



**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA**

**FACULTAD DE INGENIERÍA**

**PROYECTO DE INVESTIGACIÓN**

**APP DE ADQUISICIÓN DE DATOS PARA OBTENER FOTOGRAFÍAS DE ALTA CALIDAD**

**PREVIO A LA OBTENCIÓN DEL TÍTULO**

**INGENIERO MECATRÓNICA**

**PRESENTADO POR:**

**21641156 LUIS EDUARDO FLORES OLIVA**

**ASESOR: JOSÉ LUIS ORDOÑEZ**

**CAMPUS SAN PEDRO SULA; ABRIL, 2021**

## DEDICATORIA

*A mis padres, quienes me apoyan desde siempre y nunca dejan de creer en mí:*

*Eduardo Flores y Luisa Oliva*

## **AGRADECIMIENTOS**

Agradezco primeramente a Dios, por darme todas las oportunidades que he tenido en la vida, por darme salud y permitirme estudiar. A mis padres, Eduardo Flores y Luisa Oliva, por todo el apoyo, sacrificio y amor incondicional que me brindan, que a pesar de lo que pase siempre están para mí. A Sophia Ayestas, por apoyarme en cada momento, levantar mi ánimo después de frustraciones con clases y proyectos, por creer en mí cuando yo dudaba. A mis amigos de carrera con los cuales pase por tantas experiencias y siempre están apoyándome y esperando lo mejor para mí. Al Ing. José Luis Ordóñez, por el tiempo dedicado para que mi proyecto de investigación se pudiera realizar con éxito. A todos los catedráticos que me hicieron disfrutar sus clases y aprender cosas nuevas.

## **RESUMEN EJECUTIVO**

Este documento trata sobre un proyecto que da solución a un problema cotidiano en muchos proyectos, una solución para dejar de usar cámaras de mala calidad debido al presupuesto ajustado. Muchos proyectos en todo el mundo usan cámaras para realizar ciertos trabajos como recopilar datos, identificación, grabación y más, y muchos de estos proyectos pueden usar cámaras de baja calidad debido al hecho de que las cámaras de gran calidad suelen ser muy caras y requieren mucha potencia de procesamiento. Con esta aplicación hecha, no es necesario comprar cámaras para sus proyectos, le da la opción de usar la cámara de su teléfono para tomar fotografías de lo que desea. Esta idea vino con el hecho de que hoy en día los teléfonos tienen mejores cámaras cada año que pasa, por lo que si tienes un teléfono con una gran cámara te permitirá tomar excelentes fotografías sin tener que comprar una cámara cara. Lo único es que esta aplicación es solo para teléfonos Android, no funcionará con iPhone. Este artículo habla sobre la creación de una aplicación en MIT APP Inventor que permite que los teléfonos se comuniquen con un Arduino para tomar fotografías mientras viajan con un robot. La aplicación también usa más sensores además de la cámara del teléfono, como el GPS y el acelerómetro en el interior.

## **ABSTRACT**

This document is about a project that gives a solution to an everyday problem in many projects, a solution to stop using bad quality cameras because of tight budget. Many projects around the world use cameras to do certain jobs like gather data, identification, recording and more, and many of these projects might use low quality cameras since great quality cameras are usually expensive and require a lot of processing power. With this app made there is no need to buy cameras for your projects, it gives you the option to use your phone's camera to take pictures of what you want. This idea came with the fact that nowadays phones have better cameras each year that passes by, so if you have a phone with a great camera, it will make possible to take excellent pictures without having to buy an expensive camera. The only thing is that this app is just for Android phones, it will not work with iPhone. This paper talks about the creation of an app in MIT APP Inventor that lets phones to communicate with an Arduino to take pictures while in trips with a robot. The app also uses more sensors besides the phone's camera, like the GPS and accelerometer inside.

# ÍNDICE DE CONTENIDO

I Introducción.....	18
II Planteamiento del problema.....	19
2.1 Precedentes del problema.....	19
2.2 Definición del problema .....	20
2.3 Justificación .....	20
2.4 Preguntas de investigación .....	21
2.5 Objetivos.....	21
2.5.1 Objetivo general .....	21
2.5.1 Objetivos específicos .....	21
III Marco Teórico.....	22
3.1 Ingeniería Mecatrónica.....	22
3.1.1 Sistemas mecatrónicos.....	23
3.2 Microcontroladores .....	24
3.2.1 Microcontroladores Arduino.....	27
3.2.2 Microcontroladores PIC .....	28
3.3 Robótica.....	30
3.4 Adquisición de datos .....	38
3.4.1 Comunicación .....	43
3.4.2 Base de datos.....	45
3.4.3 Aplicaciones Android .....	48
3.5 Visión .....	51
3.5.1 Cámara .....	53

3.5.2 Posicionamiento de cámara .....	54
IV Metodología .....	55
4.1 Enfoque .....	55
4.2 Variables .....	55
4.3 Técnicas e instrumentación .....	56
4.4 Materiales .....	56
4.5 Metodología de estudio .....	56
4.5.1 Incremento 1: Adquisición de Imágenes .....	57
4.5.2 Incremento 2: Adquisición de datos GPS .....	58
4.5.3 Incremento 3: Medir Distancia y Cambios En Estética .....	59
4.6 Cronograma .....	61
V Resultados y Análisis .....	62
5.1 Primer Incremento .....	65
5.1.1 Comunicación Bluetooth .....	71
5.1.2 Utilizar Múltiples Pantallas Para La App .....	72
5.1.3 Utilización de Cámara .....	72
5.1.4 Canvas .....	76
5.2 Segundo Incremento .....	77
5.2.1 Sensor GPS .....	86
5.3 Tercer Incremento .....	88
5.3.1 Sensor Acelerómetro .....	95
5.2.3 Podómetro .....	100
5.2.2 Sensor de Temperatura .....	103

5.2.3 Medición de distancia .....	104
5.4 Pruebas.....	110
5.4.1 Permisos de Aplicación .....	118
VI Conclusiones.....	119
VII Recomendaciones .....	120
Bibliografía.....	121

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Un microcontrolador automotriz típico con diseño de bloques para soportar múltiples protocolos de comunicación, varios motores de cálculo, seguridad y características de seguridad. ....	27
Ilustración 2 Un prototipo integrado para el robot de cuatro ruedas.....	32
Ilustración 3 El robot VineRobot, un robot agricultor móvil. ....	33
Ilustración 4 El robot Tega el cual fue diseñado para interacciones .....	36
Ilustración 5 Escena III del epílogo R.U.R. ....	36
Ilustración 6 Diagrama del sistema del dispositivo móvil para el monitor de señal de potencia. ....	39
Ilustración 7 Arquitectura del sistema de adquisición de datos, donde se encuentra la placa SD registrado a través de una línea de datos única en un archivo .CSV: .....	40
Ilustración 8 Aquí se muestra la ventana de reporte final del APP. ....	50
Ilustración 9 (a) La imagen original. (b) Objeto con color específico.....	52
Ilustración 10 En ese estudio realizado se muestra el ángulo de visión de campo de una cámara estéreo usando el método de multi cambio. ....	53
Ilustración 11 Variable dependiente y variables independientes. ....	55
Ilustración 12 Fotografía de la APP del primer incremento. ....	58
Ilustración 13 Fotografía de la app en el segundo incremento .....	59
Ilustración 14 Fotografía de la APP del tercer incremento.....	60
Ilustración 15 Cronograma de trabajo. ....	61
Ilustración 16 Algoritmo de funcionamiento de la app.....	63
Ilustración 17 Modelo de App Final.....	64
Ilustración 18 Imagen del primer incremento de la app.....	65

Ilustración 19 Primer incremento.....	66
Ilustración 20 Algoritmo de comunicación. ....	67
Ilustración 21 Algoritmo de cámara.....	67
Ilustración 22 Algoritmo de almacenamiento de imágenes.....	68
Ilustración 23 Algoritmo de visualización de imagen. ....	68
Ilustración 24 Pruebas del primer incremento realizadas.....	69
Ilustración 25 Resultado de pruebas del primer incremento.....	70
Ilustración 26 Lógica utilizada con el Arduino .....	71
Ilustración 27 Programaciones de la cámara disponibles en MIT APP Inventor.....	73
Ilustración 28 Extensión de MIT APP Inventor "TaifunCamera". ....	74
Ilustración 29 Configuraciones permitidas por la extensión TaifunCamera. ....	75
Ilustración 30 Imagen donde se muestra la app para visualizar el canvas. ....	76
Ilustración 31 Imagen de la app en su segundo incremento.....	77
Ilustración 32 Segundo incremento.....	78
Ilustración 33 Algoritmo de comunicación. ....	79
Ilustración 34 Algoritmo de GPS. ....	80
Ilustración 35 Algoritmo de configurabilidad. ....	81
Ilustración 36 Pruebas del segundo incremento realizadas con teléfono móvil 1.....	82
Ilustración 37 Resultado de pruebas del segundo incremento realizadas en teléfono móvil 1. ...	83
Ilustración 38 Pruebas del segundo incremento realizadas en teléfono móvil 2.....	84
Ilustración 39 Resultado de pruebas del segundo incremento realizadas en teléfono móvil .....	85
Ilustración 40 Bloque de programación principal para utilizar el sensor de ubicación. ....	86
Ilustración 41 Tercer Incremento de la app. ....	88

Ilustración 42 Algoritmo implementación de acelerómetro.....	89
Ilustración 43 Algoritmo de implementación de fórmula matemática de Haversine. ....	90
Ilustración 44 Algoritmo de configurabilidad tercer incremento. ....	91
Ilustración 45 Captura de pantalla en teléfono móvil al presionar mostrar datos.....	92
Ilustración 46 Captura de pantalla en el teléfono móvil al presionar ocultar datos.....	93
Ilustración 47 Pruebas del tercer incremento realizadas. ....	94
Ilustración 48 Resultado de pruebas del tercer incremento.....	95
Ilustración 49 Opciones principales del acelerómetro.....	96
Ilustración 50 Ejemplo básico para obtener datos de aceleración en X. ....	97
Ilustración 51 Imagen de la app. ....	98
Ilustración 52 Opciones del sensor Podómetro. ....	100
Ilustración 53 Función del sensor podómetro "WalkStep". ....	101
Ilustración 54 App a la hora de prueba. ....	102
Ilustración 55 Pruebas con la app en otro teléfono móvil. ....	103
Ilustración 56 Vista de tramo recorrido para pruebas #1.....	105
Ilustración 57 Vista de tramo recorrido para pruebas #2.....	106
Ilustración 58 Vista de tramo recorrido para pruebas #3.....	107
Ilustración 59 Vista de tramo recorrido para pruebas #4.....	108
Ilustración 60 Teléfonos móviles utilizados para hacer pruebas. ....	110
Ilustración 61 Teléfono móvil utilizado primer, colocado de forma horizontal. ....	111
Ilustración 62 Fotografía obtenida al tener el teléfono móvil con rotación de pantalla activada. .....	112
Ilustración 63 Fotografía obtenida con rotación de pantalla desactivada. ....	113

Ilustración 64 Fotografía capturada por el otro teléfono móvil durante pruebas. ....	114
Ilustración 65 El segundo teléfono móvil a la hora de pruebas con la app, este colocado en forma vertical. ....	115
Ilustración 66 Almacenamiento de las fotografías capturadas.....	116
Ilustración 67 Almacenaje en tercer incremento.....	117

## ÍNDICE DE TABLAS

Tabla 1 Comparación de ubicaciones de GPS capturadas de diferentes fuentes. .... 87

Tabla 2 Comparación de distancias medidas entre la app y Google Maps..... 109

## ÍNDICE DE ECUACIONES

Ecuación 1 Formula de Haversine.....	104
--------------------------------------	-----

## I INTRODUCCIÓN

Un problema en proyectos de investigación actualmente es que en muchas ocasiones personas utilizan cámaras de baja calidad o incrementan el costo del proyecto debido a la compra de una cámara. Con la app a desarrollar se solucionará el problema permitiendo que personas utilicen su dispositivo móvil para capturar fotografías en sus proyectos. De esta investigación nace la interrogante, ¿por qué utilizar una cámara del teléfono en vez de una cámara especial para el trabajo? Debido a que hoy en día los teléfonos celulares salen al mercado con cámaras cada vez mas profesionales esto permite que el teléfono celular sea mas que suficiente o que incluso sobrepase expectativas en el desempeño de una cámara en un proyecto. La calidad de la fotografía dependerá del teléfono móvil utilizado debido a que hay diferentes gamas de teléfonos como gama baja, media y alta pero debido a que cada año la tecnología de cámaras en teléfonos avanza esto permite que se obtengan imágenes deseadas incluso en un teléfono de gama baja. La app igual ofrecerá al usuario funcionalidad de GPS con el cual se obtendrán las coordenadas GPS de donde se captura la imagen e incluso se utilizará el GPS para obtener diferentes coordenadas para poder utilizar la fórmula matemática de Haversine, la cual es utilizada para calcular distancias utilizando coordenadas de dos puntos en el mapa, y de esta manera obtener con la app distancias recorridas en viajes realizados por proyectos.

## II PLANTEAMIENTO DEL PROBLEMA

### 2.1 PRECEDENTES DEL PROBLEMA

Se realizó una cámara estereó que se activa al detectar movimiento para una experiencia de captura en 3D. Donde debido al alto precio en buen equipo de cámaras, para realizar un sistema así se tuvo que desarrollar un sistema de cuerpo de cámara que genera puntos nube 3D en tiempo real de bajo costo y eficiente en consumo de energía (Islam et al., 2018). Se hicieron pruebas con diferentes cámaras para la tele operación del robot y se observaron los diferentes resultados de diferentes cámaras, se encontró que agregando un lente ojo de pescado a una cámara de bajo presupuesto mejora bastante la calidad dejando buenos resultados (Shiroma et al., 2004). Pero viendo esto se observa que las cámaras en robots son caras si buscamos de buena calidad de imagen y alta resolución.

En dos proyectos de trimestres anteriores que se realizaron, en donde ambos eran robots, uno tomaba fotografías y el otro utilizaba un módulo GPS, en estos la calidad del sensor y las fotografías no fueron buenas, donde el sensor GPS no tenía precisión y la calidad de imágenes era muy baja.

Factores como luz, fondo, sombras, ruido y problemas de conexión afectan la calidad de la imagen (Zhonghan Zhang et al., 2019).

En problemas de conexión entre un APP realizada para proyecto, En este proyecto se quieren detectar problemas mencionados anteriormente por medio de un sistema diseñado y mostrado en el APP. El sistema utiliza Arduino para enviar información por medio de bluetooth a un dispositivo Android. El escenario se basa en sistema automático de mantenimiento en tiempo real donde se muestra información detallada de problemas (Gino Alvarado et al., 2017).

## **2.2 DEFINICIÓN DEL PROBLEMA**

A la hora de realizar un robot móvil con cámara normalmente se usan cámaras de baja calidad, sin embargo, se pueden implementar cámaras que capturen imágenes en altas resoluciones, pero esas cámaras son bastante caras y usualmente requieren alto poder de procesamiento (Shiroma et al., 2004). Hasta las cámaras de tele operación tienen sus diferentes FOV, resolución, ángulos de vista y etc.

Pueden haber problemas de comunicación o retrasos en proyectos en la parte de comunicación de un APP con un Arduino o algún otro microcontrolador (Gino Alvarado et al., 2017).

## **2.3 JUSTIFICACIÓN**

Con este proyecto se da solución al problema en costos sobre comprar una cámara de alta calidad para un robot, ya que será posible utilizar la cámara de un celular con el robot, y como se sabe hoy en día los celulares tienen cámaras con excelente calidad fotográfica. La cámara del teléfono será utilizada por medio de un APP que se desarrollará la cual estará conectada con el robot. Esto ayudara a tener un mejor monitoreo de lo que el robot observara debido a que no se usara una cámara de baja calidad para tomar las imágenes. Esta APP ayudara a personas que tengan un presupuesto ajustado y no quieran gastar gran cantidad de dinero en cámaras para robots y proyectos.

El propósito de utilizar un microcontrolador con la aplicación es que de esa manera se pueden realizar una infinidad de programaciones y escenarios diferentes para que la aplicación capture una fotografía, por ejemplo un robot que quiera utilizar la aplicación cuando su sensor de temperatura esta en cierto nivel y requiera una fotografía para monitoreo, o que con un sensor ultrasónico al estar a cierta distancia de algún objeto este requiera que la app tome una fotografía.

De igual la aplicación permitirá utilizar sensor de GPS y acelerómetro del teléfono móvil los cuales tienen buena precisión ya que estos se utilizan para navegación en teléfonos cada día por personas en el mundo.

## **2.4 PREGUNTAS DE INVESTIGACIÓN**

- 1) ¿Qué problemas hay normalmente con el uso de cámaras en robots?
- 2) ¿Cuáles son los parámetros para que la app tome fotografías?
- 3) ¿Cuál método es el adecuado a utilizar para medir recorridos con la aplicación?

## **2.5 OBJETIVOS**

### 2.5.1 OBJETIVO GENERAL

Diseño y elaboración de un APP para Smartphones Android con el cual por medio de esta se utilizará la cámara del teléfono para obtener las fotos que se necesiten y se guarden en el teléfono.

### 2.5.1 OBJETIVOS ESPECÍFICOS

- 1) Determinar problemas con respecto a calidad fotográfica que se encuentran con cámaras en proyectos de robótica.
- 2) Determinar los parámetros de sensores implementados a cumplir por la aplicación para capturar fotografías.
- 3) Determinar la manera adecuado para medir distancias de recorrido con la aplicación.

## III MARCO TEÓRICO

### 3.1 INGENIERÍA MECATRÓNICA

La carrera de ingeniería mecatrónica es un área con múltiples disciplinas donde se combinan varias ingenierías, entre estas: la ingeniería eléctrica, la computación y la mecánica, y la combinación de estas refiere a un set de habilidades necesarias en lo contemporáneo, en la industria de fabricación automatizada avanzada. La mecatrónica es una palabra de moda en la industria con robótica e ingeniería electromecánica. La robótica, sistemas de control, y sistemas electromecánicos todos están en la mecatrónica (*Michigan Technological University, 2021*).

En la ingeniería mecatrónica se observa la variedad de elementos que se aprenden y ponen en práctica, esto abre puertas en tantos lados gracias a la cantidad de cosas que un ingeniero puede hacer, igual cabe mencionar que un ingeniero mecatrónico puede usar softwares de diseño 3D, como Fusion 360 y SolidWorks, estos ayudan de una gran manera en proyectos y trabajos a crear piezas y prototipos necesarios. También está la programación que se aprende en la ingeniería mecatrónica, se pueden aprender varios lenguajes de programación los cuales luego permiten al ingeniero hacer proyectos de APPs, softwares y automatización en cosas, para la automatización entran los microcontroladores de los cuales se hablara más adelante.

¿Y qué hacen estos ingenieros mecatrónicos? Estos especialistas pueden hacer muchas cosas a través de múltiples disciplinas en ingenierías e igual tienen una base sólida que los ayuda a crecer.

Los especialistas mecatrónicos saben de fundamentos de la ingeniería mecánica y eléctrica, hablan ambos lenguajes, eso permite que trabajen en ambos ámbitos (*Michigan Technological University*, 2021).

### 3.1.1 SISTEMAS MECATRÓNICOS

Dentro de la ingeniería mecatrónica están los sistemas mecatrónicos, donde el diseño y desempeño de estos dependen de muchos factores y subsistemas como engranajes, componentes hidráulicos y motores eléctricos. Estos últimos son considerados como un dispositivo principal en la conversión de potencia electromecánica, así su diseño se vuelve de prima importancia. En este paper también dice que el manejo efectivo del calor en componentes es una de las principales consideraciones en un diseño eficiente que se debe tomar en cuenta en los sistemas mecatrónicos (Assaad et al., 2016).

Que un sistema mecatrónico tenga eficiencia térmica es algo tan importante que todo ingeniero debe saber ya que cuando los componentes en un proyectos o dispositivo se calientan mucho estos no funcionan igual, se sabe que entre más alta la temperatura de un sistema mecatrónico, mas es el desempeño que se pierde.

En sistemas mecatrónicos, la conversión de potencia produce perdidas de calor y estas eventualmente provocan que la temperatura del sistema incremente. Estas pérdidas impactan el desempeño del sistema de dos maneras; directamente donde el calor es transmitido al sistema por una de los fenómenos de transferencia de calor( conducción, convección y radiación); o indirectamente impactando la eficiencia y por lo tanto, el torque transmitido o el consumo eléctrico (Assaad et al., 2016).

Un punto esencial en investigación hoy en día es el problema de la formalización de sistemas complejos. Estos sistemas no pueden ser controlados sin una etapa preliminar de modelado. Actualmente, el riesgo de una decisión incorrecta sea tomada excede más allá del costo de construcción del modelo (Zamula & Kavun, 2017).

Dentro de los sistemas mecatrónicos también se encuentran la intralogística, la cual permite realizar cosas como optimizar, automatizar, integrar y manejar el flujo de información y más cosas dentro de las paredes de un centro de distribución.

Debido a la gran cantidad de desarrollos en avances de tecnologías mecatrónicas, las redes robóticas han sido posibles en muchas aplicaciones incluyendo monitoreo ambiental, vigilancia de campos, búsqueda y recates de objetivos, detección e interceptación de intrusos, exploración de regiones inseguras y más (Hu et al., 2013).

La intralogística ha dependido de los productos mecatrónicos por muchas décadas. Sin embargo, en publicaciones no se enfocan mucho la intralogística hacia el lado de los productos mecatrónicos. La modularización y plataformas de producto han sido ampliamente aceptadas en productos de clientes como un medio para permitir ambos, la individualización y la reducción de costos simultáneamente. Esta estrategia ha sido bien aceptada en producción de masas de dispositivos como computadoras personales, electrodomésticos y más. Sin embargo, con respecto a plantas industriales, las cuales pueden funcionar por muchas décadas, un factor crucial siguiente a costos de inversión es el costo total mínimo de propiedad el cual se puede obtener parcialmente al tener por medio de la utilización óptima del espacio. En ese momento entra la intralogística, la cual específicamente en sistemas de almacenamiento, una combinación de producción en masa se aproxima y la individualización es necesaria para alcanzar ambos los costos iniciales bajos y ganancias optimizadas (Sadlauer, 2019).

### **3.2 MICROCONTROLADORES**

Los microcontroladores se utilizan en tantos productos electrónicos y proyectos hoy en día, estos son algo tan importante para la ciencia e ingenierías. ¿Pero que es un microcontrolador?

Un microcontrolador es un circuito integrado compacto diseñado para gobernar una operación específica en un sistema incrustado. Un microcontrolador típico incluye un procesador, memoria, entradas y salidas en un chip (Ben Lutkevich, 2019).

Estos microcontroladores se encuentran en vehículos, máquinas de oficinas, maquinas industriales, equipos médicos, robots, televisores, computadoras, consolas de videojuegos, teléfonos celulares y una amplia lista con más.

Los microcontroladores son clave en las ingenierías como la mecatrónica, biomédica, sistemas computacionales, electrónica y otras. Ya que con estos se logra realizar proyectos de investigación, proyectos para clases e incluso proyectos en el trabajo profesional. De igual manera se puede utilizar para realizar cosas personales como automatizar partes del hogar, hoy en día solo se compran los materiales necesarios en Amazon y eso da la facilidad de obtener lo que se necesita para por ejemplo automatizar lavamanos en el hogar para evitar el malgasto del agua, o automatizar luces y así ahorrar energía eléctrica. Como se observa hay tanto que se puede realizar por medio de estos dispositivos, estos van acompañados de programación y hardware. Ahora la pregunta es, ¿cómo funcionan?

Los microcontroladores se encuentran dentro de un sistema para controlar cierta función en un dispositivo. Esto lo hace interpretando datos que recibe por medio de sus periféricos I/O usando su procesador central. Esta información temporal que el microcontrolador recibe es almacenada en su memoria de datos, donde el procesador la accesa y usa instrucciones almacenadas en su memoria de programación para descifrar y aplicar los datos recibidos. Luego utiliza sus periféricos I/O para comunicar y promulgar la acción apropiada. Los microcontroladores son utilizados en una basta cantidad de dispositivos y sistemas. Incluso los dispositivos usualmente utilizan múltiples de estos para trabajar varias tareas diferentes a la vez. Por ejemplo, están los carros que tienen varios microcontroladores con los que se controlan los sistemas de frenos, control de tracción, inyección de combustible, control de suspensión entre otros. En donde estos microcontroladores se comunican entre sí para informar acciones correctas (Ben Lutkevich, 2019).

Los microcontroladores cada vez se están volviendo más complejos debido a la integración de más núcleos al procesador, incrementos en AMS (señales análogas y mixtas), manejos de potencia complejos, variedad de interfaces de comunicación, soluciones de hardware innovativas como

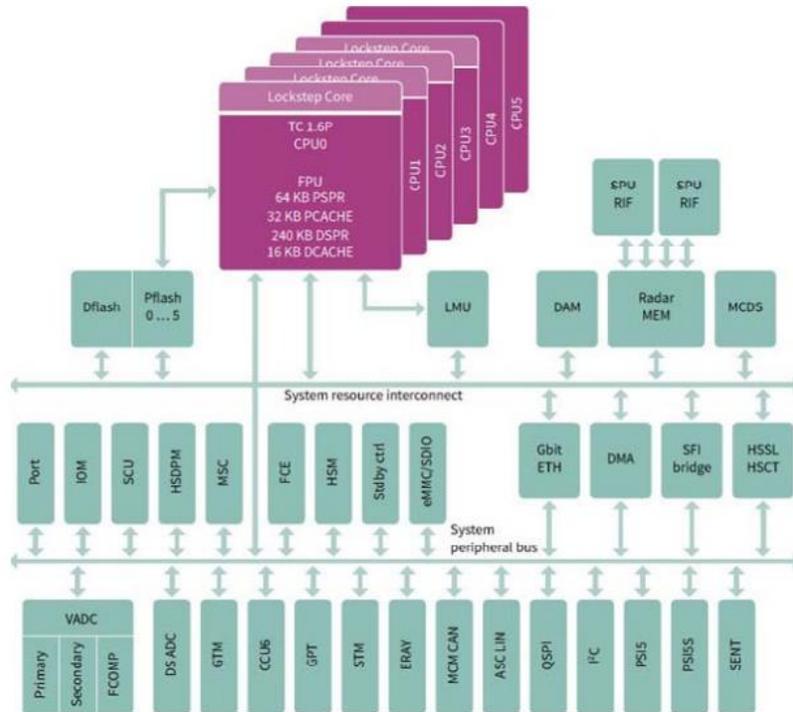
ADAS (Sistemas asistentes de manejo avanzado) y más módulos configurables. (Moharikar & Guddeti, 2017).

Actualmente la mayoría de microcontroladores son programados por medio de alguna variante del lenguaje de programación C. El lenguaje de programación C provee un balance amigable entre el control del hardware del microcontrolador que tiene el programador y la eficiencia del tiempo que usa a la hora de escribir su programación (Barrett, 2020).

Un microcontrolador es esencialmente un microprocesador que es equipado con circuitos Input/Output (I/O) y periféricos incorporados, permitiendo una interface directa con dispositivos del mundo real como un switch, solenoide neumático, motores, sensores y otros dispositivos (Burhan et al., 2016).

Los microcontroladores son uno de los dispositivos más importantes a la hora de implementar comunicación y sistemas de control electrónicos. Un microcontrolador es un tipo de microprocesador que necesita un soporte mínimo de chips. Estos son capaces de almacenar y correr programas (Premadi et al., 2009).

En la ilustración 1 se muestra el número y diversidad de bloques de diseño en un microcontrolador automotivo están incrementando.



**Ilustración 1 Un microcontrolador automotriz típico con diseño de bloques para soportar múltiples protocolos de comunicación, varios motores de cálculo, seguridad y características de seguridad.**

Fuente: (Moharikar & Guddeti, 2017).

### 3.2.1 MICROCONTROLADORES ARDUINO

El Arduino es un microcontrolador basado en C, el cual provee una interfaz amigable para el usuario para desarrollar un programa de una manera más rápida y eficaz, transformar el programa a código de maquina y luego cargar el código de la maquina en el procesador del Arduino de una manera práctica (Barrett, 2020).

En la Universidad Tecnológica Centroamericana en San Pedro Sula, Honduras, se utiliza Arduino en ciertas clases en ingenierías, sobre todo en la mecatrónica donde se llevan varias clases de programación y en proyectos se utilizan, estos abren la mente a estudiantes y motivan a buscar cómo hacer más proyectos desde el hogar hasta para clases. El Arduino brinda la facilidad que es fácil de adquirir y hay una gran librería en internet e incluso videos explicando cómo usar cierto sensores y actuadores e incluso de comer hacer proyectos en específico.

El concepto de Arduino es un hardware de código abierto desarrollado por el equipo visionario de Arduino conformado por Massimo Banzi, David Cuartilles, Tom Igoe, Gianluca Martino y David Mellis en Ivrea, Italia. La meta de este equipo era desarrollar un hardware microcontrolador y software que fueran fáciles de usar para que ese poder de procesamiento estuviera listo para quien sea (Barrett, 2020).

Una gran ventaja de Arduino es que a pesar de que hay varios modelos, todos utilizan la misma programación y el mismo software entonces esto permite a usuarios migrar sus avances y desarrollos de un dispositivo a otro de una forma bien simple (Amestica et al., 2019).

Toda la línea de productos de Arduino está basada en el concepto visionario de un software y hardware de código abierto. Los desarrollos en hardware y software están compartidos abiertamente entre usuarios para estimular nuevas ideas y avances en el concepto de Arduino (Barrett, 2020).

### 3.2.2 MICROCONTROLADORES PIC

El microcontrolador PIC fue desarrollado en el año 1993 por la empresa microchip technology. El término PIC significa controlador de interfaz periférica y originalmente fue desarrollado para soportar computadoras PDP para controlar sus dispositivos periféricos. Estos dispositivos son muy rápidos y fáciles de ejecutar un programa comparado a otros microcontroladores (Admin, 2019).

PIC que pertenece a la familia de los microcontroladores es fabricado por Microchip Technology Inc., el líder en fabricación de microcontroladores y semiconductores análogos. PIC simplifica el diseño de lógica y sistema de control, permitiendo el diseño de comportamientos complejos a ser incrustados en una pieza de instrumentos electrónicos (Burhan et al., 2016).

Usar PIC es más complejo que Arduino ya que requiere mucha investigación al haber poca información obtenida en comparación. Este microcontrolador es utilizado en la clase de Microprocesadores en la Universidad Tecnológica Centroamericana, clase que la llevan los estudiantes de ingeniería en mecatrónica. Se ve cómo utilizar estos microcontroladores por medio de Proteus una app para simulación y MikroC que es donde se hace la programación a usar. Luego se realizan proyectos con estos.

Las PIC igual se pueden utilizar en gran variedad de proyectos y trabajos, hay tantos proyectos de investigación en revistas al igual que papers utilizando esos microcontroladores. Uno de los proyectos es "Hardware Implementation of Single-Phase Fullbridge VSI Switched by PSO Based SHEPWM Signals Using Embedded PIC Microcontroller", en el cual con esta PIC se generaba el SHEPWM usando valores digitales almacenados de angulos de conmutación calculados en varios valores de Mis. La PIC que se utilizó en este proyecto fue la PIC16F676 la cual es usada como el controlador incrustado para la fase única VSI. Es un microcontrolador CMOS de 14 pines, flash-based 8 bits. Sus características principales son un reloj de oscilación de 0 a 20 MHz, 12 pines I/O, 10-bit ADC, memoria de programación de 1024 palabras, 64 bytes de SRAM y 128 bytes de memoria de datos EEPROM (Ganguly et al., 2017).

Estos microcontroladores pueden trabajar con PLC. Un PLC es un dispositivo como computadora especializado el cual es usado en procesos industriales de control, este reemplaza bancos de relees electromagnéticos (Sarma & Bordoloi, 2017).

En el proyecto este microcontrolador PIC fue utilizado para generar la conmutación de señales para una fase única VSI. Este microcontrolador es programado para producir la señales SHEPWM para los cuatro switches del invertir (Ganguly et al., 2017).

El proyecto donde se usó PLC con PIC se llama "Speed Control of A DC Motor Using PLC and PIC Microcontroller". En este proyecto los motores los controla un controlador PID simulados en una plataforma de PLC de Siemens. Y el microcontrolador PIC es utilizado para integrar un numero de funciones como métodos ópticos para la medición y mostrar la velocidad del motor, también para dar retroalimentación e implementar el controlador PID al proyecto. El motor es activado por PWM que es mandado por el microcontrolador. El controlador PID es programado en el microcontrolador PIC y el control de velocidad es realizado por medio de una técnica de PWM (Sarma & Bordoloi, 2017).

Microchip es el único proveedor de estos semiconductores innovativos dentro de 8-, 16- y 32-bit MCUs, DSCs y MPUs, proveyendo la última selección de desempeño escalable. Tampoco hay necesidad de aprender un nuevo ambiente de nuevo desarrollo o empezar los códigos desde cero dentro del ecosistema de Microchip ya que como hay herramientas comunes de influencia a través

de múltiples diseños. Esto ayuda a preservar la inversión en el desarrollo del código reusando el firmware de aplicación, incluso cuando el diseño requiere un cambio (Microchip, n.d.).

### **3.3 ROBÓTICA**

La robótica es algo sumamente importante en nuestras vidas, cada día que pasa se hacen avances en este ámbito a nivel global y esta se encuentra en muchos lados, hay robots pequeños como los que aspiran hogares, al igual que hay robots grandes que ayudan a limpiar el hogar, sin embargo la robótica no solo es en sí robots, también entran otras cosas como prótesis ya sea de brazo, mano y piernas, también hay peleas de robots donde pelean hasta que un robot no pueda seguir, hay peleas sumo de robot donde se ponen ambos en un ring circular y el primero en dar vuelta al oponente o sacarlo de la pista gana. En si la robótica está en muchos lados y a medida pasen los años cada vez será más el impacto que tendrán en el mundo y más la necesidad humana de un robot.

Los robots industriales son bien interesantes, esos robots usualmente son desarrollados de acuerdo a la metodología convencional de desarrollo de software de códigos en desarrollo para controlar hardware, compilar y ejecutar códigos. Los robots industriales pueden ser desarrollados fácilmente con una metodología de desarrollo porque ellos no tienen que considerar el ambiente a su alrededor (Cho et al., 2017).

Los robots humanoides por otra parte son un área de investigación relativamente nueva ya que se aspira a desarrollar sistemas robóticos antropomórficos con el objetivo de ser utilizados en público y ambientes de hogar. En contraste con un manipulador industrial, un robot humanoide es esperado que interactúe y coopere con seres humanos (Folgheraiter & Aubakir, 2018).

En la robótica también están los exoesqueletos, estos a comparación de vehículos con ruedas, son dispositivos asistivos que tienen muchas ventajas potenciales, así como permitir que el usuario atraviese superficies de terreno irregulares (Walsh et al., 2007).

Los robots en agricultura están mostrando buenos resultados. Sin embargo, los robots agrícolas siguen siendo experimentales y aún no está cerca el día en que los implementen en operaciones a grandes escalas. Organizaciones a través del mundo están experimentando con

robots móviles, manipuladores, humanoides y drones en múltiples tareas de agricultura. Los robots están siendo probados en cosechas, recoger, pastoreo de animales y otras actividades agrícolas (Hajjaj & Sahari, 2016).

Los robots de agricultura cada vez llaman más la atención a investigadores. Estos robots tienen potencial reemplazando el trabajo a mano en tareas simples y repetitivas. También se cree que los robots de agricultura pueden mejorar la precisión de operación para tareas específicas. Muchos tipos de estos robots han sido mencionados en otros proyectos como robots para recoger frutas, robots para el control de hierba, robots de monitoreo y más (Z. Fan et al., 2017).

Los robots para agricultura, los cuales pueden operar en varios terrenos duros, poseen un set único de desafíos de ingeniería y tecnología que usualmente no están presente para los robots usados dentro de casas y edificios. Estos desafíos específicamente orientados a la agricultura se centran en las actividades de navegación específica, procesamiento de imágenes y manejar terrenos peligrosos (Hajjaj & Sahari, 2016).

Los robots agricultores móviles son partes importantes de ellos robots agricultores. Muchos prototipos de robots móviles agricultores han sido un éxito como el "BoniRob" y "Bin-Dog". Pero el diseño e implementación de este tipo de robot aun es una pregunta abierta. Muchos de los prototipos reportados comparten las mismas plataformas móviles con los robots de industria o plataformas comerciales de investigación, otros que diseñar mecanismos específicos y soluciones de motor de acuerdo a los requerimientos de escenarios agricultores. Como resultado, robots móviles agricultores existentes sufren de una forma inapropiada y transitabilidad insuficiente. Por eso diseñaron un robot para solucionar este problema el cual tiene una estructura compacta y motores potentes, este puede viajar de manera inteligente en campos abiertos de terreno y realizar tareas con ayuda de sus sensores de percepción ambiental y actuadores. (Z. Fan et al., 2017).



**Ilustración 2 Un prototipo integrado para el robot de cuatro ruedas.**

Fuente: (Z. Fan et al., 2017).

Se observa el robot "Agrobot", realizado para varios propósitos. Desde el inicio los jóvenes que trabajaron en este robot se dividieron el trabajo en dos partes principales; unos en diseñar el robot y otros en desarrollar un sistema de nube que se conectaría al robot y esto a granjeros vía teléfonos celulares. Este robot lo hicieron por menos de su presupuesto de 2000\$ y lo podían controlar por medio de un control inalámbrico. Lo interesante es que ellos cumplieron con el robot agricultor solo que les quedó inconcluso que no lograron terminar de conectar el robot a la nube. (Durmus et al., 2015)

En ese mismo año otro equipo desarrollo un robot llamado "Agribot", el cual fue diseñado para realizar varias tareas. Su diseño era hexápodo así que el robot podía moverse fácilmente en cualquier dirección. Este robot se enlazaba al mundo con Wifi, así se conectaba a su operador, también el robot por medio de Wifi se podía conectar a otros robots y coordinar las tareas. Este robot hacia las tareas de una manera excelente pero su limitación fue la dependencia de Wifi. Esto solo permitía al robot operar en unas áreas limitadas (Amer et al,2015).

En la ciudad de Sapporo, Japón se inició un taller de robots llamado "Bio-Robotics III" donde el enfoque principal de estos talleres era promover un intercambio de descubrimientos científicos en el uso de robótica y maquinaria automatizada en sistemas de bio producción a nivel mundial (Noguchi & Murase, 2008).

En un estudio realizaron una mano robótica antropomórfica llamada "OCU-Hand" la cual fue hecha con un peso y tamaño para compararse a una mano humana y así también tener la misma efectividad, esta mano consta de 5 dedos y 15 grados de libertad. La mano se controlaba por medio de un guante liviano y cabe destacar que era tele operada (Mahmoud et al., 2012).

En 2014 un equipo realizo un robot para viñedo, este robot se llamaba "VineRobot" el cual estaba diseñado para ayudar con la producción y agricultura de vino por "the Seventh Framework Program". Este robot fue hecho para andar libre en el viñedo de forma autónoma recolectando datos en el estado del viñedo, así como el desarrollo vegetativo, el estado del agua, la composición de las uvas y otros datos importantes. Las imágenes adquiridas y los datos generados por el robot eran transferidos de manera inalámbrica a los trabajadores de las uvas y a técnicos que monitoreaban el robot en tiempo real. Luego estos datos eran procesados en tiempo real y usados para enfocarse en esfuerzos y recursos, como el equipo y mano de obra, determinar donde era necesitado en el área y así cumpliendo el significado del término "Precision Agriculture" (Tardaguila et al,2014).



**Ilustración 3 El robot VineRobot, un robot agricultor móvil.**

Fuente: (Tardaguila et al,2014).

"Precision Agriculture" puede ser definido como la estrategia de manejo que utiliza información tecnológica para traer datos de múltiples fuentes para influir en decisiones asociadas con la producción de cosechas. Esta estrategia ha sido desarrollada y adoptada no solo por países como los Estados Unidos, la Unión Europea y Australia donde tienen grandes áreas de granjas, sino que también en países como Japón, China, Corea donde sus granjas son pequeñas (Li & Chung, 2015).

Otro grupo de investigadores discutían que incorporar muchos sistemas electrónicos en robots de agricultura incrementaría sus funciones, pero que esto haría un gran incremento en tamaño, costos y peso del robot al igual que en su fiabilidad. Entonces propusieron que para que un robot agricultor fuera exitoso, este tendría que tener un balance entre funcionalidad y complejidad. Desarrollaron su propia arquitectura de sistema llamada el proyecto "RHEA". El cual se ocupa del diseño y control de robots, comunicación Wifi, interface de software con usuarios humanos, básicamente todos los puntos más importantes en robots de esta área (F. Emmi et al, 2014).

Hay muchas cosas interesantes en la robótica y esto solo es en la parte enfocada a agricultura. Algo interesante del ultimo robot mencionado es que el robot a pesar de andar en todo el campo haciendo sus tareas, el no interrumpía a los trabajadores del viñedo si no que más bien el hacia sus tareas de manera independiente.

Una de las principales razones detrás de buscar la automatización de granjas es para ahorrar tiempo y energía requerida para realizar esas tareas tan repetitivas e incrementar la productividad del manejo tratando cada cosecha individual utilizando el concepto de "precisión farming". En un proyecto de investigación de uso el "Agribot" mencionado previamente para realizar la función de sembrar semillas lo cual no es tarea fácil (Naik et al., 2016).

Ahora están los robots móviles autónomos los cuales se consideran así si se pueden mover a donde sea sin importar el cambio de ambientes. Para que un robot haga eso se requiere que el camino sea planeado. Igual debe de poder re planear su camino en caso de que necesite evitar ciertos obstáculos que pueden ser encontrados. Los robots móviles obtienen el conocimiento de su ubicación y su ambiente y planean una trayectoria en línea para seguir el objetivo que se está moviendo. Un robot de cuatro ruedas para seguir un objetivo móvil en ambientes dinámicos debe ser diseñado usando una técnica de planeación de caminos RRT. Para diseñar tal robot, se requiere un método para computar un camino dinámico y también para replantearlo donde este el obstáculo. El sistema requiere un sensor para sentir esa información. El camino requerido para que el robot se mueva es entrada al planeador de camino y este computa la velocidad y manejo de la velocidad correspondiente del camino. El camino actual a ser recorrido y la posición actual de la planta es comparada y el error es dado al controlador el cual causa el movimiento del robot hacia el siguiente o dirección del objetivo (Jayasree et al., 2017).

Se observa que para que un robot pueda explorar de manera autónoma este debe llevar múltiples sensores y programación compleja así este funciona de la mejor forma para evitar obstáculos y llegar a su destino. Hoy en día se busca hacer robots autónomos para industrias, proyectos de exploración de áreas, mantenimiento e incluso más cosas.

Un proyecto interesante del cual se hablaron datos arriba es el de un robot tipo carro el cual es diseñado para ambientes dinámicos usando una técnica de planeamiento de caminos RRT el cual incluye RRT y suavizado. Utilizando un esquema, un camino óptimo es obtenido de manera eficaz y rápida. Se probaron varios escenarios con diferentes posiciones del robot para verificar las maneras como se movía y como replanteaba caminos este robot. Esta técnica utilizada les dio buenos resultados, pero cabe aclarar que hay muchas técnicas que se pueden utilizar para el planeamiento de caminos (Jayasree et al., 2017).

Un robot "QuicaBot", el cual es un robot automatizado para garantizar la calidad de construcción de lugares el cual observa huecos, grietas, uniformidad, alineaciones y problemas de inclinación. El objetivo principal de este robot es sistematizar el trabajo manual de inspección convirtiéndolo en algo automatizado, lo cual resulta en reportes de inspección más confiables, también acelerando el proceso de inspección resultando a la vez en reducción de costos (Yan et al., 2017).

Incluso existen robots sociales. El robot "Tega" es un robot social creado por un diverso equipo de ingenieros, desarrolladores de software y artistas en el grupo de robots personales en el MIT Media Lab. Es un robot con piel de peluche, apariencia con color vivo, y fue desarrollado para tener interacciones largas con niños. El movimiento de este robot fue inspirado por el principio de animación "squash and stretch", creando movimientos naturales y orgánicos mientras manteniendo la cuenta de actuador baja. "Tega" tiene 5 grados de libertad en los cuales incluye mover la cabeza hacia arriba y abajo, inclinar la cintura hacia derecha e izquierda, inclinar cintura hacia enfrente y hacia atrás, mover el cuerpo hacia arriba o abajo, y el cuerpo hacia derecha e izquierda. Esas uniones son combinacionales y permiten que el robot se exprese de una manera más consistente, rápida y confiable (Westlund et al., 2016).



**Ilustración 4 El robot Tega el cual fue diseñado para interacciones con niños pequeños.**

Fuente: (Westlund et al., 2016).

Es fascinante como este robot tenía una gran cantidad de expresiones faciales y movimientos del cuerpo que podían ser disparados en el robot, él se podía reír, emocionarse y frustrarse. También podía hablar por medio de audio pregrabados y más cosas.

El robot puede correr de manera autónoma o puede ser operado remotamente por una persona por medio de una interfaz de teleoperación. También puede operar en baterías hasta por seis horas antes de necesitar ser recargado, lo cual permite una forma más fácil de probarlo en el campo (Westlund et al., 2016).

Hoy en día existen robots autónomos que son actores en teatros robóticos, ante los ojos de los roboticistas una obra R.U.R. (Rossum's Universal Robots) de un escritor checo Karel Capek es visto como el mensajero de la nueva era robótica (Petrovic et al., 2019).



**Ilustración 5 Escena III del epílogo R.U.R.**

Fuente: (Petrovic et al., 2019).

Es increíble como la humanidad ha logrado tantos avances con la robótica hoy en día que permite que se pueda encontrar robots donde sea, en agricultura, hogares, industria 4.0, supermercados y mucho más.

Un roboticista podría estar de acuerdo con la obra de Karel Capek's, la cual representa un pilar en la robótica, aunque los personajes principales en la obra, humanoides robóticos, no existen excepto en la imaginación humana. Cuando Capek describe el robot en su obra, él tiene en mente un androide, una máquina que se ve, trabaja y piensa como un humano, con el potencial de sobrepasar a un humano física e intelectualmente. El teatro con robótica representa un polígono excelente para probar hoy en día tecnologías de interacción humano-robot. Estos métodos que resuelven problemas de manera efectiva puede ser también aplicado en casas, lugares públicos y hasta en las instalaciones de producción de la industria 4.0 donde robots con habilidades de comunicación avanzada y habilidades cognitivas deben compartir el espacio de trabajo con humano cohabitantes (Petrovic et al., 2019).

En otro proyecto de robótica se habla de cómo muchas personas sufren de deformidades de mano, como la osteoartritis, fracturas, ligamentos rotos, dislocaciones, artritis y más, y como avances recientes en accesorios y tecnología de la internet de las cosas IOT pueden aliviar muchos de estos problemas brindando una solución más exacta, confiable y automatizada para el proceso de cuantificación y rehabilitación (Rashid & Hasan, 2019).

En otro estudio se demuestra, por primera vez, la viabilidad de la monitorización de la actividad SC utilizando NIRS en modo de transmisión en un modelo animal grande, donde no es necesaria la alineación perfecta del emisor y receptor de luz, gracias a la propiedad de difusión de los medios biológicos. Esto contribuye a un avance hacia el uso de Internet de Things (IoT) para aplicaciones médicas, mediante la monitorización del SC durante procesos altamente invasivos, como la estabilización de la columna, en forma de implantes y otras cirugías, como la de la aorta, con el uso de catéteres mínimamente invasivos específicos (Tsiakaka & Feruglio, 2019).

### **3.4 ADQUISICIÓN DE DATOS**

Los sistemas de adquisición de datos son utilizados ampliamente en aplicaciones industriales y de laboratorio, así como control, monitoreo, medidas y pruebas, automatización, y más. El objetivo principal de los sistemas de adquisición de datos es medir el fenómeno físico, convirtiendo las señales análogas en señales digitales y luego envían todos los datos recolectados para un mejor análisis (Velasquez-Aguilar et al., 2017).

Estos sistemas de adquisición de datos se encuentran en tantos lugares, pero donde más se ven es en industrias, estos normalmente los vemos con PLC en SCADA donde se logran ver varios valores de motores y sensores gracias a estos sistemas. De igual manera se pueden utilizar en fábricas para monitoreo de temperaturas, proyectos para poder monitorear y almacenar datos específicos y más cosas. En los sistemas de adquisición de datos es necesario tener una base de datos para ahí almacenar información recibida para monitorear en el momento o monitorear después.

También hay tarjetas comerciales de adquisición de datos las cuales son diferenciadas por sus viabilidades así como frecuencia de muestreo, escala de señal adquirida, requerimientos y potencia, pero generalmente tienen un precio bien elevado y necesitan una computadora en el sitio de recolección de datos(Velasquez-Aguilar et al., 2017).

En un proyecto se usó una DAQ la cual es una tarjeta de adquisición de datos, ese estudio tenía como objetivo presentar un método simple basado en un algoritmo con cruce por cero para detectar la frecuencia de la señal de poder. En el estudio primero se usó el algoritmo Fourier como un filtro digital para remover las señales de la las partes de coseno y seno de la frecuencia fundamental. Luego la tecnología de algoritmo de cruce por cero es aplicada a las señales de seno y coseno para calcular la frecuencia de señal de la onda de seno. Ahí se planea utilizar la tarjeta de adquisición de datos, utilizando la potencia adquirida por la DAQ se envían a un servidor en la nube por un dispositivo Android para analizar la operación de las señales capturadas y finalmente los dispositivos móviles son usados para monitorear los datos medidos (Y. Chen et al., 2016).



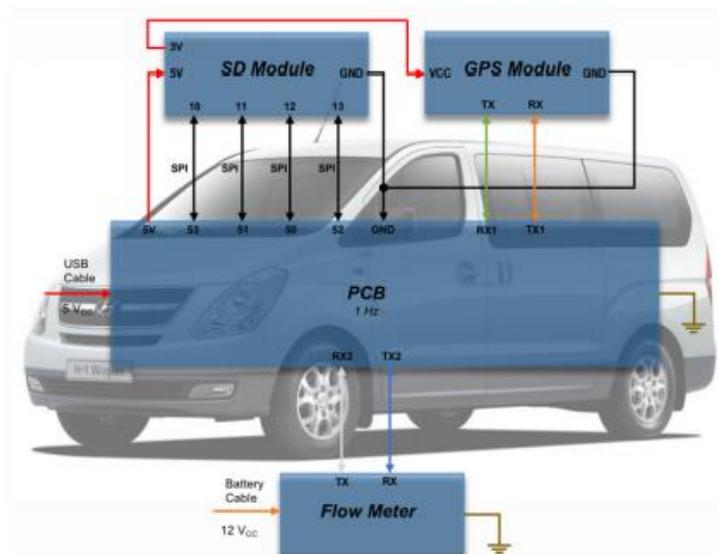
**Ilustración 6 Diagrama del sistema del dispositivo móvil para el monitor de señal de potencia.**

Fuente: (Y. Chen et al., 2016).

Un sistema para adquisición de datos fue construido el cual integra información de coordenadas GPS, velocidad del vehículo y altitud, todo esto obtenido por el módulo GPS y también los datos instantáneos de consumo del medidor de flujo de Diesel. Estas dos fuentes de datos fueron unidas por medio de un puerto de datos serial con el uso de una placa de microcontrolador Arduino, es cual en el tiempo maneja la adquisición de datos y graba los datos en archivos de texto en un formato .CSV en una tarjeta de memoria usando un módulo SD. La frecuencia de adquisición de datos es alcanzada alrededor de 1Hz (Baquero-Larriva & Álvarez-Coello, 2018). En la ilustración 7, esta de forma detallada la arquitectura del sistema de adquisición de datos.

Los sistemas para adquisición normalmente no son económicos entonces siempre hay personas creando sus sistemas por medio de proyectos o experimentos, en otro estudio esta un sistema para adquisición de datos de bajo costo para registrar el consumo de combustible en vehículos con motor Diesel.

Ese estudio es bien interesante como se decidió medir los datos del combustible ya que como se sabe en el mundo de la mecatrónica hay tantas formas de hacer proyectos, literal un proyecto se puede hacer de múltiples formas, todo depende del investigados y su manera de trabajar.



**Ilustración 7 Arquitectura del sistema de adquisición de datos, donde se encuentra la placa SD registrado a través de una línea de datos única en un archivo .CSV: hora (hh: mm: ss), posición (latitud, longitud), altitud sobre el nivel del mar (m), velocidad instantánea (Km / h), caudal instantáneo (l / h) y caudal acumulado (l).**

Fuente: (Baquero-Larriva & Álvarez-Coello, 2018).

En ese estudio del sistema para adquisición de datos para registrar consumo de combustible se creó un nuevo sistema. Este sistema integraba la información resultante de medidas instantáneas y directas del consumo instantáneo del combustible Diesel con información adquirida por el GPS el cual era usado con un microcontrolador Arduino, los cuales son de fuente abierta, de bajo costo y están disponible a nivel mundial. Adicionalmente este sistema es fácil de replicar y ser instalado en gran variedad de vehículos usado para transporte. Una ventaja potencial de ese sistema de adquisición de datos, es el hecho de que las medidas son tomadas directas del flujo de combustible, no como otros dispositivos que calculan el consumo de combustible indirectamente por medio de otros parámetros del motor (Baquero-Larriva & Álvarez-Coello, 2018).

Los sistemas de adquisición de datos han sido utilizados en un gran rango de aplicaciones, así como en la industria de biomédica y mecatrónica y así también ha podido ser usada como una herramienta de enseñanza. En el ambiente académico, los sistemas de adquisición de datos

pueden ser una de las herramientas de enseñanza que puede ayudar a demostrar lo práctico detrás de teorías. La demanda por un sistema de adquisición de datos de bajo costo es alta para el ámbito educativo, sin embargo, los sistemas de adquisición de datos en el mercado son caros. En adición a ser muy caros estos sistemas, muchos de los sistemas de adquisición de datos que están disponibles en el mercado no son de fuente abierta y por lo tanto requieren software propietario para poder ser utilizados. Estos tipos de software usualmente son caros y tiene problemas propietarios (Haizad et al., 2016).

Con un enfoque en resolver problemas de alto costo, gran consumo de potencia, cobertura incompleta, sitios de operación aburridos y abstractos de un sistema para adquisición de datos de datos micro climáticos agrícola en el mercado, se implementó un sistema para adquisición de datos portátil de bajo costo, de poco consumo de potencia, de operación simple y humanista el cual se puede monitorear desde teléfonos Android, tecnología del internet de las cosas (IOT), tecnología bluetooth, tecnología de sensores, y tecnología de adquisición de información. El sistema creado en este proyecto consiste de tres módulos funcionales principales en los cuales está una terminal de adquisición de datos en primer plano, un programa de aplicación celular para dispositivos Android y un centro de control de datos en el fondo. La terminal de adquisición de datos manda datos micro climáticos procesados y perceptivos a los teléfonos móviles vía bluetooth, luego la APP terminal del teléfono analiza, procesa, almacena y muestra los datos micro climáticos y los envía a un centro de control de datos por medio de comunicación GPRS (Gao & Yao, 2015).

Este estudio de sistemas de adquisición de datos del diseño e implementación de un sistema para adquisición de datos basado en la plataforma de Android para monitoreo de datos micro climáticos agrícolas. El cual fue creado debido a que los sistemas de adquisición de datos en el mercado son de precio alto y consumen bastante potencia.

Esta terminal para adquisición de datos es responsable de recolectar información micro climática agrícola, como la temperatura atmosférica, humedad atmosférica, presión atmosférica, la temperatura del suelo, la humedad del suelo, el valor de pH del suelo y otros datos como la velocidad del viento, la dirección, PM2.5, la intensidad de la radiación solar, la intensidad fotosintética efectiva de la radiación, la concentración del dióxido de carbono y más. Luego la

terminal de adquisición de datos analiza, procesa, almacena los datos brutos resultantes y transmite los datos a la terminal móvil vía bluetooth donde permite verlo desde el dispositivo Android. La terminal móvil es la responsable de recibir datos de la terminal de adquisición, haciendo que los parámetros actuales se muestren de manera visual y visualización de la curva de parámetros históricos dinámicamente; Setear o chequear la función del sistema y el estado de la terminal de adquisición; enviando datos al centro de control de datos en tiempo real. Luego el centro de control de datos recolecta, analiza, almacena, y muestra los datos recolectados (Gao & Yao, 2015).

El uso de un sistema de adquisición de datos (DAQ) es muy importante en el proceso de la industria ya que permite una señal análoga desde un sistema físico para ser convertida en una señal digital para el procesamiento por computadora y viceversa. Sin embargo, muchas de las DAQs disponibles en el mercado hoy en día son complejas y caras (Haizad et al., 2016).

En un estudio en el año 2016 se presenta un desarrollo de un sistema de adquisición de datos simple, de bajo costo y de fuente abierta basado en un microcontrolador para el proceso de automatización y control, este se hizo con un Arduino Mega 2560.

La necesidad de sensores en el proceso de automatización en la industria cada vez incrementa más y más y a un paso rápido. Esto es debido al desarrollo de plantas requiriendo altos grados de automatización, en el cual los sensores son necesarios para alimentar el sistema de control de las plantas con información crucial. La solución tradicional para operar una planta de procesos que depende de tecnologías análogas se están volviendo obsoletas ya que los usuarios ahora prefieren técnicas de medición innovativas y tecnologías digitales. La integración de esas tecnologías digitales en proceso de planta requiere el diseño de una planta más sofisticada. Estas plantas tienen una gran cantidad de instrumentos digitales instalados para recolectar información para la automatización que conlleva a una abundancia de datos digitales inundando el sistema. Y una manera de manejar estos datos es por medio de la adquisición de datos (DAQ). La adquisición de datos (DAQ) es el núcleo de cualquier sistema digital para interfaz de mundo real. Estos sistemas de adquisición de datos permiten a las señales de proceso para la medición de dispositivos ser recolectadas, procesadas y analizadas con el propósito de control, almacenaje, presentación y más (Haizad et al., 2016).

Luego en este mismo estudio se habla sobre las primeras tres categorías de sistemas de adquisición de datos comerciales en la industria, lo cual es muy importante conocerlo bien para entender más del tema.

La primera categoría de sistemas de adquisición de datos es el sistema a base de microcontrolador integrado. Este sistema tiene la ventaja de la portabilidad ya que está basada en instrumentos autónomos. La arquitectura fija del diseño incrustado también incrementa la robustez de la DAQ y su tolerancia a fallas como los cortos circuitos. Este tipo de DAQ es el más adecuado a ser utilizado en ambientes peligrosos. La segunda categoría es el sistema de adquisición modular que es conectado a un bus local especializado como extensiones PCI para "Instrumentation (PXI)". Este sistema de adquisición de datos puede ser más potente que el mencionado anteriormente. También es más adecuado para ser utilizado en un ambiente que tiene necesidad de un crecimiento futuro. La tercera y última categoría es basada en computadora, este tipo de sistema de adquisición de datos es conectado a una computadora de monitoreo donde el usa el poder de procesamiento de la computadora. El sistema de adquisición de datos actuara como un acondicionador de señal antes de que los datos sean procesados por la computadora conectada para realizar la manipulación de datos deseada, almacenaje y visualización. El sistema de adquisición de datos que está basado en computadora existe en dos formas, módulos externos e internos. El módulo interno es una tarjeta de extensión que es conectada a un puerto PCI de la computadora. Mientras que el módulo externo se conecta por medio de un bus serial universal (USB) (Haizad et al., 2016).

### 3.4.1 COMUNICACIÓN

Los sistemas para adquisición de datos necesitan alguna forma de comunicación para sus datos ya que los datos convertidos de señales análogas a digitales deben almacenarse en algún lado, una base de datos y para ello se necesita comunicación. Estos datos pueden ser transmitidos por tecnologías de conectividad inalámbricas como Bluetooth, Wifi y otros.

En un estudio de sistema de adquisición de datos hablan de la parte de la comunicación, donde hablan de la amplia adaptación de tecnologías inalámbricas como la tecnología Bluetooth y XBee. En los últimos se han usado para aplicaciones industriales, científicas y domésticas, gracias a las

tecnologías de comunicación inalámbricas han emergido aplicaciones móviles. Están los módulos OEM de serie Digi XBee que implementan la radio IEEE 802.15.4 y el protocolo de redes ZigBee los cuales se han vuelto muy populares en desarrollo de sistemas de aplicación. En el estudio se usa el XBee Pro S1, este módulo tiene un alto nivel de protocolos de comunicación usados para crear "Wireless Personal Area Networking" (WPAN). Las distancias de transmisión de este dispositivo van desde 10 hasta 100 metros y esta distancia depende de la salida de potencia y ciertas características ambientales (Velasquez-Aguilar et al., 2017).

La comunicación por medio de Bluetooth debe ser utilizada en estudios, proyectos o trabajo que no requieren un gran rango de espacio ya que este método no tiene tanto alcance. Esta forma de comunicación es bastante utilizada para controlar dispositivos con aplicaciones ya sea por medio de teléfonos celulares o computadoras. Como estudiante de universidad de la carrera de ingeniería mecatrónica he sentido preferencia a trabajar con módulos bluetooth que otro tipo de módulos de comunicación.

El módulo Bluetooth utilizado es un convertidor de interfaz serial a adaptador de Bluetooth. Este módulo puede ser definido por comandos AT, y tiene dos modos, el modo maestro y el modo esclavo, pero solo puede ser un modo a la vez y una vez está en proceso de comunicación no se puede cambiar en el proceso (Velasquez-Aguilar et al., 2017).

En el estudio se utilizó un módulo ZigBee el cual es una especificación basada en IEEE 802.15.4 para conjunto de protocolos de comunicación de alto nivel. El ZigBee define tres diferentes tipos de dispositivos los cuales son el coordinador, el enrutador y el dispositivo final. Las redes ZigBee siempre van a tener un único dispositivo coordinador, el cual empieza la red, seleccionando el canal y la ID de red de área personal (PAN). El enrutador es un nodo con características completas ZigBee. Este dispositivo puede unirse a redes existentes y mandar, recibir y enrutar información. Y por último está el dispositivo final el cual es una versión reducida del enrutador. Este dispositivo puede unirse a redes existentes y enviar y recibir información (Choden et al., 2019).

En este estudio se observa como se realizó un robot de rescate controlado por medio de un control remoto utilizando comunicación ZigBee que como mencionado anteriormente está basado en protocolos de comunicación de alto nivel.

XBee es un nombre de marca de una familia de factor de forma compatible con módulos de radio de Digi International. XBee usa el estándar ZigBee y lo añade a él y lo envuelve en su propio pequeño y nítido paquete. Los módulos de radio XBee pueden funcionar en tres diferentes maneras, el modo transparente, el modo comando y el modo de interfaz de programación de aplicación (API). El modo transparente es el estado default para los módulos XBee. Este modo es llamado transparente porque el módulo simplemente pasa la información al mismo tiempo que la está recibiendo. En el modo comando, el módulo XBee envía los comandos de configuración a los ajustes de configuración. Y el modo de interfaz de programación de aplicación (API) requiere que la comunicación con el módulo sea por medio de una interfaz estructurada (Choden et al., 2019).

### 3.4.2 BASE DE DATOS

Una base de datos es una colección de datos estructurada organizada los cuales normalmente están almacenados en computadoras. Una base de datos usualmente es controlada por un sistema de manejo de base de datos (Oracle, n.d.).

La pérdida permanente y temporal de almacenamiento de datos es requerida en cualquier información funcional de sistemas de procesamiento, la cual hasta ahora ha sido cumplida por memorias basadas en CMOS como SDRAM, DRAM y memorias flash. La brecha de velocidad y memoria se ha vuelto un cuello de botella en el desempeño de un sistema crítico, y ahí entran los NVM para sistemas. NVM es memoria no volátil la cual permite alta velocidad y desempeño en manipulación de datos (A. Chen, 2016).

Hay una gran cantidad de tipos de base de datos que tienen diferentes maneras de manipular datos lo cual afecta en el desempeño de transacciones a la hora de tratar con información almacenada. También es muy importante para compañías manejar la información de manera rápida, así no correr riesgo de perder alguna operación por cosas como mal desempeño de una base de datos, y en la misma manera, necesitan operar de manera rápida manteniendo a la vez la integridad de la información (Monjaras et al., 2019).

En el pasado, muchas aplicaciones utilizaban ampliamente bases de datos relacionales para recolectar información. La base de datos relacional se asemejaba a la estructura básica en forma

de una tabla la cual contenía una fila y una columna. Esta se podía ordenar por medio de un lenguaje de consulta SQL para insertar, borrar, actualizar y consultar datos. También es una tecnología que puede escalar, realizar y entregar necesidades de disponibilidad continua para soporte en línea aplicaciones. Adicionalmente las bases de datos relacionales han tenido problemas manteniéndose al día con la oleada de la modernización (Puangsaijai & Puntheeranurak, 2017).

La opción de base de datos más viable para aplicaciones en páginas web y teléfonos celulares deben soportar la cantidad de datos que incrementan de gran manera a la vez que el tipo de información con más variedad como datos sin estructura, datos estructurados y datos semi estructurados. Sin embargo, es necesario tener un sistema de base de datos la cual pueda procesar grandes cantidades de información de manera rápida y con situaciones flexibles. Las bases de datos relacionales son los tipos de base de datos más populares desde su aparición en los 1970s y son críticas en todas las compañías por el hecho de que manejan su información día a día. Pero gracias a su crecimiento exponencial de usuarios en el internet, la cantidad de datos crece exponencialmente. Y ahí es donde las bases de datos relacionales no logran manejarlo de la mejor manera, haciendo que su desempeño disminuya (Monjaras et al., 2019).

Estas bases de datos son usadas en muchas organizaciones de varias naturalezas desde las últimas tres décadas, así como en educación, salud, negocios y muchas más aplicaciones. Las bases de datos SQL son diseñadas para manejar estructuras de datos y mostrar desempeños muy grandes. ACID el cual consiste en atomicidad, consistencia, aislamiento y durabilidad, es una propiedad de las bases de datos relacionales la cual es utilizada para manejar la integridad y consistencia de los datos. Técnicas físicas de base de datos son usadas para incrementar el desempeño de las bases de datos relacionales. Espacios de tabla los cuales también son llamados sub folders son una de las técnicas físicas de bases de datos usada por la base de datos SQL de Oracle. Estos espacios de tabla (Tablespaces) son utilizados para almacenar datos de manera local en diferentes archivos de datos. Hoy en día gran cantidad y varia naturaleza de datos son generados por varias organizaciones, así como videos, imágenes, blogs y otros. Y esa cantidad de datos no puede ser manejada de manera eficiente por una base de datos SQL. La bases de datos NoSQL son usadas

en el proceso para procesar y analizar esa gran cantidad de datos de manera eficiente (Khan et al., 2019).

Las bases de datos relacionales son destacadas ya que organizan los datos en tablas que son mantenidas en relaciones entre ellas. Otro aspecto importante de este tipo de base de datos es que pueden manejar propiedades transaccionales completamente. Pero como hay problemas en el desempeño de las bases de datos relacionales en la parte de manejar grandes cantidades de datos, las bases de datos NoSQL fueron la nueva solución para este problema (Monjaras et al., 2019).

Las bases de datos SQL son bases de datos de escalabilidad vertical y pueden procesar cierta cantidad de datos. Tecnología nueva, avances y técnicas de almacenamiento son introducidas para administrar conjunto de datos de repuesto, grandes y conectados. Así que las bases de datos NoSQL son usadas para procesar gran cantidad de datos de manera eficiente, estas bases de datos son de escalabilidad horizontal. BASE (Basically Available State and Eventually Consistent) lo cual significa estado básicamente disponible y eventualmente consistente, es la propiedad principal de las bases de datos NoSQL. Esta es utilizada para procesar grandes cantidades de sets de datos conectados de manera eficiente (Khan et al., 2019).

Las bases de datos NoSQL son más rápidas a la hora de tratar grandes cantidades de datos gracias ya que al tener menos procesos involucrados a la hora de manipular datos las hace más rápidas, a diferencia de las bases de datos SQL. Pero cabe mencionar que las bases de datos NoSQL no pueden lidiar con la gestión de propiedades ACID (Atomacity, Consistency, Isolation, Durability) completamente para las transacciones de bases de datos, propiedades que las bases de datos SQL pueden manejar de manera apropiada, haciendo esta una de las características y razones de la popularidad de las bases de datos relacionales. Dentro de las bases de datos NoSQL, hay cuatro tipos de bases de datos, la base de datos NoSQL document based, NoSQL Key-Value, NoSQL Column Store y NoSQL Graph Based. Pero gracias a la falta de desempeño que tienen las bases de datos relacionales para manejar grandes cantidades de datos mientras mantienen transacciones de propiedades ACID, hay una necesidad parar resolver ese problema. Ahí es donde las bases de datos en memoria entran. La característica principal de este tipo de base de datos es

que pueden manejar datos dentro de la memoria principal (RAM) lo cual las hace más rápidas que las tradicionales. También pueden ser bases de datos tipo SQL o NoSQL (Monjaras et al., 2019).

Se menciona bastante de las bases de datos y como están las relacionales y no relacionales, igualmente las SQL y NoSQL donde vemos que las NoSQL son las más rápidas con bastantes datos de igual manera esto las hace ser bases de datos más eficientes, pero su desventaja es que no puede lidiar con las propiedades ACID.

Por ejemplo, puede haber una base de datos SQ que pueda almacenar datos en la memoria principal y que mantenga sus propiedades transaccionales ACID gracias a sus propiedades relacionales, por otro lado, hay bases de datos NoSQL que pueden almacenar datos en su memoria principal igualmente. Debido a que hay una gran cantidad de tipos de bases de datos existente, es difícil saber y entender todas ellas. Pero es muy importante para los usuarios saber sobre estas categorías de bases de datos para saber cuándo usar una categoría específica para un escenario específico, así poder obtener el mejor desempeño para el manejo de datos (Monjaras et al., 2019).

### 3.4.3 APLICACIONES ANDROID

Las aplicaciones Android generalmente son desarrolladas por un desarrollador individual o un pequeño equipo de desarrolladores, y los desarrolladores pueden que no tengan experiencia probando app de Android o que no tengan experiencia probando ningún sistema de software. Aun así una persona que no tenga el conocimiento básico de Android puede crear apps de Android por medio de varias herramientas generadoras de apps disponibles en el mercado (Jha et al., 2019).

Hoy en día los celulares sirven para tantas cosas, desde calculadora, reloj, calendario, cronómetros, temporizadores, despertador, linterna, llamadas, reproductor de música y video, dispositivo para redes sociales y más cosas, pero a la vez de esas cosas básicas, los celulares permiten utilizar aplicaciones creadas para facilitar cosas, están las aplicaciones de redes sociales que permiten a las personas entretenerse, comunicarse con otras personas, ver noticias y más e igual aplicaciones de banco en los cuales uno puede revisar el estado de la cuenta, movimientos, transacciones,

cantidad de dinero en la cuenta y más, así como están esas aplicaciones, se pueden crear aplicaciones para estudios, proyectos y trabajos con los softwares y conocimientos debidos.

El avance en herramientas y técnicas para probar aplicaciones de Android aún debe ponerse al día, en parte debido a la rápida evolución de la plataforma de Android. A pesar de los avances recientes en herramientas y tecnologías para probar apps de Android, una gran numero de desarrolladores de apps prefieren probar sus productos de manera manual debido a varios desafíos utilizando las técnicas y herramientas disponible en el mercado. Uno de los mayores desafíos enfrentados por los desarrolladores de apps a la hora de probar sus aplicaciones son las limitaciones de tiempo, problemas de compatibilidad, falta de exposición, complejidad de las herramientas y la falta de experiencia (Jha et al., 2019).

Utilizar smartphones basados en Android es muy conveniente y ha hecho a Android muy popular alrededor del mundo. Hasta ahora Java ha dominado los lenguajes de C# y Corona para el desarrollo de apps de Android, mientras que Kotlin parece ser el futuro para el desarrollo de apps para Android. Kotlin provee extras adicionales para el desarrollo de apps para Android, pero aun así hay algunos agujeros que proveen una forma para que las amenazas de seguridad lo penetren. Una de las amenazas más comunes es "App Repackaging", el cual es un ataque peligroso y malicioso que explota la plataforma de fuente abierta (Shah et al., 2018).

Las aplicaciones de Android son implementadas en Java, y así heredan el modelo de excepción de Java. Java tiene tres tipos de excepciones, "RuntimeException" que son las excepciones que son tiradas durante la operación normal de la máquina virtual de Java cuando el programa viola las limitaciones semánticas. Esta "Error", el cual representa serios problemas que una aplicación responsable no debería de intentar atrapar. Y la tercera excepción es "Checked Exception", estas excepciones son requeridas para ser declaradas en un método o una cláusula e indicar las condiciones que un programa de cliente razonable querría atrapar (L. Fan et al., 2018).

Los sistemas operativos móviles han evolucionado un montón en la última década. Una gran variedad de sistemas operativos está disponible en el mercado. Con más de 2 billones de dispositivos activos al mes, como dijo Google I/O 2017, Android es uno de los sistemas operativos preferidos. Android es un sistema operativo móvil para smartphones y tabletas, construido en una

fundación de Linux. Android provee un marco de aplicación muy enriquecido para desarrollar aplicaciones. A los desarrolladores de APP de les provee un rango de lenguajes orientados a objetos para motivos de desarrollo, incluyendo Java, C#, Python, Corona, y muchos más. Java es el primer lenguaje usado por la mayoría de los desarrolladores de apps para Android (Shah et al., 2018).

En un proyecto se querían detectar problemas eléctricos en una motocicleta utilizando una app de Android, mostrar las fallas en la APP optimizarían el tiempo de respuesta y trabajo del técnico ya que la app seria optima. El sistema usa un Arduino para mandar información por medio de bluetooth al dispositivo Android. El escenario es basado en un sistema de mantenimiento automático en tiempo real el cual brinda información detallada de los problemas registrados. Luego del diseño y desarrollo del sistema hicieron pruebas experimentales con cada componente y falla eléctrica construido en una motocicleta genérica y en diferentes marcas y modelos. El App tenía una interfaz detallada y cumplía con los requerimientos. El sistema desarrollado tenía una capacidad de detectar fallas en el sistema de ignición de la motocicleta, fallas en el sistema CDI, fallas en el alternador, fallas en el rectificador, fallas en la batería y todo se comprobó por medio de pruebas con motocicletas de diferentes marcas, así como Suzuki, Daytona, Tundra y Traxx (G. Alvarado et al., 2017).



**Ilustración 8 Aquí se muestra la ventana de reporte final del APP.**

Fuente: (G. Alvarado et al., 2017).

En un estudio se utilizó una placa electrónica IOIO la cual añade características de hardware I/O a computadoras o aplicación Android. Esa placa puede ser controlada desde distancia usando bluetooth, así que es posible programarla y controlarla por medio de una app en un smartphone Android. Esta placa IOIO tiene una librería de Java y es compatible con el sistema operativo Android, así que es posible crear apps de Android que controlen diferentes pines de esta placa. Lo cual conlleva al siguiente punto en el cual programar Android es complejo, así que sería bueno hacer esta tarea más simple. Ahí entra la plataforma de App Inventor, la cual es una plataforma online la cual permite al usuario crear aplicaciones de Android solo conectando bloques. Este concepto está basado en una interfaz de bloques. Ahí no hay código de java por escribir, no es necesario considerar limitaciones de Android, y la interfaz de diseño y escritura es simple e intuitiva. Con esa plataforma incluso niños pueden crear proyectos interesantes. las características existentes hacen los componentes complejos de interfaz más fáciles, tal como leer sensores, cámara, micrófonos y mas (Chtourou et al., 2016).

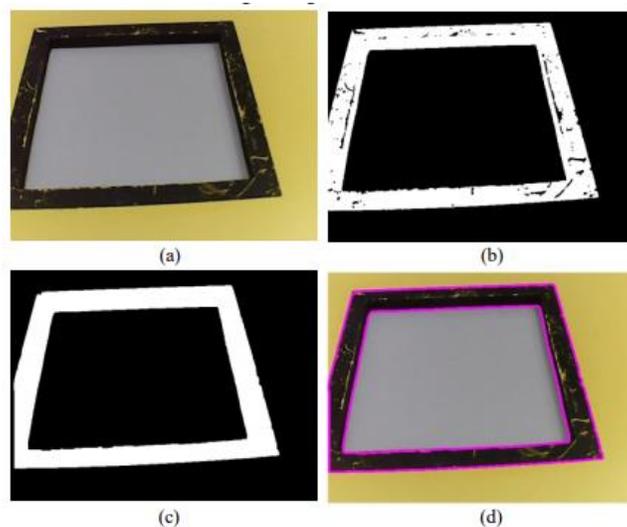
Kotlin es un lenguaje basado en JVM desarrollado por JetBrains. Kotlin fue creado con desarrolladores de Java en mente, y con IntelliJ como su principal IDE de desarrollo. Su meta principal es proveer una alternativa para todas las plataformas que actualmente usan Java el cual es más nulo seguro, más productivo y más conciso(Shah et al., 2018).

### **3.5 VISIÓN**

Al momento de utilizar cámaras en un proyecto es necesario tener una buena visión, en la visión se incluyen lo que son cámaras, vistas, posicionamiento de cámaras y más. Es necesario también tener una buena calidad de imagen para tener los mejores resultados y una mejor experiencia.

En un estudio se determinó que debido a factores como la iluminación, el fondo, sombras, la imagen original usualmente es una imagen con ruido y conexiones discontinuas. En orden para obtener imágenes sin ruido con características de objetivo claro, los siguientes procesos son realizados en cada fotograma de imágenes adquiridas. Se extienden de seis a 10 pixeles al borde de la imagen para prevenir que falte el contorno del eje, luego debido al poder limitado de computación de una Raspberry Pi, el tamaño de la imagen es reducido para incrementar la velocidad de calculación. Luego se usa un desenfoque gaussiano usando un núcleo gaussiano 3x3

para hacer la textura de la imagen más suave. Luego se convierten las fotos del espacio RGB al espacio HSV, el cual hace la distinción del color más fácil. La H, S y V de la imagen y el diccionario de preconfiguración de color desarrolla operaciones bit, los pixeles con valores de H, S y V están retenidas dentro de un límite específico, luego extrayendo un objeto de color específico. Luego se usa un núcleo 3x3 para realizar una operación abierta y cerrada, la cual puede remover ruido esporádico y llenar el punto de rompimiento en la conexión cerrada. Y finalmente se extrae el contorno del objeto usando una función de OpenCV. Después de todo esto se obtiene un contorno claro de información del objeto deseado (Z. Zhang et al., 2019).



**Ilustración 9 (a) La imagen original. (b) Objeto con color específico. (c) Después de la operación de apertura y cierre. (d) Contornos del objeto.**

Fuente: (Z. Zhang et al., 2019).

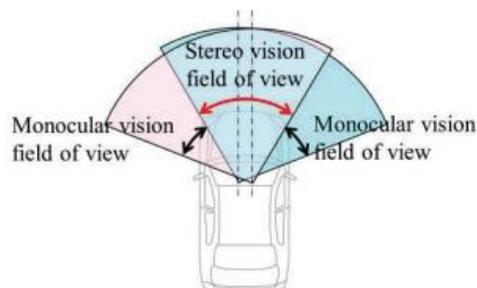
En otro estudio sobre la evolución de la calidad visual del contenido proyectado hablan de las apariencias globales de la imagen donde dice que investigaciones existentes de estética en imágenes muestran que las estadísticas globales de imagen pueden servir como buenas indicaciones de la calidad de imagen. Entonces ellos computaron esas estadísticas de la imagen como características incluyendo el matiz promedio, saturación, intensidad y valor de brillo en la imagen (Le et al., 2017).

### 3.5.1 CÁMARA

Hoy en día las cámaras son tan necesarias en tanto lados, se utilizan en teléfonos celulares para tomar fotos, videos, se utilizan para filmar películas y comerciales y más cosas. El enfoque de cámaras a realizar es en proyectos y trabajos, las cámaras para proyectos normalmente no son muy buenas, pero esto es debido a que las buenas cámaras son muy caras y entonces los proyectos o salen bien caros o requieren a cámaras de baja calidad para poder recortar costos. Pero lo que se puede hacer para solucionar este problema es utilizar la cámara del teléfono móvil ya que hoy en día estas son excepcionales. Hay cámaras de teléfonos celulares que sobrepasan la calidad de cámaras profesionales de tienen años en el mercado, esto si cabe decir que los teléfonos como mejores cámaras tienen valor elevado, pero es más conveniente pagar por un teléfono caro que funciona para múltiples cosas que una cámara muy cara solo para imágenes.

En un estudio se utilizó un sistema de cámara estéreo, el cual consistía de un set de dos cámaras de las cuales se obtenían imágenes que podían ser usadas para medir distancia hacia un objeto deseado calculando la disparidad del objeto entre las dos imágenes. Debido a que la información de distancia puede ser obtenida, se han utilizado una gran cantidad de estas cámaras en vehículos, vigilancia y aplicaciones robóticas. Muchos fabricantes de autos venden sus vehículos con este sistema de cámaras el cual ayuda a tener un buen soporte de sistema de frenos y un sistema de control de cruce adaptivo (Kobayashi et al., 2020).

Estas cámaras tienen un alto grado de campo de visión por lo cual ayuda a detectar ciclistas y personas que crucen en las calles.



**Ilustración 10 En ese estudio realizado se muestra el ángulo de visión de campo de una cámara estéreo usando el método de multi cambio.**

Fuente: (Kobayashi et al., 2020).

### 3.5.2 POSICIONAMIENTO DE CÁMARA

El posicionamiento óptimo de cámara es una de las muchas aplicaciones prácticas de un núcleo de problema completos NP en el campo combinatorial de optimización. En una forma genérica, el problema OCP se relaciona al posicionamiento y puesta de cámaras individuales para que la red total sea capaz de cubrir un área dada mientras teniendo un set de limitaciones de aplicación específicas y optimización de un objetivo. El problema de posicionamiento óptimo de cámaras está en el núcleo de muchos diseños, dado el costo o alcance del objetivo y un set de limitaciones operacionales, el diseño de un algoritmo efectivo que determine las ubicaciones de cámara y orientaciones ha sido de un tipo de interés para investigadores y otros (Kritter et al., 2019).

Diseño de app para adquisición de datos.

## IV METODOLOGÍA

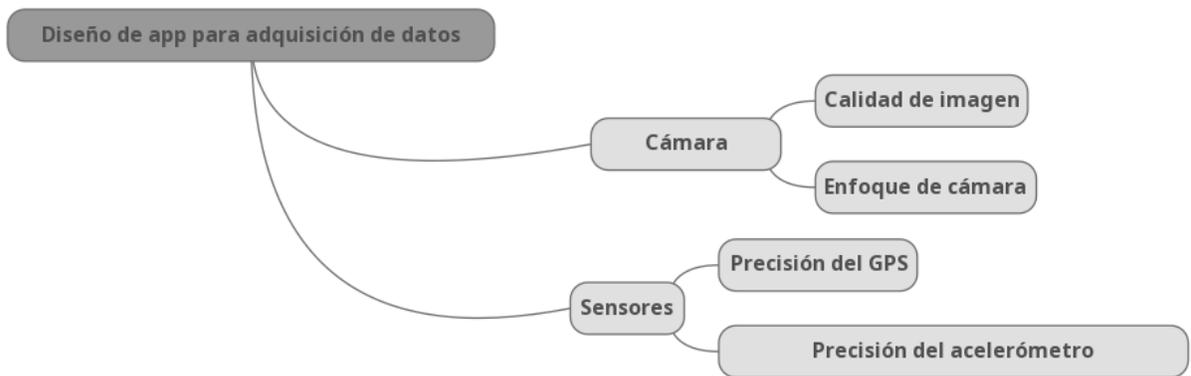
En este capítulo se hablará sobre el enfoque del proyecto, las variables de investigación que contiene, las dependientes e independientes, las técnicas e instrumentos aplicados, materiales a utilizar, la metodología de estudio, los resultados y por último el cronograma de actividades.

### 4.1 ENFOQUE

En enfoque de la investigación es de carácter cuantitativo ya que en este proyecto se recolectan datos y se almacenan. En esta investigación se obtienen datos por medio de un teléfono celular que se comunica con un robot, y esto con ayuda de un APP.

### 4.2 VARIABLES

En toda investigación es importante tener claro las variables del mismo, las dependientes y las independientes. Las variables independientes son variables en las cuales sus valores se pueden modificar en el proyecto y las dependientes son las que su valor depende de cómo de modifican las variables independientes.



**Ilustración 11 Variable dependiente y variables independientes.**

Fuente: (Propia,2021).

En la ilustración 11 se observa las variables dependientes e independientes de la investigación, donde la variable independiente es el diseño de software de adquisición de datos ya que es objetivo principal de la investigación, luego están las variables dependientes donde está la

cámara y los métodos de comunicación, en la cámara puede variar, dependiendo del teléfono celular a ser utilizado, su resolución y los píxeles de la cámara.

### **4.3 TÉCNICAS E INSTRUMENTACIÓN**

Para esta investigación de un APP para adquisición de datos por medio de un smartphone se ha utilizado varias fuentes de programación para comprender y realizar de manera adecuada la misma. Para crear la aplicación de adquisición de datos se utilizará una página web especial para crear APPS para Android la cual se llama MIT APP INVENTOR, esta página web brinda la capacidad de crear APPS móviles a muchas personas a nivel mundial ya que no es necesario saber programar en java o Kotlin para poder usarla, cabe mencionar que si es necesario saber de lógica de programación para tener buen entendimiento de las funciones del software. Los datos a almacenar serán almacenados en una base de datos compatible con MIT APP INVENTOR.

### **4.4 MATERIALES**

Los materiales esenciales para la realización de la investigación son los siguientes:

- Modulo bluetooth: este dispositivo se usará conectado a un Arduino para permitir la comunicación entre el APP y el Arduino.
- Arduino: es el microcontrolador a usar en la investigación con la cual se enviará datos a la aplicación para que esta trabaje.
- Smartphone Android: dispositivo donde se utilizará la APP a realizar con la investigación.

### **4.5 METODOLOGÍA DE ESTUDIO**

La metodología a utilizar con esta investigación es la metodología incremental, esta metodología es un acercamiento intuitivo al modelo de cascada. Hay múltiples iteraciones de ciclos pequeños que incluyen requerimientos, diseño, desarrollo, pruebas, cada una produciendo un prototipo del software. Las iteraciones subsecuentes mejoran el prototipo anterior (Ganney et al., 2020).

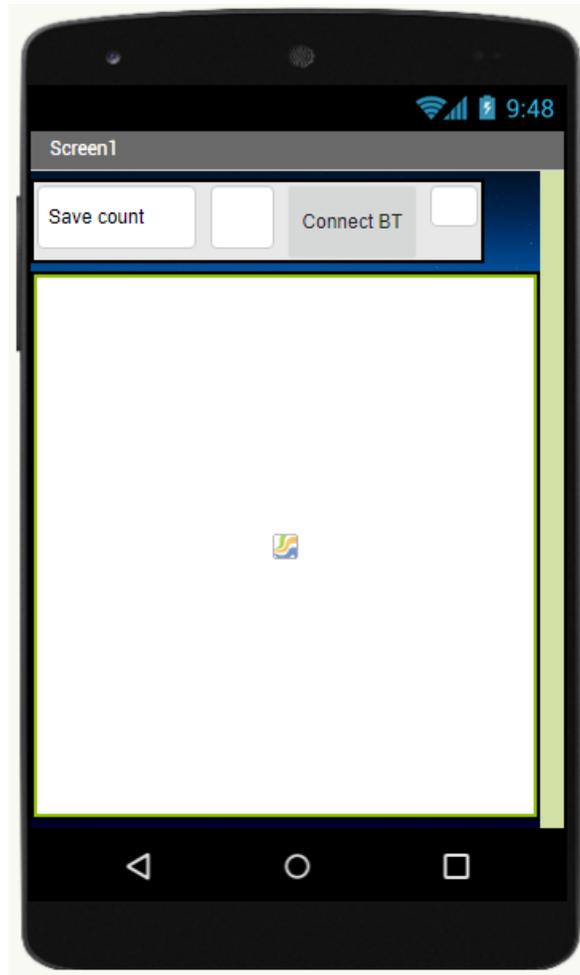
En esta metodología incremental se agregan funciones o mejoras al primer incremento para aumentar la utilidad del proyecto, y cada vez más se acerca a un objetivo final.

#### 4.5.1 INCREMENTO 1: ADQUISICIÓN DE IMÁGENES

En este primer incremento se utiliza MIT APP INVENTOR para diseñar y crear la aplicación para el proyecto, se programa y pone un diseño específico.

En el desarrollo para este incremento se realiza una app en este software donde se añaden textos, títulos, botones y funcionalidades de cámara y bluetooth a la app, luego se arma la lógica de programación por medio de bloques de funciones específicas. A la vez se realiza la programación en software de Arduino con el cual se establece la comunicación con la app y en el mismo código se ponen comandos para que el Arduino permita que la app tome fotos deseadas. Se conecta el Arduino a una computadora para cargar la programación e igual se le conecta con sus respectivos cables el módulo al Arduino y finalmente se conecta la app con el módulo bluetooth permitiendo comunicación entre teléfono y Arduino.

En validación se hacen pruebas múltiples veces probando la efectividad de la app y el código de programación, así verificando que la app haga su trabajo adquiriendo las fotografías deseadas.



**Ilustración 12 Fotografía de la APP del primer incremento.**

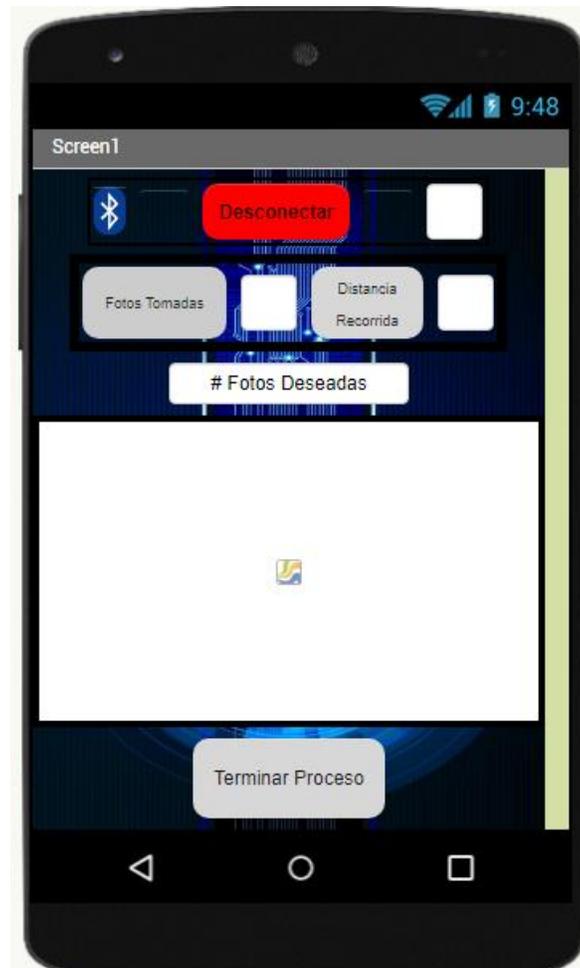
Fuente: (Propia,2021).

En la ilustración anterior se muestra la app en el primer incremento donde se observan lo añadido originalmente a la app, luego en la siguiente ilustración se muestra

#### 4.5.2 INCREMENTO 2: ADQUISICIÓN DE DATOS GPS

En el segundo incremento se registran datos de ubicación en la APP. En el desarrollo de este segundo incremento se le añaden funciones de GPS al MIT APP INVENTOR e igual se añaden más bloques de programación para añadir la característica de saber ubicación con la APP. Para así al momento de tomar cierta foto se guarde la ubicación de donde se tomó la foto.

Para la validación del segundo incremento se realizaron varias pruebas, probando que cada vez que la APP capture una foto también se guarde la ubicación GPS y así verificar que todo está bien.



**Ilustración 13 Fotografía de la app en el segundo incremento**

Fuente: (Propia,2021).

#### 4.5.3 INCREMENTO 3: MEDIR DISTANCIA Y CAMBIOS EN ESTÉTICA

En el tercer incremento se registrará la distancia recorrida del robot por medio de la app y el sensor "pedometer", el cual mide pasos a la vez que distancia recorrida. También se implementó sensor acelerómetro para registrar si el robot está en movimiento y así determinar si tomar o no

la foto. Y por último en este incremento se mejoró de una gran manera la estética de la aplicación así dejándola agradable para el usuario.

Para la validación de este tercer incremento se hacen pruebas para medir los sensores implementados y se hace comparación en estética y funcionamiento a su vez.

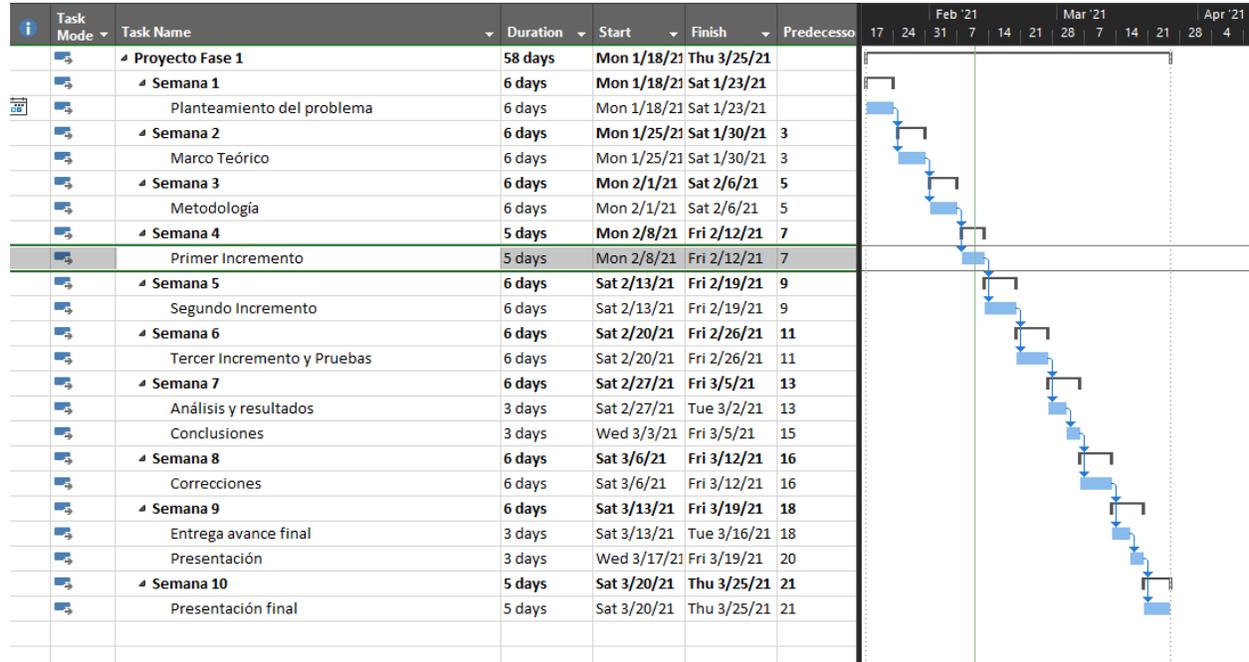


**Ilustración 14 Fotografía de la APP del tercer incremento.**

Fuente: (Propia,2021).

## 4.6 CRONOGRAMA

En la siguiente ilustración se muestra el cronograma de actividades de cómo se llevará a cabo el proyecto de investigación que tienen una duración de 10 semanas de trabajo.



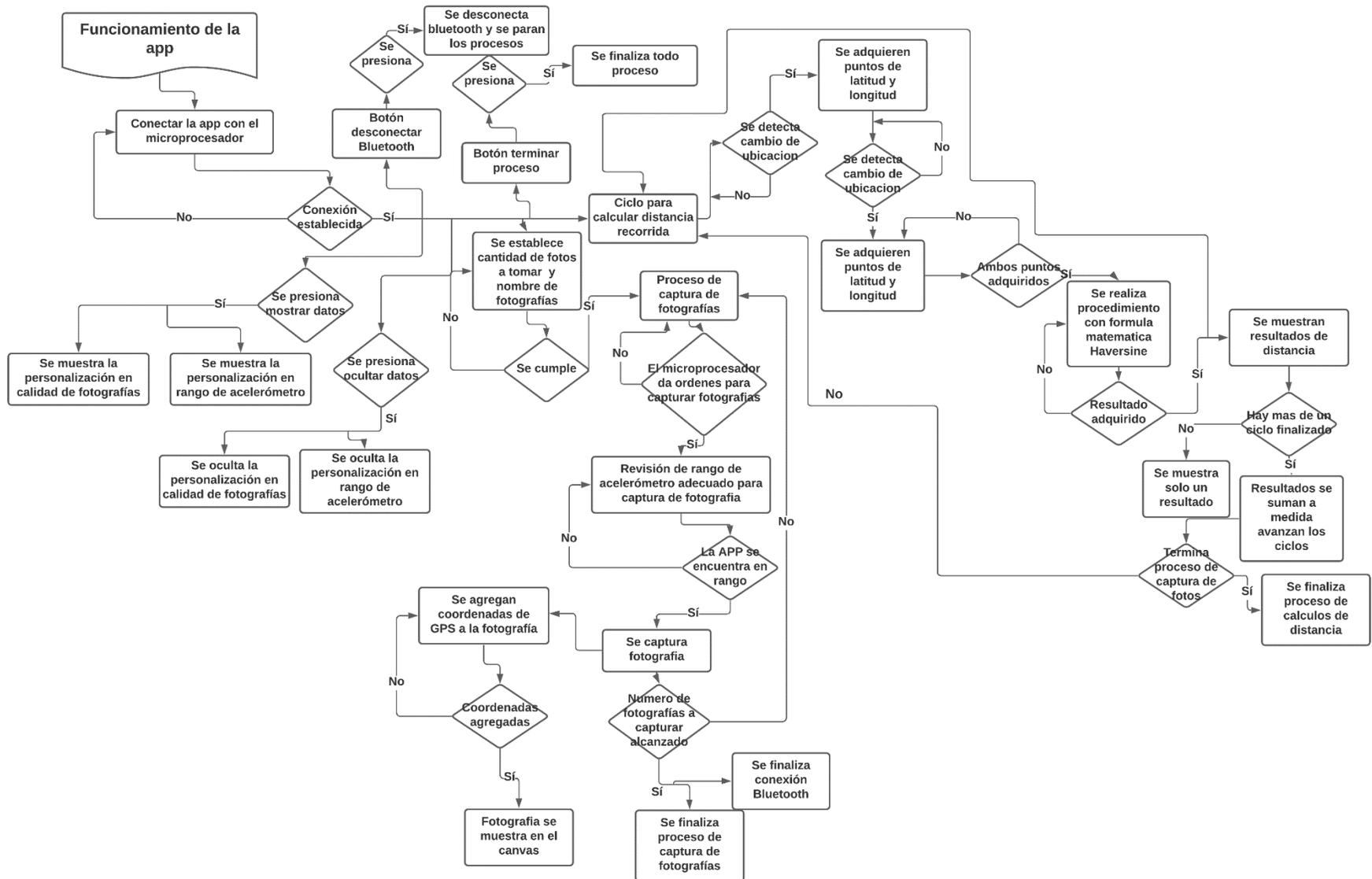
**Ilustración 15 Cronograma de trabajo.**

Fuente: (Propia,2021).

## V RESULTADOS Y ANÁLISIS

En este capítulo se detallará los resultados obtenidos durante la elaboración del proyecto y a la hora de probarlo en carretera, se verán fotos y análisis de por qué ciertas cosas si funcionan bien, porque otras no, y más.

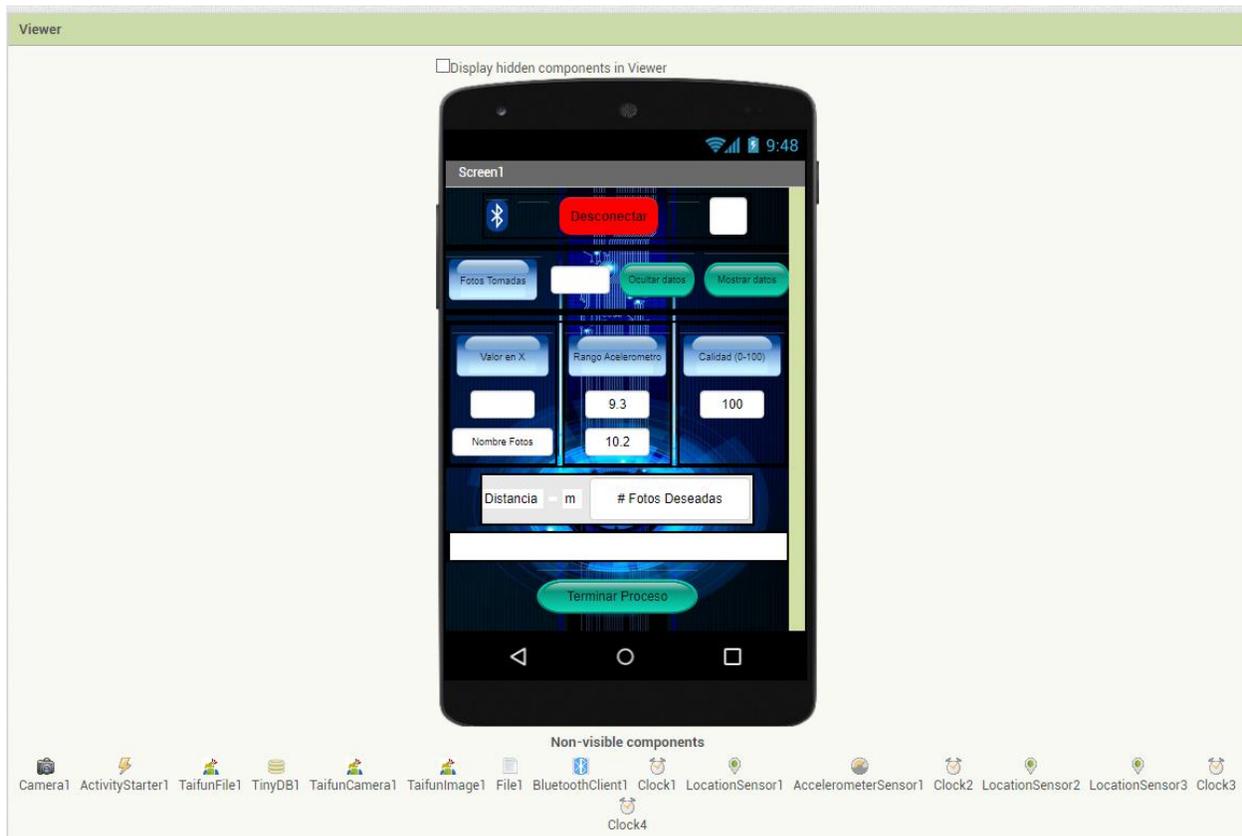
Con respecto al marco teórico se obtuvo una gran cantidad de información útil para el proyecto realizado, desde información de sensores, apps, sistemas para adquisición de datos, otros proyectos científicos y tecnológicos realizados y más. Con la investigación realizada se descubrió que el software más adecuado para realizar la app para el proyecto era MIT APP Inventor, en esta plataforma se realizó el proyecto. MIT APP Inventor fue creada con el propósito de abrir puertas a personas que quieren empezar a realizar aplicaciones ya sea para su trabajo, proyectos o investigaciones. De igual manera se logró entender para que se utilizan los softwares o apps para adquisición de datos y como facilitan al usuario y creador poder obtener varia información en un solo lugar. Se obtuvo de igual manera información y experiencia de ingenieros con sus proyectos y como las cámaras son un factor que depende de variedad de cosas, en como el costo de la cámara afecta en la calidad de imágenes a obtener en un proyecto. Y finalmente se descubrió una formula científica la cual fue clave para lograr obtener distancia recorrida en el proyecto por medio de coordenadas de latitud y longitud brindadas por el GPS en la app. En la ilustración 16 se muestra el algoritmo de funcionamiento de la app.



**Ilustración 16** Algoritmo de funcionamiento de la app

Fuente: (Propia,2021).

La Ilustración 17 muestra la app final, donde se observan ciertos botones para conectar y desconectar el bluetooth, unos datos mostrados, caja de texto para poner la cantidad de fotos deseadas a tomar en el viaje del robot, la imagen en el canvas y un botón de terminar proceso.



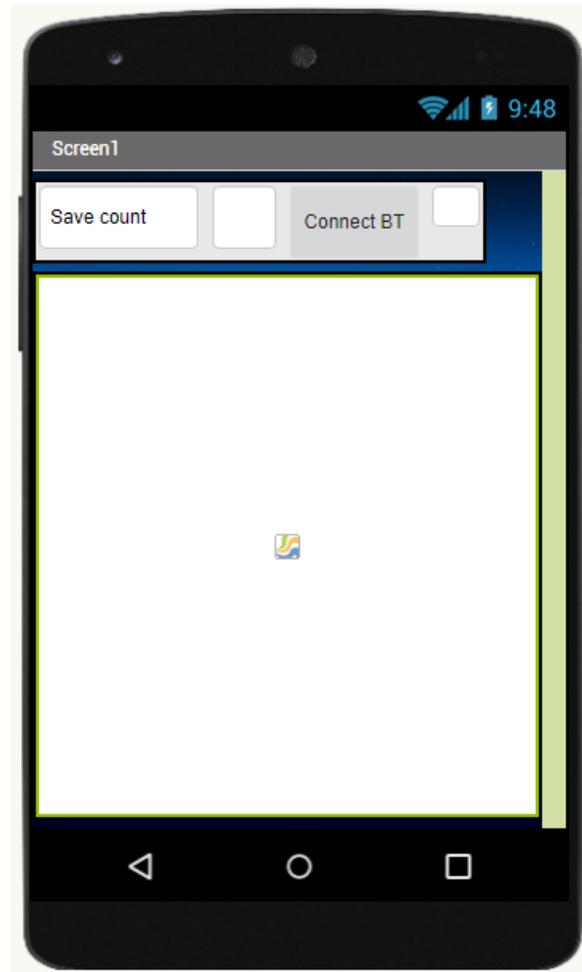
**Ilustración 17 Modelo de App Final**

Fuente: (Propia,2021).

En el transcurso del proyecto se empezó en el primer incremento el cual era sobre empezar la app con ciertas funcionalidades, se le agrego para tomar fotos de manera autónoma, esto para que el robot sostuviera el teléfono y el teléfono opere sin necesidad de interacción humana. En ese mismo incremento se estableció conexión entre la app y un Arduino con cierto código el cual le dice a la app cuando tomar la foto.

## 5.1 PRIMER INCREMENTO

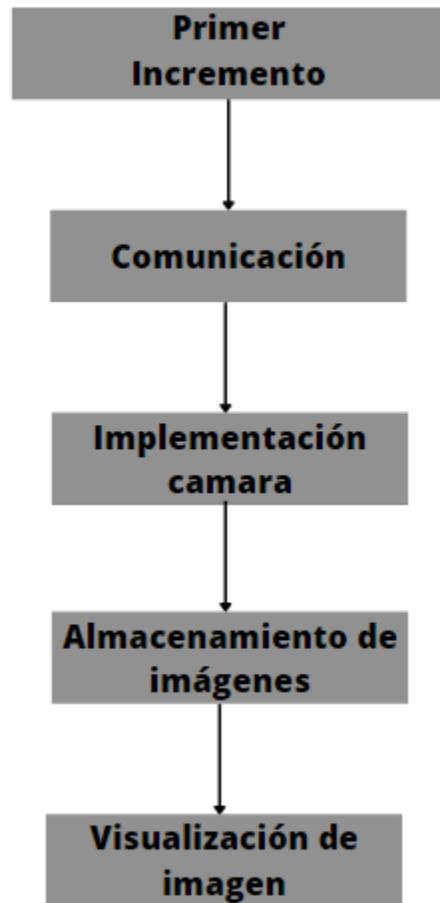
En el primer incremento se realizaron pruebas, sin embargo, este incremento no tenía mucha funcionalidad. La app solo permitía la conexión con el Arduino por medio de bluetooth y con una programación establecida la foto se tomaba de manera autónoma. En esta etapa no hubo ningún problema ya que todo funciono como se debía.



**Ilustración 18 Imagen del primer incremento de la app.**

Fuente: (Propia,2021).

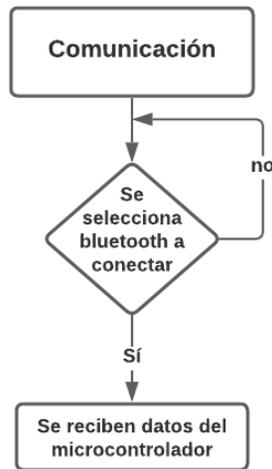
En la ilustración 19 se muestra cómo se inició la app con la metodología incremental, donde lo mostrado fue el inicio de la app y a medida se avanza en el documento se mostrarán los demás incrementos.



**Ilustración 19 Primer incremento.**

Fuente: (Propia,2021).

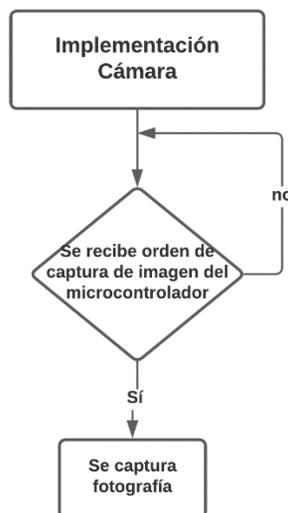
En la ilustración 19 se muestra por medio de la metodología incremental lo agregado al primer incremento del proyecto, en las siguientes ilustraciones se mostrarán algoritmos de funcionamiento de cada una de las características mostradas en la ilustración 19.



**Ilustración 20 Algoritmo de comunicación.**

Fuente: (Propia,2021).

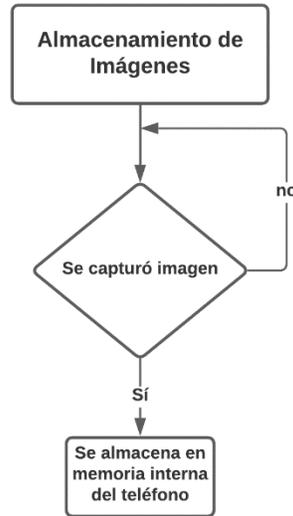
En la ilustración 20 se muestra el algoritmo de comunicación donde se configuró que, al conectar la app con el microcontrolador, la app reciba datos. En la ilustración 21 se muestra el algoritmo de cámara con para capturar una fotografía es necesario que la app reciba la orden del microcontrolador.



**Ilustración 21 Algoritmo de cámara.**

Fuente: (Propia,2021).

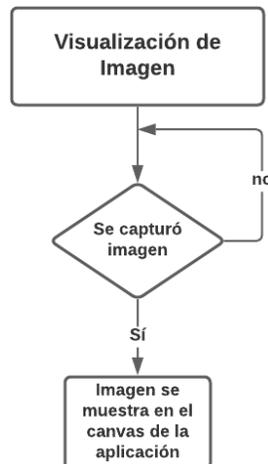
En la ilustración 22 se muestra el algoritmo de almacenamiento de imágenes donde si se captura una imagen esta será almacenada en el teléfono móvil.



**Ilustración 22 Algoritmo de almacenamiento de imágenes.**

Fuente: (Propia,2021).

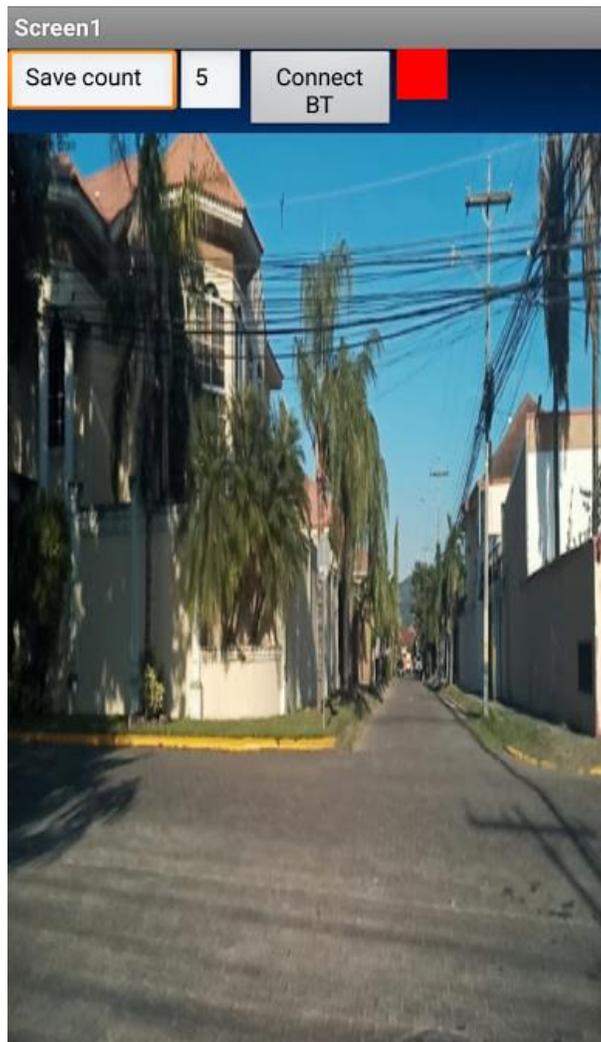
En la ilustración 23 se muestra el algoritmo de visualización de imagen de imagen donde después de cada captura de imagen, esta misma se muestra sobre el canvas añadido a la app.



**Ilustración 23 Algoritmo de visualización de imagen.**

Fuente: (Propia,2021).

En este primer incremento todo funciono de manera deseada, donde la app capturaba fotos en momentos ordenados por un Arduino, en la siguiente ilustración se muestra una captura de pantalla en la app desde un teléfono móvil al haber terminado una de las pruebas.



**Ilustración 24 Pruebas del primer incremento realizadas.**

Fuente: (Propia,2021).

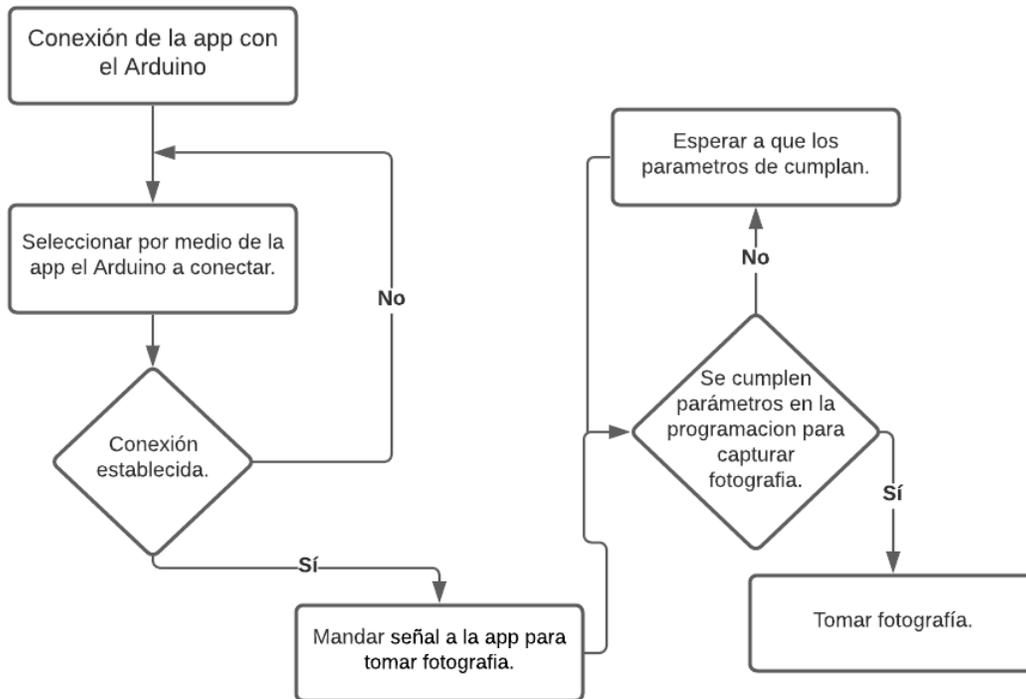
En la siguiente ilustración se muestra el resultado de la última imagen capturada por la sesión de pruebas en la ilustración anterior.



**Ilustración 25 Resultado de pruebas del primer incremento.**

Fuente: (Propia,2021).

### 5.1.1 COMUNICACIÓN BLUETOOTH



**Ilustración 26 Lógica utilizada con el Arduino**

Fuente: (Propia,2021).

En la ilustración 26 mostrada anteriormente se muestra el código utilizado por el Arduino con el cual se logra que la app tome foto en un momento específico. Es un código sencillo pero que puede ser modificado al gusto para que se tomen fotos en diferentes ocasiones e igual se puede usar esta app no solo para robots, sino que incluso para algún sistema de vigilancia y más, todo depende de la creatividad del que trabaja.

En el código lo que se hace es que con un temporizador de 15 segundos por el medio de bluetooth se manda un numero a la app, en este caso se manda el número 1, cuando la app recibe este número cada 15 segundos esta toma una fotografía. Este código se puede hacer más complejo, pero ya que el propósito del proyecto es crear una app con cierta funcionalidad y no hacer un robot u otro dispositivo para controlarla.

Entre los cambios que se le pueden hacer al código de Arduino para usar la app son infinitos, hay tantas maneras de programar y tantos proyectos diferentes en los cuales se usan códigos y sensores distintitos. Por ejemplo, la manera más común de cómo se podría modificar el código sería cambiando el temporizador. Otro ejemplo puede ser que el robot o dispositivo al recibir el valor de algún sensor se encargue de mandar un 1 a la app para que esta tome la foto en ese momento deseado y así la app toma la foto, podría ser con un sensor de proximidad que cuando el robot detecte algo a cierta distancia él ordene a la app tomar una foto, también podría ser un robot para monitoreo de temperatura de máquinas y al detectar que cierta maquina esta con altas temperaturas este se encargue de ordenar a la app que tome una foto de la maquina o máquinas.

### 5.1.2 UTILIZAR MÚLTIPLES PANTALLAS PARA LA APP

En la parte estética de la app no se pudo realizar de más de una pantalla ya que con MIT APP Inventor aún no hay una manera de poder mantener una conexión bluetooth al cambiar de página en la app, entonces lo que sucedía a la hora de experimentar con diferentes pantallas en el proyecto era de que la conexión entre el Arduino y la app se perdía lo cual es un gran problema. Debido a ese problema la app se tuvo que dejar en su versión final de una pantalla. Si se pueden realizar apps con múltiples páginas, pero si el proyecto depende de una conexión bluetooth ahí se hace un problema.

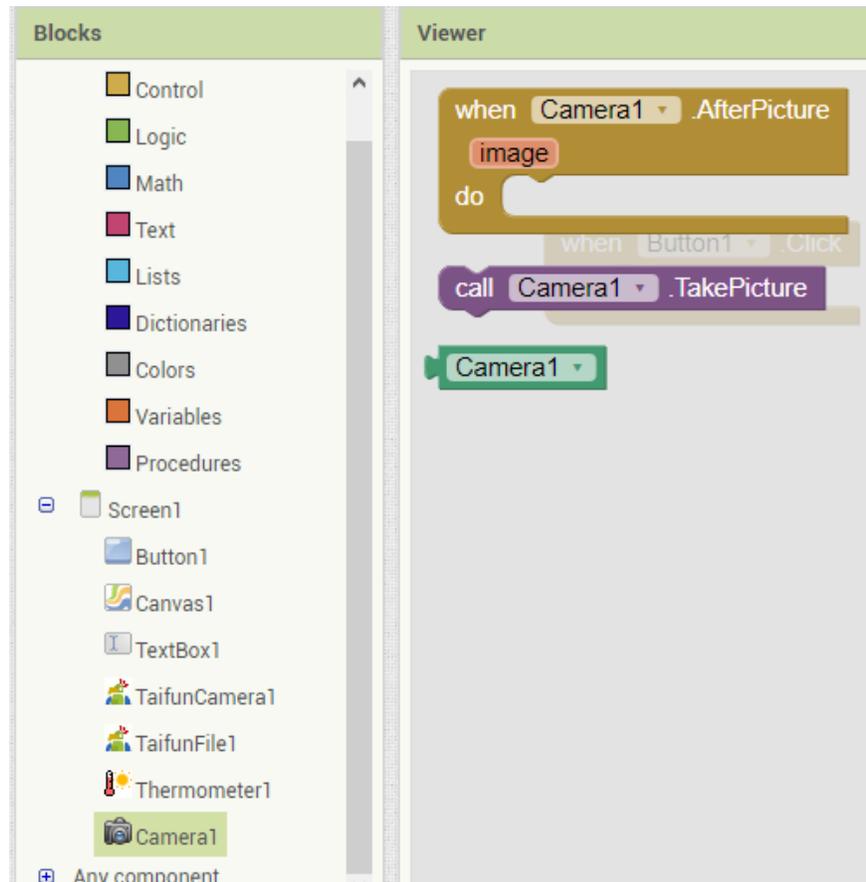
### 5.1.3 UTILIZACIÓN DE CÁMARA

El enfoque principal del proyecto es hacer una app de adquisición de datos con el cual por medio de una app un robot obtiene fotos. El uso de la cámara de manera autónoma era lo más fundamental del proyecto para lograr su objetivo.

El uso de la cámara fue abordado desde el primer incremento ya que sin esta funcionalidad el proyecto sería fallido, se empezaron a hacer programas enfocados al uso de cámara en un principio, en los cuales le lograba abrir la cámara con la app, pero no se tomaba la foto al menos que hubiera interacción humana con la app.

Se intentaron bastantes configuraciones en la programación y ajustes en la opción de cámara que brinda MIT APP Inventor, después de ciertos días fallidos en el cual no se lograban tomar la foto

de manera autónoma, investigando en cientos de páginas webs y artículos se descubrió que hasta hoy en día MIT APP Inventor no tiene esa opción de utilizar la cámara de manera autónoma.



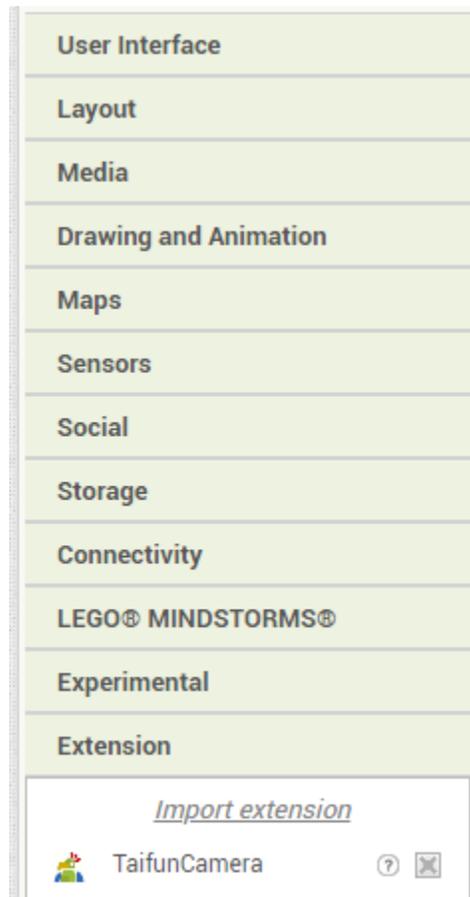
**Ilustración 27 Programaciones de la cámara disponibles en MIT APP Inventor.**

Fuente: (Propia,2021).

En la ilustración 27 se muestra lo limitada que es la opción de cámara en MIT APP Inventor, la cual solo permite tomar fotos de manera manual.

Luego después de días buscando una solución se descubrió que hay desarrolladores de extensiones para MIT APP Inventor, estos desarrolladores lo que hacen es brindar soluciones para usuarios de MIT APP Inventor, gracias a su conocimiento y experiencia en programación ellos desarrollan extensiones para diferentes motivos, hay extensiones para establecer una conexión bluetooth sin interacción humana, para el uso de wifi, para alarmas, para capturas de pantallas y más.

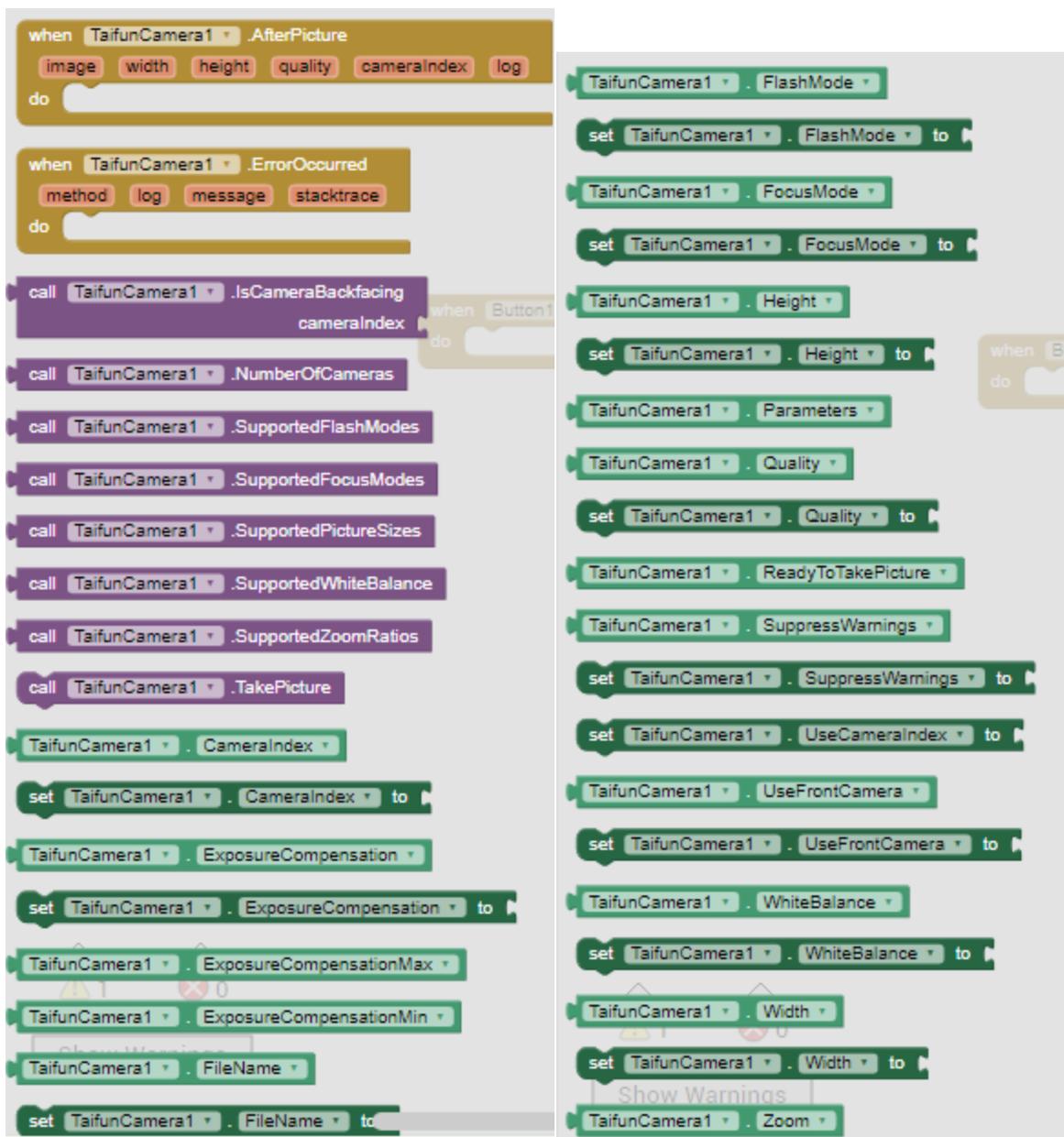
Al descubrir de la existencia de dichas extensiones se empezó a investigar en varias páginas si existía alguna extensión para lograr utilizar la cámara de un teléfono celular de manera autónoma y se llegó a una página donde finalmente se encontró la solución a ese problema. Esta extensión es mostrada en la siguiente ilustración y tiene el nombre de "TaifunCamera" extensión, esta extensión fue desarrollada por un ingeniero alemán que actualmente reside en Costa Rica.



**Ilustración 28 Extensión de MIT APP Inventor "TaifunCamera".**

Fuente: (Propia,2021).

Esta extensión obtenida se añadió a MIT APP Inventor por medio de la opción también visible en la ilustración anterior que dice "Import Extension". De igual manera ahí es donde se importan las extensiones conseguidas en diferentes lados para realizar cosas adicionales que MIT APP Inventor no permite realizar.



**Ilustración 29 Configuraciones permitidas por la extensión TaifunCamera.**

Fuente: (Propia,2021).

En las ilustraciones anteriores se muestra la variedad de configuraciones que nos permite utilizar esta extensión con MIT APP Inventor. Si se hace una comparación con la opción de cámara que nos brinda MIT APP Inventor en la ilustración 20 se puede observar una gran diferencia, se puede cambiar desde la cámara a utilizar, opción útil para teléfonos con múltiples cámaras traseras, se

puede cambiar zoom a la hora de tomar una fotografía, cambiar el ancho y altura de la foto, la calidad de imagen etc.

#### 5.1.4 CANVAS

La opción de canvas que ofrece MIT APP Inventor está orientada a dibujar en la pantalla y grabar lo dibujado o escrito, sin embargo, para el proyecto se utilizó esta opción para mostrar ahí la imagen tomada por la app, la app se encarga de tomar la foto y luego mostrarla en el canvas y luego la guarda en el teléfono celular.



**Ilustración 30 Imagen donde se muestra la app para visualizar el canvas.**

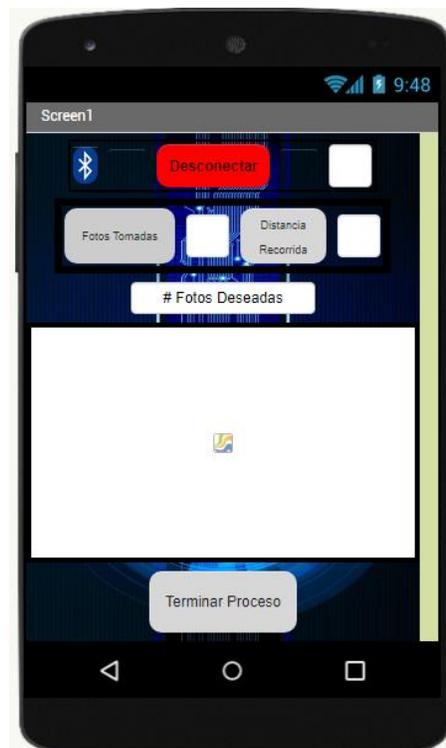
Fuente: (Propia,2021).

En la ilustración 30 se muestra la app y esta misma muestra una imagen capturada, esta imagen esta dibujada sobre el canvas, lo cual se hace por medio de una opción en MIT APP Inventor. De igual manera en el canvas se escribe la dirección del GPS y así cada imagen que se guarda en el teléfono celular se guarda de igual forma con ubicación GPS.

## 5.2 SEGUNDO INCREMENTO

En el segundo incremento del proyecto no se cambió nada en la estética del proyecto ya que este incremento consistía en agregar la funcionalidad de ubicación GPS con la app. Solo se agregaron bloques de programación para hacer funcionar esta característica agregada.

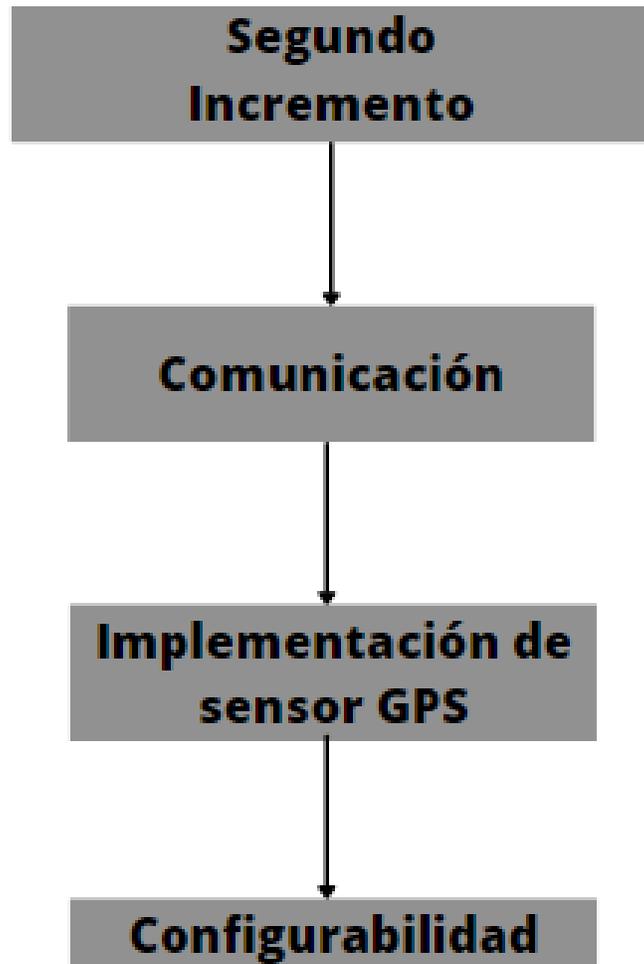
En la parte visual los datos de ubicación GPS se mostraban en el canvas añadido a la imagen capturada por la aplicación.



**Ilustración 31 Imagen de la app en su segundo incremento.**

Fuente: (Propia,2021).

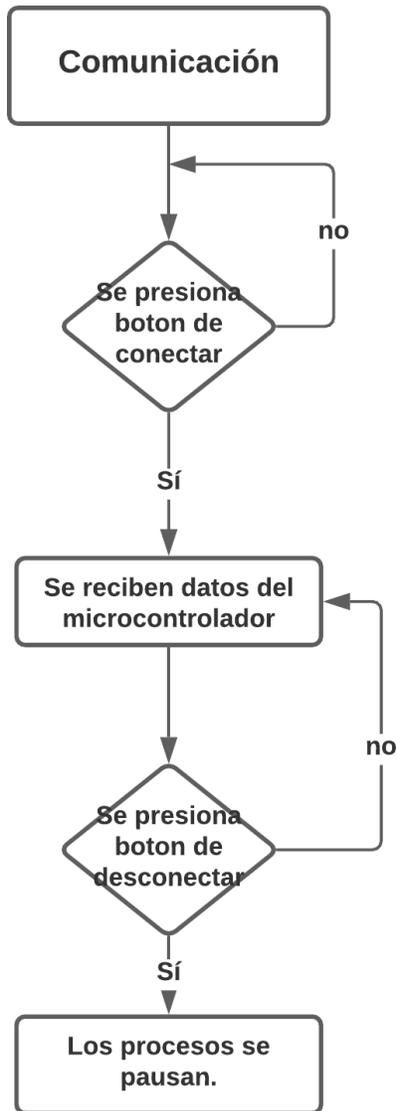
En la ilustración 32 se muestra lo realizado para el segundo incremento, donde se añade el sensor GPS al igual que configurabilidad.



**Ilustración 32 Segundo incremento.**

Fuente: (Propia,2021).

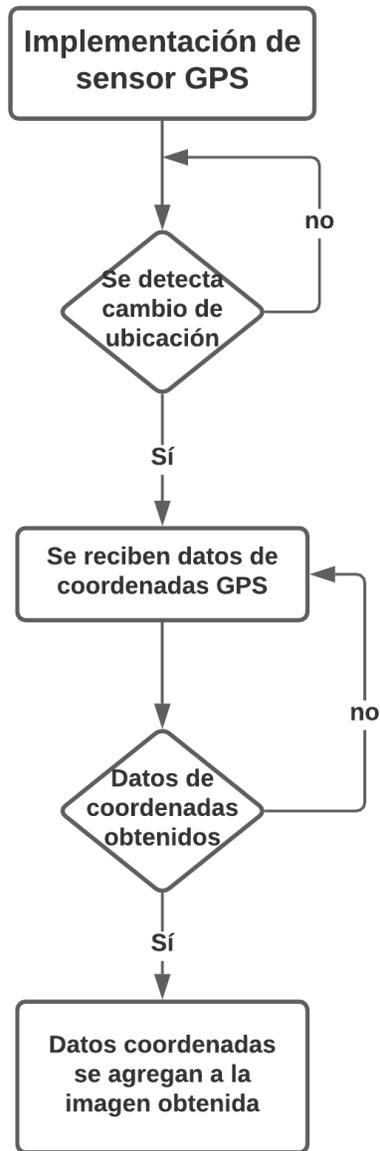
En la ilustración 33 se muestran el algoritmo de comunicación donde se le añadió a la app la opción de termina la conexión bluetooth con el microcontrolador.



**Ilustración 33 Algoritmo de comunicación.**

Fuente: (Propia,2021).

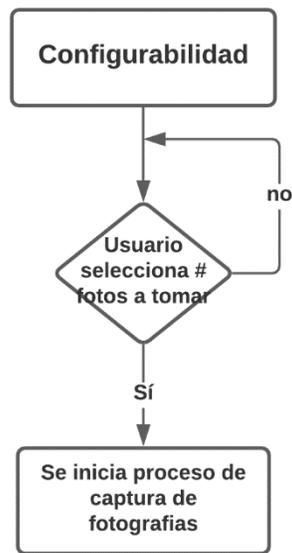
En la ilustración 34 se muestra el algoritmo de implementación del sensor de GPS a la app, en el cual el GPS obtiene coordenadas de la ubicación actual en cada cambio de ubicación de la app.



**Ilustración 34 Algoritmo de GPS.**

Fuente: (Propia,2021).

En la ilustración 35 se muestra la configurabilidad añadida a la app en el segundo incremento, donde la app permite al usuario decidir el número de fotografías a tomar en el proceso.

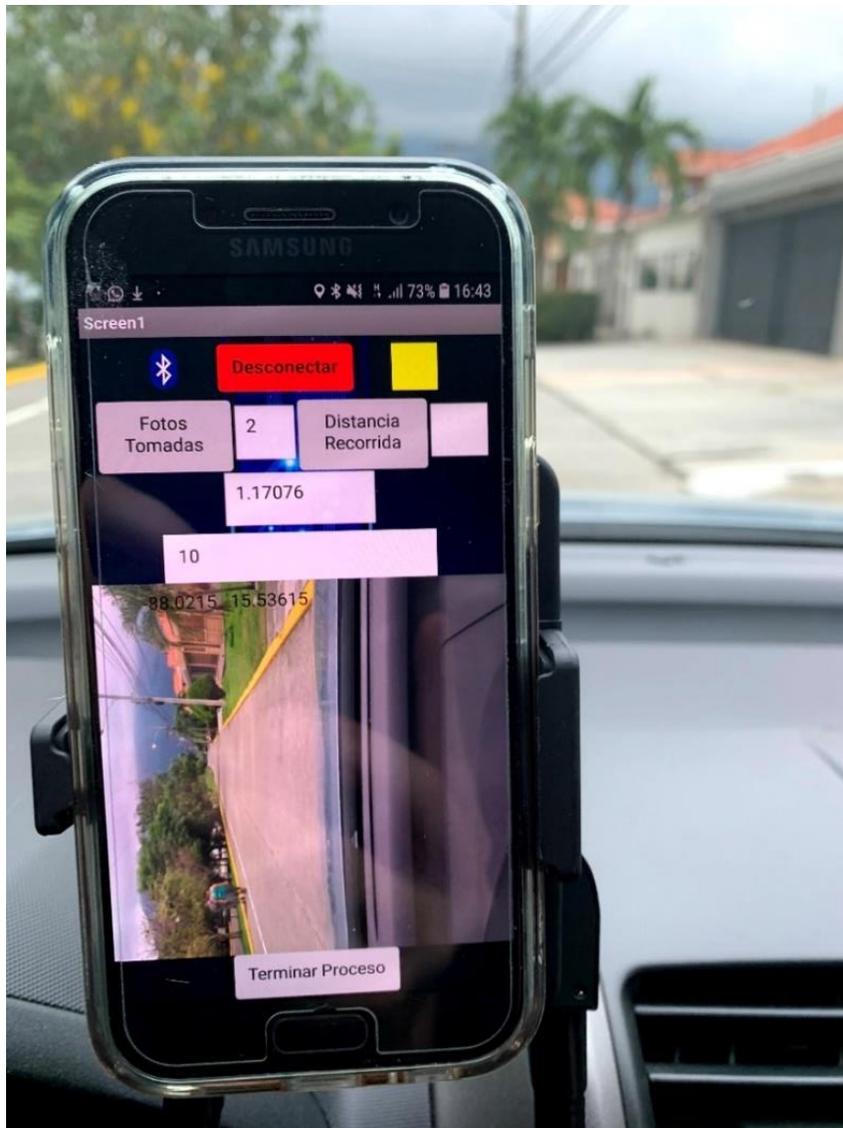


**Ilustración 35 Algoritmo de configurabilidad.**

Fuente: (Propia,2021).

En esta segunda etapa el único detalle encontrado fue que al iniciar la app no registra una ubicación GPS, esta se registra hasta que el teléfono cambia de ubicación, es decir, cuando ya el robot está en movimiento con el teléfono móvil colocado en su lugar. Luego de ya estar en movimiento el teléfono móvil con la app abierta todo funciona de la manera esperada.

En el segundo incremento se realizaron pruebas con dos teléfonos móviles en los cuales se muestran pruebas en las ilustraciones siguientes.



**Ilustración 36 Pruebas del segundo incremento realizadas con teléfono móvil 1.**

Fuente: (Propia,2021).

En la ilustración 36 se muestra prueba realizada por teléfono móvil 1 donde se observa que el teléfono móvil está posicionado de manera vertical por lo cual en la ilustración siguiente se observa que la imagen esta de manera horizontal.



**Ilustración 37 Resultado de pruebas del segundo incremento realizadas en teléfono móvil**

**1.**

Fuente: (Propia,2021).

En la ilustración 37 se puede observar la imagen obtenida donde se muestra de manera horizontal debido al posicionamiento del teléfono móvil y de igual manera se muestra en la imagen las coordenadas del GPS implementado en este incremento. En la siguiente ilustración se muestra prueba realizada con otro teléfono móvil donde está posicionado de manera horizontal.



**Ilustración 38 Pruebas del segundo incremento realizadas en teléfono móvil 2.**

Fuente: (Propia,2021).

En la ilustración 38 se muestra la app en su segundo incremento en una prueba realizada por el teléfono móvil dos, este se observa en manera horizontal lo cual nos da como resultado fotografías en forma deseada como en la ilustración siguiente.



**Ilustración 39 Resultado de pruebas del segundo incremento realizadas en teléfono móvil**

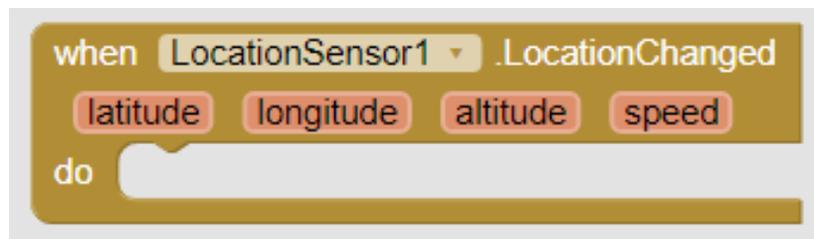
**1.**

Fuente: (Propia,2021).

### 5.2.1 SENSOR GPS

MIT APP Inventor por medio de los sensores que permite usar brinda un sensor de ubicación con el cual se haría el segundo incremento el cual consistía de guardar la ubicación de cada foto tomada, cuando se tomara una foto en ese instante igual se tomaría la ubicación con el sensor, guardando la longitud y latitud de donde fue tomada la fotografía.

Este sensor tiene múltiples opciones y bloques de código por usar, sin embargo, solo se usaron unos, esta desde habilitar el sensor, adquirir longitud, latitud y más. En la ilustración siguiente se muestra el bloque principal que se debe usar para utilizar este sensor en proyectos.



**Ilustración 40 Bloque de programación principal para utilizar el sensor de ubicación.**

Fuente: (Propia,2021).

De los problemas que causo este sensor fue que al recién abrir la aplicación no toma la ubicación, es necesario cambiar de ubicación para que el sensor empiece a marcar cambios en la app, este sensor como en la ilustración anterior permite adquirir datos de latitud, longitud, altitud y velocidad, de los cuales en el proyecto solo se utilizaron la latitud y longitud.

La ubicación obtenida por este sensor GPS utilizado en la app resulto ser muy eficiente y exacto, se realizaron varias comparaciones entre la app, Google Maps y My GPS Coordinates en los cuales las latitudes y longitudes en los lugares específicos a medir fueron muy similares y hasta casi las mismas.

En la tabla siguiente se muestran los resultados de las pruebas de exactitud del sensor GPS, donde se muestran los resultados de tres ubicaciones diferentes obtenidas por 2 apps diferentes y la app realizada.

App		My GPS Coordinates		Google Maps	
latitud	longitud	latitud	longitud	latitud	longitud
15.53631	-88.02256	15.536138	-88.021480	15.5361012	-88.0214923
15.53792	-88.02761	15.537776	-88.027711	15.5378436	-88.0276866
15.5243	-88.02723	15.524300	-88.027036	15.5242755	-88.0270549

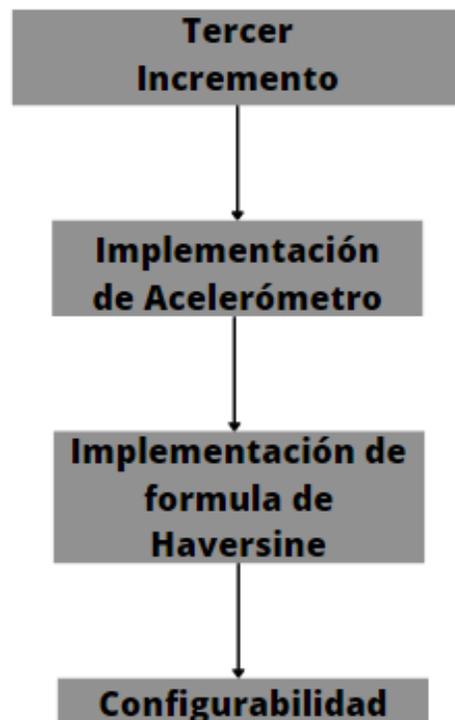
**Tabla 1 Comparación de ubicaciones de GPS capturadas de diferentes fuentes.**

Fuente: (Propia,2021).

### 5.3 TERCER INCREMENTO

En el tercer y último incremento se hicieron varias mejoras al proyecto y de igual manera fue donde más resultados importantes se consiguieron acerca de sensores nuevos a implementar. En este incremento se implementaron el acelerómetro, el podómetro, el termómetro, de los cuales el termómetro quedó fuera del proyecto ya que no hay forma de utilizar termómetro con teléfonos móviles actuales, los únicos teléfonos que permiten utilizar esa opción son dos teléfonos antiguos de Samsung que traían un sensor de temperatura ambiental.

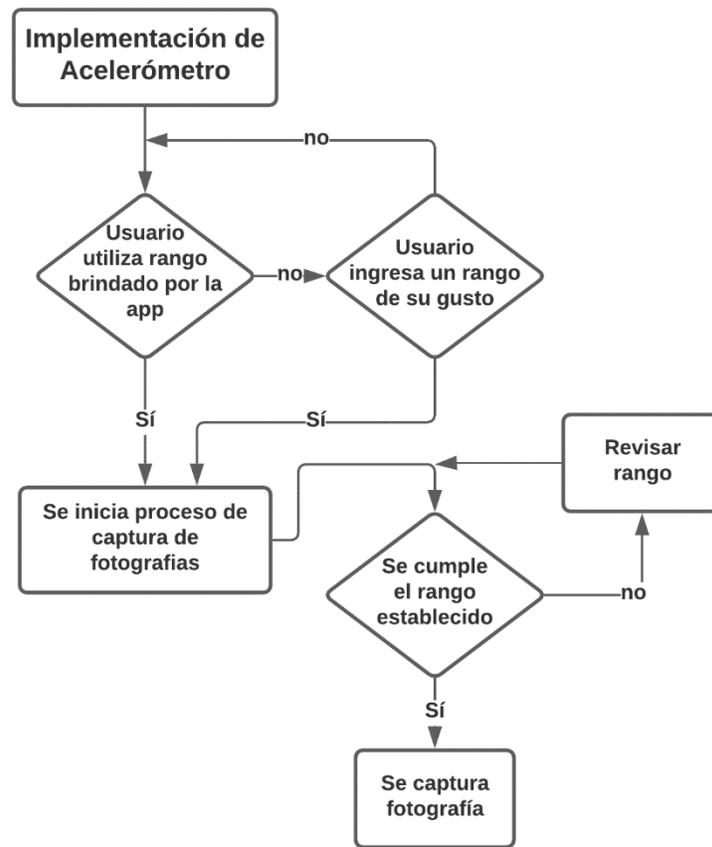
En la ilustración 41 se muestra el tercer incremento de la app donde se observan lo añadido a la app en el incremento.



**Ilustración 41 Tercer Incremento de la app.**

Fuente: (Propia,2021).

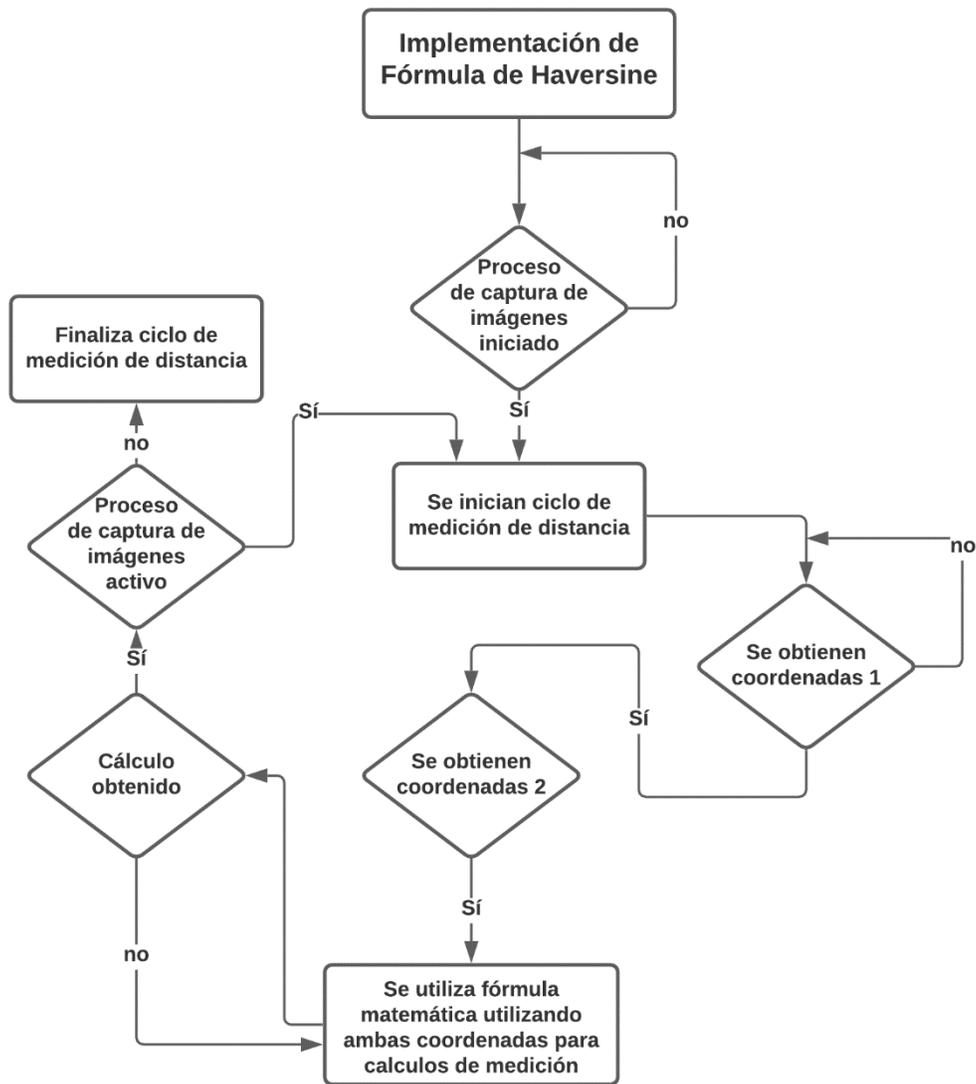
En la ilustración 42 se muestra el algoritmo de implementación de acelerómetro a la app. En el cual este sensor da un rango a utilizar con la app para establecer métricas.



**Ilustración 42 Algoritmo implementación de acelerómetro.**

Fuente: (Propia,2021).

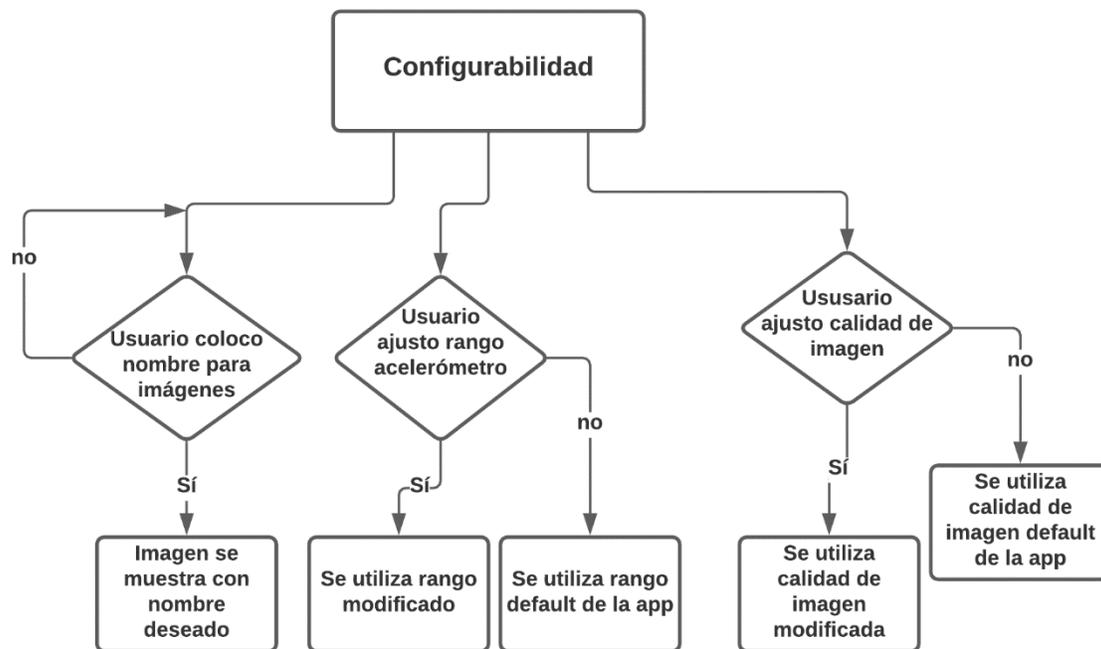
En la ilustración 43 se muestra el algoritmo de implementación de la fórmula matemática de Haversine, con la cual se calculan distancias por medio de la utilización del GPS en la app.



**Ilustración 43 Algoritmo de implementacion de fórmula matemática de Haversine.**

Fuente: (Propia,2021).

En la ilustración 44 se muestra el algoritmo de configurabilidad del tercer incremento donde se permite al usuario cambiar calidad de imágenes, rango de acelerómetro y nombre de almacenamiento de imágenes.



**Ilustración 44 Algoritmo de configurabilidad tercer incremento.**

Fuente: (Propia,2021).

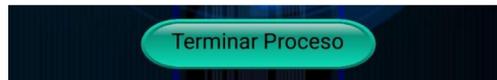
Se realizaron de igual manera cambios estéticos en el tercer incremento para una app de buena calidad amigable y agradable para el usuario.



**Ilustración 45** Captura de pantalla en teléfono móvil al presionar mostrar datos.

Fuente: (Propia,2021).

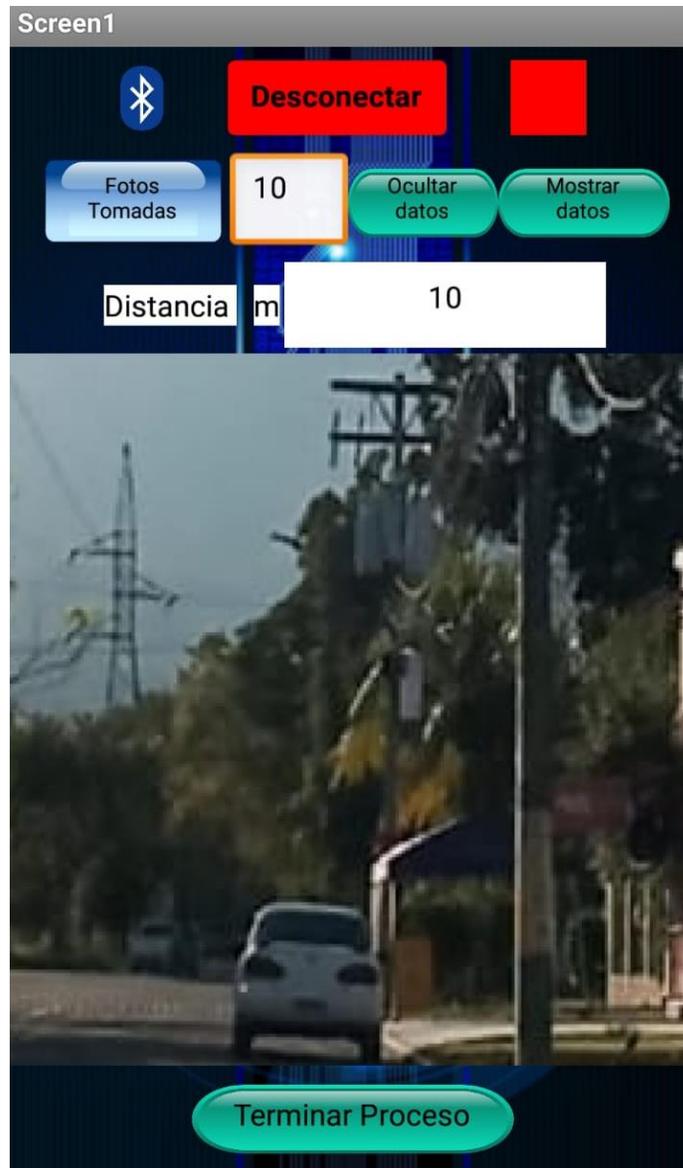
En la ilustración anterior se muestra la app en el tercer incremento cuando se presiona la opción de mostrar datos, luego en la siguiente ilustración se muestra con la opción de ocultar datos. Esto se realizó con el propósito de hacer la app más amigable y experimentable para el usuario.



**Ilustración 46 Captura de pantalla en el teléfono móvil al presionar ocultar datos.**

Fuente: (Propia,2021).

En la ilustración 46 se muestra como se ve la app en este último incremento, más adelante se detallará sobre las pruebas realizadas y se mostrará en una imagen la app en teléfonos móviles. Como se observa en este incremento se puede monitorear las fotos tomadas, permite conectarse y desconectarse al Arduino por bluetooth, permite escribir la cantidad de fotos deseadas por el usuario, se muestra también distancia recorrida, el canvas donde se visualiza la foto tomada junto con los datos de ubicación y por último un botón para terminar el proceso general.



**Ilustración 47 Pruebas del tercer incremento realizadas.**

Fuente: (Propia,2021).

En la ilustración anterior se muestra la app en el tercer incremento durante una prueba, ahí observamos el número de fotografías capturadas. En la ilustración siguiente se muestra la última fotografía capturada en una de las pruebas en el tercer incremento.

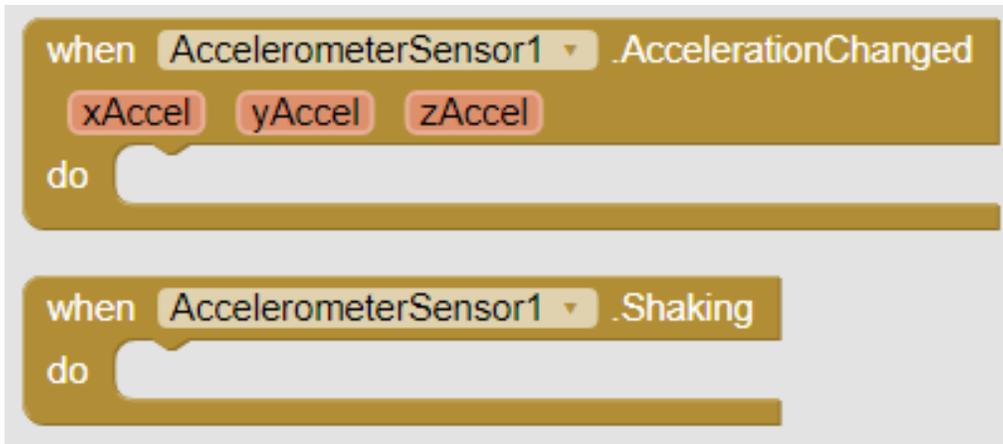


**Ilustración 48 Resultado de pruebas del tercer incremento.**

Fuente: (Propia,2021).

### 5.3.1 SENSOR ACCELERÓMETRO

A medida el proyecto iba avanzando se buscaron mejoras y adiciones para que el proyecto fuera cada vez más completo, con más funciones, y ahí se decidió empezar a utilizar el acelerómetro el cual ayudaría a que la app tuviera una mejor idea de cuando tomar foto, es decir, que si la app detectaba que el robot estaba en movimiento esta misma no tomaría la foto hasta que el robot estuviera estático, así garantizando que la foto no saliera movida.



**Ilustración 49 Opciones principales del acelerómetro.**

Fuente: (Propia,2021).

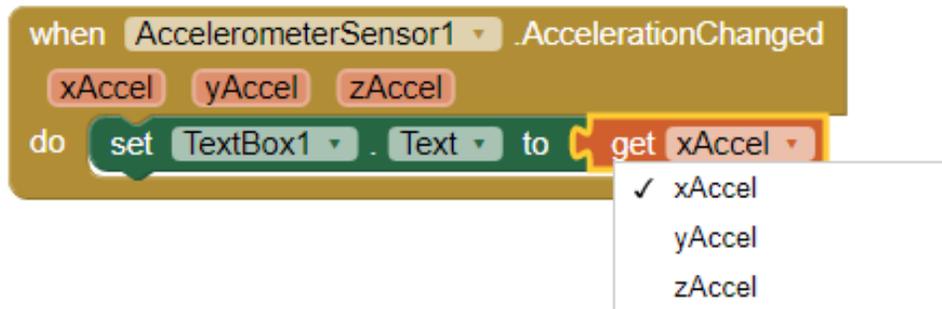
En la ilustración anterior se muestran dos de las opciones principales que nos brinda MIT APP Inventor con el sensor acelerómetro. Esas no son las únicas opciones que brinda el sensor, pero son las principales.

La primera opción que se empezó a utilizar en el proyecto fue la de "Shaking", el problema con la opción de "Shaking" era que requería cierta fuerza para que la app supiera que estaba temblando el teléfono, lo que pasaba era que la app no iba a detectar normalmente que temblaba el teléfono por el hecho que el robot no necesariamente movería tanto el teléfono a la hora de estar en movimiento.

Cabe aclarar que esta opción si se recomendaría probar con proyectos diferentes para confirmar si sería posible su uso, pero en este proyecto no se tenía el robot para hacer las pruebas, toda prueba se hizo en un automóvil.

Luego se procedió a utilizar la otra opción que dice "AccelerationChanged", esta se encargaba de detectar hasta los más mínimos movimientos del teléfono celular. Esta opción del acelerómetro permite obtener los datos de aceleración en X, Y y Z, los cuales se utilizaron todos para realizar pruebas con la app.

En la ilustración a continuación se muestra una parte del código donde lo que hace es obtener la aceleración en X del teléfono celular, la cual se guarda en un texto y luego ese texto se utiliza en otras partes de la programación de la app para determinar el movimiento. Igualmente, en la ilustración 50 se muestra que el sensor brinda la opción de utilizar datos en X, Y y Z.



**Ilustración 50 Ejemplo básico para obtener datos de aceleración en X.**

Fuente: (Propia,2021).

El sensor de acelerómetro que brinda MIT APP Inventor fue utilizado en la aplicación e igual se hicieron pruebas con él en físico, funciona de buena manera lo único es que no es tan fácil utilizarlo, no es fácil en el sentido que al sacar un rango de números que oscilan al tener el teléfono celular estático en cierta posición, con el más mínimo cambio en inclinación del teléfono celular, ya sea hacia arriba o hacia abajo, hace que el rango en donde el teléfono esta estático cambie.

En la app se programó para que el teléfono tome fotos en cierto rango de aceleración en x ya que con eso se determina si hay movimiento o no, sin embargo, al estar utilizando el app en un vehículo en el campo o carretera hay que considerar algunas cosas como que al estar en movimiento en una superficie plana funciona perfecto ya que el rango no cambia, pero al estar en una subida o bajada ese rango cambia y ya no funciona de la manera programada el sensor.

La app fue probada en una colonia con terreno plano y terreno irregular y al estar en terreno plano funciono de manera perfecta, al estar en movimiento el carro no se tomaban fotografías y al estar estático el carro si se tomaban las fotografías. Luego al estar en algunas inclinaciones el rango cambiaba y la app no recibía los números del rango establecido entonces no podía tomar fotografías al estar estático el carro por el hecho que estaba en otro rango.

Otra consideración a la hora de realizar pruebas es que al menos en un automóvil, donde el motor tiende a darle vibraciones al auto incluso al estar estático, esto dio problemas ya que a pesar de estar estático el carro el acelerómetro recibía cambios en movimiento en algunas ocasiones, por lo cual algunas veces al estar estático el automóvil la app aun así no tomaba la fotografía.



**Ilustración 51 Imagen de la app.**

Fuente: (Propia,2021).

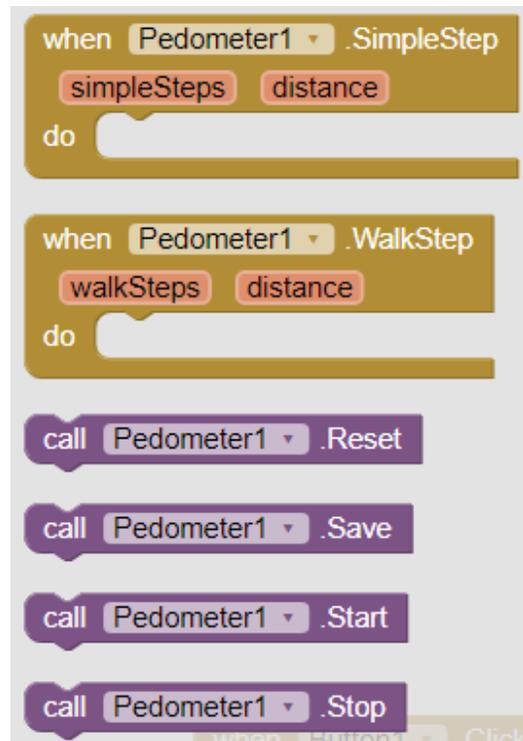
En la ilustración anterior se muestra una imagen de un teléfono celular con la app realizada, en esta fotografía se muestra el teléfono móvil en un automóvil. También se observa que el camino es una superficie plana por lo cual justo en el momento se estaba tomando una fotografía ya que el automóvil estaba estático. Los datos del acelerómetro se pueden observar en el cuadro de texto de la tercera fila, en esa fila solo hay un cuadro de texto y ahí es donde está el rango de los valores obtenido por el acelerómetro en X. El rango utilizado en la programación para que el teléfono móvil tomara fotografías era entre 9.3 a 10.2.

En proyectos si existe la duda de que opción del acelerómetro utilizar en MIT APP Inventor es recomendado utilizar ambas opciones para descubrir cual es la que funciona mejor de acorde al proyecto, por ejemplo, si el robot anda en terrenos inestables es probable que se utilice de una manera adecuada la opción de "Shaking" del acelerómetro y de esa manera no sería necesario andar poniendo rangos en la programación para determinar si se está en movimiento o no.

Para evitar el problema donde las fotografías no se capturaban debido a las vibraciones del vehículo lo que se realizó fue cambiar la sensibilidad del acelerómetro fijándolo en una configuración con menor sensibilidad. De igual manera los rangos de acelerómetro variaron un poco en diferentes teléfonos móviles probados, con lo cual se llegó a obtener un rango efectivo para múltiples teléfonos móviles el cual está entre 9.3 hasta 10.2.

### 5.2.3 PODÓMETRO

Casi en etapas finales del proyecto siempre con el pensamiento de innovar y brindar más funcionalidad a la app se pensó en una manera de medir distancia recorrida en los viajes para tomar fotografías. Se investigo y busco en las opciones que brinda MIT APP Inventor para medir distancias y la única opción encontrada fue el podómetro.

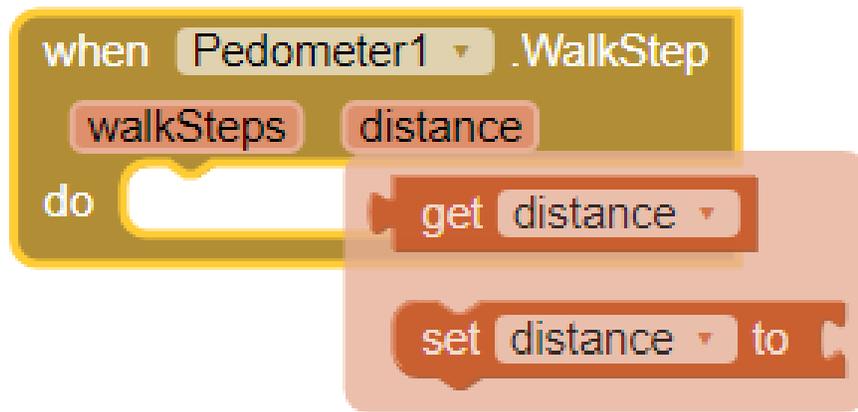


**Ilustración 52 Opciones del sensor Podómetro.**

Fuente: (Propia,2021).

En la ilustración anterior se muestran algunas de las opciones de programación que MIT APP Inventor brinda a creadores de aplicaciones. De los más importantes son las dos opciones en naranjas que son las formas de adquirir datos con el sensor, ambas opciones fueron probadas y se hablara de las mismas adelante. También están las opciones de Start y Stop que son importantes ya que en la programación de la app es necesario especificar cuando se empieza a utilizar el sensor y cuando se termina de utilizar.

A continuación, se muestra una ilustración de cómo obtener la distancia por medio de la opción "WalkStep" que nos brinda el sensor podómetro en MIT APP Inventor.



**Ilustración 53 Función del sensor podómetro "WalkStep".**

Fuente: (Propia,2021).

En el proyecto se utilizaron ambas opciones con la app, la opción de WalkStep y la opción de SimpleStep, ambas utilizadas para obtener distancias recorridas. En la ilustración 40 mostrada arribase muestra donde nos permite adquirir los datos de distancia y esa distancia se puede registrar por ejemplo en una caja de texto.

Ambas opciones del podómetro fueron probadas en carretera con la app y los resultados no fueron nada exactos, este sensor fue hecho para medir pasos entonces era de imaginarse que no registraría bien la distancia al probar la app en un vehículo, y tal como lo imaginado en las pruebas se comprobó que no es factible utilizar este sensor para medir distancias, solo si son distancias caminadas por el usuario.

En la ilustración 41 a continuación se muestra la app en momentos de pruebas donde se puede observar la distancia recorrida la cual está en metros, y esta distancia recorrida solo registro 21.9 metros después de haber recorrido más de 600 metros en automóvil. En este teléfono móvil de la ilustración se utilizó la opción de simple step.



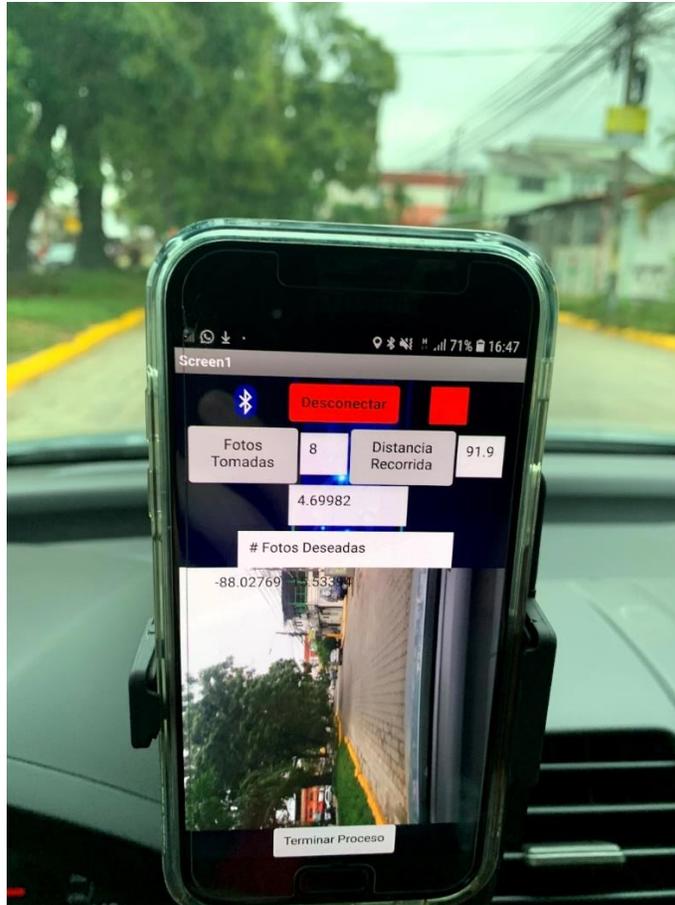
**Ilustración 54 App a la hora de prueba.**

Fuente: (Propia,2021).

En la ilustración 54 se muestra la app al final del proceso, donde ya se habían fotografiado las 10 imágenes deseadas por el usuario, ahí se muestra lo mencionado anteriormente de que en todo el viaje solo se registraron 21.9 metros de recorrido.

En la ilustración siguiente se muestra la app en otro dispositivo celular en el cual en vez de utilizar la opción de simple step se utilizó la opción de walk step. En la ilustración se puede observar los datos de la aplicación como la aceleración en x, fotografías tomadas y distancia recorrida.

En ese viaje para tomar fotografías se puede observar que solo eran 8 fotografías a tomar deseadas y que en el viaje la app solo registro 91.9 metros de recorrido, en la cual se recorrido más de 500 metros.



**Ilustración 55 Pruebas con la app en otro teléfono móvil.**

Fuente: (Propia,2021).

Al ver las dos ilustraciones, la 54 y 55, se observa que, en ambas opciones del podómetro, la simple y walk step, no funcionan de una manera óptima para el proyecto ya que este sensor este hecho para medir pasos y distancia de acuerdo a pasos. Cabe sacar a luz que la opción de walk step funciono un poco mejor ya que registro ms distancia que simple step, pero de igual manera los datos no estuvieron ni cerca de lo real.

## 5.2.2 SENSOR DE TEMPERATURA

En la metodología se procuraba usar un sensor de temperatura para medir la temperatura interna del teléfono ya que al estar tomando fotografías por mucho tiempo el teléfono se calentaría y por eso se quería poner en la app un sistema de protección con este sensor. Investigando sobre cómo

usar el sensor y haciendo pruebas se descubrió que MIT APP Inventor no permite usar el sensor de temperatura interno de teléfonos celulares, lo que permite es utilizar sensor de temperatura para medir la temperatura del ambiente y sin embargo no todo teléfono trae un sensor que le permita hacer eso, investigando se descubrió que solo dos modelos antiguos de Samsung tenían ese sensor, entonces no se logró utilizar sensor de temperatura.

En la ilustración siguiente se muestra una de las maneras en las cuales se hicieron pruebas del sensor de temperatura, se realizaron varias hasta que al no ser exitoso el uso del sensor se descubrió que era imposible utilizar tal sensor con los teléfonos celulares en los que se hicieron las pruebas.

### 5.2.3 MEDICIÓN DE DISTANCIA

En la medición de distancia se implementó primero es sensor podómetro en la aplicación, sin embargo, este sensor no funciona de la manera adecuada ya que es un sensor creado para medir pasos. Al momento de hacer pruebas las medidas de distancia con este sensor eran muy distintas a las distancias medidas en Google Maps. Luego se implementó una fórmula matemática de Haversine y por medio de esta se logró medir distancias de una manera exacta como se observa en los resultados siguientes.

Para medir distancias por medio de GPS se utilizó una fórmula matemática de Haversine con la cual se comparan dos puntos diferentes con GPS y con estos se calcula una distancia recorrida.

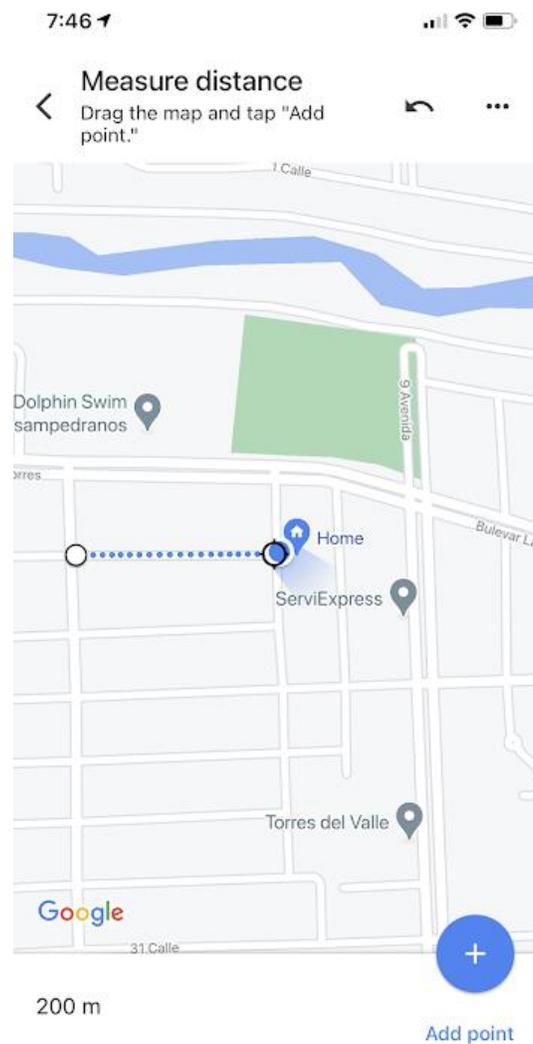
$$d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

#### **Ecuación 1 Fórmula de Haversine.**

Fuente: (Gilbert,2019)

En la ecuación 1 se muestra la fórmula matemática utilizada donde  $\phi$  es la latitud y  $\lambda$  es la longitud, en la fórmula se muestra que se utiliza una latitud y longitud 1 al igual que una latitud y longitud 2.

Luego esta fórmula se implementó en la programación en un procedimiento donde cada cierto tiempo se obtendría una distancia. Se tomaron mediciones en un tramo de 200 metros mostrado en la siguiente ilustración.

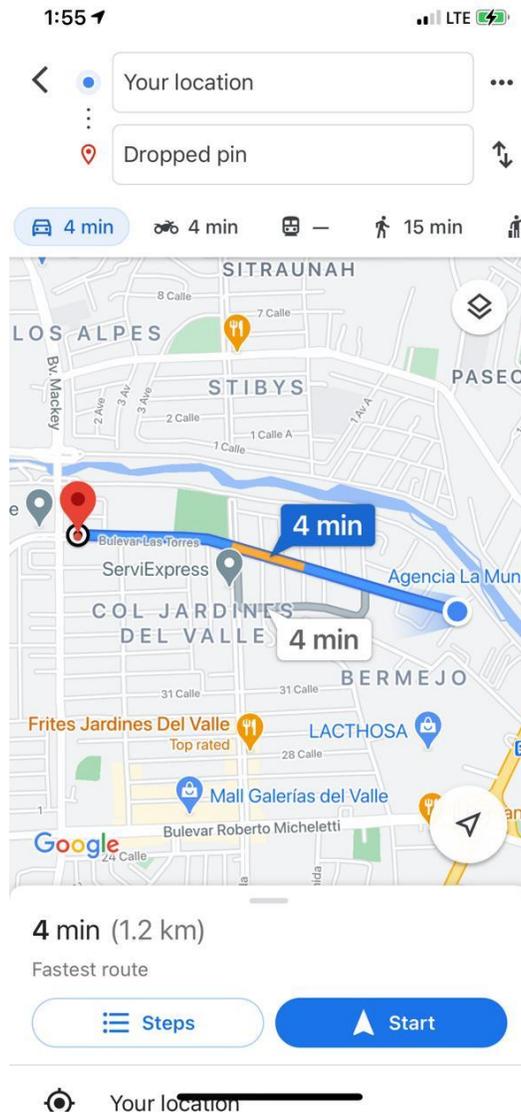


**Ilustración 56 Vista de tramo recorrido para pruebas #1.**

Fuente: (Propia,2021).

En la ilustración 57, 58 y 59 se muestran diferentes tramos donde de igual manera se realizaron pruebas de medición de distancia por medio de la app. En el tramo numero dos se realizaron

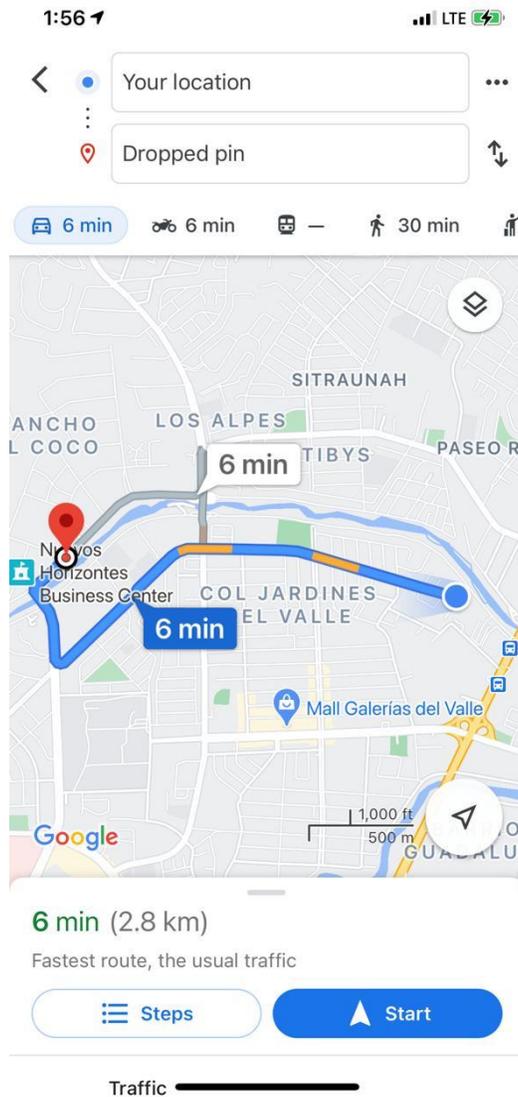
varias pruebas, el tramo tenía una distancia de 1.2km, en donde la app en todas las pruebas midió una distancia recorrida de 1.2km.



**Ilustración 57 Vista de tramo recorrido para pruebas #2.**

Fuente: (Propia,2021).

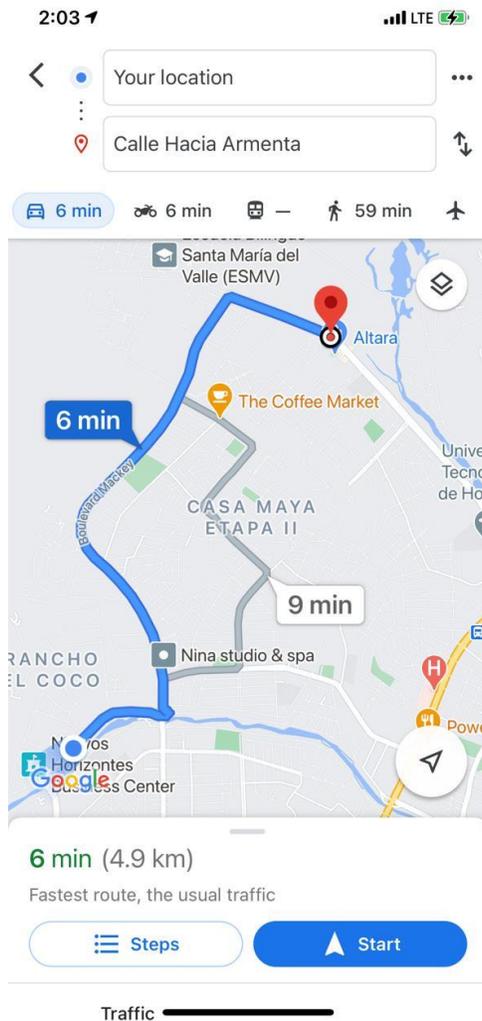
En el tramo número tres se realizaron varias pruebas, el tramo tenía una distancia de 2.8km, en donde la app en todas las pruebas midió una distancia recorrida de 2.8km.



**Ilustración 58 Vista de tramo recorrido para pruebas #3.**

Fuente: (Propia,2021).

Finalmente, en el tramo número cuatro también se realizaron varias pruebas, el tramo tenía una distancia de 4.9km, en donde la app en todas las pruebas midió una distancia recorrida de 4.9km. Dando un excelente resultado esperado en la efectividad de medición de distancias obtenido por la app mediante el uso de la fórmula matemática de Haversine.



**Ilustración 59 Vista de tramo recorrido para pruebas #4.**

Fuente: (Propia,2021).

Luego se compararon distancias recorridas en diferentes viajes. Estas pruebas se realizaron en el tramo de la ilustración 46. En la siguiente tabla se muestran 4 viajes realizados con la app y las distancias recorridas por la misma donde se muestra una buena exactitud de distancia medida por la app.

<b>APP</b>	<b>Google Maps</b>
199.91152 m	200 m
1.207 km	1.2 km
2.810 km	2.8 km
4.920 km	4.9 km

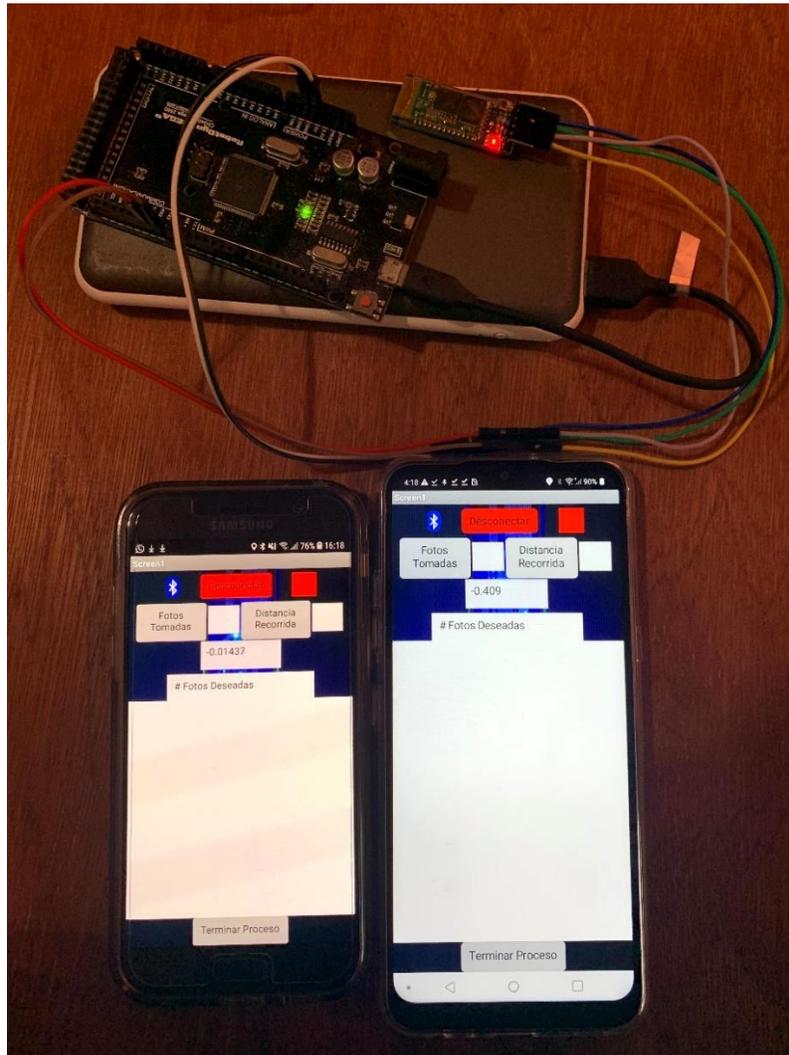
**Tabla 2 Comparación de distancias medidas entre la app y Google Maps.**

Fuente: (Propia,2021).

Se observa por medio de las pruebas que la distancia medida es muy exacta y esto permite su uso dentro de la app para medir cada viaje realizado a la hora del uso de la app.

## 5.4 PRUEBAS

Para las pruebas del prototipo final se utilizaron dos teléfonos móviles, un Samsung A5 y un LG K41s, ambos llevaban la app y se utilizaron uno primero y luego el otro, esto en un proceso repetitivo.



**Ilustración 60** Teléfonos móviles utilizados para hacer pruebas.

Fuente: (Propia,2021).

En la ilustración anterior se muestran los teléfonos utilizados, ambos se utilizaron en múltiples viajes donde tomaban las fotos deseadas por el usuario en cada proceso. No hubo diferencia al usar la app en diferentes teléfonos móviles, ambos tuvieron el desempeño deseado, lo que cambia entre teléfonos móviles es la calidad de la cámara.

Al momento de realizar pruebas se utilizaron dos teléfonos, un teléfono a la vez el cual iba sostenido con un sostenedor de teléfonos e iba conectado por medio de comunicación bluetooth al Arduino.

Para la primera prueba en ambos teléfonos en la app se estableció una cantidad a tomar de 10 fotografías. El primer teléfono utilizado fue el LG donde el teléfono se colocó de manera horizontal para la hora de las pruebas, una vez conectado el teléfono móvil al Arduino se continuo a hacer la prueba poniendo el vehículo en movimiento, como lo esperado la app no capturaba fotografías mientras el carro estaba en movimiento ya que por medio de la programación de la app se estableció que si el acelerómetro detectaba que el teléfono móvil no estaba estático no se debía capturar fotografías.



**Ilustración 61 Teléfono móvil utilizado primer, colocado de forma horizontal.**

Fuente: (Propia,2021).

Como se observa en la ilustración numero 61 el teléfono móvil se colocó en el tablero del vehículo en manera horizontal. Un problema encontrado al poner el teléfono móvil en forma horizontal es que el teléfono a utilizar debe tener la rotación de pantalla desactivada ya que tener la app en forma horizontal como en la ilustración 62 nos da fotografías como resultado en la ilustración 49.



**Ilustración 62 Fotografía obtenida al tener el teléfono móvil con rotación de pantalla activada.**

Fuente: (Propia,2021).

En la ilustración anterior se muestra las fotografías capturadas en la primera prueba con el teléfono móvil debido a la rotación de la pantalla. Con esto se llegó a la conclusión que en la prueba con el siguiente teléfono móvil se desactivaría la opción de rotación de pantalla.



**Ilustración 63 Fotografía obtenida con rotación de pantalla desactivada.**

Fuente: (Propia,2021).

Como se observa en la ilustración 63 ya al desactivar la opción de rotación de pantalla en el teléfono móvil las fotografías salen de la manera deseada. De igual manera se puede observar en la fotografía los datos de ubicación GPS obtenidas en la app.

-88.02769 15.53394



**Ilustración 64 Fotografía capturada por el otro teléfono móvil durante pruebas.**

Fuente: (Propia,2021).

En la ilustración 64 se puede observar una de las fotografías capturadas por la app en el otro teléfono móvil. La fotografía como se puede observar se ve aceptable con el detalle de que sale horizontal la fotografía en vez de vertical, esto se debe a que el teléfono móvil en la ilustración 52 al momento de esa prueba fue colocado de forma vertical en vez de horizontal, con esto podemos ver que la manera adecuada es posicionar el teléfono móvil de manera horizontal con la rotación de pantalla desactivada.



**Ilustración 65 El segundo teléfono móvil a la hora de pruebas con la app, este colocado en forma vertical.**

Fuente: (Propia,2021).

Las fotografías capturadas se iban almacenando en los teléfonos a la hora de hacer las pruebas, cada foto con un nombre diferente, de acorde al número de fotografía capturada.



**Ilustración 66 Almacenamiento de las fotografías capturadas.**

Fuente: (Propia,2021).

En la ilustración anterior se muestra cómo se almacenaron las fotografías capturadas por el teléfono móvil utilizado en las pruebas, en la ilustración se logra observar que cada imagen tiene su nombre diferente para poder diferenciarlas de una manera más simple.



**Ilustración 67 Almacenaje en tercer incremento.**

Fuente: (Propia,2021).

En la ilustración 67 se muestra como en el tercer incremento la app permite guardar las fotografías capturadas con nombres específicos. En la ilustración se observan pruebas donde se utilizaron los nombres: fotos y Nombre Fotos.

En estas pruebas realizadas en automóvil se descubrió que el acelerómetro es bien sensible al movimiento del teléfono móvil entonces muchas veces al estar estático el automóvil, el acelerómetro de igual manera detectaba movimiento debido a las vibraciones ocasionadas por el motor y esto no permitía que la app capturara fotografías en algunas ocasiones.

El problema anterior con el acelerómetro se solucionó reduciendo la sensibilidad del sensor acelerómetro, con ese cambio el acelerómetro funciono de la manera visualizada de un principio.

En el teléfono móvil LG utilizado en las pruebas se utilizó también la opción de WalkStep del podómetro, y en el teléfono Samsung utilizado se utilizó la opción SimpleStep del mismo sensor. En ambos casos utilizar este sensor con la app para medir distancias recorridas fue ineficiente ya que los sensores están fabricados para medir distancia de personas caminado, no en un vehículo. En ambas opciones a la hora de pruebas se registraron mucho menos metros de lo recorrido.

Este problema de la distancia se solucionó gracias a una fórmula matemática llamada Haversine, la cual permite calcular distancia por medio de dos puntos en el GPS donde utilizando ambas longitudes y latitudes se obtiene una distancia.

#### 5.4.1 PERMISOS DE APLICACIÓN

Es de suma importancia brindar permisos a la app en el teléfono móvil, previamente a iniciar pruebas la app en teléfonos móvil Samsung no estaba capturando fotografías, con eso se descubrió que al entrar en la configuración del teléfono móvil donde están los permisos de aplicación la app no tenía permiso de cámara. Luego se le concedió permiso a la app y se capturaban fotografías sin ningún problema.

## VI CONCLUSIONES

1. Actualmente entre los problemas más comunes en cámaras colocadas en robots, es mala imagen debido a utilizar cámaras de bajo costo, lo cual con la app se soluciona el problema al utilizar una buena cámara con teléfonos móvil y a la vez permite la utilización de sensores extra como el acelerómetro y GPS.
2. Se añadió el parámetro de rango de acelerómetro a la app para evitar que la app tomara fotografías estando en movimiento, garantizando de esa manera que las fotos obtenidas sean enfocadas y adecuadas.
3. Al no haber sensores dedicados para medición de distancias en MIT APP Inventor se investigó e implemento la fórmula matemática de Haversine con la cual se utilizan dos puntos en el mapa para calcular distancia. La app por medio de la función GPS añadida obtenía latitud y longitud en ambos puntos y de esa manera se utilizó la formula.

## VII RECOMENDACIONES

1. Se recomienda utilizar la app para proyectos debido a que muchas personas pueden tener una forma de capturar fotografías al alcance de sus manos, al utilizar la app se evita tener que comprar una cámara y ciertos sensores.
2. Se recomienda cambiar el rango del acelerómetro en la app si el teléfono no está capturando fotografías a la hora de estar estático, esto se debe a que no todos los teléfonos tienen el mismo sensor y esto crea variaciones.
3. Se recomienda utilizar la fórmula de Haversine para calcular distancias en proyectos donde no se desee comprar algún sensor de mediciones de distancias específico, esto gracias a que la fórmula nos permite obtener resultados adecuados.

## BIBLIOGRAFÍA

- Admin. (2019). PIC Microcontroller: Architecture and Its Applications. *WatElectronics.Com*.  
<https://www.watelectronics.com/pic-microcontroller-architecture-and-applications/>
- Alvarado, G., Reyes, A., & Ordoñez, M. (2017). Desing of system for electrical faults diagnostic in multibranch motorcycle, using free hadware and android APP. *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 27–31.  
<https://doi.org/10.1109/ICEECCOT.2017.8284639>
- Alvarado, Gino, Reyes, A., & Ordonez, M. (2017). Desing of system for electrical faults diagnostic in multibranch motorcycle, using free hadware and android APP. *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 27–31. <https://doi.org/10.1109/ICEECCOT.2017.8284639>
- Amestica, O. E., Melin, P. E., Duran-Faundez, C. R., & Lagos, G. R. (2019). An Experimental Comparison of Arduino IDE Compatible Platforms for Digital Control and Data Acquisition Applications. *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 1–6.  
<https://doi.org/10.1109/CHILECON47746.2019.8986865>
- Assaad, B., El kadri Benkara, K., Friedrich, G., & Michon, A. (2016). An advanced simulation tool for considering the electric machines thermal performance in mechatronic systems. *2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM)*, 197–202.  
<https://doi.org/10.1109/MECATRONICS.2016.7547141>
- Baquero-Larriva, A., & Alvarez-Coello, G. (2018). Low Cost Data Adquisition System to Register Fuel Consumption in Diesel Engine Vehicles. *2018 Congreso Internacional de Innovación y Tendencias En Ingeniería (CONIITI)*, 1–4. <https://doi.org/10.1109/CONIITI.2018.8587057>
- Barrett, S. F. (2020). Arduino II: Systems. *Synthesis Lectures on Digital Circuits and Systems*, 15(2), 1–291. <https://doi.org/10.2200/S01024ED1V01Y202006DCS059>

Ben Lutkevich. (2019). *What is a Microcontroller and How Does it Work?* IoT Agenda. <https://internetofthingsagenda.techtarget.com/definition/microcontroller>

Burhan, I., Azman, A. A., & Othman, R. (2016). Electro pneumatic trainer embedded with programmable integrated circuit (PIC) microcontroller and graphical user interface platform for aviation industries training purposes. *IOP Conference Series: Materials Science and Engineering*, 152, 012065. <https://doi.org/10.1088/1757-899X/152/1/012065>

Chen, A. (2016). A review of emerging non-volatile memory (NVM) technologies and applications. *Solid-State Electronics*, 125, 25–38. <https://doi.org/10.1016/j.sse.2016.07.006>

Chen, Y., Shen, H., Chen, H., & Hsu, C. (2016). Low Cost Arduino DAQ Development and Implementation on an Android App for Power Frequency Measurement. *2016 International Symposium on Computer, Consumer and Control (IS3C)*, 99–102. <https://doi.org/10.1109/IS3C.2016.36>

Cho, Y., Choi, J., Choi, J., & Ryoo, Y.-J. (2017). Robot Software Platform for IoT-Based Context-Awareness. *International Journal of Humanoid Robotics*, 14(02), 1750012. <https://doi.org/10.1142/S0219843617500128>

Choden, Y., Raj, M., Wangchuk, C., Singye, P., & Muramatsu, K. (2019). Remote Controlled Rescue Robot Using ZigBee Communication. *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 1–5. <https://doi.org/10.1109/I2CT45611.2019.9033924>

Chtourou, S., Kharat, M., Amor, N. B., Jallouli, M., & Abid, M. (2016). Development of an Android Service to add IOIO Hardware Features to Android Apps. *2016 IEEE International Conference on Computer and Information Technology (CIT)*, 100–103. <https://doi.org/10.1109/CIT.2016.13>

Fan, L., Su, T., Chen, S., Meng, G., Liu, Y., Xu, L., Pu, G., & Su, Z. (2018). Large-Scale Analysis of Framework-Specific Exceptions in Android Apps. *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, 408–419. <https://doi.org/10.1145/3180155.3180222>

Fan, Z., Qiu, Q., & Meng, Z. (2017). Implementation of a four-wheel drive agricultural mobile robot for crop/soil information collection on the open field. *2017 32nd Youth Academic Annual*

*Conference of Chinese Association of Automation (YAC)*, 408–412.  
<https://doi.org/10.1109/YAC.2017.7967443>

Folgheraiter, M., & Aubakir, B. (2018). Design and Modeling of a Lightweight and Low Power Consumption Full-Scale Biped Robot. *International Journal of Humanoid Robotics*, 15(05), 1850022. <https://doi.org/10.1142/S0219843618500226>

Ganguly, S., Sadhukhan, A., Gayen, P. K., & Dolui, S. (2017). Hardware implementation of single-phase full-bridge VSI switched by PSO based SHEPWM signals using embedded PIC microcontroller. *2017 IEEE Calcutta Conference (CALCON)*, 341–346.  
<https://doi.org/10.1109/CALCON.2017.8280752>

Ganney, P. S., Pisharody, S., & Claridge, E. (2020). Chapter 9—Software engineering. In A. Taktak, P. S. Ganney, D. Long, & R. G. Axell (Eds.), *Clinical Engineering (Second Edition)* (pp. 131–168). Academic Press. <https://doi.org/10.1016/B978-0-08-102694-6.00009-7>

Gao, C., & Yao, K. (2015). The Design and Implementation of Portable Agricultural Microclimate Data Acquisition System Based on Android Platform. *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, 1, 210–213.  
<https://doi.org/10.1109/ISCID.2015.275>

Haizad, M., Ibrahim, R., Adnan, A., Chung, T. D., & Hassan, S. M. (2016). Development of low-cost real-time data acquisition system for process automation and control. *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, 1–5.  
<https://doi.org/10.1109/ROMA.2016.7847826>

Hajjaj, S. S. H., & Sahari, K. S. M. (2016). Review of agriculture robotics: Practicality and feasibility. *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 194–198.  
<https://doi.org/10.1109/IRIS.2016.8066090>

Hu, J., Xu, J., & Xie, L. (2013). Cooperative Search and Exploration in Robotic Networks. *Unmanned Systems*, 01(01), 121–142. <https://doi.org/10.1142/S2301385013500064>

Islam, B., Islam, M. T., & Nirjon, S. (2018). Demo Abstract: A Motion-Triggered Stereo Camera for 3D Experience Capture. *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 134–135. <https://doi.org/10.1109/IPSN.2018.00030>

Jayasree, K. R., Jayasree, P. R., & Vivek, A. (2017). Dynamic target tracking using a four wheeled mobile robot with optimal path planning technique. *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, 1–6. <https://doi.org/10.1109/ICCPCT.2017.8074365>

Jha, A. K., Kim, D. Y., & Lee, W. J. (2019). A Framework for Testing Android Apps by Reusing Test Cases. *2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 20–24. <https://doi.org/10.1109/MOBILESoft.2019.00012>

Khan, W., Ahmad, W., Luo, B., & Ahmed, E. (2019). SQL Database with physical database tuning technique and NoSQL graph database comparisons. *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 110–116. <https://doi.org/10.1109/ITNEC.2019.8729264>

Kobayashi, M., Yamazaki, K., Caballero, F. G., Osato, T., Endo, T., Takemura, M., Nagasaki, T., & Shima, T. (2020). Wide Angle Multi-Shift Stereo Camera with Monocular Vision. *2020 IEEE International Conference on Consumer Electronics (ICCE)*, 1–4. <https://doi.org/10.1109/ICCE46568.2020.9043079>

Kritter, J., Brévilliers, M., Lepagnot, J., & Idoumghar, L. (2019). On the real-world applicability of state-of-the-art algorithms for the optimal camera placement problem. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1103–1108. <https://doi.org/10.1109/CoDIT.2019.8820295>

Le, H., Marshall, C., Doan, T., Mai, L., & Liu, F. (2017). Visual Quality Assessment for Projected Content. *2017 14th Conference on Computer and Robot Vision (CRV)*, 225–231. <https://doi.org/10.1109/CRV.2017.47>

Li, M., & Chung, S.-O. (2015). Special issue on precision agriculture. *Computers and Electronics in Agriculture*, 112, 1. <https://doi.org/10.1016/j.compag.2015.03.014>

Mahmoud, R., Ueno, A., & Tatsumi, S. (2012). *AN INTERACTIVE TELE-OPERATED ANTHROPOMORPHIC ROBOT HAND: OSAKA CITY UNIVERSITY HAND*. 16.

Michigan Technological University. (2021). What Is Mechatronics?  
<https://www.mtu.edu/mechatronics/what-is/>

Microchip. (n.d.). *Microcontrollers and Microprocessors | Microchip Technology*. Retrieved January 29, 2021, from <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors>

Moharikar, P., & Guddeti, J. (2017). Automated test generation for post silicon microcontroller validation. *2017 IEEE International High Level Design Validation and Test Workshop (HLDVT)*, 45–52. <https://doi.org/10.1109/HLDVT.2017.8167462>

Monjaras, A., Bcndezu, E., & Raymundo, C. (2019). Decision Tree Model to Support the Successful Selection of a Database Engine for Novice Database Administrators. *2019 8th International Conference on Industrial Technology and Management (ICITM)*, 353–357. <https://doi.org/10.1109/ICITM.2019.8710696>

Naik, N. S., Shete, V. V., & Danve, S. R. (2016). Precision agriculture robot for seeding function. *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2, 1–3. <https://doi.org/10.1109/INVENTIVE.2016.7824880>

Noguchi, N., & Murase, H. (2008). Special issue on bio-robotics. *Computers and Electronics in Agriculture*, 63(1), 1. <https://doi.org/10.1016/j.compag.2008.02.005>

Oracle. (n.d.). *What is a database?* Retrieved January 31, 2021, from <https://www.oracle.com/database/what-is-database/>

Petrovic, D., Kicinbaci, L., Petric, F., & Kovaic, Z. (2019). Autonomous Robots as Actors in Robotics Theatre—Tribute to the Centenary of R.U.R. *2019 European Conference on Mobile Robots (ECMR)*, 1–7. <https://doi.org/10.1109/ECMR.2019.8870908>

Premadi, A., Ab-Rahman, M. S., Chuan, N. B., Saupe, M. N. M., & Jumari, K. (2009). Application of PIC microcontroller for online monitoring and fiber fault identification. *2009 International*

*Conference on Electrical Engineering and Informatics*, 463–467.  
<https://doi.org/10.1109/ICEEI.2009.5254693>

Puangsaikai, W., & Puntheeranurak, S. (2017). A comparative study of relational database and key-value database for big data applications. *2017 International Electrical Engineering Congress (IEECON)*, 1–4. <https://doi.org/10.1109/IEECON.2017.8075813>

Rashid, A., & Hasan, O. (2019). Wearable technologies for hand joints monitoring for rehabilitation: A survey. *Microelectronics Journal*, 88, 173–183. <https://doi.org/10.1016/j.mejo.2018.01.014>

Sadlauer, A. (2019). Mechatronic product platforms: An intralogistics case study. *2019 20th International Conference on Research and Education in Mechatronics (REM)*, 1–5. <https://doi.org/10.1109/REM.2019.8744100>

Sarma, U., & Bordoloi, P. K. (2017). Speed Control of A DC Motor Using PLC and PIC Microcontroller. *2017 14th IEEE India Council International Conference (INDICON)*, 1–6. <https://doi.org/10.1109/INDICON.2017.8488046>

Shah, Y., Shah, J., & Kansara, K. (2018). Code obfuscating a Kotlin-based App with Proguard. *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAIECC)*, 1–5. <https://doi.org/10.1109/ICAIECC.2018.8479507>

Shiroma, N., Sato, N., Chiu, Y., & Matsuno, F. (2004). Study on effective camera images for mobile robot teleoperation. *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*, 107–112. <https://doi.org/10.1109/ROMAN.2004.1374738>

Tsiakaka, O., & Feruglio, S. (2019). Toward the monitoring of the spinal cord: A feasibility study. *Microelectronics Journal*, 88, 145–153. <https://doi.org/10.1016/j.mejo.2018.01.026>

Velasquez-Aguilar, J. G., Aquino-Roblero, F., Limon-Mendoza, M., Cisneros-Villalobos, L., & Zamudio-Lara, A. (2017). Multi-channel Data Acquisition and Wireless Communication FPGA-Based System, to Real-Time Remote Monitoring. *2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, 181–186. <https://doi.org/10.1109/ICMEAE.2017.39>

Walsh, C. J., Endo, K., & Herr, H. (2007). A QUASI-PASSIVE LEG EXOSKELETON FOR LOAD-CARRYING AUGMENTATION. *International Journal of Humanoid Robotics*, 04(03), 487–506. <https://doi.org/10.1142/S0219843607001126>

Westlund, J. K., Lee, J. J., Plummer, L., Faridi, F., Gray, J., Berlin, M., Quintus-Bosz, H., Hartmann, R., Hess, M., Dyer, S., dos Santos, K., Adalgeirsson, S. O., Gordon, G., Spaulding, S., Martinez, M., Das, M., Archie, M., Jeong, S., & Breazeal, C. (2016). Tega: A social robot. *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 561–561. <https://doi.org/10.1109/HRI.2016.7451856>

Yan, R., Kayacan, E., Chen, I.-, & Tiong, L. K. (2017). A novel building post-construction quality assessment robot: Design and prototyping. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6020–6023. <https://doi.org/10.1109/IROS.2017.8206499>

Zamula, A., & Kavun, S. (2017). Complex systems modeling with intelligent control elements. *International Journal of Modeling, Simulation, and Scientific Computing*, 08(01), 1750009. <https://doi.org/10.1142/S179396231750009X>

Zhang, Z., Jiang, Y., Zhang, C., Zhang, C., & Li, X. (2019). The Optimization of Localization and Navigation for Vision-based Robot. *2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, 180–181. <https://doi.org/10.1109/ICTA48799.2019.9012838>

Zhang, Zhonghan, Jiang, Y., Zhang, C., Zhang, C., & Li, X. (2019). The Optimization of Localization and Navigation for Vision-based Robot. *2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, 180–181. <https://doi.org/10.1109/ICTA48799.2019.9012838>