



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN PROTOTIPO INALÁMBRICO PARA LA  
DETECCIÓN DE OBSTÁCULOS Y SEÑALÉTICA DIRIGIDO A  
PERSONAS CON DISCAPACIDAD VISUAL.

AUTOR

Cristian Patricio Guachamin Ayala

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN PROTOTIPO INALÁMBRICO PARA LA DETECCIÓN DE  
OBSTÁCULOS Y SEÑALÉTICA DIRIGIDO A PERSONAS CON DISCAPACIDAD  
VISUAL.

Trabajo de Titulación presentado en conformidad a los requisitos establecidos  
para optar por el título de Ingeniero en Redes y Telecomunicaciones.

Profesor Guía

MSc. Carlos Carrión Betancourt

Autor

Cristian Patricio Guachamin Ayala

Año

2020

## **DECLARACION DEL PROFESOR GUIA**

“Declaro haber dirigido el trabajo, Implementación de un prototipo inalámbrico para la detección de obstáculos y señalética dirigido a personas con discapacidad visual, a través de reuniones periódicas con el estudiante Cristian Patricio Guachamin Ayala, en el semestre 202020, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”



Carlos Carrión Betancourt  
MSc. En Telecomunicaciones y Telemática  
CI: 1103738074

## DECLARACION PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, Implementación de un prototipo inalámbrico para la detección de obstáculos y señalética dirigido a personas con discapacidad visual, de Cristian Patricio Guachamin Ayala, en el semestre 202020, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

A handwritten signature in blue ink, appearing to read 'Ivan', is positioned above a horizontal line. A long, thin stroke extends from the bottom of the signature, crossing the line and continuing downwards.

Iván Ricardo Sánchez Salazar

Magister en Calidad, Seguridad y Ambiente

C.I:18030456142

## **DECLARACION DE AUTORIA DEL ESTUDIANTE**

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”



---

Cristian Patricio Guachamin Ayala

C.I: 1721996369

## **AGRADECIMIENTOS**

Agradezco a mi dios que ha proporcionado salud y vida para poder llevar a cabo la culminación de mi carrera académica, a mi madre Luz Ayala y mi padre Julio Guachamin, por su apoyo constante.

## **DEDICATORIA**

Este proyecto va dedicado a mis padres y mi familia quienes con su apoyo incondicional me permitieron cumplir una meta más en la vida.

## RESUMEN

El presente proyecto previo a la obtención del título de Ingeniero en Redes y telecomunicaciones, se propone en el desarrollo del prototipo inalámbrico para la detección de obstáculos y señalética dirigido a personas con discapacidad visual, el cual se enfoca en brindar una herramienta para la movilidad de personas con discapacidad visual en un perímetro urbano.

Se realizara un análisis de la identificación de la cobertura adecuada detectar obstáculos que presentan las personas con discapacidad visual además se identificara que señaléticas son las más esenciales para su movilidad dentro de un perímetro urbano.

El prototipo cumple con los objetivos principales que son mediante un sensor de proximidad informar al cliente que obstáculo puede presentarse a lo largo del diámetro de cobertura del sensor y mediante una cámara se identificara señales de advertencia o de prevención para que el usuario pueda anticiparse con tiempo reduciendo cualquier evento que puede afectarlo al momento de desplazarse.

La comunicación de los dispositivos de detección será de vía inalámbrica eliminando la necesidad de tener dispositivos cableados lo cual genera incomodidad al usuario al usar el sistema de detección, con un sistema de alimentación para que se cargue la batería de los dispositivos, mediante una aplicación móvil se conectara el sistema para enviar la información al usuario mediante una señal auditiva o señal de vibraciones.

## **ABSTRACT**

The present project prior to obtaining the title of Network and Telecommunications Engineer is proposed in the development of the wireless prototype for the detection of obstacles and signage aimed at people with visual disabilities, which focuses on providing a tool for mobility of visually impaired people in an urban perimeter.

An analysis of the identification of adequate coverage is performed to detect obstacles that people with visual disabilities present, and the most important sound signaling for their mobility within an urban perimeter is identified.

The prototype meets the main objectives that are by means of a proximity sensor informing the client that obstacles can occur along the sensor coverage diameter and through a camera, warning or prevention signals are identified so that the user can anticipate the time reducing any event that may affect it when traveling. The communication of the detection devices will be via wireless eliminating the need to have wired devices which generates discomfort to the user when using the detection system, with a power system to charge the battery of the devices, through a mobile application the system is connected to send the information to the user by means of an audible signal or vibration signal.

# ÍNDICE

Introducción.....	1
Antecedentes .....	1
Alcance del Proyecto.....	2
Justificación .....	2
Objetivo General .....	3
Objetivos Específicos.....	4
Metodología.....	4
<b>1. Capítulo I. Marco Teórico .....</b>	<b>4</b>
1.1 Visión artificial.....	5
1.2 Tiflotecnología .....	6
1.3 Discapacidad visual en el ser humano .....	6
1.4 Sensores.....	7
1.4.1 Características de los sensores. ....	7
1.4.2 Clasificación de sensores.....	8
1.4.3 Sensores de presencia o proximidad .....	8
1.4.4 Sensor ultrasónico.....	9
1.5 Placa arduino.....	10
1.5.1 Modulo bluetooth.....	11
1.5.2 Arduino IDE.....	11
1.5.3 Software Fritzing .....	12
1.6 APP inventor.....	12
1.6.1 Componentes de App Inventor.....	13
1.6.2 Ventana de diseño App Inventor .....	14
1.6.3 Ventana de bloques App Inventor .....	15
1.7 Raspberry Pi .....	18
1.7.1 Cámara Pi .....	18
1.7.2 Entorno de programación Anaconda.....	19
1.7.3 Software Tensorflow.....	20
1.7.4 Librerías pre entrenadas .....	20

1.7.5	Software Liclipse .....	21
1.7.6	Software Pycharm .....	22
1.7.7	Entorno Opencv .....	22
2.	<b>Capítulo II. Variables</b> .....	23
2.1	Variables de Entrada y Salida .....	23
2.2	Variables del Sistema .....	24
2.2.1	Variables de entrada .....	25
2.2.2	Variables de salida .....	27
2.3	Arquitectura detección de obstáculos.....	28
2.4	Arquitectura detección de señalética.....	30
2.5	Tareas a describir .....	31
3.	<b>Capítulo III. Diseño</b> .....	31
3.1	Aspectos esenciales: .....	32
3.2	Casos de Uso .....	33
3.3	Diagrama General .....	34
3.4	Sistemas .....	35
4.	<b>Capítulo IV. Desarrollo</b> .....	36
4.1	Comentarios en el código .....	36
4.2	Detector de obstáculos.....	36
4.2.1	Implementación de la App.....	37
4.2.2	Código de la App.....	38
4.2.3	Código del circuito detector de obstáculos.....	41
4.2.4	Código de conexión <i>bluetooth</i> .....	43
4.3	Detector de señalética.....	43
4.3.1	Código detección de señalética.....	44
4.3.2	Entrenamiento de modelo predictivo de señalética.....	46
4.3.3	Código circuito audio detección de señalética .....	51
4.4	Funcionamiento del prototipo .....	54
5.	<b>Conclusiones y Recomendaciones</b> .....	59
5.1	Conclusiones .....	59
5.2	Recomendaciones.....	59

6. Referencias .....	61
----------------------	----

# Introducción

## Antecedentes

Según la O.M.S. (Organización Mundial de la Salud), el 15 por ciento de la población mundial está afectado por alguna discapacidad física, psíquica o sensorial que dificulta su desarrollo personal y su integración social, educativa o laboral. Tal porcentaje equivale a 900 millones de personas. En Ecuador existen alrededor 435865 personas con discapacidad visual correspondiente al 11.90 % de la poblacional nacional. Las últimas investigaciones indican que el número de personas que padecen ceguera evitable va en aumento. Los costos que suponen la educación, la rehabilitación y la pérdida de productividad por causa de la ceguera tienen importantes repercusiones en las personas, las familias, las comunidades y las naciones. Las personas con ceguera total o con poca visión usualmente tienen problemas para manejarse fuera de entornos conocidos. De hecho, el movimiento físico es uno de los desafíos más grandes para las personas invidentes, explica World Access fortheBlind. Viajar o simplemente caminar por una calle llena de gente puede generar grandes dificultades. Por esta razón, muchas personas con poca visión caminan junto a un amigo o familiar que los ayude a conducirse en entornos desconocidos. De la misma forma, las personas invidentes deben aprender todos los detalles de su hogar. Los obstáculos grandes como mesas y sillas deben permanecer en un lugar para prevenir lesiones. Si una persona invidente vive con otras personas, cada miembro del hogar debe, diligentemente, mantener los pasillos libres de obstáculos y todos los objetos deben permanecer en su lugar. Por lo cual, determinada la tecnología se ha enfocado en contribuir con estas personas para de alguna manera ayudar a que tengan una mejor vida, como por ejemplo desarrollando sistemas de movilidad que facilitan el desplazamiento de las personas y reducen las barreras arquitectónicas. En sentido, surge la idea de seguir aportando a las personas con discapacidad visual, mediante el desarrollo de este proyecto.

## **Alcance del Proyecto**

El presente proyecto, consiste en el desarrollo de un prototipo que permite detectar obstáculos e identificar señalética, para que faciliten la movilidad de las personas con discapacidad visual.

Se realizará un análisis de las diferentes tecnologías existentes para el uso y control de sensores de proximidad al igual que dispositivos que permiten identificar señalética. Mediante envío de señales auditivas se informará al usuario final cual es la distancia hacia el obstáculo o señalética permitiendo reaccionar para limitar accidentes que podría tener la persona con discapacidad visual.

Como siguiente punto se investigaran las tecnologías inalámbricas que permitirán crear la comunicación con el circuito detector de obstáculos y señalética, sin que generen una afectación o sean invasivos con el usuario.

La funcionalidad del prototipo busca detectar obstáculos que cubran el rango de cobertura del sensor y la cámara con la característica que depende de la definición que otorgue la cámara, con la investigación se determinará que dispositivos cumplen con los requerimientos de identificación de señalética y objetos.

Para la implementación se determinará un lugar estratégico para que el usuario lo pueda usar, intentando que no genere incomodidad al usarlo.

Finalmente, se realizarán las pruebas de rigor del prototipo para verificar su funcionamiento.

## **Justificación**

El principio de identificar objetos de manera auditiva se llamada ecolocación, permitiendo detectar el objeto mediante la localización por el eco, la mayoría de

las personas con discapacidad visual usan esta técnica para detectar obstáculos a corta distancia sea aplaudiendo con las palmas de las manos o al golpear sus muslos genera un eco el cual permite detectar que tan cercano se encuentra un objeto.

La movilidad de las personas con discapacidad visual y la falta de una infraestructura que mejore las condiciones de desplazamiento además de señalética que no es adaptada para informar de alguna afectación que pueda presentarse han generado una barrera hacia este grupo de personas con esta discapacidad limitando un desenvolvimiento en sus actividades diarias.

Al implementar este prototipo se busca generar un plus adicional en relación a las herramientas de movilidad que actualmente se tienen como el bastón blanco y bastón plegable, lo cual mejorara sus actividades diarias.

Este dispositivo es para facilitar el desplazamiento de las personas no videntes, en el Ecuador existen programas que ayudan de una manera sustancial al entorno de personas con discapacidad visual sin embargo no cubren al 100 % su necesidad es ahí en donde se pretende 3 agregar un elemento adicional para que puedan desplazarse de una manera más fácil, actualmente en el mercado nacional no se ha diseñado un dispositivo con sensor para facilitar su movilidad.

Este proyecto también es diseñado para evitar accidentes que ocurren diariamente por la ineficiencia de los recursos que tienen las personas con discapacidad visual para desplazarse, este planteamiento es importante en especial para las personas que tengan esta discapacidad como para la sociedad en general.

## **Objetivo General**

Implementar un prototipo detector de obstáculos y señalética para personas con discapacidad visual, mediante el principio de teledetección.

## Objetivos Específicos

- ✓ Analizar las tecnologías relacionadas con proyectos destinados a facilitar la movilidad de personas con discapacidad visual.
- ✓ Diseñar el sistema de detección de obstáculos mediante sensores ultrasónicos.
- ✓ Diseñar el sistema de reconocimiento de señalética.
- ✓ Implementar la solución que integre la detección de obstáculos y reconocimiento de señalética.
- ✓ Realizar las pruebas de funcionamiento del prototipo.

## Metodología

El proyecto es de carácter exploratorio, se centra en buscar un sistema simple, con control digital que detecte obstáculos y señalética, que ayuda a las personas con discapacidad visual a desplazarse en su vida diaria. Más en concreto, detectar obstáculos y señalética por medio de sensores ultrasónicos y una cámara digital, los cuales dan señales por medio de vibraciones y audio, cuando detectan la proximidad de un obstáculo, a una distancia determinada. Esto implica:

- ✓ Encontrar en el mercado un sensor ultrasónico y una cámara que den una calidad de señales suficientemente buenas. Para ello debemos tomar en cuenta las características técnicas que contiene el ultrasónico y la comunicación que debe tener la cámara para transmitir a un receptor la información recopilada e informar al usuario mediante un audio la señalética identificada.
- ✓ El circuito que incorpore el sistema de detección y señalética deber ser inalámbrico incorporando tecnologías que actualmente se usan, siendo tecnologías innovadoras.

## 1. Capítulo I. Marco Teórico

En el presente capítulo se da a conocer las metodologías al igual que las tecnologías que contribuyeron a la ejecución del presente proyecto de titulación, se mostrara como se desarrolló el proyecto con los diseños propuestos y seleccionados además de las pruebas de rigor aplicadas para el cumplimiento de los objetivos trazados.

## 1.1 Visión artificial

En la actualidad los avances tecnológicos se han incrementado de manera exponencial enfocados en el internet de las cosas y la inteligencia artificial es ahí en donde mediante una cámara se pueden procesar millones de datos, identificar objetos, personas, animales.

En la actualidad el procesamiento de imágenes es fundamental para entornos de seguridad de acceso, desde los años 50 con el estallido de las computadores el ser humano ha buscado que las computadores procesen, analicen, diagnostiquen datos, con el fin de comprender el comportamiento humano para facilitar su diario vivir, en la actualidad la visión artificial permite realizar actividades que el ser humano por sí solo no podría como el control de calidad en una fábrica de productos masivos al detectar si alguno registra una falla o no, cosas que el ser humano no podría realizar por el volumen de objetos a detectar.



*Figura 1.* Ejemplo del uso de visión artificial  
Tomado de (Automatización Industrial, 2020)

## 1.2 Tiflotecnología

Se origina del griego *Tiflos* que significa ciego, y se enfoca en el conjunto de los recursos tecnológicos encaminados en mejorar la calidad de vida de las personas que sufren de discapacidad visual.

Los avances tecnológicos fomentan un ambiente más fácil y cómodo para el ser humano sin embargo existen ciertos grupos de personas que no son considerados al momento de crear un producto o servicio como por ejemplo gafas de realidad virtual para jugadores en línea, sin embargo existen personas que no las pueden usar como personas con discapacidad visual, la Tiflotecnología busca incluir a todos en los avances tecnológicos que se dan día a día.



*Figura 2.* Computadora personal para persona con discapacidad visual.

Tomado de (Comuniello, DeSantis, & Recine, 2020).

## 1.3 Discapacidad visual en el ser humano

En la actualidad el número de personas que sufren de discapacidad visual es de aproximadamente de 1300 millones de personas a nivel mundial según datos proporcionados por la Organización mundial de la salud.

Según la organización mundial de la salud, clasifica en dos grupos las

afectaciones por la discapacidad visual:

Deficiencia de visión de lejos:

- ✓ Leve: agudeza visual inferior a 6/12.
- ✓ Moderada: agudeza visual inferior a 6/12.
- ✓ Grave: agudeza visual inferior a 6/60.
- ✓ Ceguera: agudeza visual inferior a 3/60.

Deficiencia visual de cerca:

Agudeza visual de cerca inferior a N6 o N8 a 40 cm con la corrección existente.

Este tipo de clasificación no solo depende de factores físicos sino también de factores en relación a tratamientos que permitan reducir el impacto de afectación u objetos que limiten la discapacidad con rehabilitación como el bastón blanco, bastón plegable. Tomado de (salud, 2018)

## **1.4 Sensores**

Son dispositivos creados para detección de acciones o estímulos de manera externa permitiendo generar una reacción o respuesta dependiendo de la configuración o programación establecida previamente.

Los sensores permiten realizar un seguimiento de un fenómeno en específico los cuales generalmente son usados en la industria, sin embargo también se los puede usar para tareas comunes para registrar la velocidad de un automóvil como lo es el velocímetro, además sensores de temperatura para un refrigerador por ejemplo permitiendo validar a que temperatura se debe refrigerar los alimentos. Tomado de (Barbera, 2018)

### **1.4.1 Características de los sensores.**

Permiten definir el comportamiento de los sensores, en relación a la señal que se desea procesar o analizar.

- ✓ Rango
- ✓ Exactitud
- ✓ Reproducibilidad
- ✓ Resolución
- ✓ Error
- ✓ Sensibilidad
- ✓ Estabilidad

(Pallás Areny, 2005)

#### **1.4.2 Clasificación de sensores**

La aplicación que se pretende aplicar define cual es el tipo de sensor más adecuado para detectar, registrar y analizar la señal que se desea procesar, la alimentación que recibe y los datos con los que va a interactuar, estableciendo a continuación los más importantes:

- ✓ Sensores de temperatura
- ✓ Sensores de humedad
- ✓ Sensores de presión
- ✓ Sensores de posición
- ✓ Sensores ópticos
- ✓ Sensores de movimiento
- ✓ Sensores de presencia o proximidad.

(Barbera, 2018)

#### **1.4.3 Sensores de presencia o proximidad**

Este tipo de sensores permiten detectar un objeto físico dependiendo del rango de sensibilidad del sensor al igual que su posición, al enviar una señal de salida se espera la respuesta de la señal con el principio de un eco el cual permite identificar que tan cercano se encuentra el objeto a detectar sin necesidad que el sensor interactúe físicamente con el objeto. (Barbera, 2018)



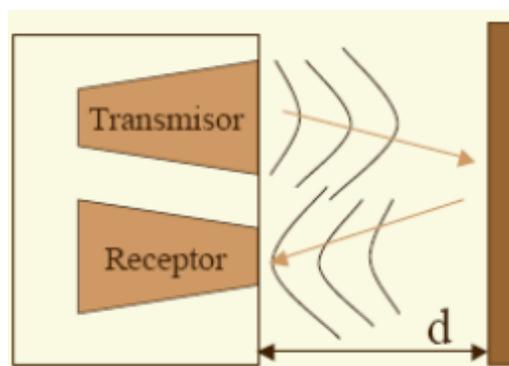
*Figura 3 Sensor de proximidad*

Tomado de (NaylampMechatronic, 2020)

#### **1.4.4 Sensor ultrasónico**

El sensor ultrasónico lo que hace es enviar una señal de sonido que trabaja en una frecuencia de 40 KHz siendo imposible para el ser humano percibir estas señales a través de su oído, el sensor puede trabajar en lugares en los cuales no es necesario que exista luz, puede trabajar en áreas oscuras ya que no perjudica su funcionamiento.

El funcionamiento de estos sensores se basa en medidores de distancia, en donde el emisor envía una señal que rebota sobre el objeto a detectar permitiendo recibir una señal de respuesta la cual permite determinar qué tan alejado está el objeto. (Pepperl, 2003)



*Figura 4 Principio funcionamiento sensor ultrasónico*

Tomado de (Pérez, 2020)

Los sensores industriales proporcionan una fiabilidad y presión del objeto a detectar exacta sin embargo para aplicaciones que no necesitan tanta precisión se puede optar por sensores que tienen características básicas, como lo es el sensor ultrasónico hc-sr04, proporcionando una detección que ronda los 2 cm hasta los 4,5 m, enviando pulsos en alta frecuencia formado por 2 cilindros los que se encargan de emitir la señal ultrasónica.



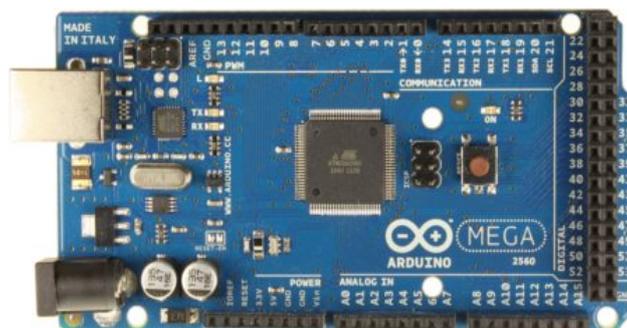
*Figura 5 Sensor ultrasónico hc-sr04 arduino*

Tomado de (Electroniclab, 2020)

## 1.5 Placa arduino

La placa arduino es microcontrolador la cual posee entradas y salidas tanto analógicas como digitales, esta placa permite trabajar en un entorno abierto basado en Java.

Este dispositivo permite crear proyectos enfocados en la electrónica de una manera fácil y sencilla debido a que maneja un código abierto permite crear aplicaciones con variedad de dispositivos electrónicos y digitales.



*Figura 6 Placa arduino Mega*

Tomado de (González, 2020)

### 1.5.1 Modulo bluetooth

Módulo de conexión vía *bluetooth* el cual viene configurado como esclavo, permite la vinculación con dispositivos arduino, se puede variar el código de vinculación la velocidad nombre, tiempo de retardo etc.



*Figura 7 Modulo bluetooth arduino*

Tomado de (NaylampMechatronic, 2020)

### 1.5.2 Arduino IDE

El *Software* arduino, programa desarrollado para cargar código establecido en lenguaje java compatible con arduino para establecer características y directrices de funcionamiento en dispositivos arduino.



*Figura 8 Logo Arduino*

Tomado de (Arduino, 2020)

### 1.5.3 Software Fritzing

Este software permite simular circuitos con componentes de arduino, es muy útil para identificar cuáles son los componentes compatibles con cada una de las placas arduino y sus módulos respectivamente.



*Figura 9 Pantalla de simulación Fritzing*

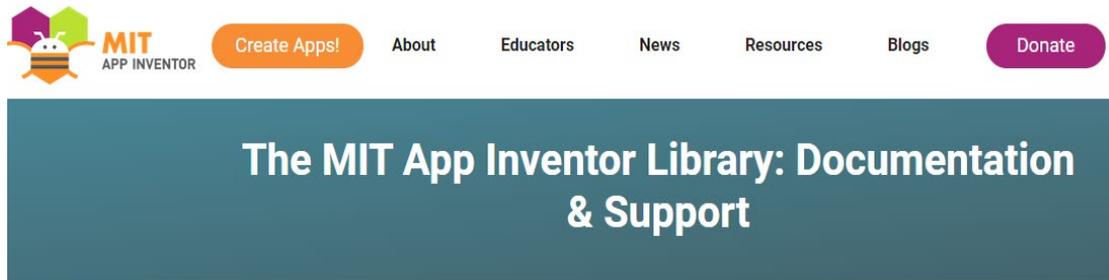
Tomado de (Foundation, s.f.)

## 1.6 APP inventor

Es un entorno de desarrollo creado por Google, el cual permite crear aplicaciones para dispositivos con sistema operativo Android, posee un conjunto de herramientas que se enlazan mediante bloques para generar la aplicación deseada la cual es gratis y accesible desde cualquier navegador web sin necesidad de instalar el software en la máquina físicamente.

Su simplicidad al usarla ha generado un incremento en el desarrollo de aplicaciones, su apertura al público se realizó en el año 2008, permitiendo vincular a personas que no necesariamente poseen un grado alto en programación.

Se basa en JavaScript y sus librerías son distribuidas por el MIT con una licencia libre, mediante el uso de bloques permite programar la interfaz lógica del programa, sin necesidad de interactuar con el código y es compatible con cualquier dispositivo con sistema Operativo Android.



## Getting Started with App Inventor

[Setup computer and phone/tablet or emulator for live testing as you build](#)

[Preview the IDE](#)

[Watch a beginner video tutorial to get started](#)

[Take the next step with another tutorial](#)

[Compatibility Issues with Releases of Android and App Inventor](#)

*Figura 10 Página Web principal App Inventor*

Tomado de (Technology, 2020)

### 1.6.1 Componentes de App Inventor

La aplicación app inventor permite acceder desde cualquier navegador web que esté conectado a internet, el acceso a la aplicación no requiere de un pago o suscripción es totalmente gratis al igual que el acceso a sus librerías y videos tutoriales o programas en el repositorio, en entorno app inventor está compuesto por lo siguiente:

Diseño.- es el lugar en el cual se desarrolla la parte visual de nuestra aplicación móvil, donde diseñamos la presentación de la app, la cual contiene 5 secciones.

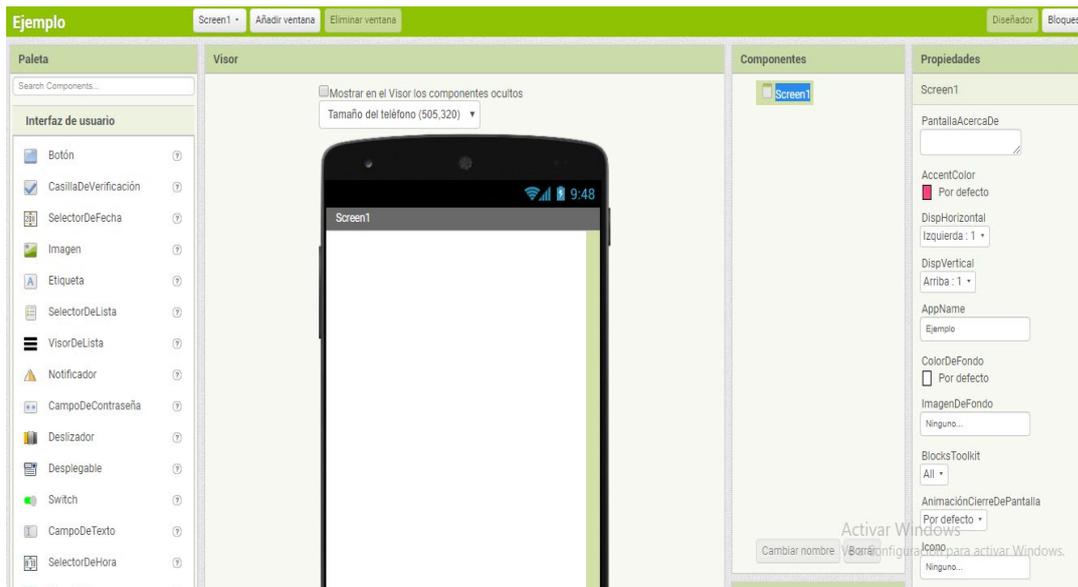


Figura 11 Ventana de diseño App Inventor

Tomado de (Technology, 2020)

Bloques.- es la sección en la cual se procede con la programación basada en bloques en la cual se unen las interfaces que darán funcionalidad a la app.

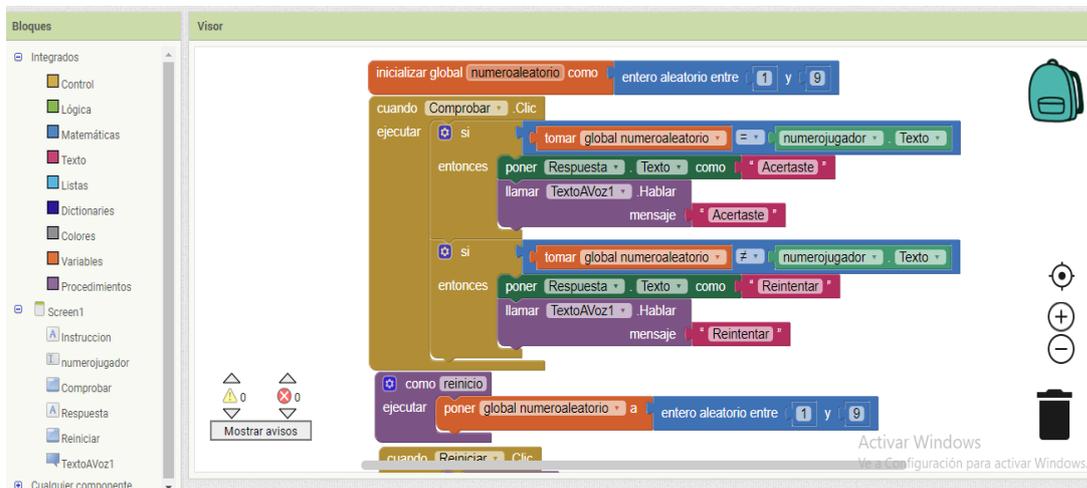


Figura 12 Ventana de bloques App Inventor

Tomado de (Technology, 2020)

## 1.6.2 Ventana de diseño App Inventor

En la venta diseño es en donde se va a proceder a definir la interfaz gráfica o

de presentación de la aplicación web, en esta ventana se encuentran 5 secciones:

- ✓ Paleta.- en esta sección se tienen la interfaz del usuario, los componentes básicos de diseño como los son botones, campo de texto, campo de contraseña, selección de lista, imagen, visor de lista etc.
- ✓ Visor.- en la sección visor se tiene la pantalla de visualización de la app, en donde se incluyen los componentes y se puede dar forma a la interfaz gráfica de la app.
- ✓ Componentes.- aquí se valida cuáles son los componentes seleccionados y que ya fueron agregados a nuestra interfaz gráfica de la app.
- ✓ Propiedades.- en esta sección se da las propiedades de cada uno de los componentes como la posición, color de fondo, tamaño de fuente, color de fuente.
- ✓ Medios.- si es necesario agregar una imagen o algún adjunto que sea necesario para el diseño de la interfaz gráfica lo realizaremos desde esta sección.



*Figura 13 Secciones de la ventana diseño App Inventor*

Tomado de (Technology, 2020)

### **1.6.3 Ventana de bloques App Inventor**

En la ventana de bloques se tiene las herramientas necesarias para aplicar la funcionalidad de la aplicación es en este lugar en el cual se programa la app con la ayuda de bloques los cuales permiten generar la parte lógica de programación la cual es transparente para el usuario.

Esta ventana está compuesta de las secciones bloques y visor las cuales en este entorno permiten establecer la programación de la funcionalidad de la aplicación:

Bloques.- en bloques se tiene los bloques de uso general para programar como control, lógica, matemáticas, texto, listas, variables.

Los bloques control definen cual es la acción que se debe realizar en relación a sucesos que se presentan como si se tratase de una estructura de pregunta.



*Figura 14 Bloques de control App Inventor*

Tomado de (Technology, 2020)

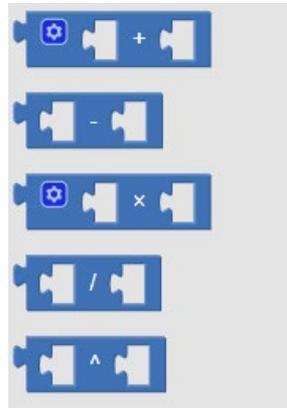
Los bloques lógica entrega la respuesta en relación al bloque de control.



*Figura 15 Bloques de lógica App Inventor*

Tomado de (Technology, 2020)

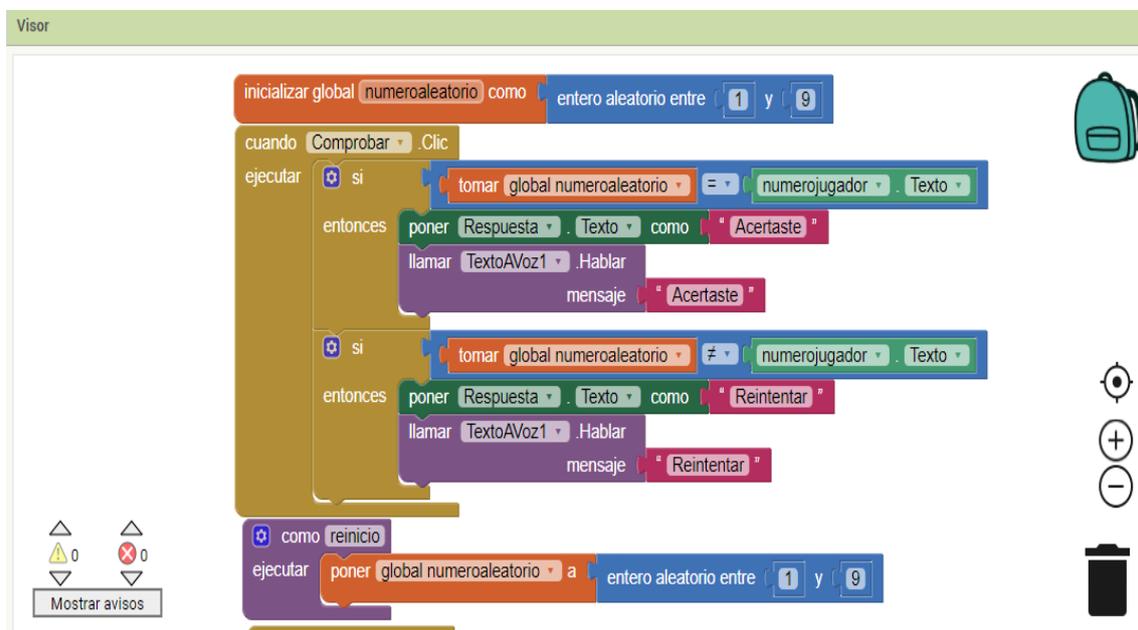
Los bloques que corresponden a matemáticas se relacionan con fórmulas matemáticas básicas y avanzadas.



*Figura 16 Bloques de matemática App Inventor*

Tomado de (Technology, 2020)

Visor.- el visor es la parte en la cual se agregaran los bloques conforme se va generando la funcionalidad de la aplicación.



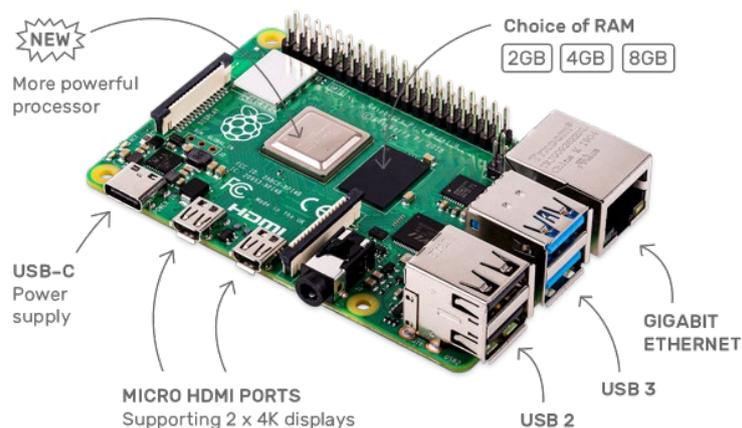
*Figura 17 Sección visor ventana bloques App Inventor*

Tomado de (Technology, 2020)

## 1.7 Raspberry Pi

Es una placa ordenador de reducido tamaño de código abierto, es capaz de soportar la conexión de componentes como teclado, mouse, monitor, tarjeta de red, mediante sus periféricos, el cual tiene la opción de interactuar con sistemas operativos abiertos entre los más conocidos como lo *Raspbian*, *Noobs* y *Ubuntu*.

Su tamaño permite que sea facil de manipular y la opcion a integrar sistemas operativos de codigo abierto permite generar un entorno de desarrollo para programacion y proyectos avanzados como lo son Inteligencia Artificial, Vision Artificial, Internet de las cosas.



*Figura 18.* Placa Raspberry pi 4

Tomado de (FOUNDATION, 2020)

El entorno de Raspbian el sistema operativo oficial de Raspberry es orientado en su gran mayoría a programación vía terminal mediante comandos y con un bajo grado de programación mediante interfaz gráfica.

### 1.7.1 Cámara Pi

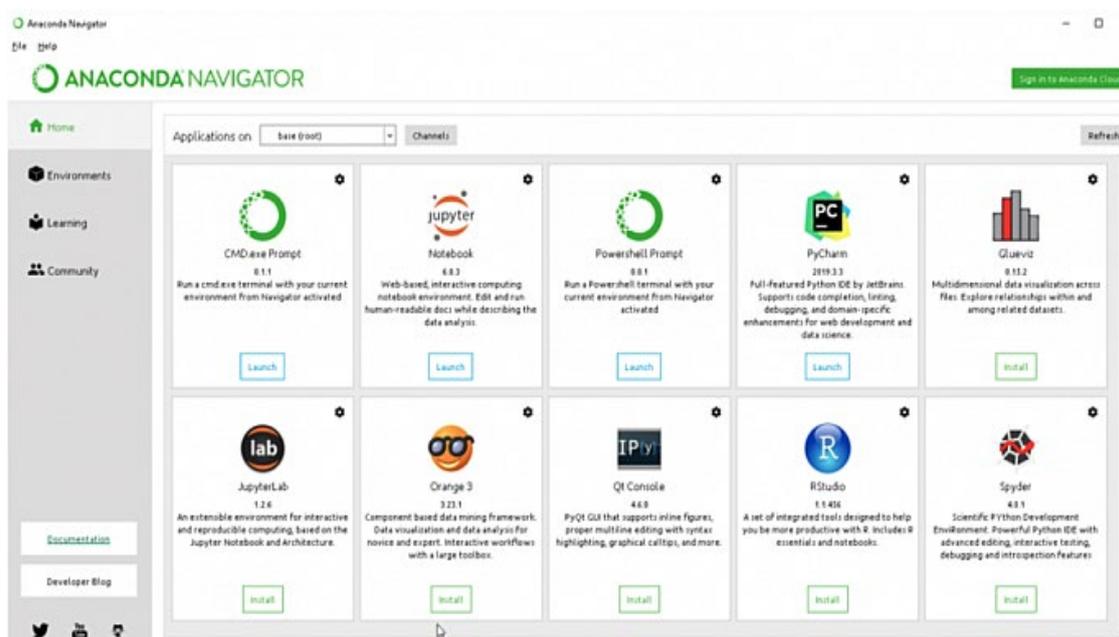
Es una cámara diseñada específicamente para Raspberry pi, la cual es configurada desde la interfaz de Raspberry, su funcionamiento es el mismo de cualquier cámara web con una resolución de 5 megapíxeles.



*Figura 19. Cámara Pi Raspberry*  
Tomado de (FOUNDATION, 2020)

### 1.7.2 Entorno de programación Anaconda

Es un entorno de programación basado en Python, orientado a Machine Learning en el cual se pueden manejar altos volúmenes de información, predicción de datos, compatibles para sistemas operativos como Windows, Linux y MacOs.



*Figura 20. Entorno de programación Anaconda*  
Tomado de (Inc., 2020)

### 1.7.3 Software Tensorflow

Se define como una biblioteca virtual de código abierto de computación utilizando un flujo de datos para construir y establecer redes neuronales las cuales permiten detectar, correlacionar, censar el razonamiento y aprendizaje que usan los seres humanos.



*Figura 21.* Logo de Tensorflow

Tomado de (Tensorflow, 2020)

Este tipo de herramienta fue lanzada por google y está enfocado para desarrollar aplicaciones de inteligencia artificial, permite relacionar letras, números, imágenes con fines de predicción estadística.

### 1.7.4 Librerías pre entrenadas

En línea se dispone de material diseñado para visión artificial los cuales funcionan para acoplarlos con códigos propios orientados a diseños de aplicaciones orientadas a detección de imágenes.

# TensorFlow Model Garden

## Bienvenido al jardín modelo para TensorFlow

TensorFlow Model Garden es un repositorio con varias implementaciones diferentes de modelos y soluciones de modelado de última generación (SOTA) para usuarios de TensorFlow. Nuestro objetivo es demostrar las mejores prácticas para modelar para que los usuarios de TensorFlow puedan aprovechar al máximo TensorFlow para su investigación y desarrollo de productos.

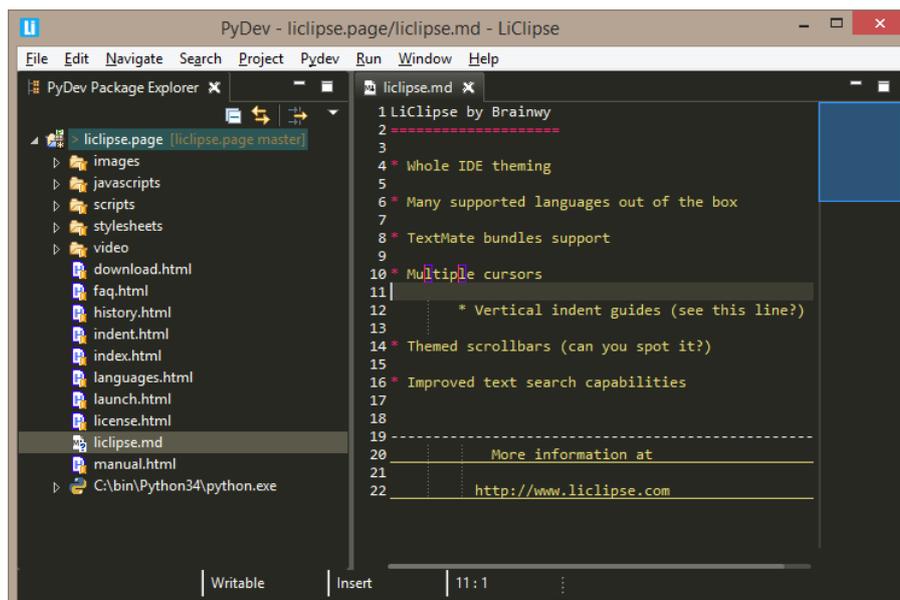
Directorio	Descripción
<a href="#">oficial</a>	<ul style="list-style-type: none"> <li>• Una colección de implementaciones de ejemplo para modelos SOTA que utilizan las últimas API de alto nivel de TensorFlow 2</li> <li>• Mantenedas, respaldadas y actualizadas oficialmente con las últimas API de TensorFlow 2 de TensorFlow</li> <li>• Razonablemente optimizado para un rendimiento rápido sin dejar de ser fácil de leer</li> </ul>

Figura 22. Librerías tensorflow repositorio modelos

Tomado de (GitHub, 2016)

### 1.7.5 Software Liclipse

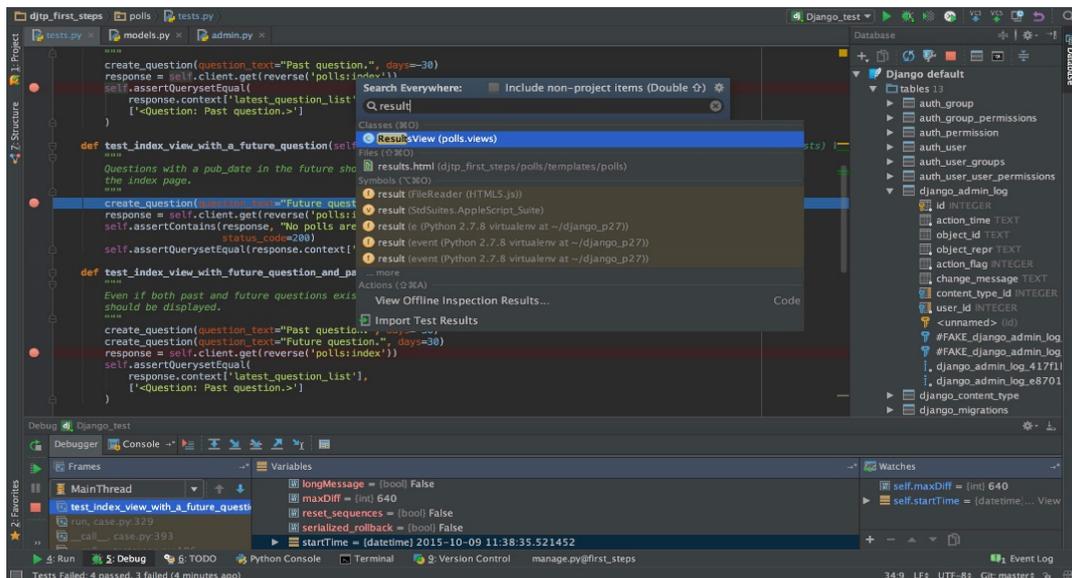
Entorno de desarrollo virtual para desarrollar aplicaciones basadas en Python, mediante este entrono se pueden crear aplicaciones que combinan tanto Openvc, Numpy y Python ya que permite importar todas las librerías relacionadas.



*Figura 23.* Pantalla principal programa Lidlipse.  
Tomado de (Ltda., 2020)

### 1.7.6 Software Pycharm

Entorno de desarrollo para Python, el software permite generar aplicaciones en un entorno integrado el cual asocia varias librerías para su vinculación de manera adecuada con Python.



*Figura 24.* Pantalla principal programa Pycharm.  
Tomado de (s.r.o., 2020)

### 1.7.7 Entorno Opencv

Librería de software libre orientado a Machine Learning y visión artificial, esta librería dispone de aproximadamente 2000 algoritmos, con el fin de identificar, analizar, generar estadísticas, de rostros, objetos, movimientos de acciones humanas.



*Figura 25.* Logo de Opencv-Python

Tomado de (team, 2020)

## **2. Capítulo II. Variables**

En el presente capítulo se detallaran cuales se consideran como principales salidas y entradas para la implementación de prototipo definiendo cuales son las principales señales que recibirá el sistema y cuál puede ser la respuesta más adecuada para que el usuario presente más facilidad al desplazarse.

Realizando un análisis del entorno en cual se desenvuelve cotidianamente una persona con discapacidad visual y que beneficios se obtendrán con el prototipo a implementar.

### **2.1 Variables de Entrada y Salida**

Las variables que se analizaran son las que el usuario necesita detectar para que no presente accidentes o que le proporcionen ayuda para poder movilizarse, por lo cual se han definidos 2 grupos:

Obstáculos: definiendo a obstáculo como todo cuerpo físico de cualquier material que impide avanzar o pasar desde un lugar a otro.



*Figura 26.* Imagen de un obstáculo para una persona con discapacidad visual  
Tomado de (Federico Domenech S.A., 2013)

Señaléticas: se entiende por señalética a una información o comunicación visual de símbolos que permiten que el usuario decida en relación a la información que esta proporciona, advirtiéndolo de un peligro o guiándole para llegar a su destino.

## **2.2 Variables del Sistema**

El lugar en el que se pretende implementar el prototipo va encaminada hacia un perímetro urbano en el que generalmente el usuario se desenvuelve, este prototipo es una herramienta más en los diferentes mecanismos de ayuda que poseen las personas con discapacidad visual, como lo son el bastón blanco y el bastón plegable.

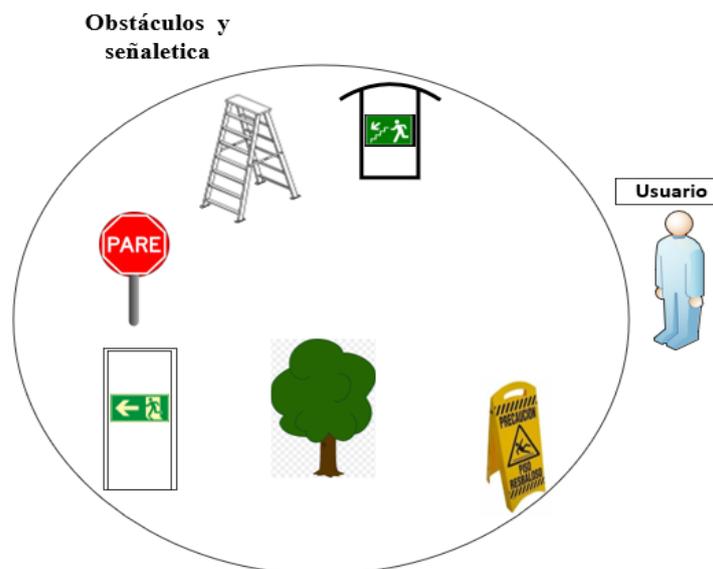


Figura 27. Diagrama de macro proceso prototipo detección de obstáculos y señalética.

Tomado (Guachamin, 2020)

### 2.2.1 Variables de entrada

La variable de entrada es todo lo que puede detectar nuestro sistema para el caso se divide en dos secciones obstáculos y señalética.



*Figura 28.* Diagrama de variables de entrada  
Tomado (Guachamin, 2020)

VARIABLES DE ENTRADA OBSTÁCULOS. - son todos los obstáculos que el usuario no puede detectar sea con sus sentidos o con la ayuda del bastón blanco, es decir superficies por arriba de la detección de bastón.



*Figura 29.* Imagen de un obstáculo para una persona con discapacidad visual  
Tomado de (Agencia EFE, 2013)

VARIABLES DE ENTRADA SEÑALÉTICA. - las señales de advertencia en relación a peligro, advertencia o de ayuda para personas como por ejemplo un disco pare, gradas, paso elevado, salida de emergencia, las opciones que estén expuestas para ayudar al desplazamiento de personas.



*Figura 30.* Imagen de una señalética de advertencia.  
Tomado de (L., 2020)



*Figura 31.* Imagen de una señalética de servicios baños.  
Tomado de (L., 2020)



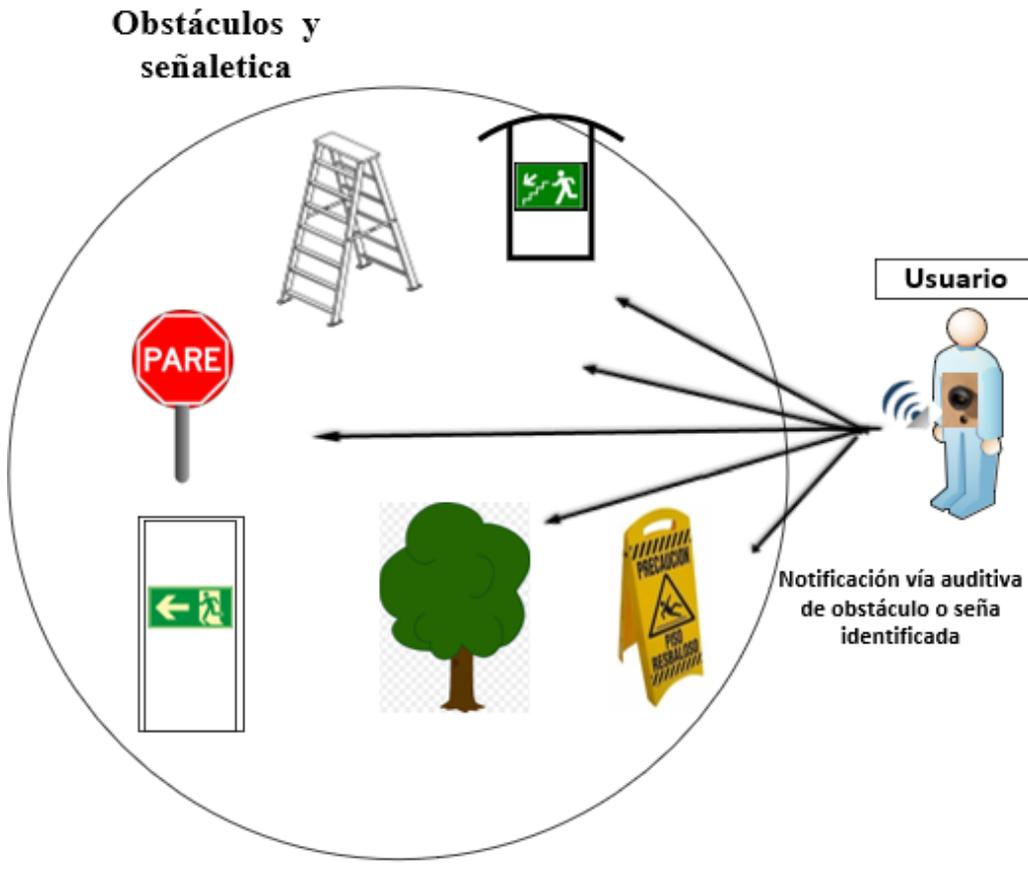
*Figura 32.* Imagen de una señalética de servicios baños.  
Tomado de (L., 2020)

### **2.2.2 Variables de salida**

Las variables de salida es lo que nuestro sistema proporciona al usuario para que evite el obstáculo o tenga precaución en caso de ser una señal de advertencia o informativa.

Al tratarse de una persona con discapacidad visual se ha optado por enviar la información correspondiente vía audio mediante la app móvil en el caso del

detector de obstáculos y en el caso de la señalética mediante una bocina la cual identifica la señal e informa al usuario cual se pudo reconocer.



*Figura 33.* Diagrama de variables de salida de prototipo.

Tomado de (Guachamin, 2020)

### **2.3 Arquitectura detección de obstáculos.**

Para el detector de obstáculos se optó por realizarlo en una placa arduino en la cual se implementará el sensor ultrasónico para detectar obstáculos y el modulo bluetooth para que se envíe la información hacia nuestra app móvil y desde ahí se envíe la información al usuario.

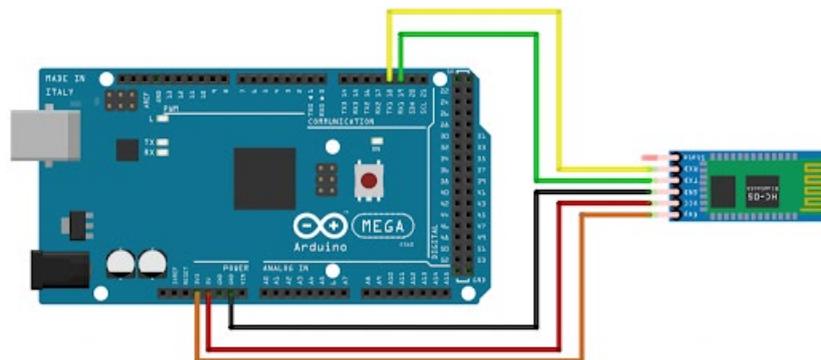


*Figura 34.* Placa arduino mega seleccionada para montar el circuito de detección.

Tomado de (Guachamin, 2020)

Se eligió la placa arduino por sus referencias en proyectos implementados en clases previas ya que se posee conocimiento de las aplicaciones que se pueden realizar al igual que también sus limitantes.

El modulo bluetooth permite la vinculación del dispositivo móvil con la placa arduino para que la comunicación sea inalámbrica y se reduzca el uso de cables.



*Figura 35.* Modulo bluetooth conexión arduino mega.

Tomado de (Guachamin, 2020)

El sensor ultrasónico permite detectar los obstáculos que se presentan a lo largo de su línea de vista con un rango de cobertura desde los 5mm hasta 2 metros, la precisión depende del obstáculo.

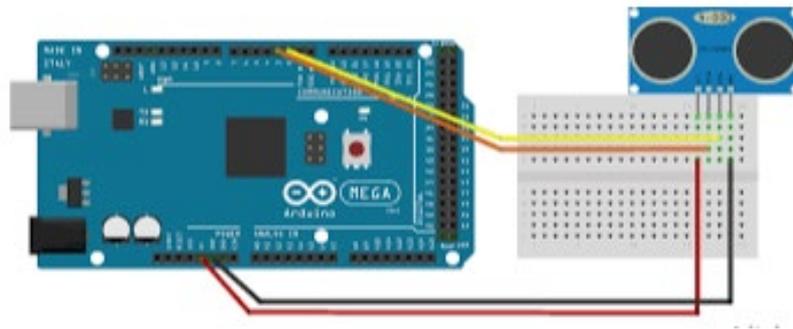


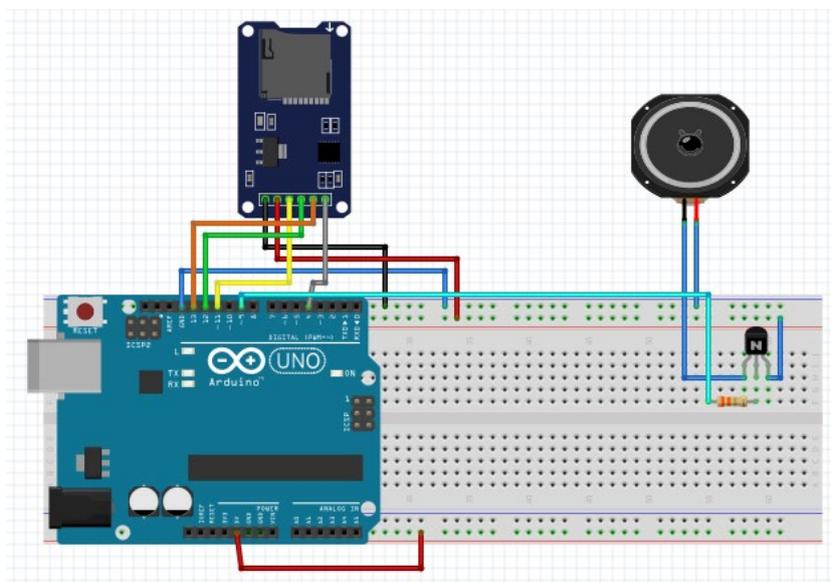
Figura 36. Conexión sensor ultrasónico conexión arduino mega.

Tomado de (Guachamin, 2020)

Una vez que se ha realizado la prueba de funcionamiento por individual de cada componente se procede con conexión en conjunto.

## 2.4 Arquitectura detección de señalética.

Para la señalética se optó por realizar pruebas de primer lugar en la placa Raspberry Pi en cual se simulo identificación de imágenes mediante la herramienta tensorflow la cual permite identificar objetos mediante un entrenamiento de algoritmos, existen algunos que ya se encuentran pre entrenados sin embargo fue necesario establecer desde cero su funcionalidad, debido a que el previo aprendizaje de esa herramienta no se lo poseía.



*Figura 37.* Conexión circuito audio para detección de señalética arduino uno.  
Tomado de (Guachamin, 2020)

Mediante la herramienta *Pycharm* y *Liclipse* se ha realizado varias simulaciones para identificación de señalética, para mediante un algoritmo previamente entrenado para que sea capaz de identificar señaléticas que son primordiales para las personas que poseen discapacidad visual.

## **2.5 Tareas a describir**

La tarea de procesamiento de entrenamiento se la ha realizado mediante *opencv-python*, *tensorflow* tanto en el pc con sistema operativo Windows como en el miniordenador *Raspberry* el cual usa las mismas herramientas para entrenamiento de algoritmos de reconocimiento de objetos.

La arquitectura que se definió lo componen el ordenar encargado de realizar la identificación de imágenes sea mediante la cámara web o cámara pi en el caso de arduino el cual detecta las imágenes.

Para la información hacia el usuario acerca de cuál fue la señalética que identifico se ha optado por un circuito armado con la placa arduino que cumple con la funcionalidad de recibir los datos vía serial y mediante un *if* al identificar el dato que recibe envía una alerta de cuál es la señal que se ha podido identificar.

## **3. Capítulo III. Diseño**

En el presente capítulo se detallará el diseño del circuito una vez ya definido los componentes a implementar en el capítulo anterior, en este capítulo se integrarán los dispositivos seleccionados, además de realizar las primeras pruebas de conexión, permitiendo validar que existe compatibilidad entre los componentes tanto en hardware como software.

### 3.1 Aspectos esenciales

Se optó por las herramientas más comunes usadas para programar y diseñar e implementar, los dispositivos más conocidos actualmente para realizar proyectos enfocados en detección de objetos y visión artificial son mediante el uso del miniordenador *Raspberry* y también se lo puede hacer con sistema operativo Windows en plataformas como *Pycharm*, *Liclipce*, para el presente proyecto se inició con *Raspberry* que existe una gran variedad de proyectos elaborados en este entorno lo que permite tener mayores fuentes de consulta.

Mediante el entorno en *Raspberry* se procedió con varios programas guía que permitieron vincularme con el entorno.

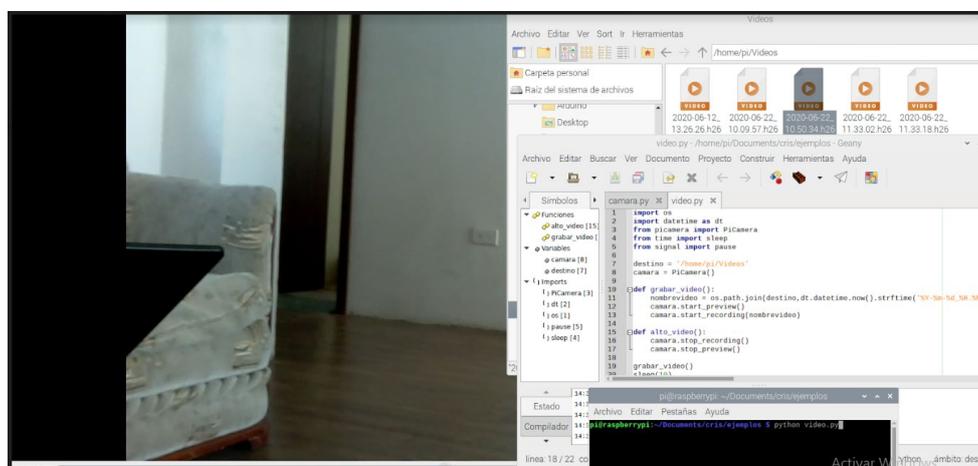
```

pi@raspberrypi:~/Documents $ ls
cris deteccion_objetos-master detector
pi@raspberrypi:~/Documents $ cd cris/
pi@raspberrypi:~/Documents/cris $ ls
ejemplos
pi@raspberrypi:~/Documents/cris $ cd ejemplos/
pi@raspberrypi:~/Documents/cris/ejemplos $ ls
arduino_opencv          camara.py  detectar.py  video.py
arduino_opencv.py      captura.py practica.py
pi@raspberrypi:~/Documents/cris/ejemplos $

```

Figura 38. Diseño y aplicación de programas detección en *Raspberry pi*.

Tomado de (Guachamin, 2020)



*Figura 39.* Ejemplos de proyectos iniciales para aplicar detección de objetos.  
Tomado de (Guachamin, 2020)

### 3.2 Casos de Uso

Los escenarios en los cuales se puede comprobar el funcionamiento del prototipo es cuando el usuario se encuentre en movimiento o se encuentre cercano al obstáculo, debido a que la información de obstáculo detectado va a estar siempre enviando avisos se sugiere que si el cliente no se va a movilizar y no ve la necesidad de identificar algún obstáculo o señalética, desactive el prototipo.

*Tabla 1.* Tabla de posibles casos que pueden presentarse al detectar.

<b>Caso</b>	<b>Detección obstáculo</b>	<b>Detección señalética</b>
<b>En el caso de ser una pared</b>	Se informa al usuario de la detección del obstáculo el cual tiene un rango de cobertura de 2 metros.	Circuito de detección no envía ninguna notificación o aviso.
<b>En el caso de un poste</b>	Se detecta obstáculo en rango de cobertura y ancho de obstáculo superior a 5 cm.	Circuito de detección no envía ninguna notificación o aviso.
<b>En el caso de detectar un objeto robusto de hasta 10 cm de ancho.</b>	Se informa al usuario de la detección del obstáculo el cual tiene un rango de cobertura de 2 metros.	Circuito de detección no envía ninguna notificación o aviso.
<b>En caso de una señal como pare, piso resbaloso, no ingrese, gradas, salida, servicios baños y</b>	Detector de obstáculo envía aviso de distancia de disco pare.	Se envía aviso de la señal detectada mediante circuito de audio.

prohibido el ingreso.		
-----------------------	--	--

### 3.3 Diagrama General

En las siguientes graficas se muestra un esquema básico del funcionamiento de prototipo, el cual se compone de 2 componentes el detector de obstáculos y el detector de señalética, los cuales son interconectados de manera inalámbrica para no generar incomodidad al usuario.

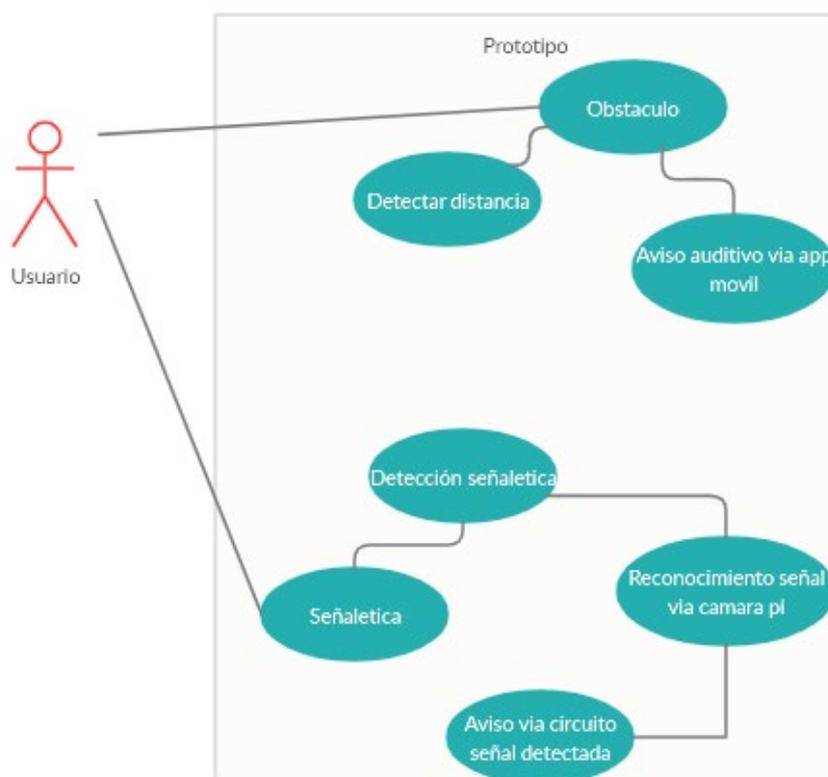


Figura 40. Diagrama de casos de uso.

Tomado de (Guachamin, 2020)

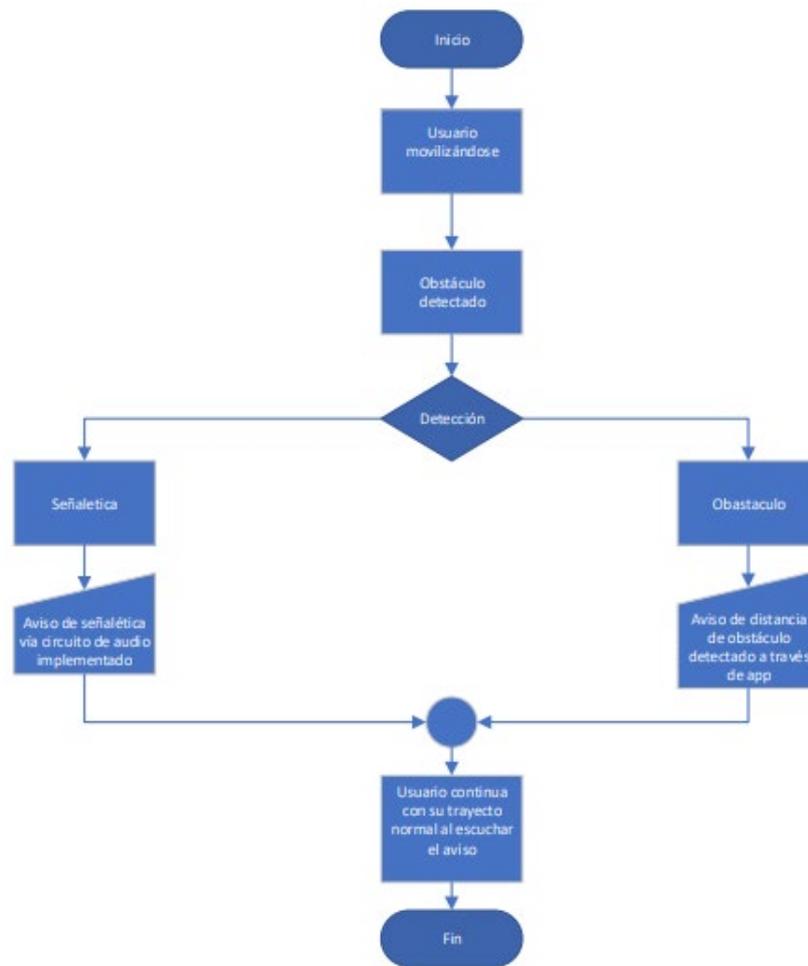


Figura 41. Diagrama de flujo de funcionamiento de prototipo.

Tomado de (Guachamin, 2020)

### 3.4 Sistemas

Los sistemas que se consideran para el diseño del prototipo se destacan el entorno de desarrollo mediante *Raspberry*, ya que es un entorno móvil al cual se pueden unir a sus periféricos teclado, mouse, monitor y cable de red Ethernet para el acceso a internet sin embargo también se puede desarrollar en un entorno de Windows con el uso de Anaconda Python o mediante Pycharm para la programación.

## 4. Capítulo IV. Desarrollo

En el presente capítulo se procederá a explicar la aplicación de los circuitos que contempla el prototipo propuesto, señalando las características más principales de cada componente, al igual que las simulaciones realizadas o códigos fuentes, además del acople de cada componente en el prototipo final a presentar con las pruebas de rigor.

### 4.1 Comentarios en el código

Los códigos que se implementaron en el prototipo corresponde a los sistemas que lo complementan la detección de obstáculos y la detección de señalética, los cuales trabajan en entornos diferentes como lo es Python, Arduino, App inventor, la relación de conexión que se implementa para ambos es vida inalámbrica para que el usuario no tenga la necesidad de tener cables que rodeen todo su cuerpo y lo incomoden.

### 4.2 Detector de obstáculos

La parte de detección de obstáculos fue basada en arduino y App Inventor, las cuales son herramientas de código abierto.

La selección de la aplicación App Inventor se basó de la simplicidad de la herramienta ya que la aplicación móvil no está diseñada para interactuar con su plataforma grafica lo que se busco fue funcionalidad y esta aplicación permite que el diseño basándose en las siguientes características:

- ✓ Aplicación de código abierto.
- ✓ Accesible desde cualquier navegador de internet.
- ✓ Orientada a funcionalidad y no a presentación.
- ✓ Operaciones básicas como conectar y desconectar bluetooth y enviar aviso de datos entrantes desde circuito detector de obstáculos.

- ✓ Fácil de programar sin necesidad de instalar todo el entorno en el pc.

#### 4.2.1 Implementación de la App

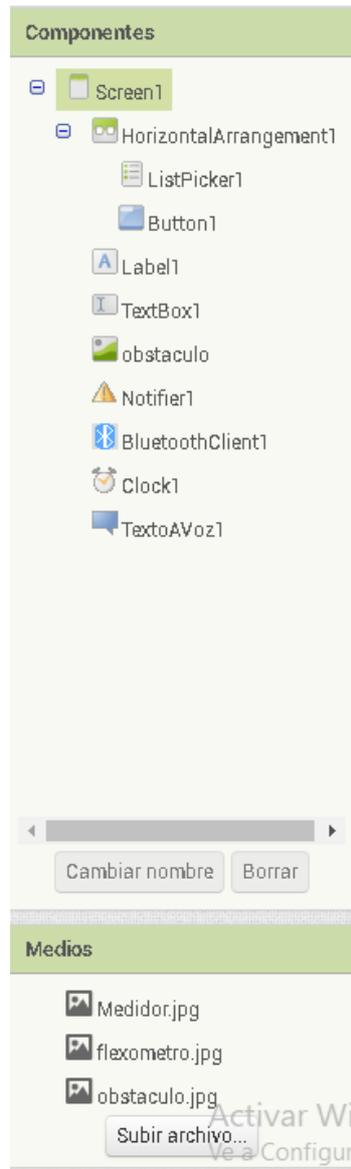
Para la implementación se procedió con una interfaz básica de una sola capa, para que si en algún punto el usuario saliese de la aplicación no se presente tanta complicación para poder acceder nuevamente.



*Figura 42.* Aplicación móvil

Tomado de (Guachamin, 2020)

Como se puede observar se dispuso únicamente de 2 botones para gestionar la conexión del móvil con el circuito arduino el cual cumplirá la función de detectar la distancia del obstáculo las mismas que el usuario recibirá en su móvil y le enviara una alerta de cuál es la distancia a la que se encuentra el obstáculo.



*Figura 43.* Componentes aplicación móvil

Tomado de (Guachamin, 2020)

Los componentes usados una vez más son botones de para la conexión, una imagen de fondo la conexión vía *bluetooth*, *labels*, y el texto a voz que es fundamental para transformar las señales enviadas desde el arduino para enviar la notificación auditiva al usuario.

#### 4.2.2 Código de la App

Mediante la unión de los bloques se da la guía de funcionamiento con el primer

segmento de bloques buscamos establecer la comunicación del bluetooth del móvil von el bluetooth de módulo arduino, señalando que cuando el usuario de clic en el *listpicker* que corresponde al botón de conectar se habilite el bluetooth del móvil.

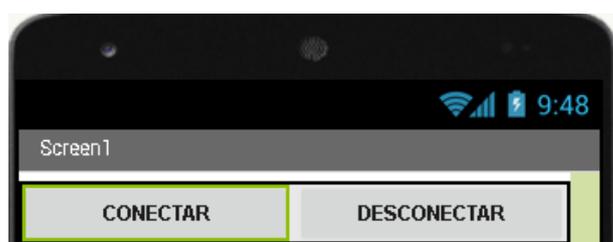


*Figura 44.* Diagrama de bloques conexión bluetooth App inventor  
Tomado de (Guachamin, 2020)

Posteriormente en el siguiente diagrama de bloques se generará la solicitud de conexión entre dispositivos al llamar al procedimiento, si el usuario da clic en aceptar se realizar la conexión y se enviara un aviso de conectado mientras que si no es vinculado de la misma manera se valida el aviso de conectado.



*Figura 45.* Diagrama de bloques confirmación conexión bluetooth  
Tomado de (Guachamin, 2020)



*Figura 46.* Diseño presentación app con *Listpicker* conectar  
Tomado de (Guachamin, 2020)

Y para la desconexión se optará por el siguiente diagrama de bloques para desconectar el usuario deberá dar clic en el botón desconectar y se envía el aviso correspondiente.

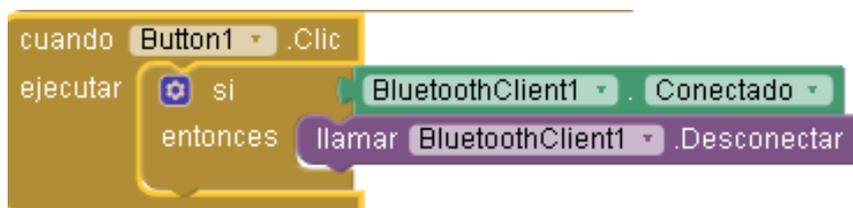


Figura 47. Diagrama de bloques desconectar bluetooth

Tomado de (Guachamin, 2020)



Figura 48. Diseño presentación app botón desconectar bluetooth

Tomado de (Guachamin, 2020)

La funcionalidad de la aplicación al recibir las señales del circuito de detección para enviar el aviso al usuario se define por el siguiente segmento de bloques.

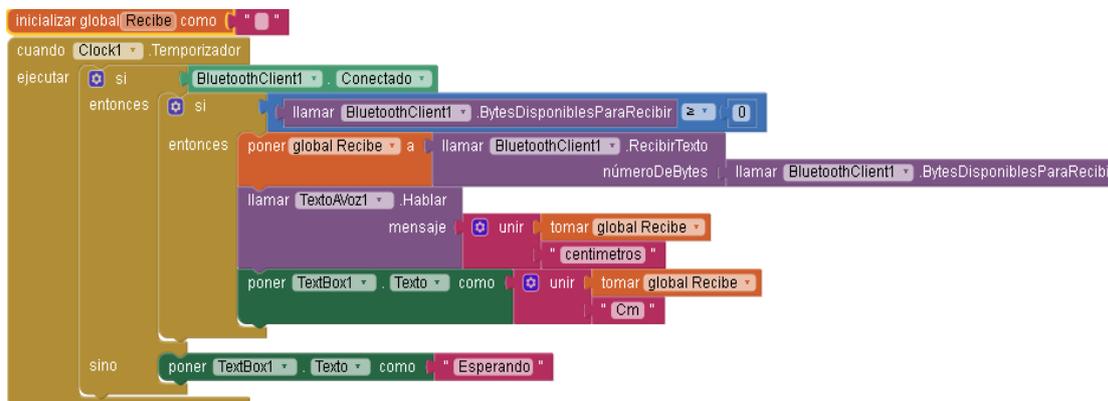


Figura 49. Diagrama de bloques recepción y tratamiento de datos.

Tomado de (Guachamin, 2020)

### 4.2.3 Código del circuito detector de obstáculos

En la placa arduino se configuro el funcionamiento del sensor ultrasónico para que se relacione la velocidad con la regresa la señal enviada permite determinar la cual es la distancia del objeto mediante la fórmula:

$$V = \frac{S}{t}$$

*Figura 50.* Formula de velocidad

Tomado de (Guachamin, 2020)

Donde:

V= velocidad

S= espacio

T= tiempo

La velocidad del sonido a la cual trabaja nuestro circuito es 343 m/s, de ahí despejamos la formula dándonos lo siguiente para poder identificar a que distancia o espacio se encuentra el objeto a detectar:

$$S = V \times t$$

*Figura 51.* Formula despeje de distancia o espacio

Tomado de (Guachamin, 2020)

```

int ecoPin= 12;
int trigPin= 13;
int ledPin= 8;

long duracion, distancia;

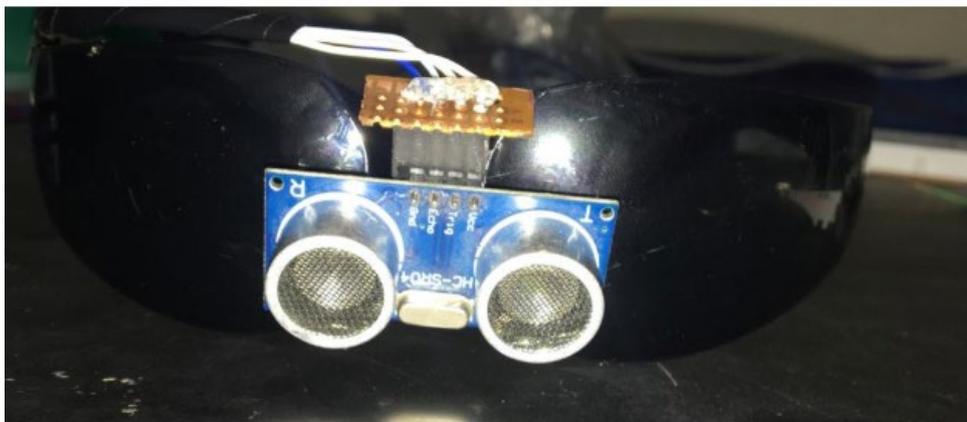
void setup(){
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT); //envia la señal de ultrasonido
  pinMode(ecoPin, INPUT); //reciba la señal de ultrasonido
  pinMode(ledPin, OUTPUT); //prende el LED
}

void loop(){
  //funcionamiento del sensor envia y recibe la señal
  digitalWrite(trigPin, LOW); delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); delayMicroseconds(5);
  digitalWrite(trigPin, LOW);
  //fin
  duracion = pulseIn(ecoPin, HIGH);
  distancia = microsecondsToCentimeters(duracion);
  Serial.print("Distancia");
  delay(3000);
  Serial.print(distancia);
  delay(3000);
  if(distancia>70){
    digitalWrite(ledPin, HIGH);
    delay(20);
  }
  if(distancia<70){
    digitalWrite(ledPin, LOW);
    delay(20);
  }
}

long microsecondsToCentimeters(long microseconds){
  return microseconds / 29 / 2;
}

```

*Figura 52.* Código detector de obstáculos Ide arduino  
Tomado de (Guachamin, 2020)



*Figura 53.* Montaje circuito detector de obstáculos  
Tomado de (Guachamin, 2020)

#### 4.2.4 Código de conexión *bluetooth*

La conexión *bluetooth* que se la realiza desde el módulo arduino debe ser la primera en ejecutarse para posteriormente correr el código detector de obstáculos, es necesario establecer una contraseña a la cual puede conectarse el dispositivo móvil.

```
#include <SoftwareSerial.h>

SoftwareSerial blue(2, 3); //Crea conexion al bluetooth - PIN 2 a TX y PIN 3 a RX

char NOMBRE[21] = "Aplicacion"; // Nombre de 20 caracteres maximo
char BPS       = '4'; // 1=1200 , 2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200
char PASS[5]   = "1234"; // PIN O CLAVE de 4 caracteres numericos

void setup()
{
  blue.begin(9600); // inicialmente la comunicacion serial a 9600 Baudios (velocidad de fabrica)

  pinMode(13,OUTPUT);
  digitalWrite(13,HIGH); // Enciende el LED 13 durante 4s antes de configurar el Bluetooth
  delay(4000);

  digitalWrite(13,LOW); // Apaga el LED 13 para iniciar la programacion

  blue.print("AT"); // Inicializa comando AT
  delay(1000);

  blue.print("AT+NAME"); // Configura el nuevo nombre
  blue.print(NOMBRE);
  delay(1000); // espera 1 segundo

  blue.print("AT+BAUD"); // Configura la nueva velocidad
  blue.print(BPS);
  delay(1000);

  blue.print("AT+PIN"); // Configura el nuevo PIN
  blue.print(PASS);
  delay(1000);
}

void loop()
{
  digitalWrite(13, !digitalRead(13)); // cuando termina de configurar el Bluetooth queda el LED 13 parpadeando
  delay(300);
}
```

Figura 54. Código conexión *bluetooth* Ide arduino

Tomado de (Guachamin, 2020)

### 4.3 Detector de señalética

Para la detección de señalética la programación se la baso en Python, mediante

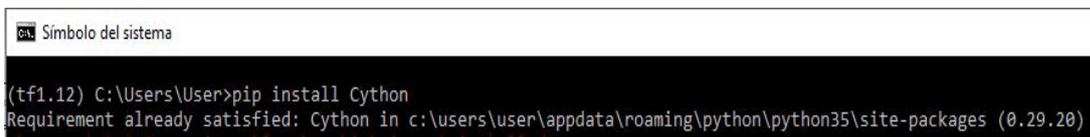
su entorno Pycharm en sistema operativo Windows, el código es abierto y puede ser usado también en el entorno de *Raspberry* para que sea portable para la propuesta, mientras que la señal auditiva se la armo en una placa arduino la cual se encarga de enviar el aviso al usuario de la señalética identificada.

### 4.3.1 Código detección de señalética

El proceso de detección de señalética registra complejidad debido a la aplicación de visión artificial que se debe usar ya que es necesario generar un modelo para identificar objetos.

Como primer punto es necesario instalar las instancias necesarias para generar el modelo predictivo como lo son las siguientes librerías:

- ✓ *Pillow*
- ✓ *Tensorflow*
- ✓ *Numpy*
- ✓ *Cv2*
- ✓ *Keras*
- ✓ *Serial*
- ✓ *Lxml*
- ✓ *Pandas*
- ✓ *Jupyter*
- ✓ *Cython*



```
Símbolo del sistema
(tf1.12) C:\Users\User>pip install Cython
Requirement already satisfied: Cython in c:\users\user\appdata\roaming\python\python35\site-packages (0.29.20)
```

*Figura 55.* Instalación Librería *Cython*

Tomado de (Guachamin, 2020)

```

Símbolo del sistema
(tf1.12) C:\Users\User>pip install pillow
Collecting pillow
  Using cached https://files.pythonhosted.org/packages/bb/f7/d644a9a65a25ff0bf43b7f69ce3ef9803742a5d6e438ddfc7e925c007e6c/Pillow-7.1.2-cp35m-win_amd64.whl

```

**Figura 56.** Instalación librería *Pillow*

Tomado de (Guachamin, 2020)

```

Símbolo del sistema
(tf1.12) C:\Users\User>pip install numpy
Requirement already satisfied: numpy in c:\programdata\anaconda3\envs\tf1.12\lib\site-packages (1.18.5)

```

**Figura 57.** Instalación librería *numpy*

Tomado de (Guachamin, 2020)

```

Símbolo del sistema
(tf1.12) C:\Users\User>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\user\appdata\roaming\python\python35\site-packages (3.0.3)
Requirement already satisfied: python-dateutil>=2.1 in c:\programdata\anaconda3\envs\tf1.12\lib\site-packages (from matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\user\appdata\roaming\python\python35\site-packages (from matplotlib)
Requirement already satisfied: numpy>=1.10.0 in c:\programdata\anaconda3\envs\tf1.12\lib\site-packages (from matplotlib) (1.18.5)

```

**Figura 58.** Instalación librería *matplotlib*

Tomado de (Guachamin, 2020)

```

Símbolo del sistema
(tf1.12) C:\Users\User>pip install lxml
Requirement already satisfied: lxml in c:\users\user\appdata\roaming\python\python35\site-packages (4.5.1)

```

**Figura 59.** Instalación de librería *lxml*

Tomado de (Guachamin, 2020)

```

Símbolo del sistema
(tf1.12) C:\Users\User>pip install jupyter
Requirement already satisfied: jupyter in c:\users\user\appdata\roaming\python\python35\site-packages (1.0.0)
Requirement already satisfied: qtconsole in c:\users\user\appdata\roaming\python\python35\site-packages (from jupyter) (4.7.5)
Requirement already satisfied: nbconvert in c:\users\user\appdata\roaming\python\python35\site-packages (from jupyter) (5.6.1)
Requirement already satisfied: ipykernel in c:\users\user\appdata\roaming\python\python35\site-packages (from jupyter) (5.3.0)

```

**Figura 60.** Instalación instancia *jupyter*

Tomado de (Guachamin, 2020)

```

Símbolo del sistema
(tf1.12) C:\Users\User>conda install -c anaconda protobuf
Collecting package metadata (current_repodata.json): failed

```

**Figura 61.** Instalación de instancia *protobuf*

Tomado de (Guachamin, 2020)

### 4.3.2 Entrenamiento de modelo predictivo de señalética.

Entrenamiento de modelo, para realizar la detección de objetos es necesario establecer un modelo previo el cual va a tener que realizar un análisis estadístico de un número de clases a las cuales vamos a detectar como:

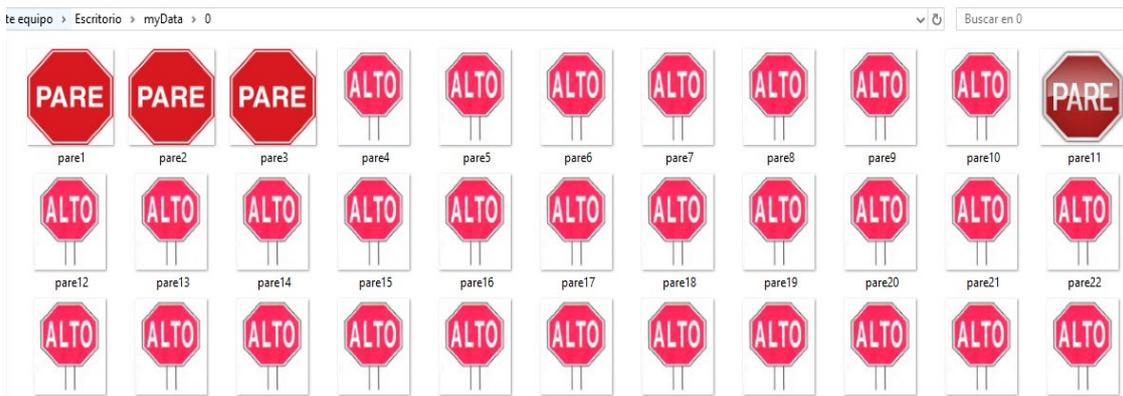


Figura 62. Gráficos de disco pare usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)



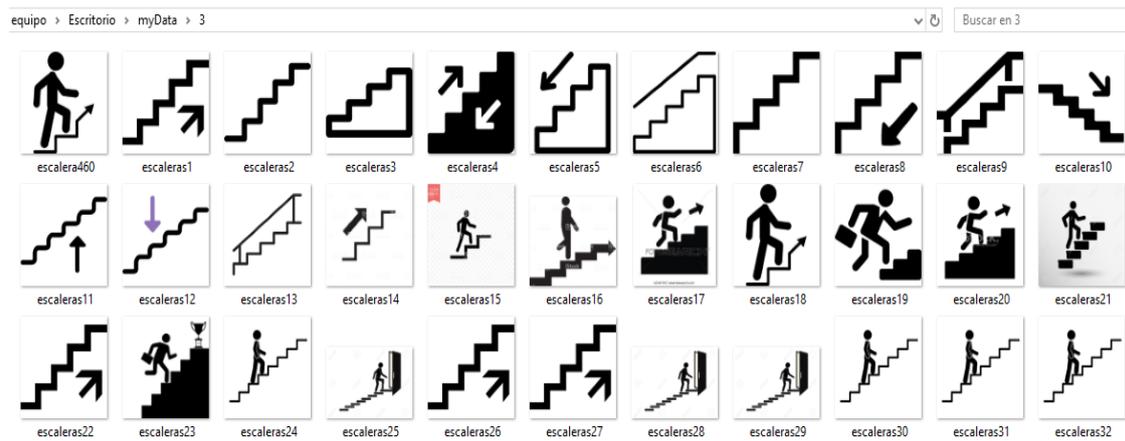
Figura 63. Gráficos de señal cruce usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)



*Figura 64.* Gráficos de señal de salida usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)



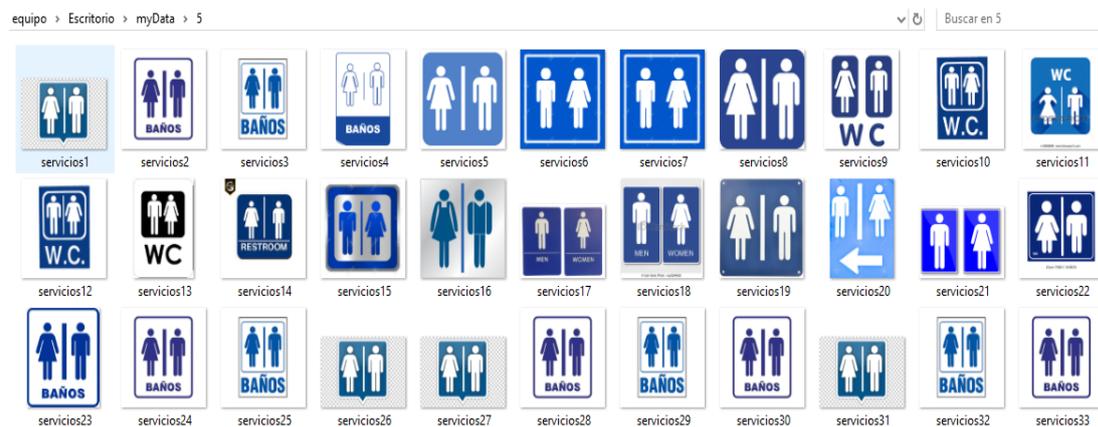
*Figura 65.* Gráficos de señal de gradas usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)



*Figura 66.* Gráficos de señal prohibido ingreso usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)



*Figura 67.* Gráficos de señal servicios baños usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)



Figura 68. Gráficos de señal ascensor usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)

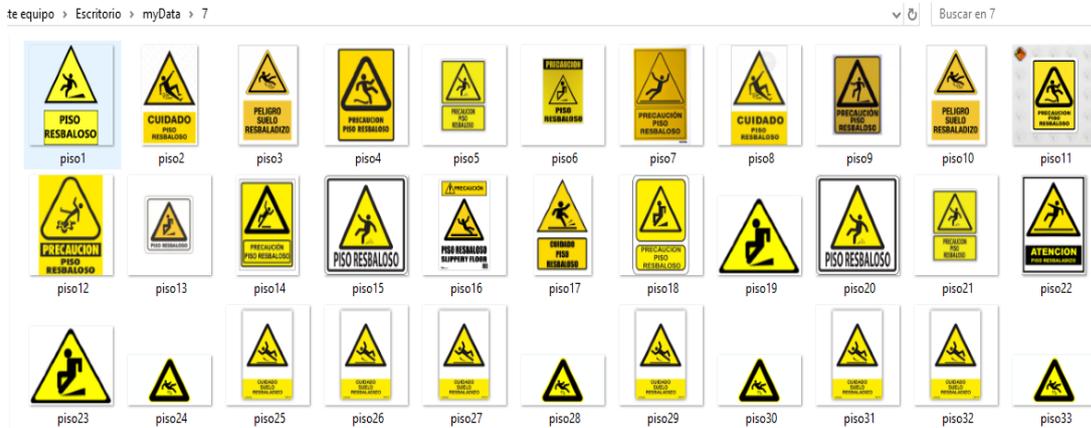


Figura 69. Gráficos de señal piso resbaloso usados para entrenamiento de modelo predictivo de detección.

Tomado de (Guachamin, 2020)

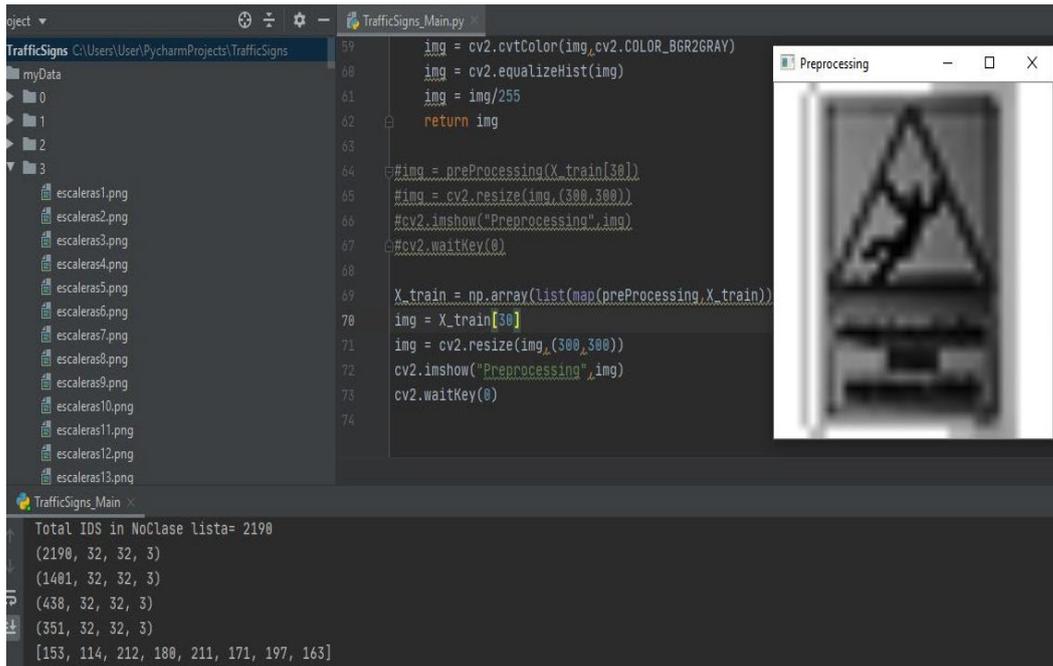
```

Project TrafficSigns
├── myData
│   ├── 0
│   ├── 1
│   ├── 2
│   ├── 3
│   └── 4
│       ├── prohibido1.png
│       ├── prohibido2.png
│       ├── prohibido3.png
│       ├── prohibido4.png
│       ├── prohibido5.png
│       ├── prohibido6.png
│       ├── prohibido7.png
│       ├── prohibido8.png
│       ├── prohibido9.png
│       ├── prohibido10.png
│       ├── prohibido11.png
│       └── prohibido12.png
TrafficSigns_Main.py
18 from keras.optimizers import Adam
19 from keras.layers import Dropout, Flatten
20 from keras.layers.convolutional import Conv2D, MaxPooling2D
21 import pickle
22
23 ##### PARAMETERS #####
24 path = 'myData'
25 testRatio = 0.2
26 valRatio = 0.2
27 imageDimensions = (32, 32, 3)
28 batchSizeVal = 50
29 epochsVal = 10
30 stepsPerEpochVal = 2000
31
32 ##### IMPORTING DATA/IMAGES FROM FOLDERS
33 count = 0
1073/2000 [=====>.....] - ETA: 6:43 - loss: 0.5758 - accuracy: 0.7892
1074/2000 [=====>.....] - ETA: 6:42 - loss: 0.5754 - accuracy: 0.7894
1075/2000 [=====>.....] - ETA: 6:42 - loss: 0.5752 - accuracy: 0.7895
1076/2000 [=====>.....] - ETA: 6:41 - loss: 0.5747 - accuracy: 0.7897
1077/2000 [=====>.....] - ETA: 6:41 - loss: 0.5742 - accuracy: 0.7898
1078/2000 [=====>.....] - ETA: 6:41 - loss: 0.5740 - accuracy: 0.7899

```

Figura 70. Código programación entrenamiento modelo de predicción señalética.

Tomado de (Guachamin, 2020)



```

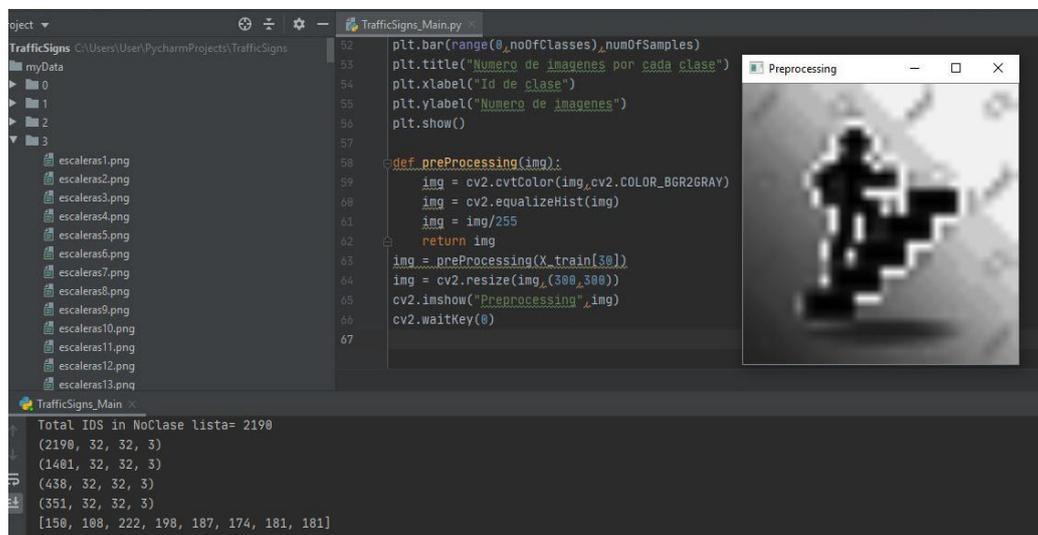
59     img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
60     img = cv2.equalizeHist(img)
61     img = img/255
62     return img
63
64     #img = preProcessing(X_train[30])
65     #img = cv2.resize(img,(300,300))
66     #cv2.imshow("Preprocessing",img)
67     #cv2.waitKey(0)
68
69     X_train = np.array(list(map(preProcessing,X_train)))
70     img = X_train[30]
71     img = cv2.resize(img,(300,300))
72     cv2.imshow("Preprocessing",img)
73     cv2.waitKey(0)
74

```

Total IDS in NoClase lista= 2190  
(2190, 32, 32, 3)  
(1401, 32, 32, 3)  
(438, 32, 32, 3)  
(351, 32, 32, 3)  
[153, 114, 212, 180, 211, 171, 197, 163]

Figura 71. Código programación entrenamiento modelo de predicción señalética, predicción señal piso resbaloso.

Tomado de (Guachamin, 2020)



```

52     plt.bar(range(0,noOfClasses),numOfSamples)
53     plt.title("Numero de imagenes por cada clase")
54     plt.xlabel("Id de clase")
55     plt.ylabel("Numero de imagenes")
56     plt.show()
57
58     def preProcessing(img):
59         img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
60         img = cv2.equalizeHist(img)
61         img = img/255
62         return img
63
64     img = preProcessing(X_train[30])
65     img = cv2.resize(img,(300,300))
66     cv2.imshow("Preprocessing",img)
67     cv2.waitKey(0)

```

Total IDS in NoClase lista= 2190  
(2190, 32, 32, 3)  
(1401, 32, 32, 3)  
(438, 32, 32, 3)  
(351, 32, 32, 3)  
[150, 188, 222, 198, 187, 174, 181, 181]

Figura 72. Código programación entrenamiento modelo de predicción señalética, predicción señal gradas.

Tomado de (Guachamin, 2020)

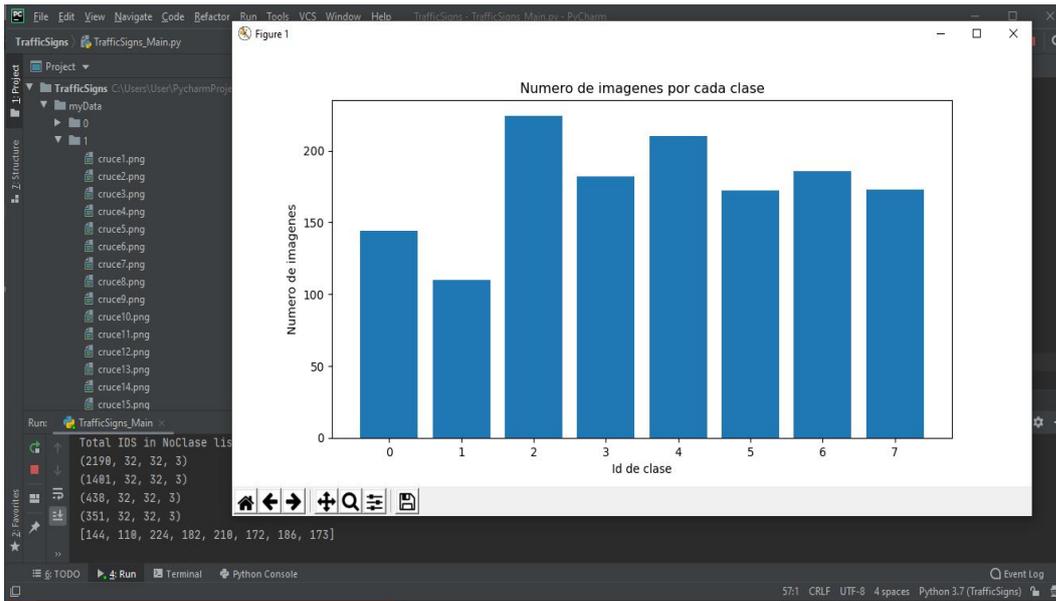


Figura 73. Estadísticas de nivel de probabilidad de detección de imágenes en base al número de muestras.

Tomado de (Guachamin, 2020)

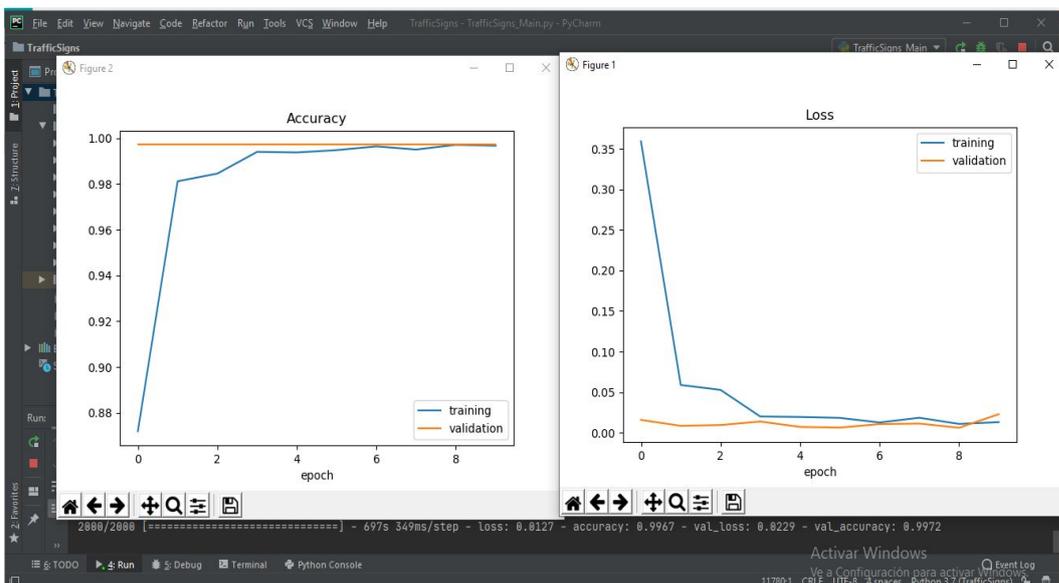


Figura 74. Resultado de ejecución de modelo de detección de señalética, se puede validar el entrenamiento entro lo perdido y detectado.

Tomado de (Guachamin, 2020)

### 4.3.3 Código circuito audio detección de señalética

Para el código se optó por trabajar en arduino el cual mediante un lector de tarjeta mirosd almacena audios de los objetos detectados que son enviados vía serial hacia el circuito elaborado.

```

#include <SD.h>
#include <TMRpcm.h>

#define Chip_SD 4

TMRpcm tmrpcm;

char lectura;

void setup(){
  Serial.begin(9600);
  tmrpcm.speakerPin = 9;
  if (!SD.begin(Chip_SD)){
    return;
  }
}
void loop (){
  delay(2400);
  Serial.begin(9600);
  if(Serial.available() >= 1){
    lectura = Serial.read();
    Serial.end();

    if (lectura == 'pare'){
      //delay(2400);
      tmrpcm.setVolume(5);
      tmrpcm.play("objpare.wav");
    }
  }
}

```

*Figura 75.* Código conexión circuito audio detección señalética.

Tomado de (Guachamin, 2020)

```

if (lectura == 'cruce'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objcruce.wav");
}

if (lectura == 'salida'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objsalida.wav");
}
if (lectura == 'gradas'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objgradas.wav");
}
if (lectura == 'salida'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objsalida.wav");
}
if (lectura == 'prohibido'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objprohibido.wav");
}
}

```

*Figura 76.* Código conexión circuito audio detección señalética.

Tomado de (Guachamin, 2020)

```

.
if (lectura == 'ascensor'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objascensor.wav");
}
if (lectura == 'pisoresbaloso'){
  //delay(2400);
  tmrpcm.setVolume(5);
  tmrpcm.play("objresbalaso.wav");
}
}}

```

*Figura 77.* Código conexión circuito audio detección señalética.

Tomado de (Guachamin, 2020)

Mediante la página web <https://audio.online-convert.com/> se puede convertir el audio pregrabado en el formato predefinido para que sea reproducido por el circuito de envío de audio de la señalética, dado que el audio viene pregrabado en la tarjeta microsd.



*Figura 78.* Página para convertir audio mp3 en WAV

Tomado de (GmbH, 2012)



*Figura 79.* Circuito emisor de aviso auditivo detector señalética.

Tomado de (Guachamin, 2020)

#### 4.4 Funcionamiento del prototipo

Al proceder con las pruebas de funcionamiento se puede validar que al estar ubicado el circuito de detección en el parte frontal de la cabeza del usuario, debido a que desde ahí se puede tener un punto estratégico para detectar obstáculos, siendo el rango de cobertura más adecuado de detección hasta de 2 metros pasado esa longitud decrece la precisión de la detección del obstáculo.

Como se puede observar en la imagen de las pruebas realizadas el usuario tiene colocado el sensor en su cabeza mientras que el dispositivo celular puede ir en un bolsillo de su camisa sin necesidad de estar expuesto, es desde ese punto en el cual el usuario puede recibir el aviso auditivo.

En relación al detector de señalética como se puede observar en la imagen ahí se tiene el circuito del envío de audio y el case con el Raspberry para procesar la detección de imágenes mediante el modelo de detección programado en Pycharm.



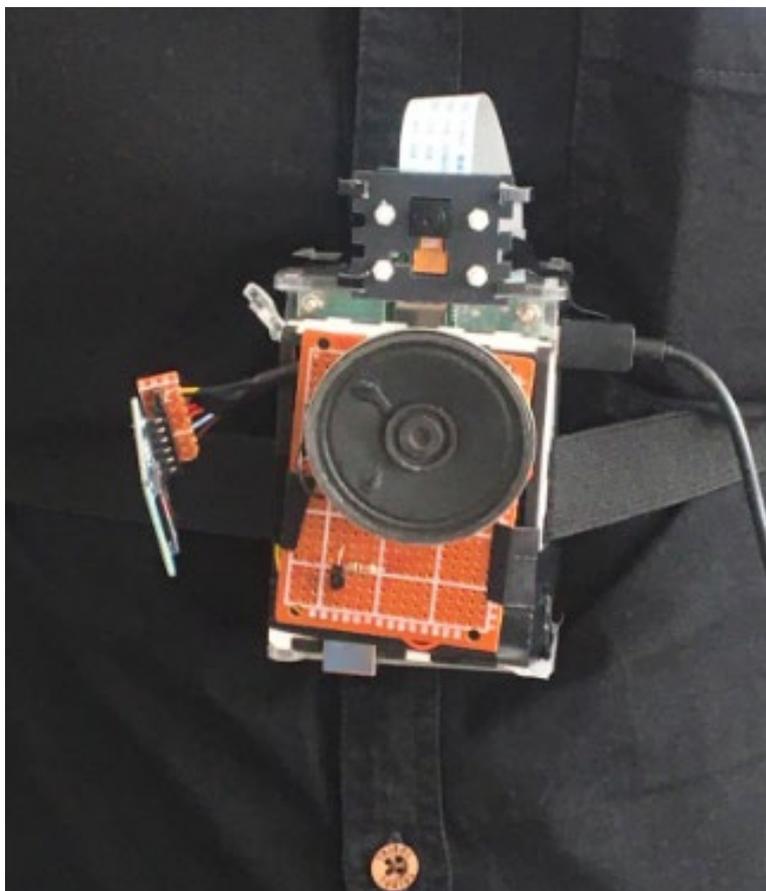
*Figura 80.* Prototipo implementado en usuario

Tomado de (Guachamin, 2020)

Como se puede apreciar en la figura el prototipo fue instalado en la parte del pecho del usuario el cual esta fijo mediante una cinta elástica la cual asegura que este fijo y no genera tanta incomodidad al usuario además si se observa un cable es el que brinda energía eléctrica el cual posteriormente puede ser sustituido por una batería la cual generaría la energía eléctrica necesaria para su funcionamiento.

Adicional se sugiere que para el aviso de detección de obstáculos que funciona mediante la app del celular, se sugiere que lo use en un punto del cual se pueda escuchar sin problema el aviso de la distancia a la cual se encuentra el obstáculo, como se puede observar en la imagen el celular se lo coloca en el bolsillo de la

camisa para que desde ahí se escuche de manera adecuada.



*Figura 81.* Prototipo implementado en usuario  
Tomado de (Guachamin, 2020)

Como se puede observar en la imagen previa se encuentra armado el circuito correspondiente a la detección de señalética con su circuito de audio para emitir el aviso, adicional se valida la tarjeta de microsd para grabar ahí el audio que será reproducido al detectar la señalética.



*Figura 82.* Prototipo implementado en usuario

Tomado de (Guachamin, 2020)

Para la detección de obstáculos se instaló del prototipo en la parte superior de su cabeza como se puede observar la cobertura de detección depende del direccionamiento de su cabeza y hacia donde apunta por lo cual es muy útil para su desplazamiento.

## **5. Conclusiones y Recomendaciones**

### **5.1 Conclusiones**

Se pudo concluir que las herramientas que actualmente poseen las personas con discapacidad visual no solucionan el problema por completo razón por la cual el prototipo implementado ayudara a desplazarse al usuario sin embargo se complementa con el uso del bastón blanco ya que existen puntos en los cuales no cubre la detección de obstáculos el prototipo.

Además se concluyó que la mejor ubicación para el sensor ultrasónico es en la parte superior del usuario específicamente en su frente ya que existen obstáculos que pueden afectar directamente al usuario como ejemplo una rama, una escalera.

También se pudo validar que para el modelo de predicción que permite detectar imágenes se puede incluir más señales que sean necesarios u otros objetos, siempre y cuando la definición de calidad de imagen de la cámara lo permita.

Finalmente se pudo concluir que existe tecnología que se puede adaptar para el segmento de personas con alguna discapacidad que no pueden utilizarlas de manera normal, como lo son aplicaciones móviles que pueden modificarse siendo incluyentes para grupos de personas que en ocasiones no son tomadas en cuenta para el lanzamiento de un producto tecnológico.

### **5.2 Recomendaciones**

Se recomienda si el usuario se encuentra descansando y no es necesario usar el prototipo, se sugiere que lo desactive ya que el detector de obstáculos siempre se encuentra detectando obstáculos y enviando avisos continuamente razón por la cual puede ser incómodo para el usuario si no se moviliza tenerlo activo.

También se recomienda si el usuario registra incomodidad al usar el prototipo debido al envío continuo de avisos auditivos cambiar el modo de aviso por

dispositivos vibratorios que replacen el sonido por intensidad en la vibración en relación a la cercanía de la detección del obstáculo.

Además se puede recomendar que si se necesita una precisión más exacta en relación a los obstáculos a detectar reemplazar el sensor ultrasónico por un sensor industrial con una sensibilidad y precisión más exacta, ya que el sensor ultrasónico seleccionado está diseñado para realizar laboratorios y prácticas.

Finalmente se recomienda al usuario determinar cuál es el lugar más adecuado en el que se siente a gusto en usar el prototipo sea en el pecho o su cabeza ya que el prototipo es inalámbrico es muy fácil cambiar la posición en la anatomía del usuario.

## 6. Referencias

- Agencia EFE, S. (2013). *EFE SALUD*. Obtenido de <https://www.efesalud.com/unbaston-inteligente-ayudara-a-los-ciegos-a-detectar-obstaculos/>
- Arduino. (2020). *Arduino*. Obtenido de <https://www.arduino.cc/en/Guide/HomePage>
- Automatización Industrial, R. e. (2020). *InfoPlc*. Obtenido de <https://www.infoplac.net/actualidad-industrial/item/102830-mercado-vision-artificial-2014>
- Barbera, J. C. (2018). *Estudio de los sensores para la detección de obstáculos aplicables a robots móviles*. Catalunya: Universidad Oberta de Catalunya.
- Comuniello, A., DeSantis, F., & Recine, G. (2020). *TecnoSalud*. Obtenido de <https://sites.google.com/a/correo.unimet.edu.ve/tecnosalud/home/tiflotecnologia>
- Electroniclab. (2020). *Tienda Electronilab*. Obtenido de <https://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>
- Federico Domenech S.A. (2013). *Las Provincias*. Obtenido de <https://www.lasprovincias.es/v/20130702/alicante/crea-baston-electronico-para-20130702.html>
- Foundation, F.-o.-F. (s.f.). *frizing*. Obtenido de <https://fritzing.org/home/>
- FOUNDATION, R. P. (2020). *Raspberry Pi 4*. Obtenido de <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
- GitHub. (2016). *Zoológico modelo de detección de Tensorflow*. Obtenido de Bienvenido al jardín modelo para TensorFlow: <https://github.com/tensorflow/models>
- GmbH, Q. M. (2012). *Online-convert.com*. Obtenido de <https://audio.online-convert.com/es/convertir-a-wav>
- González, A. G. (2020). *PanamaHitek*. Obtenido de <http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>
- Guachamin, C. (2020). *Grafico diseñado en Fritzing*. Quito.
- Inc., A. (2020). *Ananconda*. Obtenido de <https://www.anaconda.com/products/individual>
- L., B. I. (2020). *Seton*. Obtenido de <https://www.seton.es/senalizacion-adhesiva-suelo-salida-emergencia-hombre-que-corre-flecha-a-derecha.html>
- Ltda., B. S. (2020). *Liclipse*. Obtenido de <https://www.liclipse.com/>
- NaylampMechatronic. (2020). *Tienda electronica NaylampMechatronic*. Obtenido de <https://naylampmechatronics.com/sensores-proximidad/295-sensor-de-proximidad-capacitivo-ljc30a3-h-zbx.html>

- Pallás Areny, R. (2005). *SENSORES Y ACONDICIONADORES DE SEÑAL*. Madrid: S.A. MARCOMBO.
- Pepperl, F. (2003). *Sensores ultrasónicos para cualquier*. Alemania: Pepperl+Fuchs Quality.
- Pérez, D. (2020). *SENSORES DE DISTANCIA*. Obtenido de 4 Ingeniería de Telecomunicación: [http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores\\_de\\_distancias\\_con\\_ultra-sonidos.pdf](http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultra-sonidos.pdf)
- s.r.o., J. (2020). *Pycharm*. Obtenido de <https://www.jetbrains.com/es-es/pycharm/>
- salud, O. m. (2018). *Ceguera y discapacidad visual*. Obtenido de <https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- team, O. (2020). *Opencv*. Obtenido de <https://opencv.org/>
- Technology, M. I. (2020). *AppInventor*. Obtenido de <https://appinventor.mit.edu/>
- Tensorflow. (2020). *Tensorflow inicial page*. Obtenido de <https://www.tensorflow.org/>

