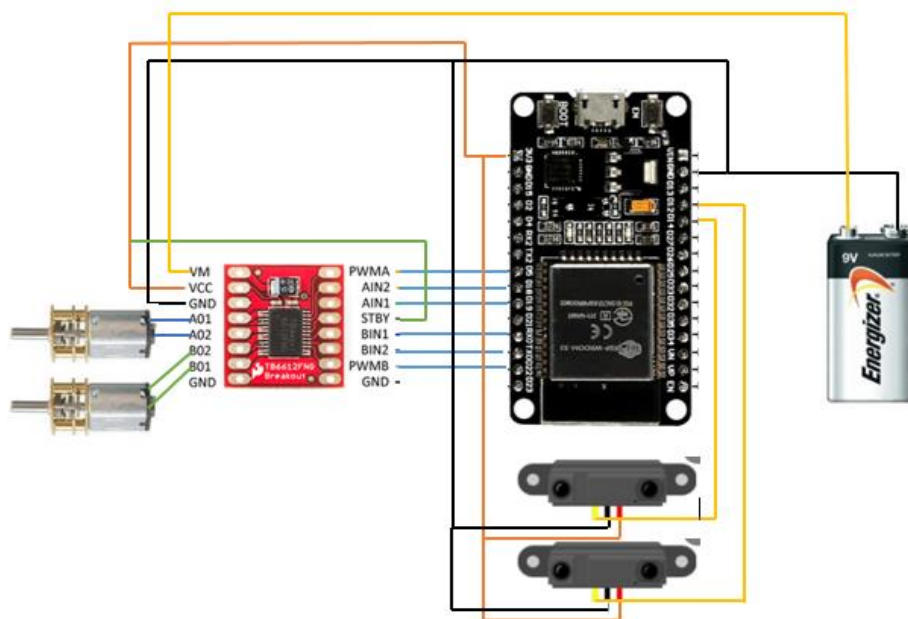


Figura 30  
CONEXIÓN DE SENSORES

### 3.2.3 DISEÑO DEL SISTEMA DE CONEXIÓN DEL ESP32 CAM

La activación de la ESP32 CAM se realiza con los pines de VCC y GND los cuales encienden al dispositivo. Esta cámara funcionará para el modo manual del robot ya que permite que el usuario pueda ver el camino que el robot está siguiendo. A continuación, se presenta la conexión de la ESP32 CAM.

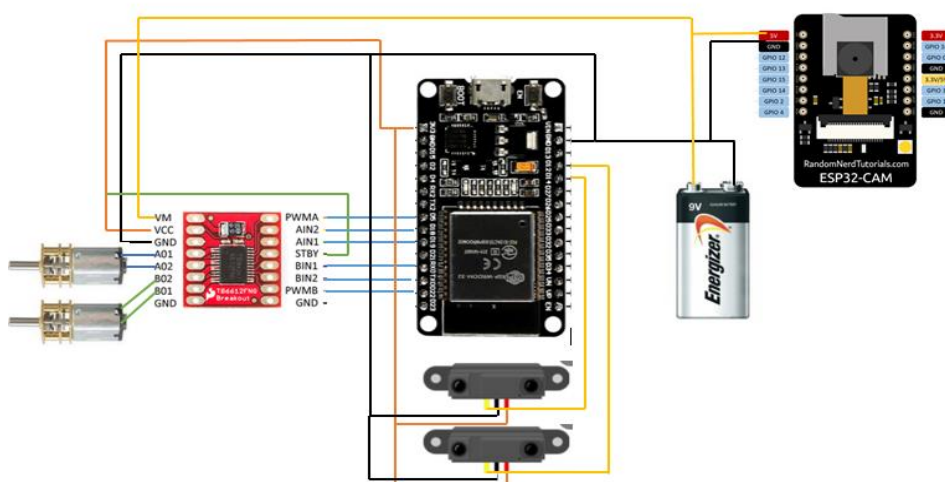
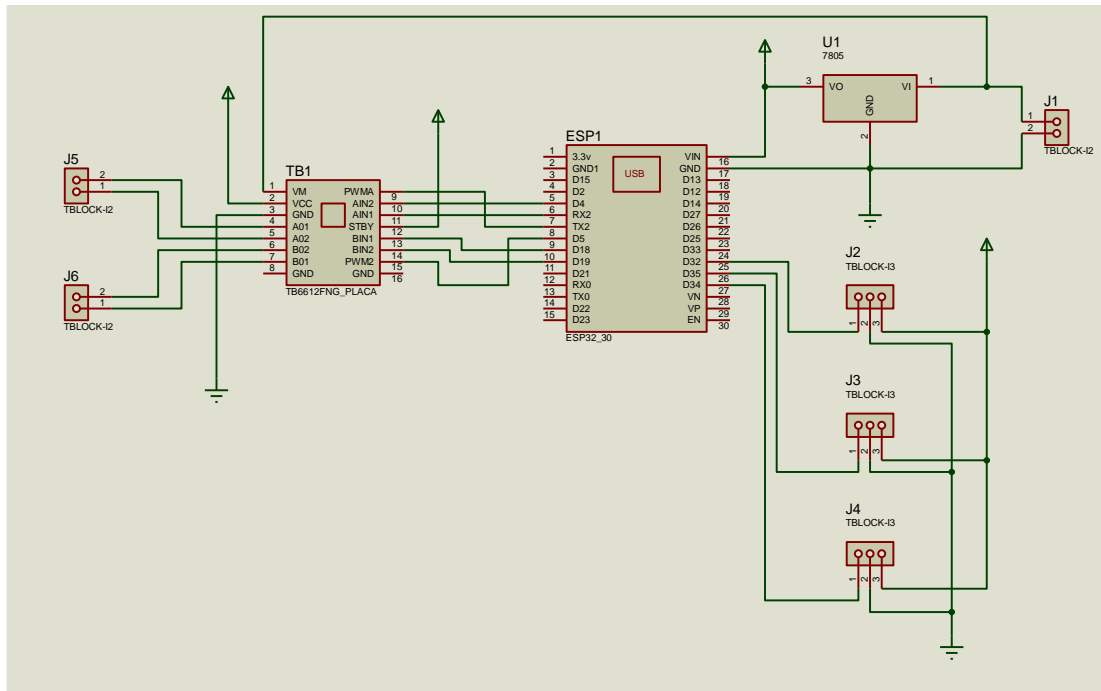


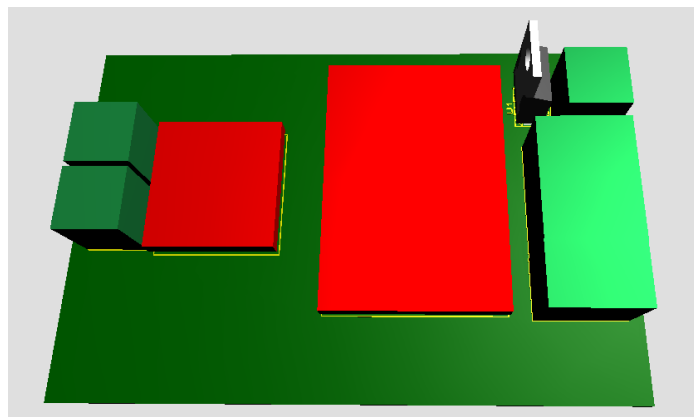
Figura 31  
CONEXIÓN DE ESP32 CAM

Con el circuito electrónico ya diseñado se crea el diagrama en el Software Proteus para su posterior implementación. A continuación, se muestra el diagrama electrónico implementado.

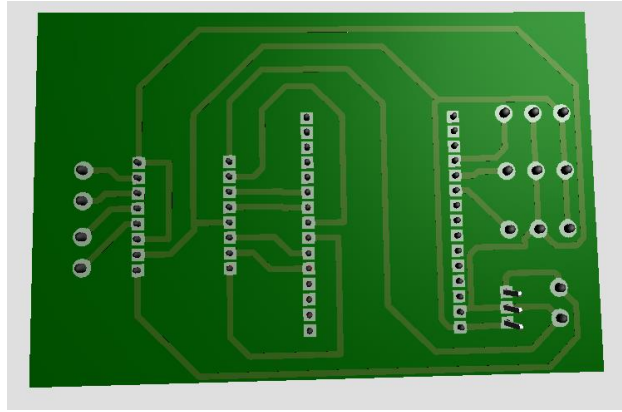


**Figura 32**  
**DIAGRAMA IMPLEMENTADO EN PROTEUS.**

Después se crea la simulación de la placa en 3D para identificar la posición de los elementos. En la siguiente figura se muestra el modelo 3D de la placa.



**Figura 33**  
**VISTA FRONTAL DE LA PLACA.**



**Figura 34**  
**VISTA POSTERIOR DE LA PLACA.**

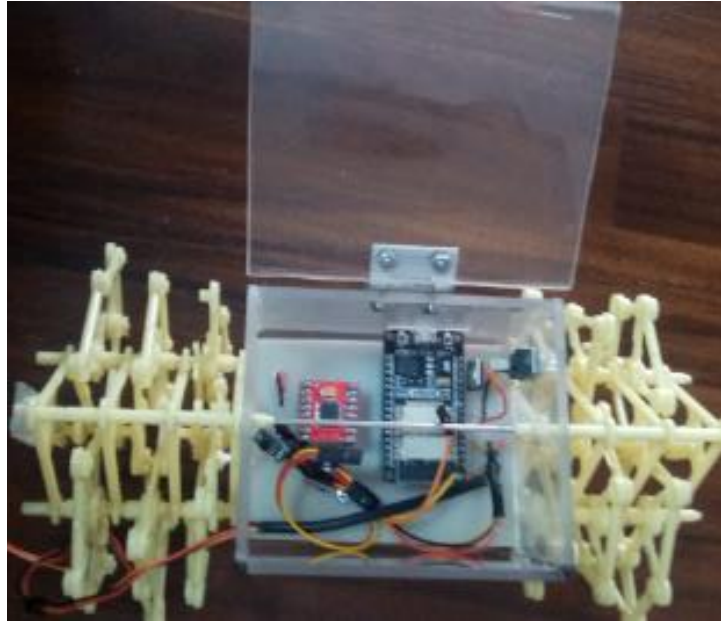
### **3.2.4 ARMADO DEL SISTEMA ELECTRÓNICO**

Con el circuito diseñado se realizó la implementación de la PCB la cual permite el control de los sensores y actuadores en la siguiente imagen se muestra una foto de la placa creada.



**Figura 35**  
**PLACA PCB**

La placa PCB se monta sobre la caja de soporte de los motores, y posterior a eso se arma una caja en acrílico la que permite proteger los circuitos electrónicos de la intemperie como se muestra en la siguiente imagen.



**Figura 36**  
**CAJA DE ACRÍLICO**

Ya instalada la caja de acrílico se realiza las perforaciones para la sujeción de la cámara en la parte frontal del robot como se muestra en la siguiente figura.



**Figura 37**  
**INSTALACIÓN DE CÁMARA FRONTAL**

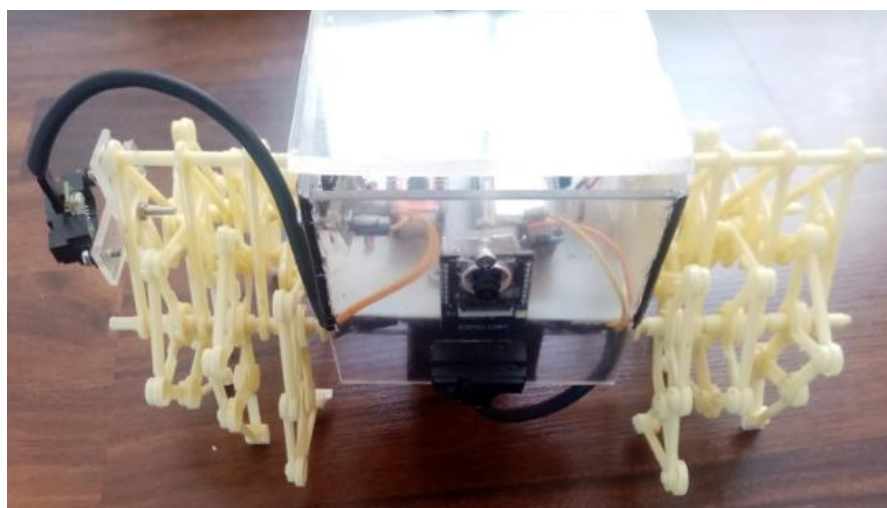


Instalada la cámara frontal se realiza el armado de la cámara en su parte lateral derecha la que permitirá la visualización de los objetos. En la siguiente figura se muestra cómo se realiza el ajuste de la cámara

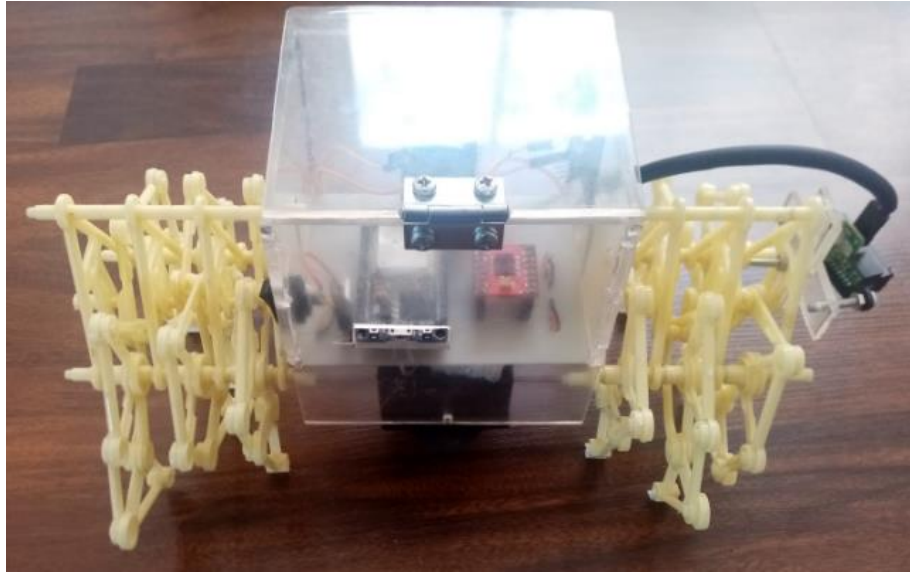


**Figura 38**  
**INSTALACIÓN DE CÁMARA LATERAL**

Finalmente se realiza la instalación de los sensores infrarrojos que permiten la detección de distancia desde el robot a los objetos en su entorno. En la siguiente figura se muestra la posición de los sensores en el robot.



**Figura 39**  
**VISTA FRONTAL**



**Figura 40**  
**VISTA POSTERIOR**

### **3.3 DESARROLLO DE LA APLICACIÓN**

En el desarrollo del software se incorpora como primera parte un interfaz para el usuario el cual permita controlar al robot de forma inalámbrica en el modo manual y mediante la cámara esté podrá ver los obstáculos que existen para su desplazamiento.

Otra parte de la aplicación será la implementación del modo automático en el cual mediante los sensores el robot mide la distancia hacia los obstáculos y se desplaza con el algoritmo de la mano derecha mediante un control PD.

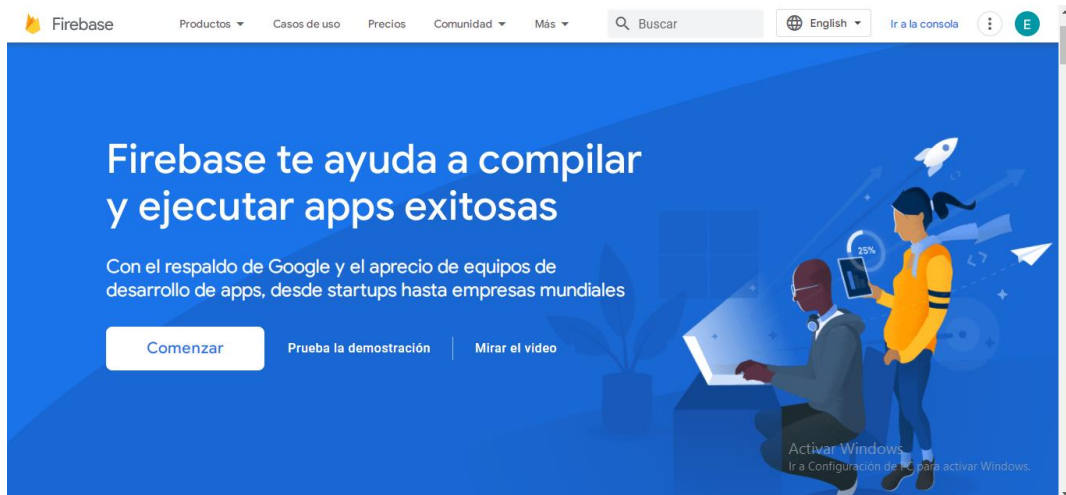
A continuación, se detallará cada uno de los funcionamientos y su programación.

#### **3.3.1 CREACIÓN DE LA BASE DE DATOS EN TIEMPO REAL**

Dentro del desarrollo una de las primeras configuraciones a realizar es la creación de la base de datos en Firebase, aquí se puede crear y configurar

una base de datos en tiempo real dentro de una nube de esta forma permitirá compartir datos entre la APP y el robot.

Para la creación de la base de datos primero se ingresa a la cuenta Google y se abre la página oficial de Firebase como se puede ver en la siguiente imagen.



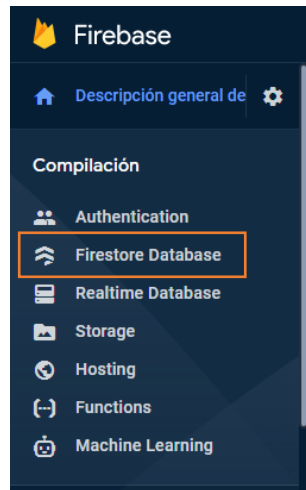
**Figura 41**  
**PÁGINA OFICIAL FIRE BASE**

Dentro de esta página se selecciona la opción de “Comenzar” esto permitirá crear un proyecto nuevo dentro de Firebase. A continuación, se muestra la otra página donde pide el nombre del proyecto aquí se eligió el nombre de “ROBOT”. En la siguiente figura se muestra el ingreso del nombre.



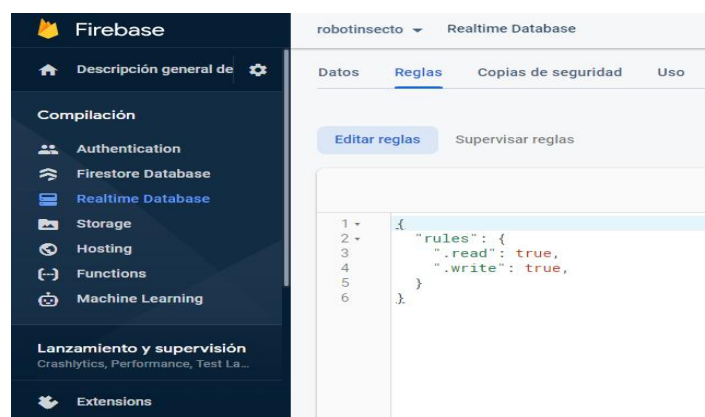
**Figura 42**  
**INGRESO DE NOMBRE DEL PROYECTO**

Creado el proyecto se selecciona la opción de “Realtime Database” esta opción permite la creación de una base de datos que se actualice en tiempo real con lo cual se puede realizar la transmisión de datos entre dispositivos que se encuentren alejados. En la siguiente figura se muestra la opción seleccionada.



**Figura 43**  
**CREACIÓN DE BASE DE DATOS EN TIEMPO REAL**

Al crear la base de datos en tiempo real se debe configurar las reglas de la base de datos para que estas permitan la escritura y lectura desde dispositivos externos. Para esto se dirige a la opción de regla y se escribe el valor de “true”. En la siguiente figura se observa la configuración de las reglas de la base de datos.

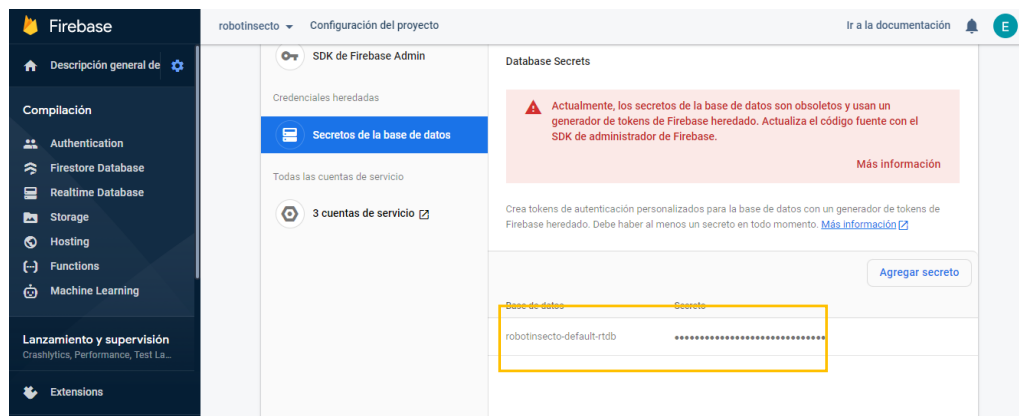


**Figura 44**  
**CONFIGURACIÓN DE LAS REGLAS DE LA BASE DE DATOS**

Después de configurar las reglas de la base de datos para utilizarla es necesario tener la clave del secreto de la base de datos. Esta clave

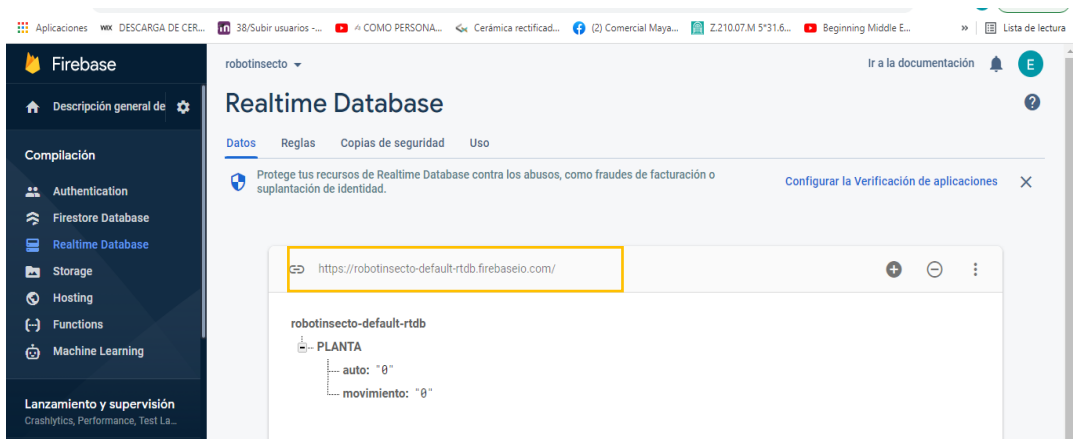
permitirá que otros software o dispositivos puedan ingresar a la base de datos de esta forma se evita que cualquier usuario pueda ingresar y modificar los datos que se están manejando.

Para esto se dirige a configuraciones del proyecto y a la opción de secretos de la base de datos, en esta pantalla se copia la clave o también llamada autorización que se encuentra bajo la palabra secreto. A continuación, se muestra la captura de pantalla de donde se copia la clave.



**Figura 45**  
**AUTORIZACIÓN DE LA BASE DE DATOS**

Con la clave de la base de datos se ingresa la opción “Realtime Database” en la cual se copia el link o host de la base de datos. Este link tiene la dirección donde se almacenan los datos. En la siguiente figura se muestra una captura de pantalla de lo antes descrito.

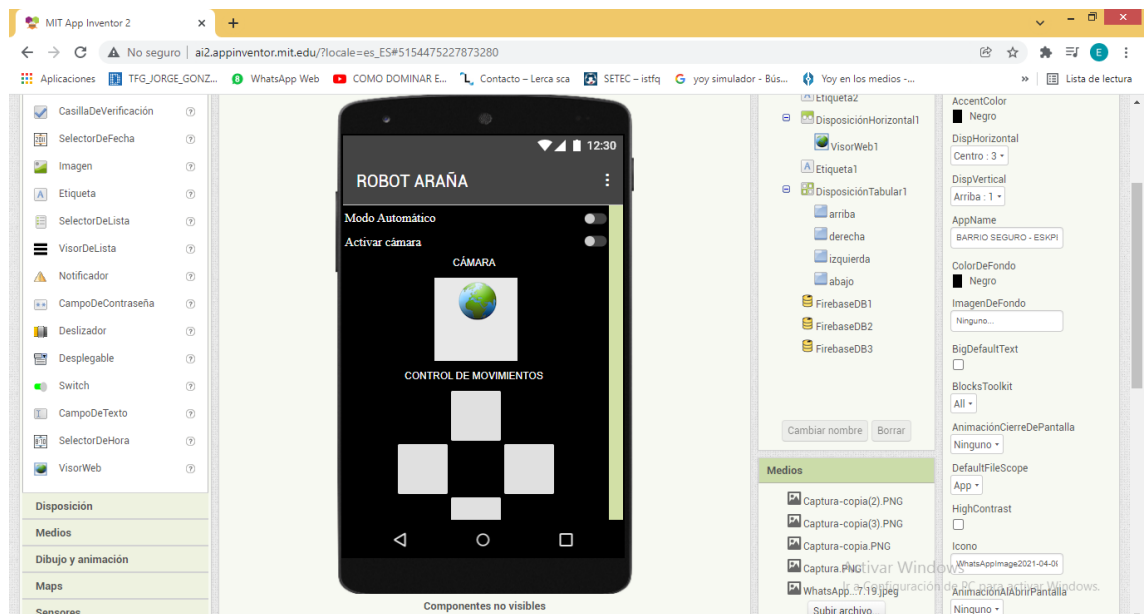


**Figura 46**  
**HOST DE LA BASE DE DATOS**

Al realizar todos estos pasos se crea una base de datos en tiempo real y se obtiene la autorización y el host. Estos datos permitirán configurar la APP y la ESP32 para que tenga acceso y puedan leer y escribir valores.

### 3.3.2 DISEÑO DEL INTERFAZ EN APP INVENTOR

Para el diseño del interfaz se utilizó APP inventor el cual permite la creación de aplicaciones celulares con conexión a una base de datos en tiempo real. A continuación, se muestra el interfaz de desarrollo de APP Inventor



**Figura 47**  
**INTERFAZ APP INVENTOR**

Dentro del interfaz de APP inventor se tiene dos partes la de programación del interfaz con el usuario y la programación en bloques para el flujo de acciones según los datos que lleguen a la aplicación.

Se detallará la configuración del interfaz con el usuario y la programación en APP inventor para el manejo del robot.

Primero se ingresó el logotipo de la universidad con la opción de imagen, con la opción de etiqueta se ingresó los nombres de los creadores. En la siguiente ilustración se muestra lo descrito.



**Figura 48**  
**LOGOTIPO Y NOMBRES DE INTEGRANTES**

En la segunda parte del interfaz con el usuario se tiene la opción de activar o desactivar el control automático. Para esto se utilizó un pulsador el cual al ser presionado una vez se activa el modo automático y se torna de color verde mostrando una leyenda que dice “ON”. Para desactivar el modo automático se mantiene presionado por un tiempo hasta que el pulsador se torne de color rojo y muestre la leyenda “OFF”. En la siguiente figura se muestra la imagen del pulsador



**Figura 49**  
**PULSADOR DE ACTIVACIÓN/DESACTIVACIÓN DEL CONTROL AUTOMÁTICO**

Finalmente se diseñó una botonera la cual se mostrará sólo cuando esté el modo manual funcionado. Dentro de esta botonera se utilizaron 4 pulsadores para manejar las direcciones de movimiento del robot arriba, abajo, izquierda y derecha. Además, se incluyó en la parte inferior una imagen que muestra la dirección del movimiento del robot según sea presionada. A continuación, se muestra la interfaz para el control en modo manual.



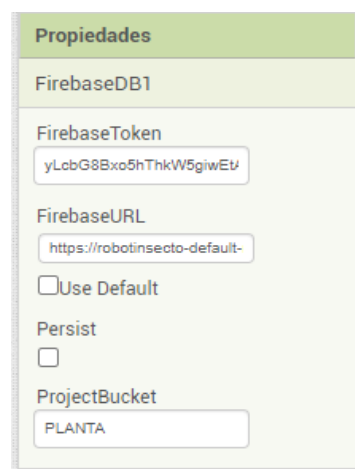
**Figura 50**  
**CONEXIÓN DE ESP32 CAM**

Con la configuración del interfaz completo se le añade la opción de Firebase DB a la APP, esto permite que la aplicación pueda utilizar funciones de programación y de manejo de datos con Firebase. A continuación, se muestra la opción de FireBase en APPInventor.



**Figura 51**  
**AÑADIR FIREBASE A APP INVENTOR**

Al tener la base de datos insertada en la APP se configura los datos de la autorización y host obtenidos en la anterior sección. En FirebaseToken se escribe la autorización y en FirebaseURL se escribe el host con estos datos se permite que la APP pueda escribir y leer datos de la base de datos antes creada. En la siguiente figura se muestra la configuración ya descrita.

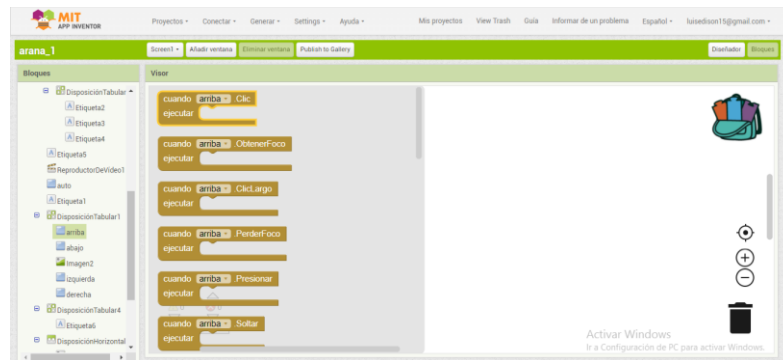


**Figura 52**  
**CONFIGURACIÓN DE LA AUTORIZACIÓN Y EL HOST**



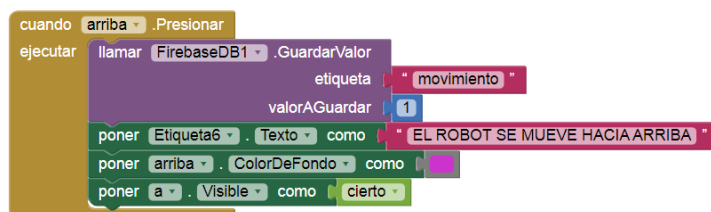
Al terminar el diseño del interfaz se procede a la programación del envío de datos. Aquí se configura el comportamiento de cada uno de los pulsadores.

Primero se configura el pulsador “arriba” del control manual. Para este pulsador se busca dentro de las opciones la sentencia “Clic” o “Presionar”, esta sentencia permitirá realizar una acción cuando se haga un clic corto.



**Figura 53**  
**SENTENCIA CLIC DEL PULSADOR ARRIBA**

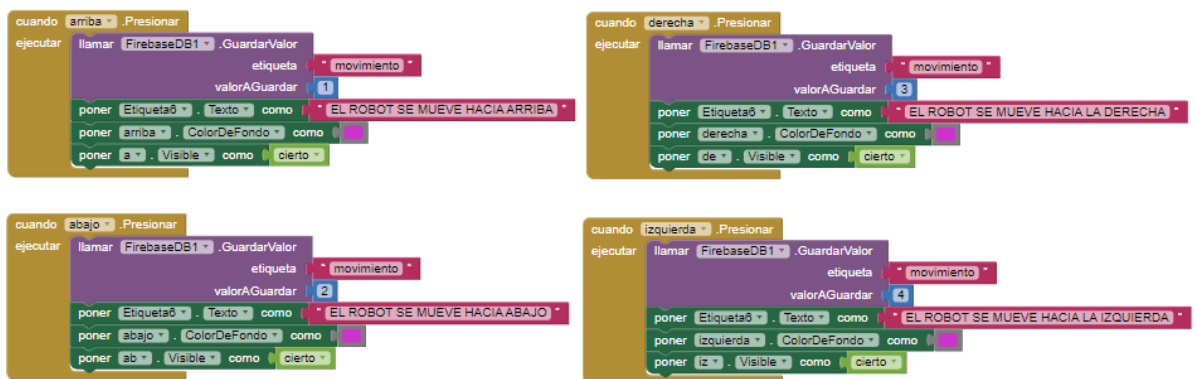
Dentro de esta sentencia se utiliza el bloque “FirebaseDB1.GuardarValor” este comando envía el valor de “1” a la base de datos en tiempo real creada en Firebase en la variable movimiento. Después se usa otro bloque el cual muestra en la “etiqueta6” el mensaje “EL ROBOT SE MUEVE HACIA ARRIBA”. Además, se incluye un tercer bloque que permite cambiar el color del pulsador al ser presionado y este tome un color morado. Finalmente, el último bloque permite que una imagen ya guardada sea visible y aparezca en la pantalla una flecha que indica hacia arriba. A continuación, se muestra la programación de lo antes descrito para el pulsador “arriba”



**Figura 54**  
**PROGRAMACIÓN DEL PULSADOR ARRIBA**

Después de configurar el pulsador “arriba” se configura los demás pulsadores los cuales tienen la misma estructura. Lo que cambia es que al presionar el pulsador “abajo” envía a la base de datos el valor de 2, al presionar el pulsador “derecha” envía el valor de 3 y al presionar el pulsador de “izquierda” envía el valor de 4.

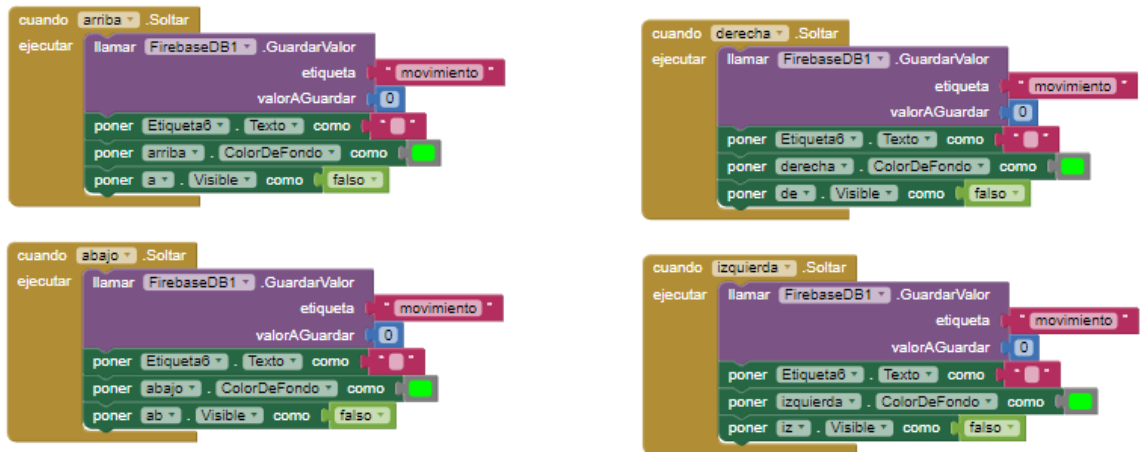
En la siguiente figura se muestra la programación de los cuatro pulsadores.



**Figura 55**  
**PROGRAMACIÓN DE LOS PULSADORES AL PRESIONAR**

Como siguiente paso se configura el comportamiento del pulsador al dejar de presionar. Para esto se selecciona la sentencia “soltar” y dentro de ella se pone el bloque que envíe un valor de 0 a la base de datos. Después que borre el mensaje y que cambie el color a verde. Finalmente, la imagen se pone en no visible para que no aparezca en la pantalla. Esta programación se repite para los cuatro pulsadores en sus respectivas sentencias.

En la siguiente figura se muestra la programación de los pulsadores al momento de soltar el pulsador.



**Figura 56**  
**PROGRAMACIÓN DE LOS PULSADORES AL SOLTAR**

Después de programar el teclado se configura el pulsador de control automático este permitirá cambiar de modo de funcionamiento del robot, para esto se escoge la sentencia “clic” del pulsador y que envíe un valor de 1 a la variable “auto” de la base de datos cuando este activo y envíe un valor de 0 cuando esté desactivado.

Además, se usa un bloque para mostrar la leyenda de “ON” o “OFF” según sea el estado de este pulsador, se configura otro bloque para que al estar en “ON” el pulsador se torne de color verde y al estar en “OFF” el pulsador se torne de color rojo. Se añade un bloque más que evita que el modo manual y automático entre en conflicto ya que si el modo automático está activo no se mostrará las teclas de dirección en la APP evitando que datos innecesarios sean transmitidos. En la siguiente figura se muestra la programación del pulsador de modo automático.



**Figura 57**  
**PROGRAMACIÓN DEL PULSADOR DE MODO AUTOMÁTICO**

### 3.3.3 PROGRAMACIÓN DE MODO MANUAL

La programación en modo manual se refiere al manejo del robot de forma radio controlada, lo que quiere decir es que mediante una APP en el celular se podrá controlar los movimientos adelante, atrás, izquierda y derecha del robot. Además, este modo dispone de dos cámaras las cuales permiten tener un video en tiempo real del entorno en el cual se desenvuelve el robot.

Para la programación del robot se inicializa la librería “wifi” la cual permite conectarse al ESP32 con una red wifi, para tener acceso a la base de datos en internet. También se configura la librería “firebaseESP32” esta librería configura el acceso a las bases de datos en tiempo real. Además, permite realizar la lectura y escritura de datos.

Después se configura el host y la autorización de la base de datos estas claves se los consiguieron en el anterior apartado en la creación de base de datos. En “Firebase\_host” se ingresa el link donde está la base de datos y en “Firebase\_auth” se escribe el secreto de la base de datos, con esto se le da acceso al ESP32 para ingresar a Firebase.

Para la configuración de la red y su clave de acceso se utiliza el comando “wifi\_ssid” donde se escribe el nombre de la red WiFi y en “wifi\_password” se escribe la clave de la red con estos datos la ESP32 puede usarlos para conectarse al router. En la siguiente imagen se muestra la programación realizada.

```
#include <WiFi.h>
#include "FirebaseESP32.h"

#define FIREBASE_HOST "robotinsecto-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "yLcbG8Bxo5hThkW5giwEtAjpg5zAQB11aj0BpPwWM"

#define WIFI_SSID "NETLIFE- Rocio"
#define WIFI_PASSWORD "1710255595"
```

**Figura 58**  
**CONFIGURACIÓN DE LIBRERÍAS Y CLAVES DE ACCESO**

En la configuración de los pines para el manejo del TB6612FNG se usa los puertos 5 y 17 como PWM y los puertos 19,18,17 y 16 como pines para el control de dirección y frenado de los motores. En la siguiente imagen se muestra la configuración de los pines.

```
// DEFINICION DE PINES
int A1 = 19;      // SALIDA 1.2
int A2 = 18;      // SALIDA 1.1
int PWMA = 5;     // PWMA

int BA = 4;       // SALIDA 2.2
int BB = 16;      // SALIDA 2.1
int PWMB = 17;    // PWMB
```

**Figura 59**  
**CONFIGURACIÓN DE PINES DEL ESP32**

Después de la configuración de los pines digitales se configura la frecuencia de la PWM en 5000Hz y se activa el canal 0 y 1. Además, se le asigna una resolución de 8 bits para el control de la PWM lo que permite variar el ancho de pulso de 0 a 255. A continuación, se presenta la programación de la PWM.

```
// CONFIGURACIONES PRINCIPALES PWM
const int freq = 5000;
const int ledChannel = 0;
const int ledChannel2 = 1;
const int resolution = 8;
```

**Figura 60**  
**CONFIGURACIÓN DE LA PWM**

Los puertos se deben poner en función de "OUTPUT" de esta forma cada puerto se configura para entregar una señal de 0 a 3.3V digital en sus salidas y en las PWM una señal de valor RMS que varía de 0 a 3.3V. Además, se configura el puerto serial a una velocidad de 115200 Baudios para realizar pruebas con los sensores y motores.

```

void setup()
{

    Serial.begin(115200);

    pinMode(A1,OUTPUT);
    pinMode(A2,OUTPUT);
    pinMode(BA,OUTPUT);
    pinMode(BB,OUTPUT);
    pinMode(PWMA,OUTPUT);
    pinMode(PWMB,OUTPUT);
}

```

**Figura 61**  
**CONFIGURACIÓN DE PUERTOS DE SALIDA**

En la configuración de los canales de salida de la PWM se usa el comando “ledcSetup” al cual se le proporciona el canal de salida en “ledchannel1”, el valor de la frecuencia en “freq” y la resolución en “resolution”. Esta configuración se realiza para los dos canales ya que cada canal controla un motor.

```

ledcSetup(ledChannel, freq, resolution);
ledcAttachPin(PWMA, ledChannel);

ledcSetup(ledChannel2, freq, resolution);
ledcAttachPin(PWMB, ledChannel2);

```

**Figura 62**  
**CONFIGURACIÓN DE CANALES DE SALIDA DE PWM**

Con la configuración de los puertos y PWM realizadas se usa el comando “wifi.begin”. Este usa la clave y el nombre de la red para conectarse, en caso de no conectarse el sistema sigue realizando la petición al moden para ingresar a la red mientras un led parpadea advirtiendo de que no existe comunicación con la red ingresada.

En la siguiente imagen se muestra el algoritmo para el ingreso del ESP32 a la red Wifi.

```

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

while (WiFi.status() != WL_CONNECTED)
{
  digitalWrite(LED_ESTADO,HIGH);
  delay(50);
  digitalWrite(LED_ESTADO,LOW);
  delay(50);
}

```

**Figura 63**  
**ALGORITMO PARA INGRESAR A LA RED WIFI**

Para ingresar a la base de datos de Fire Base se utiliza el comando “Firebase.Begin” el cual utiliza el host y la clave de autorización que se obtuvieron en los apartados anteriores. En la siguiente ilustración se muestra el comando utilizado para conectarse a la base de datos.

```

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

```

**Figura 64**  
**ALGORITMO PARA INGRESAR A FIRE BASE**

Con el programa antes descrito el robot está conectado a una red Wifi y puede recibir y enviar datos hacia la base de datos, con esto se pasa a la programación del void loop donde se programa las acciones que realiza el robot según los valores que provienen de la base de datos.

La función a programar es la obtención de valores de la base de datos y su almacenamiento en la variable “movimientos” para ingresar a Fire base y obtener estos datos se utiliza el comando “Fire.getString”. Al ingresar este comando la ESP32 ingresa a la base de datos lee el valor de “movimientos” y guarda ese valor en la memoria temporal de la ESP32. Al tener el valor almacenado este valor se compara si es igual a “1” se mueve hacia adelante, si es igual a “2” se mueve hacia atrás, si es igual a “3” se mueve hacia la izquierda, si es igual a “4” se mueve a la derecha y si el valor es igual a “0” el robot se detiene.

Para un movimiento hacia adelante se configura que los dos motores giren en un sentido horario con un PWM de 120 como se muestra en el siguiente algoritmo.

```
Firebase.getString(firebaseData, "movimiento");
if( firebaseData.stringData() == "1")
{
    digitalWrite(A1,HIGH);
    digitalWrite(A2,LOW);
    digitalWrite(BA,LOW);
    digitalWrite(BB,HIGH);

    ledcWrite(ledChannel, 120);
    ledcWrite(ledChannel2, 120);

    delay(100);
}
```

**Figura 65**  
**ALGORITMO PARA IR AL FRENTE**

Para un movimiento hacia atrás se configura que los dos motores giren en un sentido anti horario a un PWM de 120 como se muestra en el siguiente algoritmo.

```
else if( firebaseData.stringData() == "2")
{
    digitalWrite(A1,LOW);
    digitalWrite(A2,HIGH);
    digitalWrite(BA,HIGH);
    digitalWrite(BB,LOW);

    ledcWrite(ledChannel, 120);
    ledcWrite(ledChannel2, 120);

    delay(100);
}
```

**Figura 66**  
**ALGORITMO PARA IR ATRÁS**

Para un movimiento hacia la izquierda se configura que un motor gire en sentido horario y el otro en sentido anti horario a un PWM de 120.



```

else if( firebaseData.stringData() == "3")
{
    digitalWrite(A1,HIGH);
    digitalWrite(A2,LOW);
    digitalWrite(BA,HIGH);
    digitalWrite(BB,LOW);

    ledcWrite(ledChannel, 120);
    ledcWrite(ledChannel2, 120);

    delay(100);
}

```

**Figura 67**  
**ALGORITMO PARA IR A LA IZQUIERDA**

Para un movimiento hacia la derecha se configura que un motor gire en sentido horario y el otro en sentido anti horario opuesto al anterior giro y a un PWM de 120. En la siguiente figura se muestra el algoritmo.

```

else if( firebaseData.stringData() == "4")
{
    digitalWrite(A1,HIGH);
    digitalWrite(A2,LOW);
    digitalWrite(BA,LOW);
    digitalWrite(BB,HIGH);

    ledcWrite(ledChannel, 120);
    ledcWrite(ledChannel2, 120);

    delay(100);
}

```

**Figura 68**  
**ALGORITMO PARA IR A LA DERECHA**

Para frenar el robot se pone sus salidas en "LOW" y se configura el valor de PWM en 0 con lo cual el robot se detiene.

```

else if( firebaseData.stringData() == "0")
{
    digitalWrite (A1, LOW);
    digitalWrite (A2, LOW);
    digitalWrite (BA, LOW);
    digitalWrite (BB, LOW);

    ledcWrite(ledChannel, 0);
    ledcWrite(ledChannel2, 0);

    delay(100);
}

```

**Figura 69**  
**ALGORITMO PARA FRENAR**

### 3.3.4 PROGRAMACIÓN DEL MODO AUTOMÁTICO

Para la programación en modalidad automático el robot dispone de dos sensores Sharp los cuales detectan la distancia del robot hacia su lado derecho y del robot hacia el frente, mediante estos dos sensores el robot puede identificar si existen obstáculos dentro de su camino y aplicando el algoritmo de la mano derecha, este algoritmo permite desplazarse por un entorno donde no existan islas.

En el diseño del algoritmo de modo automático. Como primer paso se mide la distancia del sensor derecho hacia la pared con este valor se utiliza como referencia para desplazar a una distancia de 20 centímetros de la pared. En la siguiente figura se muestra el código para la adquisición de la distancia de la pared hacia el robot.

```

for(int i = 0; i <= 10; i++)
{
    valor = analogRead(s1);
    total= (total+valor)/11;
}

for(int i = 0; i <= 10; i++)
{
    valor2 = analogRead(s2);
    total2= (total2+valor2)/11;
}

```

**Figura 70**  
**ALGORITMO PARA MEDIR DISTANCIA CON SENSORES**

Al tener la distancia del robot hacia la pared derecha se crea un algoritmo PD el cual permite mantener la distancia de desplazamiento constante mientras el robot se desplaza cerca a la pared. Para este algoritmo se calcula el error restando la distancia medida menos el setpoint.

Después se calcula la acción de control para lo cual primero se realiza el algoritmo P este es igual a  $K_p * \text{error}$  y después se calcula la acción de control D este es igual a  $K_d * (\text{error} - \text{error}_1)$ .

Ya calculada la acción de control "u" este valor se suma a la velocidad de la llanta derecha y se resta a la velocidad de la llanta izquierda de esta forma el robot puede desplazarse en línea recta por sí solo.

```

error = distancia - d;
u = Kp*error + Kd*(error - error_1);
error_1 = error;

veld = vel + u;
veli = vel - u;

```

**Figura 71**  
**ALGORITMO PD**

Finalmente, si el robot detecta un objeto a una distancia menor a 30cm esta gira hacia la derecha evitando chocar.

```
ledcWrite(ledChannel, veld);
ledcWrite(ledChannel2, veli);
delay(100);

if(total2 <= 30)
{
    digitalWrite(A1,HIGH);
    digitalWrite(A2,LOW);
    digitalWrite(BA,LOW);
    digitalWrite(BB,HIGH);

    ledcWrite(ledChannel, 120);
    ledcWrite(ledChannel2, 120);

    delay(100);
}
```

**Figura 72**  
**ALGORITMO DE GIRO**

## 4. RESULTADOS

### 4.1 TIEMPO DE DESPLAZAMIENTO

En el presente proyecto se realizó pruebas de desplazamiento del robot en una pared de color blanca, para obtener los tiempos que se demora en una distancia de 1.20 m en diferentes calibraciones del controlador PD con una batería de 7.4V, 550mAh con un peso de 38 gramos.

Distancia	Tiempo	Velocidad
1.20	20	0.06
1.20	18	0.067
1.20	18	0.067
1.20	16	0.075

Tabla 1. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT

Se realizó pruebas de desplazamiento del robot en una pared de color café, para obtener los tiempos que se demora en una distancia de 1.20 m en diferentes calibraciones del controlador PD con una batería de 7.4V, 550mAh con un peso de 38 gramos.

Distancia	Tiempo	Velocidad
1.20	20	0.06
1.20	23	0.05
1.20	17	0.07
1.20	17	0.07

Tabla 2. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT

En el presente proyecto se realizó pruebas de desplazamiento del robot en una pared de color blanca, para obtener los tiempos que se demora en una distancia de 1.2 m en diferentes calibraciones del controlador PD con una batería de 7.4V, 1500mAh con un peso de 93 gramos.

Distancia	Tiempo	Velocidad
1.20	20	0.06
1.20	18	0.067
1.20	18	0.067
1.20	16	0.075

**Tabla 3. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT**

Se realizó pruebas de desplazamiento del robot en una pared de color café, para obtener los tiempos que se demora en una distancia de 1.20 m en diferentes calibraciones del controlador PD con una batería de 7.4V, 1500mAh con un peso de 93 gramos.

Distancia	Tiempo	Velocidad
1.20	21	0.057
1.20	20	0.06
1.20	20	0.06
1.20	18	0.066

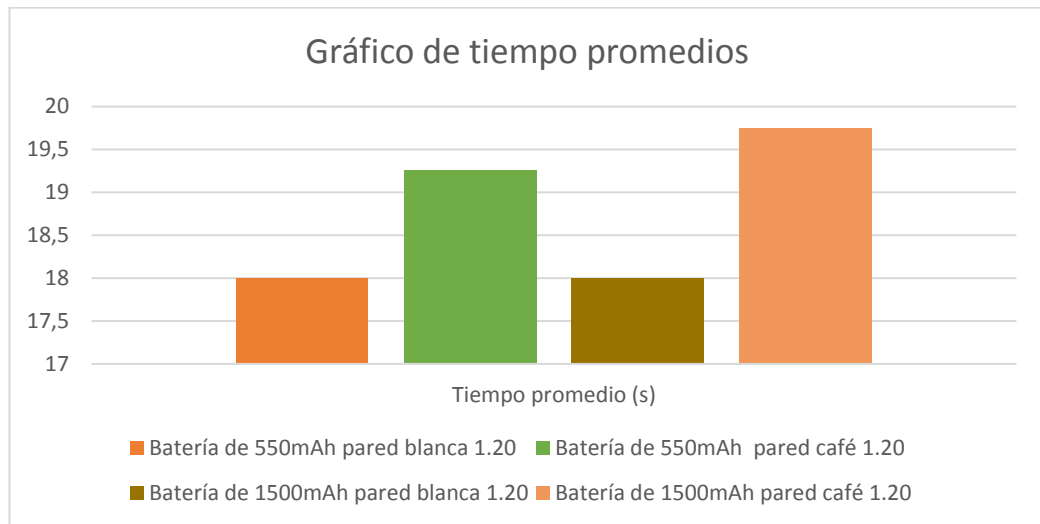
**Tabla 4. TIEMPOS DE DESPLAZAMIENTO DEL ROBOT**

Para el análisis del comportamiento del robot se realiza una tabla con los tiempos y velocidades promedio en cada una de las pruebas como se muestra a continuación.

Tipo de prueba	Distancia (m)	Tiempo promedio (s)	Velocidad promedio (m/s)
Batería de 550mAh pared blanca	1.20	18	0,0666
Batería de 550mAh pared café	1.20	19,25	0,0623
Batería de 1500mAh pared blanca	1.20	18	0,0666
Batería de 1500mAh pared café	1.20	19,75	0,0607

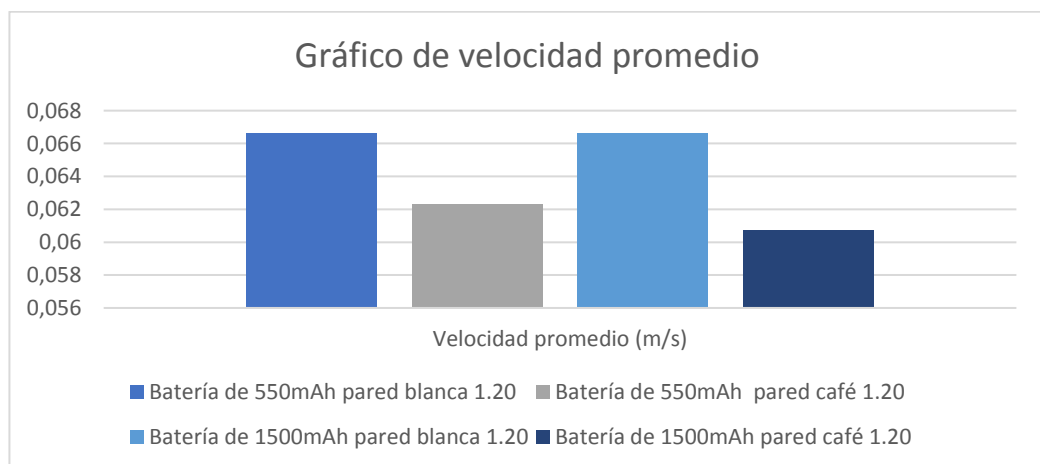
**Tabla 5. TIEMPOS Y VELOCIDADES PROMEDIO DE LAS PRUEBAS**

Con la tabla anterior se construye un gráfico de los tiempos promedios en cada una de las cuatro pruebas. Aquí se puede observar que el tiempo con la pared café aumenta eso se debe que al ser una pared de color oscuro absorbe la luz que emite los sensores y provoca que el robot se demore más tiempo en su desplazamiento ya que tiene que corregir sus movimientos.



**Figura 73**  
**TIEMPO PROMEDIO EN CADA PRUEBA**

Con la tabla anterior se construye un gráfico de velocidad promedio en cada una de las cuatro pruebas. Aquí se puede observar que la velocidad promedio con la batería de 1500mAh es menor ya que al ser más pesada provoca un desplazamiento más lento del robot.



**Figura 74**  
**VELOCIDAD PROMEDIO EN CADA PRUEBA**

## 4.2 ELABORACIÓN DE LA PROGRAMACIÓN EN ARDUINO

Para la elaboración del algoritmo de control se utilizó una ESP32 con el IDE de Arduino para su programación. Este dispositivo se conecta mediante Firebase a una App en el celular desarrollada en APP inventor



**Figura 75**  
**ROBOT ENSAMBLADO**

```
Archivo Editar Programa Herramientas Ayuda
[Icons]
robot_final
#include <WiFi.h>
#include "FirebaseESP32.h"

#define FIREBASE_HOST "robotinsecto-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "yLcbG8Bxo5hThkW5giwEtAjg5zAQB1laj0BpPwWM"

#define WIFI_SSID "NETLIFE- Rocio"
#define WIFI_PASSWORD "1710255595"

// DEFINICION DE PINES
int A1 = 19; // SALIDA 1.2
int A2 = 18; // SALIDA 1.1
int PWMA = 5; // PWMA

int BA = 4; // SALIDA 1.2
int BB = 16; // SALIDA 1.1
int PWMB = 17; // PWMA
```

**Figura 76**  
**ARDUINO IDE**



### 4.3 ELABORACIÓN DE LA PROGRAMACIÓN EN APP INVENTOR

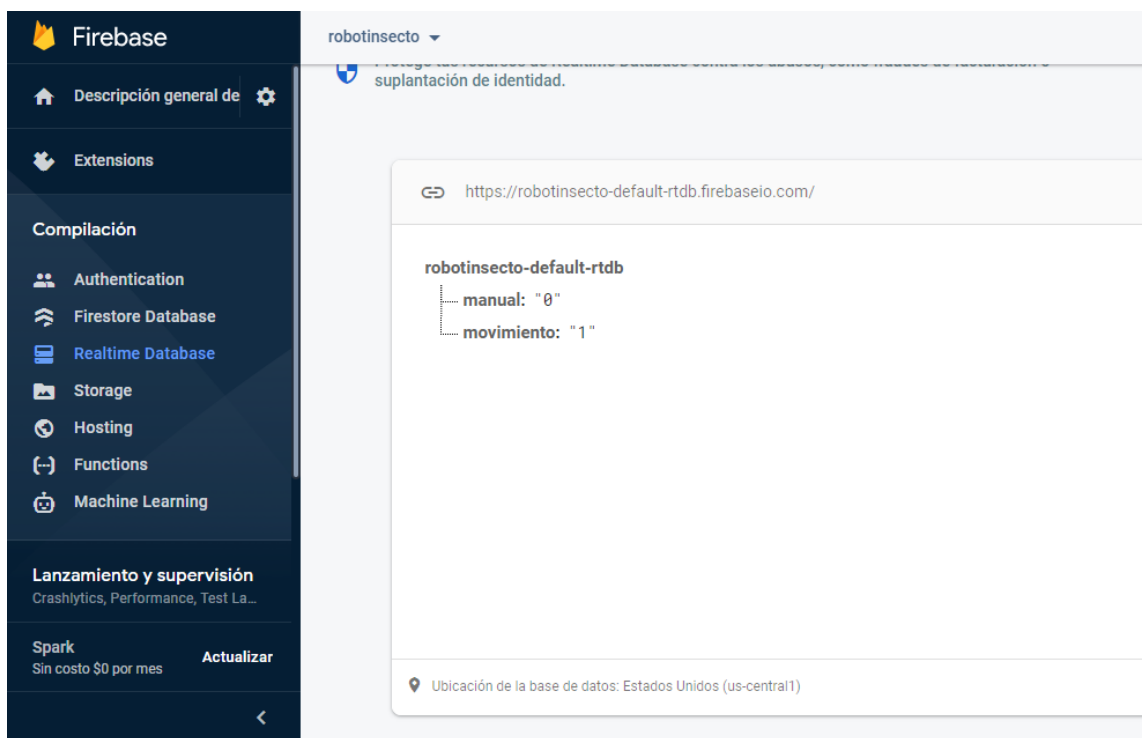
El programa en App inventor se desarrolló de forma que sea muy intuitivo y fácil de manejar por el usuario permitiendo un control rápido del robot. Para esto se realizó el diseño del interfaz con el usuario como se muestra a continuación.



Figura 77  
INTERFAZ CON EL USUARIO

## 4.4 CONFIGURACIÓN DE FIRE BASE

En Fire base se configuró las variables “manual” la cual al estar en “0” activa el modo manual y al estar en “1” activa el modo automático. Además, se tiene otra variable llamada movimientos las cual puede tomar valores de 0 a 4 las cuales permite identificar hacia donde debe dirigirse el robot. El valor de 1 es movimiento hacia adelante y el valor de 2 es movimiento hacia a atrás. El valor de 3 y 4 permiten realizar giros a la derecha e izquierda en su propio eje.



**Figura 78**  
**VARIABLES EN FIRE BASE**

## 5. CONCLUSIONES

- Se concluye que mediante el desarrollo del robot se logra un control manual y automático para la exploración de lugares desconocido.
- Se implementó una base de datos virtual en tiempo real, la cual permite ver la transmisión de datos entre el robot y la aplicación móvil.
- Se realizó la configuración de tres módulos esp32 los cuales dos funcionan para la captura de imágenes y uno para el control de motores y sensores, permitiendo que el robot pueda seguir desplazándose, aunque haya sufrido daño en una de sus cámaras.
- Se realizó la implementación de la aplicación móvil en APP Inventor lo cual permite que la aplicación pueda ser modificada tanto en su apariencia como es su funcionalidad.
- Se implementó un sistema de locomoción Theo Jansen el cual tiene 16 puntos de apoyo en el piso lo que le permite una mayor estabilidad al momento de su desplazamiento, evitando que este pueda caerse.
- Se concluye que el tipo de color de las paredes por donde se desplace el robot afecta a su desplazamiento, ya que al estar en presencia de pared de color oscuro estas absorben la luz de los sensores y provocan medidas erróneas disminuyendo la velocidad de desplazamiento.

## 6. RECOMENDACIONES

- Se recomienda el uso de baterías LiPo para aumentar el tiempo de autonomía del robot.
- Se recomienda al realizar pruebas en modo automático utilizar un ambiente controlado.
- Implementar un sistema de visión artificial para la detección de objetos y reconocimiento para usar algoritmos de movimiento.
- Se recomienda el uso de este robot un en piso sin irregularidades para evitar daño en sus patas.
- Usar red WiFi que tenga de usuario y contraseña ya que los programas del robot necesitan de estos datos para su conexión.
- Se recomienda la instalación del APK en un dispositivo Android ya que la aplicación está diseñada para este tipo de sistema operativo.
- Para el uso de las cámaras se recomienda que la computadora este en la misma red WiFi ya que estas se enlazan de forma local.

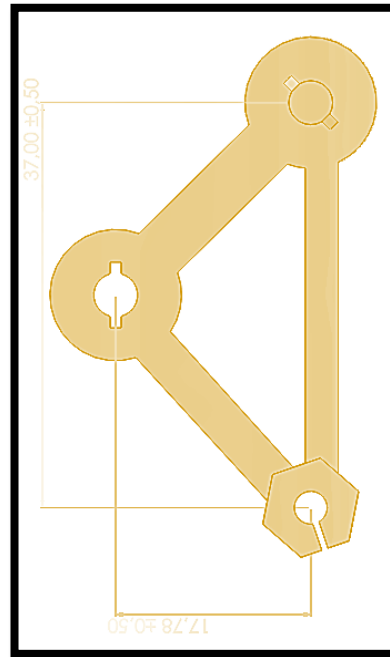
## 7. BIBLIOGRAFÍA

- Arduino. (6 de Febrero de 2020). Obtenido de Arduino:  
<https://arduino.cl/programacion/>
- Cdtecnologia. (20 de Junio de 2019). *Cdtecnologia*. Obtenido de  
<https://cdtecnologia.net/placas-de-desarrollo/1927-microcontrolador-esp32-wifi-con-camara-ov2640-0001927.html>
- Cprmerida. (16 de Septiembre de 2017). *Cprmerida*. Obtenido de  
<http://cprmerida.juntaextremadura.net/5-actividades/cursos/3325-curso-iniciacion-de-app-inventor>
- EducationPlus. (29 de Noviembre de 2018). *EducationPlus*. Obtenido de  
<https://blogsaverroes.juntadeandalucia.es/tecnocanete/2018/11/29/app-inventor/#:~:text=App%20Inventor%20es%20un%20entorno,bloques%20para%20crear%20la%20aplicaci%C3%B3n>.
- Electrostore. (15 de Agosto de 2021). Obtenido de Grupo electrostore:  
<https://grupoelectrostore.com/shop/sensores/encoders/par-de-encoders-magneticos-pololu-27-18v-para-micromotores/>
- Guerra Carmenate, J. (3 de Marzo de 2021). *Programar facil*. Obtenido de  
<https://programarfacil.com/esp8266/esp32/>
- Hubor. (13 de Marzo de 2020). *Hubor*. Obtenido de <https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/2-proteus.html#:~:text=Proteus%20es%20una%20aplicaci%C3%B3n%20para,de%20errores%2C%20documentaci%C3%B3n%20y%20construcci%C3%B3n>.
- Llamas , L. (18 de Agosto de 2016). *Luis llamas*. Obtenido de  
<https://www.luisllamas.es/arduino-motor-dc-tb6612fng/>
- Moviltronics. (5 de Febrero de 2018). *Moviltronics*. Obtenido de  
<https://moviltronics.com/tienda/modulo-driver-para-motor-tb6612fng/>

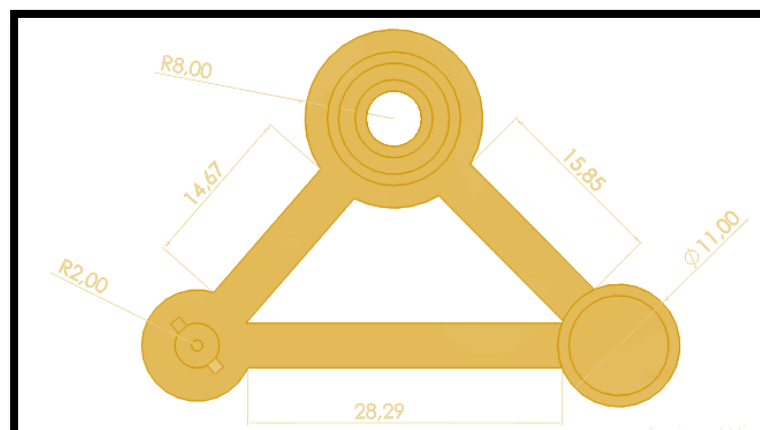
- Muradas, Y. (22 de Junio de 2021). *Openwebinars*. Obtenido de <https://openwebinars.net/blog/que-es-firebase-de-google/>
- Naylamp. (23 de Marzo de 2021). *Naylampmechatronics*. Obtenido de [https://naylampmechatronics.com/blog/55\\_tutorial-sensor-de-distancia-sharp.html](https://naylampmechatronics.com/blog/55_tutorial-sensor-de-distancia-sharp.html)
- Naylamp MEchatronics. (12 de Septiembre de 2018). *Mechatronics*. Obtenido de [https://naylampmechatronics.com/blog/55\\_tutorial-sensor-de-distancia-sharp.html](https://naylampmechatronics.com/blog/55_tutorial-sensor-de-distancia-sharp.html)
- Onieva , J., & Martinez , M. (2019). *Diseño y construcción de un mecanismo andante* . Madrid: Universidad Politécnica Madrid.
- Pérez Cardona, M. (27 de Octubre de 2021). *Oebschool*. Obtenido de <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/#:~:text=Firebase%20se%20trata%20de%20una,usuarios%20y%20ganar%20m%C3%A1s%20dinero.>
- Pololu. (5 de Marzo de 2017). *Pololu*. Obtenido de <https://www.pololu.com/category/60/micro-metal-gearmotors>
- Shunsuke , N., Mohan, E., & Masami , I. (2013). Dynamic Analysis and Modeling of Jansen Mechanism. *Sciencedirect*, 1563-1569.
- Taloselectronics. (20 de Julio de 2021). *Taloselectronics*. Obtenido de <https://uelectronics.com/producto/esp32-38-pines-esp-wroom-32/>
- Volkskrant. (5 de Septiembre de 2015). *Espacio Fundación telefonica*. Obtenido de <https://espacio.fundaciontelefonica.com/noticia/correplayas-el-articulo-de-theo-jansen-que-lo-empezo-todo/>
- Zuñiga, A. (2017). Diseño y administración de proyectos de robótica educativa. *Revista SE*, 6-27.

## 8. ANEXOS

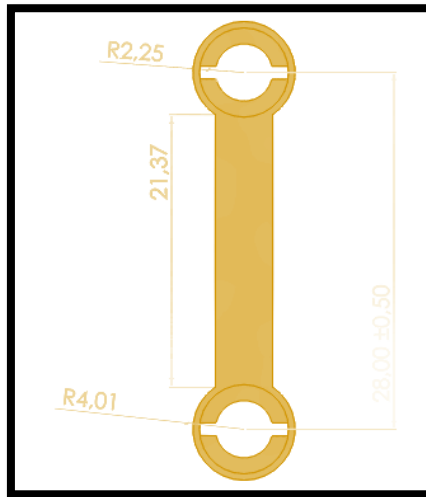
### 8.1 ANEXO A. PIEZAS DEL ROBOT



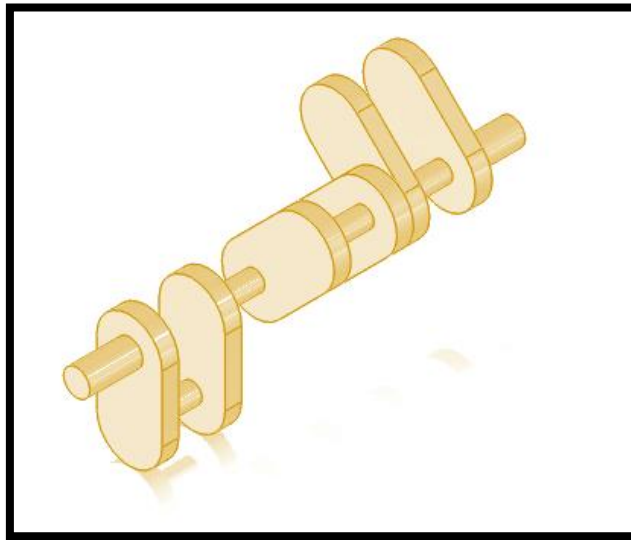
**Figura 79**  
**PIE THEO JANSEN EN SOLID WORK**



**Figura 80**  
**MUSLO THEO JANSEN EN SOLID WORK**

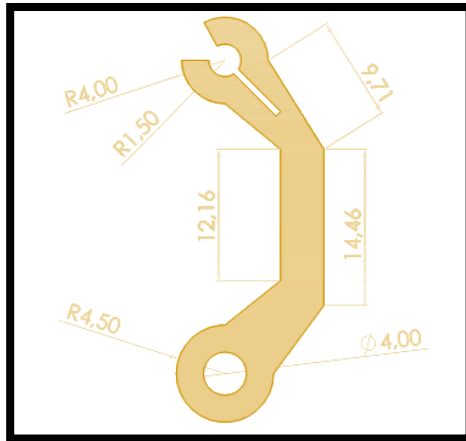


**Figura 81**  
**LIGAMENTO THEO JANSEN EN SOLID WORK**

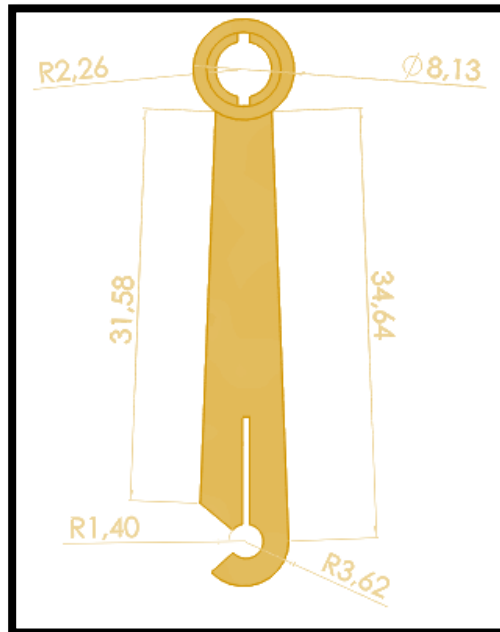


**Figura 82**  
**CIGÜEÑAL THEO JANSEN**





**Figura 83**  
**LIGAMENTO INFERIOR**



**Figura 84**  
**LIGAMENTO SUPERIOR**

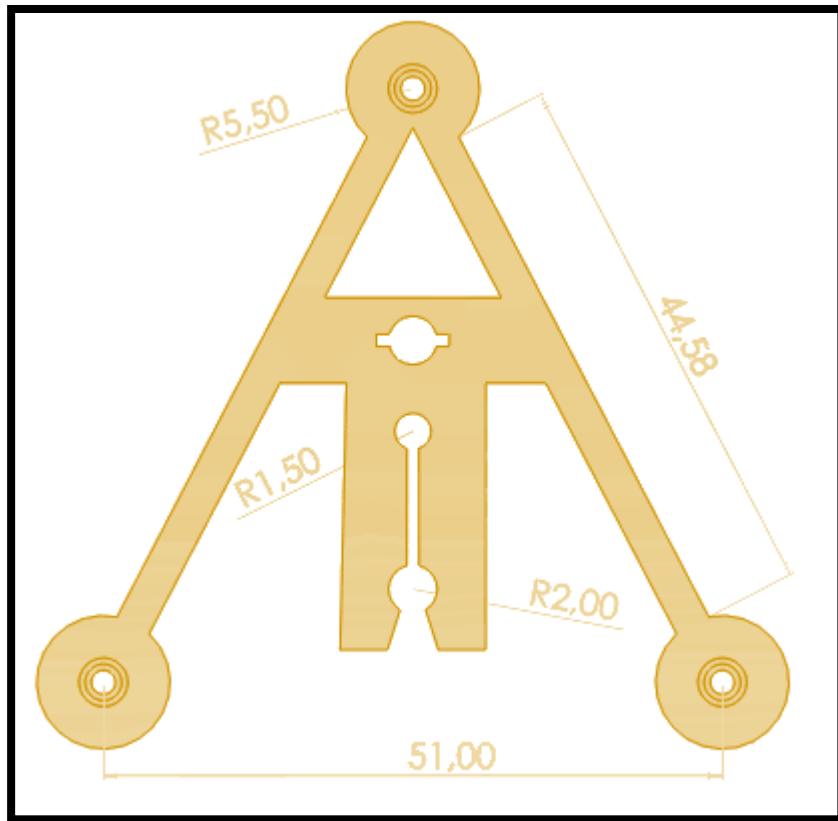


Figura 85  
SOPORTE

## 8.2 ANEXO B. PROGRAMA DE ARDUINO

```
#include <WiFi.h>

#include "FirebaseESP32.h"

#define FIREBASE_HOST "robotinsecto-default-rtdb.firebaseio.com"

#define FIREBASE_AUTH
"yLcbG8Bxo5hThkW5giwEtAjg5zAQB1laj0BpPwWM"

#define WIFI_SSID "NETLIFE- Rocio"

#define WIFI_PASSWORD "1710255595"

// DEFINICION DE PINES

int A1 = 19;    // SALIDA 1.2

int A2 = 18;    // SALIDA 1.1

int PWMA = 5;   // PWMA

int BA = 4;     // SALIDA 1.2

int BB = 16;    // SALIDA 1.1

int PWMB = 17; // PWMA
```

```
// PD
int Kp = 2.8;
int Kd = 0;
int error = 0;
int error_1 = 0;
int u = 0;

int vel = 100;
int distancia = 80;
int veld = 0;
int veli = 0;

// funcionamiento manual
int velm1 = 100;
int velm2 = 100;

// CONFIGURACIONES PRINCIPALES PWM
const int freq = 5000;
const int ledChannel = 0;
```

```
const int ledChannel2 = 1;

const int resolution = 8;

// leds de alerta

#define LED_ESTADO 2 // led de estado

FirebaseData firebaseData;

void setup()
{

    Serial.begin(115200);

    pinMode(A1,OUTPUT);
    pinMode(A2,OUTPUT);
    pinMode(BA,OUTPUT);
    pinMode(BB,OUTPUT);
    pinMode(PWMA,OUTPUT);
    pinMode(PWMB,OUTPUT);
```

```
pinMode(LED_ESTADO,OUTPUT);

ledcSetup(ledChannel, freq, resolution);

ledcAttachPin(PWMA, ledChannel);

ledcSetup(ledChannel2, freq, resolution);

ledcAttachPin(PWMB, ledChannel2);

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

while (WiFi.status() != WL_CONNECTED)
{
  digitalWrite(LED_ESTADO,HIGH);

  delay(50);

  digitalWrite(LED_ESTADO,LOW);

  delay(50);
}

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

Firebase.reconnectWiFi(true);

Firebase.setString(firebaseData,"movimiento","0");

Firebase.setString(firebaseData,"manual","0");
```

```
}
```

```
void loop()
```

```
{
```

```
    Firebase.getString(firebaseData, "manual");
```

```
    if( firebaseData.stringData() == "1")
```

```
    {
```

```
        Firebase.getString(firebaseData, "movimiento");
```

```
        if( firebaseData.stringData() == "1")
```

```
        {
```

```
            digitalWrite(A1,HIGH);
```

```
            digitalWrite(A2,LOW);
```

```
            digitalWrite(BA,LOW);
```

```
            digitalWrite(BB,HIGH);
```

```
            ledcWrite(ledChannel, 120);
```

```
            ledcWrite(ledChannel2, 120);
```

```
        delay(100);
    }

else if( firebaseData.stringData() == "2")
{
    digitalWrite(A1,LOW);
    digitalWrite(A2,HIGH);
    digitalWrite(BA,HIGH);
    digitalWrite(BB,LOW);

    ledcWrite(ledChannel, 120);
    ledcWrite(ledChannel2, 120);

    delay(100);
}

else if( firebaseData.stringData() == "3")
{
    digitalWrite(A1,HIGH);
    digitalWrite(A2,LOW);
    digitalWrite(BA,HIGH);
    digitalWrite(BB,LOW);
```



```
    ledcWrite(ledChannel, 120);  
    ledcWrite(ledChannel2, 120);  
  
    delay(100);  
}  
  
else if( firebaseData.stringData() == "4")  
{  
    digitalWrite(A1,HIGH);  
    digitalWrite(A2,LOW);  
    digitalWrite(BA,LOW);  
    digitalWrite(BB,HIGH);  
  
    ledcWrite(ledChannel, 120);  
    ledcWrite(ledChannel2, 120);  
  
    delay(100);  
}  
  
else if( firebaseData.stringData() == "0")  
{
```

```
digitalWrite(A1,LOW);
```

```
digitalWrite(A2,LOW);
```

```
digitalWrite(BA,LOW);
```

```
digitalWrite(BB,LOW);
```

```
ledcWrite(ledChannel, 0);
```

```
ledcWrite(ledChannel2, 0);
```

```
delay(100);
```

```
}
```

```
digitalWrite(LED_ESTADO,HIGH);
```

```
delay(5);
```

```
digitalWrite(LED_ESTADO,LOW);
```

```
delay(5);
```

```
}
```

```
if( firebaseData.stringData() == "0")
```

```
{
```

```
for(int i = 0; i <= 10; i++)  
{  
    valor = analogRead(s1);  
    total= (total+valor)/11;  
}
```

```
for(int i = 0; i <= 10; i++)  
{  
    valor2 = analogRead(s2);  
    total2= (total2+valor2)/11;  
}
```

```
Serial.println(d);
```

```
error = distancia - d;  
u = Kp*error + Kd*(error - error_1);  
error_1 = error;
```

```
veld = vel + u;  
veli = vel - u;
```

```
digitalWrite(A1,HIGH);  
digitalWrite(A2,LOW);
```

```
digitalWrite(BA,HIGH);
```

```
digitalWrite(BB,LOW);
```

```
ledcWrite(ledChannel, veld);
```

```
ledcWrite(ledChannel2, veli);
```

```
delay(100);
```

```
if(total2 <= 30)
```

```
{
```

```
    digitalWrite(A1,HIGH);
```

```
    digitalWrite(A2,LOW);
```

```
    digitalWrite(BA,LOW);
```

```
    digitalWrite(BB,HIGH);
```

```
    ledcWrite(ledChannel, 120);
```

```
    ledcWrite(ledChannel2, 120);
```

```
    delay(100);
```

```
}
```

```
}
```

```
}
```

## Programa de la cámara como servidor

```
#include "esp_camera.h"

#include <WiFi.h>

//

// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
//         or another board which has PSRAM enabled
//

// Select camera model
#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE
//#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

const char* ssid = "*****";
const char* password = "*****";
```

```

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;

  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;

```

```

config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
//init with high specs to pre-allocate larger buffers
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

#ifdef CAMERA_MODEL_ESP_EYE
    pinMode(13, INPUT_PULLUP);
    pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
}

```

```

    return;
}

sensor_t * s = esp_camera_sensor_get();

//initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);//flip it back

    s->set_brightness(s, 1);//up the blightness just a bit

    s->set_saturation(s, -2);//lower the saturation
}

//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

#ifdef CAMERA_MODEL_M5STACK_WIDE
    s->set_vflip(s, 1);

    s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

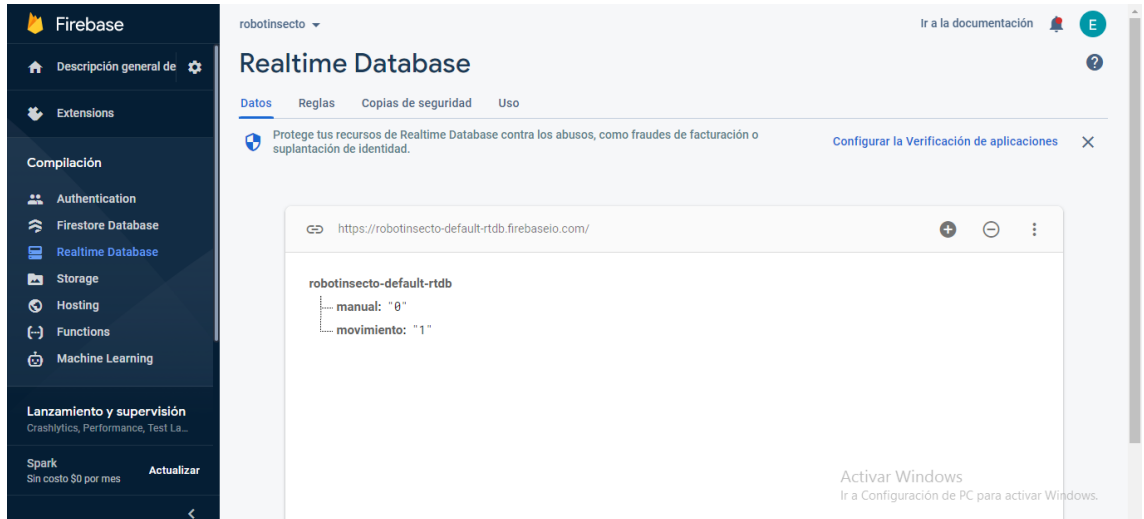
```



```
Serial.println("");  
Serial.println("WiFi connected");  
  
startCameraServer();  
  
Serial.print("Camera Ready! Use 'http://");  
Serial.print(WiFi.localIP());  
Serial.println(" to connect");  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  delay(10000);  
}
```

## 8.3 ANEXO C CONFIGURACIÓN DE FIRE BASE

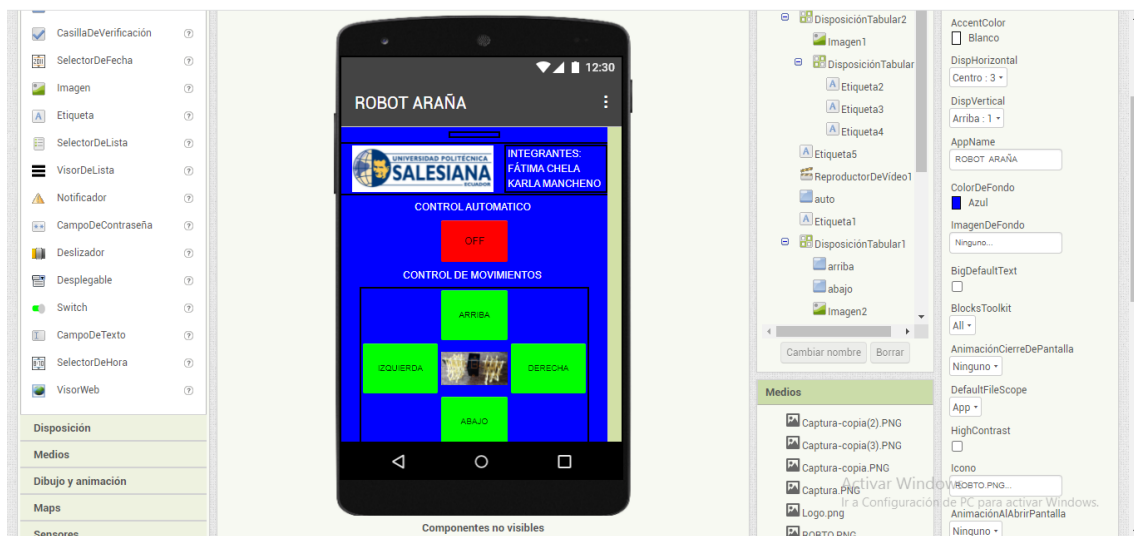
### Variables de Firebase



### Reglas de la base de datos

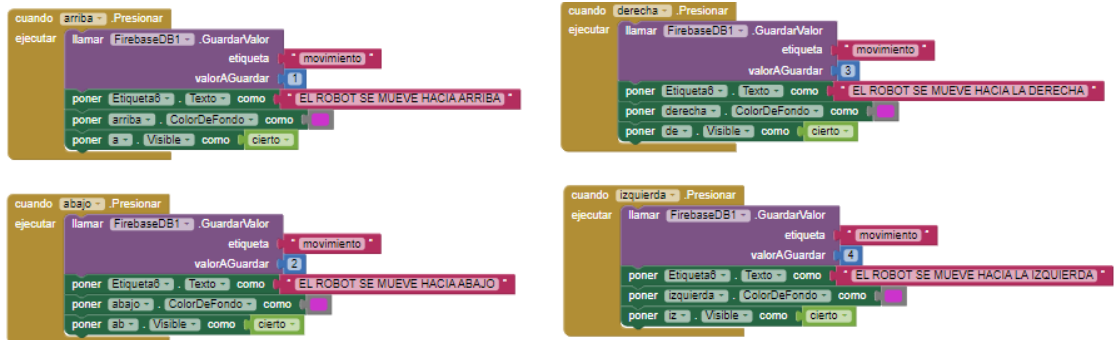


## 8.4 ANEXO D DISEÑO DE INTERFAZ APP INVENTOR

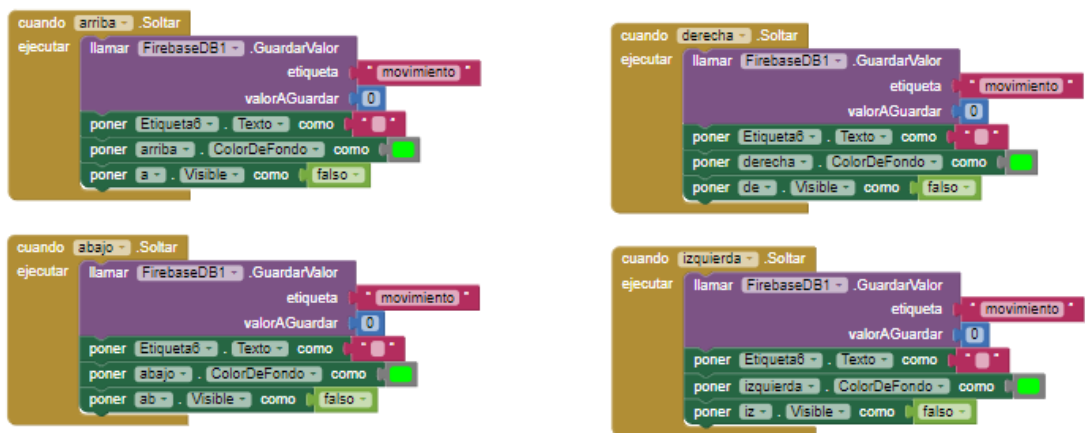


## 8.5 ANEXO E PROGRAMA DE APP INVENTOR

Acciones al presionar los pulsadores en el celular



Acciones al soltar los pulsadores en el celular



Activación de los modos manual y automático

```

cuando auto . Clic
ejecutar
  llamar FirebaseDB1 . GuardarValor
    etiqueta auto
    valorAGuardar 1
  poner auto . Texto como ON
  poner auto . ColorDeFondo como #00FF00
  poner DisposiciónTabular1 . Visible como falso

```

```

cuando auto . ClicLargo
ejecutar
  llamar FirebaseDB1 . GuardarValor
    etiqueta auto
    valorAGuardar 0
  poner auto . Texto como OFF
  poner auto . ColorDeFondo como #FF0000
  poner DisposiciónTabular1 . Visible como cierto

```

### Lectura de la hora en el celular

```

cuando Reloj1 . Temporizador
ejecutar
  poner h . Texto como
  llamar Reloj1 . FormatoDeFechaYHora
    instante llamar Reloj1 . Ahora
    pattern " MM/dd/yyyy hh:mm:ss a "

```

