



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UBICACIÓN
ESPACIAL PARA PERSONAS CON DISCAPACIDAD VISUAL
MEDIANTE IOT**

Trabajo de titulación previo a la obtención del
Título de Ingeniero/a en Electrónica y Automatización

AUTORES: FRANK ANTHONY CAYAMBE GAMARRA

TUTOR: ING. LUIS SILVINO CÓRDOVA RIVADENEIRA, MSC.

Guayaquil – Ecuador

2024 -2025

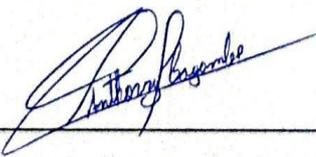
**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, **Frank Anthony Cayambe Gamarra** con documento de identificación N°
1206952564 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la
Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera
total, o parcial el presente trabajo de titulación.

Guayaquil, Marzo del año 2025

Atentamente,



Frank Anthony Cayambe Gamarra

C.I 1206952564

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

Yo, **Frank Anthony Cayambe Gamarra** con documento de identificación No. **1206952564**, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UBICACIÓN ESPACIAL PARA PERSONAS CON DISCAPACIDAD VISUAL MEDIANTE IOT**, el cual ha sido desarrollado para optar por el título de: **Ingeniero en Electrónica y automatización**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Guayaquil, marzo del año 2025

Atentamente,



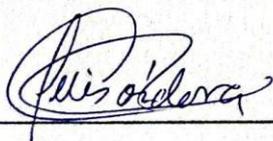
Frank Anthony Cayambe Gamarra

C.I 1206952564

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Luis Silvino Córdova Rivadeneira** con documento de identificación N° **091158939-8**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UBICACIÓN ESPACIAL PARA PERSONAS CON DISCAPACIDAD VISUAL MEDIANTE IOT**, realizado por **Frank Anthony Cayambe Gamarra** con documento de identificación N° **120695256-4**, obteniendo como resultado final el trabajo de titulación bajo la opción Trabajo de titulación que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, marzo del año 2025



Atentamente,

Ing. Luis Silvio Córdova Rivadeneira, Msc.

C.I. 091158939-8

DEDICATORIA

Esta tesis se la dedico a mi querido abuelo (José Ignacio Cayambe) aunque ha pasado tiempo desde que se fue me dejó grandes enseñanzas, una de ellas fue a luchar y seguir por lo que uno quiere.

mi querida abuelita chu (Luciana de Jesús Quinto Castro) tú que siempre estuviste pendiente de mí y orgullosa de cada logro que conseguía, me hubiera gustado que siguieras conmigo y me acompañaras en un logro más, pero, aunque te me adelantaste este logro es para ti.

Los Amo y Extraño.

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a Dios, por permitirme llegar a esta etapa final de mi carrera, por brindarme sabiduría a lo largo del camino y por darme la fortaleza necesaria para superar cada desafío sin desfallecer.

Agradezco a mi madre por siempre acompañarme y darme fortalezas en momentos tan difíciles por ser mi guía y confidente, por estar en cada momento, aunque pequeño, pero siempre alentado de seguir adelante, gracias por ayudarme a no decaer y seguir adelante.

Agradezco a mi padre por apoyarme en cumplir uno más de mis sueños por aconsejarme en momentos que lo he necesitado, por ayudarme a salir adelante, este logro También es suyo.

Gracias a todos por su amor.

RESUMEN

El próximo trabajo de investigación se concentra en desarrollar y la evaluar el sistema de navegación asistida basado en Internet de las cosas (IoT) diseñado específicamente para brindar mejoría en la autodidactación y calidad de vida de las personas con discapacidad visual en diferentes situaciones cotidianas, como en el supermercado al identificar productos y tomar decisiones de compra de forma independiente debido a la falta de acceso directo y falta de tecnología que se adapte a sus necesidades.

El prototipo propuesto es un dispositivo corporal equipado con varios sensores (por ejemplo, ultrasonidos, proximidad y GPS) que interactúan conectados a una plataforma en el Internet de las cosas. El dispositivo recopilará datos del entorno y utilizará algoritmos de procesamiento para proporcionar a los usuarios información táctil y de audio en tiempo real sobre su ubicación.

El enfoque de investigación incorporará elementos de diseño, desarrollo y evaluación, para la creación de prototipos y lenguajes de programación y plataformas IoT para implementación privada.

Los resultados de esta investigación podrían tener un impacto importante en la vida de las personas con discapacidad visual y convertirse en la base para el desarrollo de futuras soluciones tecnológicas que promuevan la integración y la autonomía.

Palabras clave: navegación IoT, sensores ultrasónicos, evaluación CAD, sensores inalámbricos, movilidad inteligente, sistema de navegación asistida.

ABSTRACT

The following research work focuses on the development and evaluation of an assisted navigation system based on the Internet of Things (IoT) specifically designed to improve the autonomy and quality of life of people with visual disabilities in different everyday situations, such as in the supermarket when identifying products and making purchasing decisions independently due to shortage of direct access and lack of technology that adapts to their needs.

The proposed prototype is a body-worn device equipped with various sensors (e.g., ultrasound, proximity, and GPS) connected to an Internet of Things platform. The device will collect data from the environment and use processing algorithms to provide users with real-time haptic and audio information about their location.

The research approach will incorporate elements of design, development and evaluation, computer-aided design (CAD) tools will be used for prototyping and programming languages and IoT platforms for private implementation.

The results of this investigation could significantly impact the lives of blind people and lay the foundation for the evolution of future technological advances that assist integration and autonomy.

Keywords: IoT navigation, ultrasonic sensors, CAD evaluation, wireless sensors, smart mobility, assisted navigation system.

ÍNDICE

RESUMEN	VII
ABSTRACT	VIII
ÍNDICE DE FIGURAS	XIV
1. INTRODUCCIÓN	- 2 -
2. PROBLEMA	- 2 -
2.1. Antecedentes	- 2 -
2.2. Importancia y alcance	- 2 -
3. DELIMITACIÓN DEL PROBLEMA	- 3 -
3.1. Temporal	- 3 -
3.2. Geográfica	- 3 -
3.3. Académica	- 3 -
4. JUSTIFICACIÓN	- 2 -
5. OBJETIVOS	- 5 -
5.1. Objetivo general	- 5 -
5.2. Objetivos específicos	- 5 -
6. MARCO TEÓRICO	- 6 -
6.1. Raspberry Pi	- 6 -
6.1.1. <i>Introducción a la Raspberry Pi</i>	- 6 -
6.1.2. Arquitectura y características	- 6 -

6.1.3.	Sistemas operativos	- 7 -
6.1.4.	Aplicaciones de la Raspberry Pi.....	- 7 -
6.1.5.	Proyectos y comunidades	- 8 -
6.1.6.	Desafíos y limitaciones.....	- 8 -
6.2.	Teachable Machine	- 9 -
6.3.	P5J.S	- 10 -
6.4.	SENSOR ULTRASONICO	- 10 -
6.4.1.	Introducción a los sensores ultrasónicos	- 11 -
6.4.2.	Principio de funcionamiento.....	- 11 -
6.4.3.	Componentes de un sensor ultrasónico	- 11 -
6.4.4.	Tipos de sensores ultrasónicos	- 12 -
6.4.5.	Aplicaciones de los sensores ultrasónicos	- 12 -
6.4.6.	Ventajas y Desventajas	- 13 -
6.5.	MODULO GPS.....	- 13 -
6.5.1.	Introducción a los módulos GPS	- 14 -
6.5.2.	Funcionamiento básico	- 14 -
6.5.3.	Componentes de un módulo GPS.....	- 14 -
6.5.4.	Tipos de módulos GPS	- 15 -
6.5.5.	Aplicaciones de los módulos GPS.....	- 15 -
6.6.	IoT O INTERNET DE LAS COSAS	- 16 -
6.6.1.	Introducción al Internet de las Cosas (IoT)	- 16 -
6.6.2.	Conceptos Fundamentales del IoT	- 17 -
6.6.3.	Arquitectura del IoT	- 17 -

6.6.4.	Aplicaciones del IoT.....	- 18 -
6.6.5.	Desafíos del IoT	- 18 -
6.7.	BATERÍAS RECARGABLES	- 19 -
6.8.	ARDUINO CLOUD	- 20 -
6.9.	CÁMARA WEBCAM.....	- 21 -
6.10.	Raspberry pi 4 Thonny	- 22 -
7.	ESTADO DEL ARTE.....	- 23 -
7.1.	Sistema IOT para asistir a personas invidentes -BLIND-N.....	- 23 -
7.2.	Diseño e implementación de un prototipo electrónico de reconocimiento de objetos y asistente de navegación para personas con discapacidad visual mediante visión artificial.	- 24 -
7.3.	Sistema domótico para personas con discapacidad visual mediante inteligencia artificial y reconocimiento de voz.	- 25 -
7.4.	Diseño e implementación de un prototipo de ubicación espacial para personas con discapacidad visual mediante IoT.	- 26 -
8.	METODOLOGÍA	- 27 -
8.1.	Diseño esquemático del proyecto	- 28 -
8.2.	Definición de variables	- 29 -
8.2.1.	Variables globales.....	- 29 -
8.2.2.	Preload	- 30 -
8.2.3.	Setup.....	- 31 -

8.2.4.	Draw	- 32 -
8.2.5.	Clasificación del video	- 32 -
8.2.6.	Manejo de resultados	- 33 -
8.2.7.	Lógica de detección y reproducción de sonidos	- 34 -
8.2.8.	Clases y sonidos	- 35 -
8.2.9.	Retardo entre clasificaciones	- 35 -
8.3.	Reconocimiento de imágenes y video	- 35 -
8.4.	Inicialización del Proyecto junto a Arduino Cloud	- 37 -
8.5.	Conexión de Arduino Cloud al módulo GPS.....	- 38 -
8.5.1.	Configuración del Software.....	- 38 -
8.5.2.	Ingreso de datos en librería.....	- 38 -
8.6.	Procedimiento de descarga de datos desde Arduino Cloud	- 38 -
8.7.	Diseño del modelo de clasificación Raspberry pi 4 Thonny	- 41 -
9.	RESULTADOS.....	- 44 -
9.1.	Desarrollo del software	- 44 -
9.2.	Integración hardware-software	- 44 -
9.3.	Pruebas y evaluación	- 45 -
9.4.	Encuestas	- 46 -
9.5.	Entrevista	- 50 -
9.6.	Análisis de Observación directa	- 52 -
9.7.	Resumen.....	- 54 -

10. CRONOGRAMA DE ACTIVIDADES	- 54 -
11. PRESUPUESTO	- 55 -
12. CONCLUSIONES	- 56 -
13. RECOMENDACIONES	- 57 -
14. BIBLIOGRAFÍA.....	- 58 -
15. ANEXOS.....	- 64 -

ÍNDICE DE FIGURAS

Figura 1. Raspberry PI 3	- 6 -
Figura 2. Teachable Machine	- 9 -
Figura 3: P5J.S	- 10 -
Figura 4: Sensor ultrasónico	- 10 -
Figura 5. Modulo Gps	- 13 -
Figura 6. IoT Internet de las cosas	- 16 -
Figura 7. Baterías recargables	- 19 -
Figura 8. Arduino Cloud	- 20 -
Figura 9. Cámara Web	- 21 -
Figura 10: Raspberry pi 4 Thonny	- 22 -
Figura 11. Ejemplo 1 de elementos de hardware	- 23 -
Figura 12. Ejemplo 2 de estado del arte	- 24 -
Figura 13. Ejemplo 3 de estado del arte	- 25 -
Figura 14. Ejemplo 4 de estado del arte	- 26 -
Figura 15: Diagrama del proceso de implementación del sistema	- 27 -
Figura 16. Diseño esquemático	- 28 -
Figura 17. Variable Classifier	- 30 -
Figura 18. Variable Preload	- 31 -
Figura 19. Variable Setup	- 31 -
Figura 20. Variable Draw	- 32 -
Figura 21: Classify Video	- 32 -
Figura 22. GotResult	- 34 -
Figura 23: Label	- 35 -

Figura 24. Entrenamiento en Teachable Machine. _____	- 36 -
Figura 25: Reconocimiento de imagen y video. _____	- 37 -
Figura 26. Datos en librería. _____	- 38 -
Figura 27. Datos de Geolocalización. _____	- 39 -
Figura 28: Prototipo completado. _____	- 40 -
Figura 29. Puerto de conexión _____	- 40 -
Figura 30. Etapa final del prototipo _____	- 40 -
Figura 31. Referencia para diseño de modelo _____	- 41 -
Figura 32. Configuración de puerto de serie _____	- 42 -
Figura 33. Función GPS de temperatura _____	- 42 -
Figura 34. Logs de demostración de hora y fecha _____	- 43 -
Figura 35. Registro de datos de temperatura _____	- 44 -
Figura 36. Resultados pregunta 1. _____	- 46 -
Figura 37. Resultados pregunta 2 _____	- 46 -
Figura 38. Resultados pregunta 3. _____	- 47 -
Figura 39. Resultados pregunta 4 _____	- 48 -
Figura 40: Resultados pregunta 5 _____	- 48 -
Figura 41: Resultados pregunta 6. _____	- 49 -
Figura 42: Resultados pregunta 7 _____	- 50 -
Figura 43. Base para la cámara del prototipo _____	- 64 -
Figura 44. Código utilizado en Rasberry pi Thonny _____	- 64 -
Figura 45. Código utilizado en PJ5 _____	- 65 -

ÍNDICE DE TABLAS

Tabla 1. Análisis de la entrevista.....	- 51 -
Tabla 2. Cronograma de actividades	- 54 -
Tabla 3. Presupuesto del proyecto	- 55 -

1. INTRODUCCIÓN

La discapacidad visual representa una de las principales barreras para la movilidad y la autonomía de quienes la experimentan, en entornos urbanos, la complejidad visual, la presencia de obstáculos imprevistos y la falta de señalización táctil adecuada dificultan significativamente la orientación y el desplazamiento de las personas ciegas, la incapacidad de percibir el entorno de manera visual limita la participación en actividades cotidianas y genera una sensación de aislamiento y dependencia (Collates Rodríguez, 2020).

El proyecto brinda como objetivo ayudar a las personas con discapacidades visuales a moverse por la ciudad utilizando las últimas tecnologías del Internet de las cosas. Estamos creando un sistema de navegación que busca ayudar a personas no videntes a desplazarse por entornos urbanos por cuenta propia. Equipado con múltiples sensores, el sistema proporcionará información continua sobre el entorno, mejorando significativamente la calidad de vida del usuario. La combinación de GPS, sensores ultrasónicos y cámaras en los dispositivos urbanos mejora la navegación y la seguridad del usuario al proporcionar información detallada sobre obstáculos y elementos visuales del entorno del vehículo, ayudando a orientarse fácilmente en espacios complejos y evitar colisiones.

La información es recibida por los sensores en tiempo real y presentado al usuario a través de una interfaz auditiva o táctil, adaptada a las necesidades específicas, esta interfaz permitirá al usuario conocer la posición actual, planificar rutas, identificar puntos de interés y recibir alertas sobre posibles obstáculos.

2. PROBLEMA

2.1. Antecedentes

Las personas con discapacidades visuales enfrentan numerosas barreras en la vida rutinaria desde obstáculos en sectores públicos, falta de señalización en braille, y la dificultad para encontrar material didáctico accesible todos debido a la falta de accesibilidad y diseño universal en los entornos construidos, específicamente, la orientación y movilidad en espacios públicos y exteriores representan un desafío significativo, a pesar de la existencia de ayudas como los bastones, estas herramientas tienen limitaciones y requieren de un entrenamiento específico. (Collates Rodríguez, 2020)

La principal problemática radica en que los entornos urbanos y los productos están diseñados principalmente para personas videntes, lo que genera dificultades para las personas ciegas en tareas cotidianas como realizar compras, cruzar calles o caminar en parques, la ausencia de señalización táctil, la complejidad de los espacios y la falta de información accesible limitan la autonomía e inclusión social. (Álvarez, 2019)

2.2. Importancia y alcance

El proyecto consiste en desarrollar un sistema portátil que, a través de sensores de movimiento como giroscopios y acelerómetros, permita controlar una silla de ruedas eléctrica existente mediante gestos.

Existe una clara necesidad de desarrollar soluciones tecnológicas innovadoras que ayuden a las personas con discapacidad visual a superar estas barreras y mejorar su calidad de vida. El prototipo de un sistema de posicionamiento espacial basado en IoT puede proporcionar información ambiental precisa y en tiempo real, favoreciendo la orientación y la movilidad.

3. DELIMITACIÓN DEL PROBLEMA

3.1. Temporal

El proyecto se desarrollará y evaluará durante un período de cuatro meses; durante este tiempo se diseñará, implementará y probará un prototipo de control electrónico de gestos. Este período de tiempo es suficiente para todas las etapas de desarrollo, incluida la investigación, el diseño, la implementación, las pruebas y los ajustes necesarios.

3.2. Geográfica

El proyecto se llevará a cabo en Guayaquil-Ecuador, el cual estará dirigido a un grupo específico de personas con discapacidad física y de movilidad que viven en esta zona.

3.3. Académica

El siguiente proyecto de técnico combina los datos adquiridos en la doctrina de la Electrónica y Automatización, aplicados al desarrollo de un sistema de control electrónico mediante gestos para personas con discapacidad visual.

4. JUSTIFICACIÓN

El permiso al avance tecnológico es un derecho fundamental que debe garantizarse a todos, incluidos los individuos con dificultad visual. Considerando este contexto, el manejo de herramientas que sean accesibles a través de la programación se convierte en una prioridad. P5.js, un marco de desarrollo basado en JavaScript centrado en la creatividad y la interactividad, combinado con Teachable Machine, la plataforma de aprendizaje automático de Google, proporciona la combinación perfecta para crear soluciones integrales.

El objetivo es desarrollar una aplicación interactiva basada en P5.js y Teachable Machine que ayude a las personas ciegas a mejorar su autonomía en el reconocimiento de objetos, colores e incluso el reconocimiento de gestos y sonidos, utilizando modelos de inteligencia artificial entrenados específicamente para satisfacer sus necesidades. La accesibilidad se logrará mediante la integración de retroalimentación de audio, reconocimiento de voz y otras interfaces adecuadas a las necesidades de cada participante.

Este enfoque no sólo promueve la inclusión digital sino que también fomenta el uso de herramientas de código abierto para desarrollar soluciones accesibles y asequibles. Además, con Teachable Machine, puedes implementar fácilmente el aprendizaje automático sin conocimientos avanzados de IA, lo que facilita su escalabilidad en una variedad de entornos educativos y de investigación.

En general, este proyecto representa un importante paso adelante en la accesibilidad tecnológica, otorgando a las personas ciegas una mayor independencia mediante la integración de inteligencia artificial y programación interactiva. Al aprovechar el poder de P5.js y Teachable Machine, estamos sentando las bases para futuras innovaciones en accesibilidad digital e inclusión social.

5. OBJETIVOS

5.1. Objetivo general

Diseñar e implementar un prototipo de ubicación espacial para personas con discapacidad visual mediante IoT.

5.2. Objetivos específicos

Realizar una investigación sobre las necesidades y retos de orientación espacial que enfrentan los individuos con discapacidad visual, mediante encuestas, entrevistas y observación directa de las interacciones en diversos entornos.

Diseñar un prototipo de dispositivo portátil con sensores de proximidad, GPS y conectividad IoT para proporcionar advertencias, guía espacial e instrucciones por voz.

Implementar un algoritmo de procesamiento de señales de los sensores y un modelo de interacción por voz para guiar y advertir al usuario sobre obstáculos y entornos próximos.

Evaluar la precisión, utilidad y facilidad de uso del prototipo con respecto a la ubicación espacial basado en IoT.

6. MARCO TEÓRICO

6.1. Raspberry Pi

Figura 1.

Raspberry Pi 4



Nota. Tomada de Suntaxi Cantuña (2019), foto de pequeño computador que correo un sistema operativo Linux.

6.1.1. Introducción a la Raspberry Pi

Raspberry Pi como se puede observar en la figura 1, se define como una línea de sistemas en chip (SBC) de mínimo presupuesto desarrollado por la Fundación Raspberry Pi en el Reino Unido. Desde su introducción, han revolucionado la educación, la investigación y el desarrollo de proyectos electrónicos gracias a su flexibilidad, bajo coste y creciente comunidad de usuarios. (Suntaxi Cantuña, 2019).

6.1.2. Arquitectura y características

La Raspberry Pi cuenta con una arquitectura basada en un procesador ARM, memoria RAM, almacenamiento interno (generalmente una tarjeta microSD) y una variedad de interfaces de entrada/salida (GPIO). Estas características la convierten en una plataforma ideal para una amplia gama de aplicaciones, desde la creación de prototipos y la automatización hasta la robótica y la Internet de las Cosas (IoT) (Orlando, 2018).

- **Procesador ARM:** Proporciona la potencia de cálculo necesaria para ejecutar sistemas operativos y aplicaciones.
- **Memoria RAM:** Almacena datos de forma temporal mientras el sistema está en funcionamiento.
- **Tarjeta microSD:** Se utiliza para el almacenamiento del sistema operativo y los datos del usuario.
- **GPIO (General Purpose Input/Output):** Permiten la interacción con el mundo físico, conectando sensores, actuadores y otros dispositivos.
- **Conectividad:** Incluye puertos USB, Ethernet, HDMI, y en algunos modelos, Wi-Fi y Bluetooth.

6.1.3. Sistemas operativos

La Raspberry Pi es compatible con una variedad de sistemas operativos, principalmente basados en Linux. Algunos de los sistemas operativos más populares incluyen:

- **Raspberry Pi OS:** Desarrollado específicamente para la Raspberry Pi, ofrece una interfaz de usuario amigable y una amplia gama de software preinstalado.
- **Ubuntu Mate:** Una distribución de Linux ligera y personalizable, ideal para usuarios que buscan un sistema operativo más familiar.
- **Kali Linux:** Orientado a la seguridad informática y la pruebas de penetración.

6.1.4. Aplicaciones de la Raspberry Pi

La versatilidad de Raspberry Pi le permite ser eficiente para una variable gama de aplicaciones:

- **Educación:** Enseñanza de programación, robótica y electrónica.

- **IoT:** Creación de dispositivos conectados a internet para automatizar tareas en el hogar, la industria y otros entornos.
- **Robótica:** Control de robots móviles, brazos robóticos y otros sistemas autónomos.
- **Emulación:** Ejecución de sistemas operativos y videojuegos retro.
- **Servidores:** Servidores web, servidores de archivos, servidores de correo electrónico.
- **Centros multimedia:** Reproductores multimedia, centros de entretenimiento en el hogar.

6.1.5. Proyectos y comunidades

La comunidad de usuarios de Raspberry Pi es muy activa y amigable. Actualmente muchos proyectos y tutoriales disponibles en Internet: Tales como proyectos simples hasta proyectos más complejos, algunas de las comunidades más importantes.

Los más comunes incluyen:

- **Raspberry Pi Foundation:** La organización detrás de Raspberry Pi proporciona muchos recursos y tutoriales.
- **Foros y comunidades en línea:** Plataformas como Raspberry Pi Forums y Reddit brindan a los usuarios un espacio para compartir conocimientos y experiencias.

6.1.6. Desafíos y limitaciones

A pesar de las ventajas, la Raspberry Pi también tiene algunas limitaciones:

- **Potencia de cálculo:** Aunque la Raspberry Pi ha realizado mejoras significativas a lo largo de los años, es posible que no sea adecuada para tareas

computacionales intensivas como el aprendizaje profundo o la representación de gráficos 3D.

- **Consumo de energía:** El consumo de energía del Raspberry Pi es un valor relativamente bajo puede ser un factor limitante en aplicaciones que requieren funcionamiento continuo durante largos períodos de tiempo.
- **Tamaño y forma:** La forma de la Raspberry Pi puede limitar su uso en algunos proyectos con espacio limitado.

6.2. Teachable Machine

Figura 2.

Teachable Machine



Nota. Modelo de Teachable Machine. Tomada de (Carranza, 2021).

Teachable Machine se define como una herramienta inteligente que es concebida de manera artificial alojándose en línea, ha sido modificada para simplificar y aumentar el proceso de creación en los modelos de automatización y aprendizaje. (Carranza, 2021)

Mediante esta herramienta nos es posible acceder con cualquier usuario y de capacitar a una computadora con la capacidad de identificar imágenes, sonidos y poses, de manera ágil y sencilla se integran en sitios webs o aplicaciones. (Perez, 2023).

6.3. P5.JS

Figura 3:

P5.JS



Nota. Tomada de Jacob (2021), imagen que muestra sensor de visión artificial equipado con múltiples funciones.

P5.JS se define como una amplia biblioteca que es parte de JavaScript, lo que se considera un lenguaje entre navegadores inteligentes, estos tienen la capacidad de ejecutar, a grandes rasgos, las páginas que se describen en HTML, donde encontramos la información y estructuración de la página, adjuntando textos, imágenes y videos.

P5.JS se describe como una manera de utilizar las tecnologías webs para manifestarnos artísticamente. En la figura 3 se puede observar. (Jacob, 2021).

6.4. SENSOR ULTRASONICO

Figura 4: *Sensor ultrasónico*



Nota. El gráfico representa una imagen de modelo ultrasónico. Tomada de Gandhi (2019).

6.4.1. Introducción a los sensores ultrasónicos

El sensor ultrasónico es dispositivo electrónico que utilizan ondas sonoras de alta frecuencia (ultrasonidos) para medir distancias, detectar objetos y realizar diversas tareas de control. Estos sensores se utilizan ampliamente en muchos campos: desde la industria y la robótica hasta la medicina y la fabricación de automóviles. (Gandhi, 2019).

6.4.2. Principio de funcionamiento

El principio básico de funcionamiento de un sensor ultrasónico se basa en emitir pulsos ultrasónicos y medir el tiempo transcurrido antes de que el eco se refleje en el objeto y regrese al sensor. Conociendo la velocidad del sonido en un medio determinado (normalmente el aire), se puede calcular la distancia a un objeto mediante la siguiente fórmula:

$$Distancia = \frac{velocidad\ del\ sonido * Tiempo\ de\ vuelo}{2}$$

El sensor consta de un transductor que actúa tanto como emisor y receptor de las ondas ultrasónicas, al emitir un pulso, el transductor entra en modo de recepción para detectar el eco, el tiempo transcurrido entre la emisión y la recepción se mide y se utiliza para calcular la distancia. (Gandhi, 2019).

6.4.3. Componentes de un sensor ultrasónico

- **Transductor:** Convierte la energía eléctrica en energía acústica y viceversa.
- **Circuito oscilador:** Genera las señales de alta frecuencia para activar el transductor.
- **Circuito de tiempo:** Mide el tiempo de vuelo del pulso ultrasónico.

- **Circuito de procesamiento:** Convierte el tiempo de vuelo en una señal de distancia legible. (Briones, 2023)

6.4.4. Tipos de sensores ultrasónicos

Existen diferentes tipos de sensores ultrasónicos, que se clasifican según la frecuencia de operación, rango de medición, forma y otros parámetros. Muchos de los más comunes son:

- **Sensor de proximidad:** detecta la presencia de objetos cercanos.
- **Sensor de nivel:** mide el nivel de líquido o sólido en el tanque.
- **Sensor de distancia:** mide con precisión la distancia a un objeto.
- **Sensor de flujo:** mide la velocidad y el flujo del líquido. (Toala, 2020)

6.4.5. Aplicaciones de los sensores ultrasónicos

Los sensores ultrasónicos tienen una amplia gama de aplicaciones, entre las que destacan:

- **Robótica:** Detección de obstáculos, navegación autónoma.
- **Automatización industrial:** Control de procesos, medición de niveles, detección de fallas.
- **Automoción:** Sistemas de estacionamiento, control de cruce adaptativo.
- **Medicina:** Ecografía, diagnóstico por imágenes.
- **Domótica:** Control de luces, persianas, sistemas de seguridad.

(Espinoza, 2023)

6.4.6. Ventajas y Desventajas

6.4.6.1. Ventajas:

- No contacto físico con el objeto a medir.
- Alta precisión en la medición de distancias.
- Insensibles a la luz ambiente y al color del objeto.
- Amplio rango de medición.

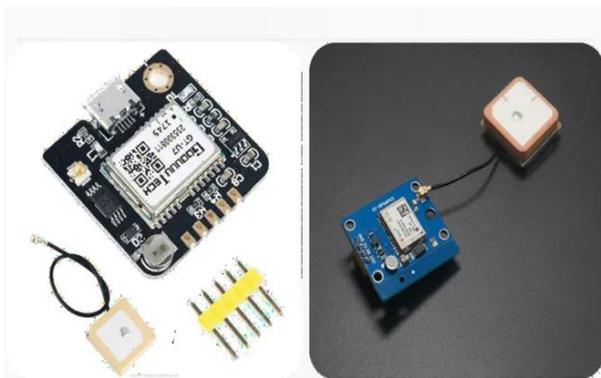
6.4.6.2. Desventajas:

- Sensibles a las interferencias acústicas.
- Pueden tener zonas muertas o ángulos muertos.
- El rendimiento se ve afectado por la temperatura y la humedad.

6.5. MODULO GPS

Figura 5.

Modulo Gps



Nota. la imagen representa un módulo GPS. Tomada de Magnetism (2023).

6.5.1. Introducción a los módulos GPS

Los módulos GPS (Sistema de Posicionamiento Global) son dispositivos electrónicos que permiten determinar la posición geográfica de un objeto en cualquier parte del mundo con una alta precisión, estos módulos han revolucionado la navegación, la logística y muchas otras aplicaciones, gracias a la capacidad de proporcionar información precisa sobre la latitud, longitud, altitud y tiempo (Magnetism, 2023).

6.5.2. Funcionamiento básico

Un módulo GPS funciona recibiendo señales de múltiples satélites GPS que orbitan la Tierra, al medir el tiempo que tardan estas señales en llegar al receptor, el módulo puede calcular la distancia de cada satélite, utilizando esta información, junto con los datos de las efemérides (posiciones de los satélites en el espacio), el módulo puede determinar la posición tridimensional en la superficie terrestre (Magnetism, 2023).

6.5.3. Componentes de un módulo GPS

- **Antena:** Capta las señales de los satélites GPS.
- **Receptor:** Procesa las señales recibidas y calcula la posición.
- **Microcontrolador:** Controla el funcionamiento del módulo y proporciona una interfaz para comunicarse con otros dispositivos.
- **Fuente de alimentación:** Proporciona de 3.3 a 5 voltios para el funcionamiento del módulo.

6.5.4. Tipos de módulos GPS

Existen diferentes tipos de módulos GPS, que se clasifican según la precisión, velocidad de adquisición de señal, consumo de energía y otras características, algunos de los tipos más comunes son:

- **Módulos GPS autónomos:** Funcionan de forma independiente y proporcionan información de posición precisa.
- **Módulos GPS diferenciales:** Utilizan señales de referencia para mejorar la precisión de la posición.
- **Módulos GPS asistidos:** Utilizan información adicional, como datos de almanacs y efemérides, para acelerar el proceso de adquisición de señal.

6.5.5. Aplicaciones de los módulos GPS

Los módulos GPS tienen una amplia gama de aplicaciones, entre las que destacan:

- **Navegación vehicular:** Sistemas de navegación por satélite, flotas de vehículos.
- **Telefonía móvil:** Localización de dispositivos móviles.
- **Geolocalización:** Aplicaciones móviles de mapas, seguimiento de objetos.
- **Agrimensura:** Levantamientos topográficos.
- **Aeronáutica:** Navegación aérea.
- **Robótica:** Navegación autónoma de robots.

6.6. IoT O INTERNET DE LAS COSAS

Figura 6.

IoT Internet de las cosas



Nota. Figura tomada de (Hat, 2023). Imagen que representa la capacidad de transmisión del IoT.

6.6.1. Introducción al Internet de las Cosas (IoT)

Internet de las cosas (IoT) es una revolución tecnológica que interconecta objetos cotidianos a Internet. Esta conectividad masiva de dispositivos ha cambiado la forma en que interactuamos con nuestro entorno, creando nuevas oportunidades en diversas áreas como la industria, la salud, la agricultura y la automatización del hogar (Reinoso, 2022).

IoT conecta "cosas" rutinarias a Internet; Desde 1990, los ingenieros informáticos han estado añadiendo sensores y semiconductores a los objetos rutinarios. Sin embargo, al principio el proceso fue difícil porque los circuitos integrados eran amplios y voluminosos. Por primera vez se han utilizado chips informáticos energéticamente eficientes, conocidos como chips, para rastrear dispositivos costosos (Hat, 2023).

6.6.2. Conceptos Fundamentales del IoT

- **Dispositivos IoT:** Son objetos físicos que incorporan sensores, actuadores y conectividad a Internet. Estos dispositivos pueden recopilar datos del entorno, realizar acciones y comunicarse con otros dispositivos.
- **Nube:** La nube actúa como un centro de almacenamiento y procesamiento de datos, permitiendo el gestionar de los datos recopilados por los dispositivos IoT.
- **Big Data:** La cantidad de datos obtenidos por los dispositivos IoT requiere de técnicas de análisis de datos a gran escala para recopilar información necesaria.
- **Inteligencia Artificial:** La IA se utiliza para analizar los datos del IoT, identificar patrones y tomar decisiones inteligentes (Reinoso, 2022).

6.6.3. Arquitectura del IoT

La arquitectura típica de un sistema IoT consta de los siguientes componentes:

- **Dispositivos:** Sensores, actuadores, microcontroladores y módulos de comunicación.
- **Gestión de dispositivos:** Plataformas que permiten la configuración, el control y la monitorización de los dispositivos IoT.
- **Comunicación:** Redes inalámbricas (Wi-Fi, Bluetooth, LoRaWAN) y protocolos de comunicación (MQTT, HTTP).
- **Nube:** Plataformas en la nube para el almacenamiento, procesamiento y análisis de datos.

- **Aplicaciones:** Interfaces de usuario y aplicaciones móviles para interactuar con los dispositivos IoT (Reinoso, 2022).

6.6.4. Aplicaciones del IoT

El IoT se utiliza en diferentes sectores:

- **Industria:** Mantenimiento predictivo, optimización de procesos, logística inteligente.
- **Salud:** Control de pacientes, telemedicina, dispositivos conectados a red.
- **Ciudades inteligentes:** Gestión del tráfico, iluminación inteligente, recolección de residuos.
- **Domótica:** Controlar la atmosfera y los sistemas de seguridad del hogar.
- **Agricultura:** Agricultura de precisión, monitoreo de cultivos, gestión del riego (Carmenate, 2023).

6.6.5. Desafíos del IoT

- **Seguridad:** El cuidado de la información y privacidad es un desafío clave, ya que los dispositivos IoT pueden ser vulnerables a ataques cibernéticos.
- **Interoperabilidad:** La falta de estándares comunes dificulta la interconexión de artefactos de distintas fábricas.
- **Escalabilidad:** Gestionar grandes cantidades de artefactos y datos requiere de infraestructuras robustas y escalables.
- **Privacidad:** El almacenamiento y el análisis de información personal plantean requerimientos éticos y legales.

6.7. BATERÍAS RECARGABLES

Figura 7.

Baterías recargables



Nota. Imagen que representa el uso de baterías recargables. Tomada de Lumpina (2023).

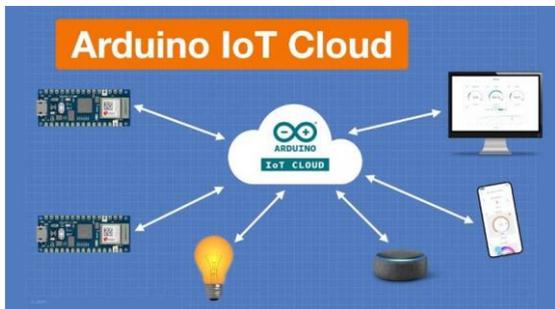
Una batería que se recarga es aquella cuya carga puede ser incorporada nuevamente en contradicción de una pila convencional que no puede recargarse. Aunque las baterías se pueden recargar, es importante recordar que este proceso no puede durar indefinidamente. Las baterías y las células solares son menos invasivas para el medio ambiente.

Estos dispositivos funcionan según el principio de reacciones electroquímicas, es decir, cuando la reacción ocurre en un sentido los materiales se agotan y para recargarlos es necesario generar una corriente eléctrica en sentido contrario para restablecerlos (Lumpina, 2023).

6.8. ARDUINO CLOUD

Figura 8.

Arduino Cloud



Nota. Figura representativa del modelo hardware de Arduino Cloud. Tomada de Sánchez (2019).

Arduino consiste en una plataforma de hardware con código abierto la cual se centraliza en microcontroladores, ofreciendo un entorno de avances diseñados para el uso de la tecnología en proyectos multifacéticos. (Sanchez, 2019).

Esta plataforma se caracteriza por múltiples componentes e interfaces integradas en un circuito, como se observa en la Figura 8. A través de los años, Arduino ha atravesado diversos avances, obteniendo como resultado la creación de distintas placas que se diferencian por la cantidad de pines analógicos, digitales, o PWM, además varía en la capacidad de memoria y en la velocidad en los microcontroladores. (Collates Rodríguez, 2020).

6.9. CÁMARA WEBCAM

Figura 9.

Cámara Web



Nota. Figura que representa el modelo característico de las Cámaras Webs. Tomada de Pesantes (2023).

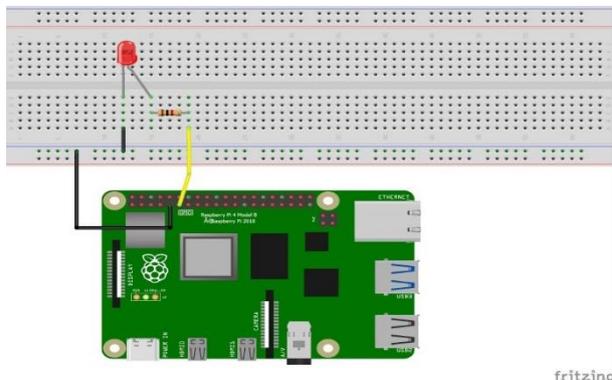
Se define como Cámara WebCam (abreviación de cámara la Wen en inglés) a aquellas cámaras que son colocadas sobre el monitor de la PC, y son capaces de enviar imágenes a través de una página web. (Pesante, 2023).

En un principio se utilizó este avance tecnológico únicamente para enviar una imagen cada determinado tiempo a un establecimiento web, donde podía ser observada por cualquier usuario. En el momento de establecerse en el mercado esta tecnología fue ampliándose en el uso, permitiendo realizar videoconferencias, videos rápidos, tomar fotografías, entre otros (Pesante, 2023).

6.10. Raspberry pi 4 Thonny

Figura 10:

Raspberry pi 4 Thonny



Nota. Modelo hardware de Raspberry pi 4 interconectado. tomada de (Reynolds, 2024).

El Raspberry pi 4 Thonny se define como prácticamente, un ordenador. Está compuesto de una placa reducida, que obtiene un bajo coste y forma de su sistema operativo con código abierto, sin embargo, no deja de ser un sistema de ordenador que debe ser adaptado a opciones básicas. Thonny, por otro lado, es el nuevo IDE oficial de Python, el cual viene incrementado con Python 3.x, facilitando la codificación inmediata (Reynolds, 2024).

Thonny consta de toda la funcionalidad que se puede necesitar para el desarrollo de aplicaciones, ofreciendo mayor facilidad dentro de los demás programadores. Su principal ventaja es que Thonny está preparado para implementar las aplicaciones que ofrece Python, sin necesidad de agregar extensiones.

7. ESTADO DEL ARTE

7.1. Sistema IOT para asistir a personas invidentes -BLIND-N

Figura 11.

Ejemplo 1 de elementos de hardware



Nota. Elementos de hardware Sistema basado en visión para asistir a personas invidentes. Tomada de Díaz, A. Campaña, S. (2021).

El sistema en desarrollo, denominado BLIND-Net, consta de una cámara RGB que se utiliza para inspeccionar el entorno o la ubicación de una persona ciega, así como una placa Jetson TX2 para procesar las imágenes capturadas. Las zonas seguras y los algoritmos de obstáculos que envían señales de advertencia en bandas táctiles permiten a las personas ciegas identificar las áreas por las que pueden pasar y las que deben evitar.

El objetivo principal de BLIND-Net es proporcionar herramientas tecnológicas de la Industria 4.0 mediante la implementación de un sistema basado en la visión que ayude a navegar en entornos interiores desconocidos, detectar obstáculos, cruzar habitaciones y encontrar objetos de interés, como puertas, sillas, escalones, ordenadores, etc., que, junto con el módulo de planificación de rutas, permite a las personas con discapacidad motriz llegar a estos objetos de forma segura.

7.2. Diseño e implementación de un prototipo electrónico de reconocimiento de objetos y asistente de navegación para personas con discapacidad visual mediante visión artificial.

Figura 12.

Ejemplo 2 de estado del arte



Nota. Estructura final del prototipo descrito. Tomada de Cadena, L, Rógel, X. (2021)

Para el desarrollo del prototipo, se optó por utilizar la placa Raspberry Pi 4 como procesador central del sistema. Este dispositivo es responsable de procesar las imágenes capturadas por la cámara web y generar mensajes de voz para notificar información al usuario. Las cámaras web están conectadas a la Raspberry Pi a través de los puertos USB 2.0, mientras que la tarjeta de sonido y los auriculares están vinculados mediante el conector de 3,5 mm.

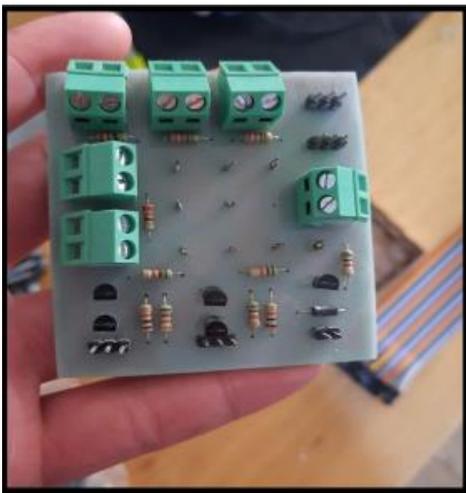
Tres botones de acción están conectados a pines de uso general, llamados GPIO, en el RPi, que funcionan como una HMI (interfaz hombre-máquina), lo que permite al usuario interactuar con el prototipo funcional. El dispositivo está equipado con un disipador de calor y un ventilador para evitar que la placa se sobrecaliente durante el funcionamiento del sistema, ya que, si excede a 85 °C, el RPi se apagará por razones de seguridad para evitar dañar el procesador.

Ambas cámaras capturan simultáneamente imágenes del entorno circundante, que sirven como datos de entrada para los algoritmos de reconocimiento de objetos y reconstrucción de imágenes. Dadas las características de la Raspberry, se decidió utilizar unos auriculares conectados directamente a la tarjeta sin utilizar un amplificador. Este tipo de auriculares consume unos 10 mA.

7.3. Sistema domótico para personas con discapacidad visual mediante inteligencia artificial y reconocimiento de voz.

Figura 13.

Ejemplo 3 de estado del arte



Nota. Diseño de placa PCB. Tomada de Lascano Solís, DA. (2023)

El proyecto de investigación se desarrolla el circuito de reconocimiento de voz usando inteligencia artificial, implementado mediante un computador Raspberry PI 4, por lo que establece un procesamiento de aprendizaje automático TensorFlow, capaz de enseñar a la red neuronal a reconocer voces específicas y así poder controlar todo el sistema, permitiendo resolver el problema con mayor calidad y menor costo, sin tener que cambiar la infraestructura de la casa controlada.

El objetivo es implantar un prototipo de sistema domótico, a través de inteligencia artificial y reconocimiento por voz, para beneficiar la calidad de vida en el hogar. Para lograr este objetivo, realizamos un análisis de la escena actual de estos individuos.

7.4. Diseño e implementación de un prototipo de ubicación espacial para personas con discapacidad visual mediante IoT.

Figura 14.

Ejemplo 4 de estado del arte



Figura 36 Vista completa del prototipo de bastón electrónico

Nota. Vista completa del prototipo de bastón electrónico. Tomada de Brito, F, Peralta, C. (2022).

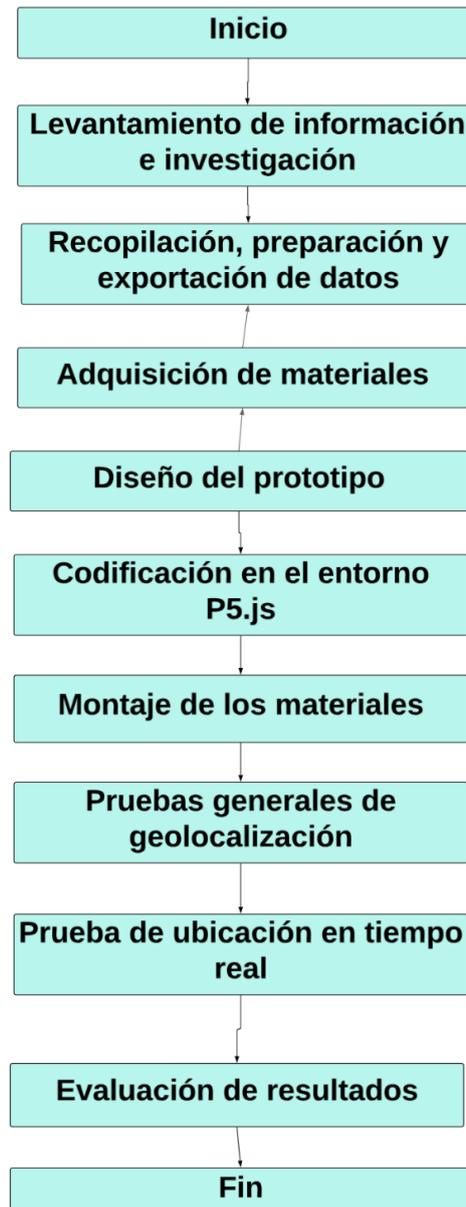
Se desarrolló un prototipo de bastón electrónico basado en la integración de sensores ultrasónicos y tecnología identificativa por radiofrecuencia (RFID).

Al realizar este trabajo se estableció un estudio preliminar sobre materiales o equipos que se puedan integrar y cumplir con los estándares esperados, lo cual será de beneficio para el beneficiario. Se desarrolló un diagrama como un mapa maestro adecuado para la dotación de personal tradicional y ubicado estratégicamente para facilitar su uso.

8. METODOLOGÍA

Figura 15:

Diagrama del proceso de implementación del sistema



De la misma forma que se realizó un análisis de los componentes que se van a utilizar en la implementación del proyecto, se analizará los datos obtenidos al momento efectuar una prueba del prototipo con los usuarios, permitiéndonos realizar mejoras y ajustes al diseño.

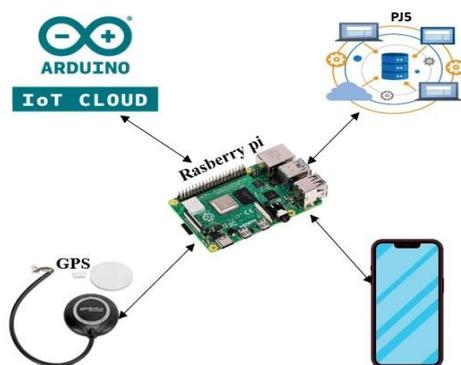
8.1. Diseño esquemático del proyecto

Para el desarrollo del modelo de imágenes, se utilizó la plataforma Teachable Machine durante su creación. Además, para la ejecución del proyecto, el diseño de la interfaz y la programación del código se empleó P5.js Web Editor, junto con el lenguaje JavaScript P5.js. La figura 16, muestra el diseño esquemático estacando la interacción entre los diferentes elementos y su integración dentro del sistema.

El funcionamiento del programa se basa en dos componentes principales: "index.html" y "sketch.js". El archivo index.html actúa como el lienzo donde se despliega el contenido de P5.js, integrando las etiquetas HTML necesarias, los enlaces a bibliotecas externas y cualquier otro elemento requerido para la visualización del modelo. Por otro lado, sketch.js contiene el código principal del proyecto, donde se define la lógica y la interacción del programa, aprovechando las funcionalidades de P5.js para proporcionar una experiencia dinámica e intuitiva al usuario.

Figura 16.

Diseño esquemático



8.2. Definición de variables

8.2.1. Variables globales

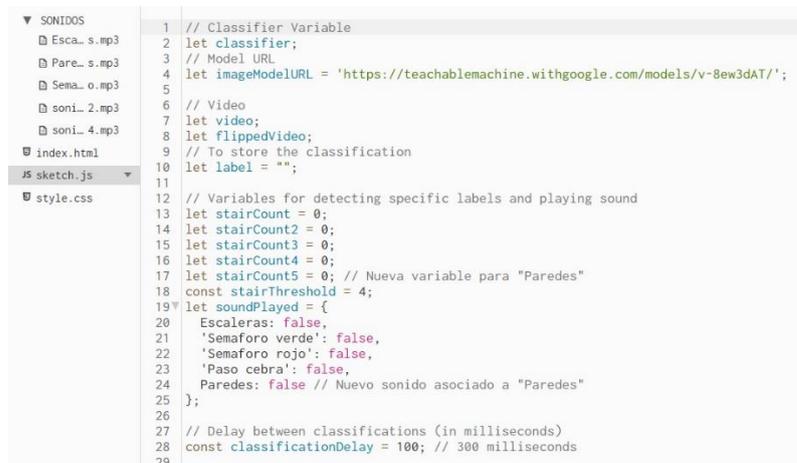
El sistema desarrollado emplea un conjunto de variables globales que permiten la gestión del modelo de clasificación de imágenes, la captura de video y la detección de objetos en tiempo real. Estas variables juegan un papel clave en la ejecución del programa, facilitando la interacción entre el usuario y el sistema.

Como se muestra en la Figura 17. Variable Classifier, se representa el almacenamiento del modelo de clasificación de imágenes, uno de los elementos esenciales para el funcionamiento del programa.

- **Classifier:** Almacena el modelo de clasificación de imágenes
- **Image Model URL:** Es la URL del modelo entrenado en Teachable Machine.
- **Video:** Almacena la captura de video de la cámara web.
- **Flipped Video:** Almacena la imagen del video volteada horizontalmente (para que se vea como un espejo).
- **Label:** Almacena la etiqueta (clase) predicha por el modelo.
- **StairCount, stairCount2, stairCount3, stairCount4, stairCount5:** Contadores para rastrear cuántas veces seguidas se ha detectado una clase específica.
- **Stair Threshold:** Umbral que define cuántas detecciones consecutivas se necesitan para realizar una acción (en este caso, reproducir un sonido).
- **SoundPlayed:** Un objeto que rastrea si un sonido ya se ha reproducido para una clase específica.
- **Classification Delay:** Retardo entre cada clasificación (en milisegundos).

Figura 17.

Variable Classifier



```
1 // Classifier Variable
2 let classifier;
3 // Model URL
4 let imageModelURL = 'https://teachablemachine.withgoogle.com/models/v-8ew3dAT/';
5
6 // Video
7 let video;
8 let flippedVideo;
9 // To store the classification
10 let label = "";
11
12 // Variables for detecting specific labels and playing sound
13 let stairCount = 0;
14 let stairCount2 = 0;
15 let stairCount3 = 0;
16 let stairCount4 = 0;
17 let stairCount5 = 0; // Nueva variable para "Paredes"
18 const stairThreshold = 4;
19 let soundPlayed = {
20   Escaleras: false,
21   'Semaforo verde': false,
22   'Semaforo rojo': false,
23   'Paso cebra': false,
24   Paredes: false // Nuevo sonido asociado a "Paredes"
25 };
26
27 // Delay between classifications (in milliseconds)
28 const classificationDelay = 100; // 300 milliseconds
29
```

8.2.2. Preload

Antes de la ejecución del programa, la función `preload()` se encarga de cargar el modelo de clasificación de imágenes y los sonidos asociados a cada clase. Esto garantiza que los recursos estén disponibles antes de que el sistema comience a procesar datos en tiempo real. En la Figura 18. Variable Preload, se muestra cómo se realiza esta carga en el código, incluyendo la inicialización del modelo y la asignación de los sonidos correspondientes a diferentes clases detectadas.

- **Preload():** Esta función se ejecuta antes de que el programa comience. Aquí se carga el modelo de clasificación y los sonidos asociados a cada clase:
- **Classifier = ml5.imageClassifier(imageModelURL + 'model.json');** Carga el modelo de clasificación desde la URL proporcionada.
- **LoadSound():** Carga los archivos de sonido que se reproducirán cuando se detecten las clases específicas.

Figura 18.

Variable Preload

```
30 // Load the model first
31 function preload() {
32   classifier = ml5.imageClassifier(imageModelURL + 'model.json');
33   // Preload the sound
34   soundFormats('mp3', 'ogg');
35   alarmSound = loadSound('SONIDOS/Escaleras.mp3');
36   alarmSound2 = loadSound('SONIDOS/sonido-2.mp3');
37   alarmSound3 = loadSound('SONIDOS/Semaforo_Rojo.mp3');
38   alarmSound4 = loadSound('SONIDOS/sonido-4.mp3');
39   alarmSound5 = loadSound('SONIDOS/Paredes.mp3'); // Nuevo sonido para "Paredes"
40 }
41
```

8.2.3. Setup

La función setup() se ejecuta una sola vez al inicio del programa y es responsable de configurar el entorno de ejecución. En la Figura 19. Variable Setup, se observa cómo se inicializa el lienzo de 640x480 píxeles, proporcionando el espacio donde se visualizará la interfaz del sistema. Además, se activa la captura de video mediante la cámara web y se ajusta su tamaño para garantizar una visualización óptima.

- Para mejorar la experiencia del usuario, la imagen del video se invierte, logrando un efecto espejo.
- Se llama a la función **classifyVideo()**, la cual da inicio al proceso de clasificación en tiempo real.

Figura 19.

Variable Setup

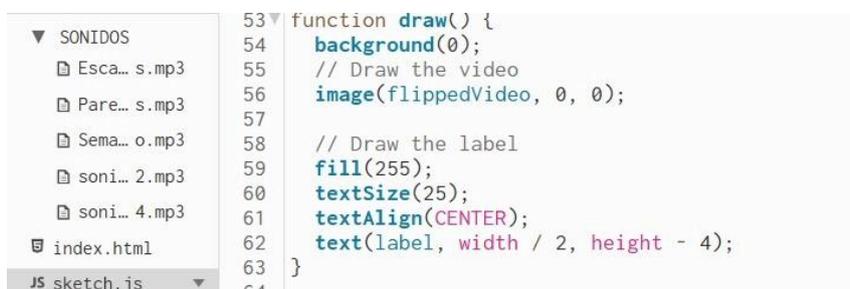
```
42 function setup() {
43   createCanvas(640, 480);
44   video = createCapture(VIDEO);
45   video.size(640, 480);
46   video.hide();
47
48   flippedVideo = ml5.flipImage(video);
49   // Start classifying
50   classifyVideo();
51 }
52
```

8.2.4. Draw

La función `draw()` tal como se muestra en la figura 20, se ejecuta de forma continua en un bucle, actualizando constantemente la visualización en el lienzo. Dentro de este proceso, se dibuja el video previamente volteado para mantener una perspectiva de espejo. Además, en la parte inferior del lienzo, se muestra la etiqueta (label) generada por el modelo, permitiendo que el usuario vea en tiempo real la clasificación realizada.

Figura 20.

Variable Draw



```
53 function draw() {
54   background(0);
55   // Draw the video
56   image(flippedVideo, 0, 0);
57
58   // Draw the label
59   fill(255);
60   textSize(25);
61   textAlign(CENTER);
62   text(label, width / 2, height - 4);
63 }
64
```

8.2.5. Clasificación del video

La función `classifyVideo()` se encarga de analizar y clasificar el fotograma actual del video en tiempo real. En la Figura 21, se muestra la estructura de esta función dentro del código, destacando su papel fundamental en la identificación de objetos en tiempo real. Para ello, primero invierte la imagen capturada, asegurando que la visualización mantenga una perspectiva de espejo. Luego, utiliza el método `classifier.classify()` para procesar el fotograma y obtener la predicción del modelo, proporcionando una clasificación precisa de los elementos detectados en la imagen.

Figura 21:

Classify Video



```
65 // Get a prediction for the current video frame
66 function classifyVideo() {
67   flippedVideo = ml5.flipImage(video);
68   classifier.classify(flippedVideo, gotResult);
69 }
70
```

8.2.6. Manejo de resultados

La función `GotResult(error, results)` se activa cuando el modelo devuelve una predicción basada en el análisis del video en tiempo real. Su propósito es gestionar los resultados obtenidos y tomar decisiones en función de la clasificación realizada. En la Figura 22. `GotResult`, se ilustra cómo el sistema maneja las predicciones, siguiendo estos pasos:

- **GotResult(error, results):** Esta función se ejecuta cuando el modelo devuelve una predicción.
- Si hay un error, se muestra en la consola.
- Si no hay error, se actualiza la etiqueta (label) con la clase predicha más confiable.
- Dependiendo de la etiqueta predicha, se incrementa el contador correspondiente (`stairCount`, `stairCount2`, etc.).
- Si un contador alcanza el umbral (`stairThreshold`) y el sonido asociado a esa clase no se ha reproducido, se reproduce el sonido y se marca como reproducido en el objeto `soundPlayed`.
- Si la etiqueta no coincide con ninguna de las clases esperadas, se reinician los contadores y el estado de los sonidos.
- Se programa una nueva clasificación después de un retardo (`classificationDelay`).

Figura 22.

GotResult

```
71 // When we get a result
72 function gotResult(error, results) {
73     // If there is an error
74     if (error) {
75         console.error(error);
76         return;
77     }
```

8.2.7. Lógica de detección y reproducción de sonidos

El código está estructurado para evitar la reproducción continua e innecesaria de sonidos. En la Figura 23. Label, se representa cómo esta lógica optimiza el funcionamiento del sistema, garantizando una detección confiable y una reproducción de sonido eficiente. Para ello, el sonido solo se activa cuando se cumplen dos condiciones específicas:

- Se detecta una clase específica un número determinado de veces consecutivas (stairThreshold), lo que garantiza mayor precisión en la identificación antes de activar el sonido.
- El sonido correspondiente a esa clase no se ha reproducido previamente (soundPlayed), evitando repeticiones innecesarias y mejorando la experiencia del usuario.

Esta lógica permite que los sonidos se activen únicamente cuando es realmente necesario, optimizando la interacción con el sistema.

Figura 23:

Label

```
79 // Update the label with the highest confidence result
80 label = results[0].label;
81
82 // Determine the label and take action
83 if (label === "Escaleras") {
84     stairCount++;
85     if (stairCount >= stairThreshold && !soundPlayed.Escaleras) {
86         alarmSound.play();
87         soundPlayed.Escaleras = true;
88     }
}
```

8.2.8. Clases y sonidos

Las clases que el modelo puede detectar son:

- Escaleras: Reproduce alarmSound (SONIDOS/Escaleras.mp3).
- Semaforo verde: Reproduce alarmSound2 (SONIDOS/sonido-2.mp3).
- Semaforo Rojo: Reproduce alarmSound3 (SONIDOS/Semaforo_Rojo.mp3).
- Paso cebra: Reproduce alarmSound4 (SONIDOS/sonido-4.mp3).
- Paredes: Reproduce alarmSound5 (SONIDOS/Paredes.mp3).

8.2.9. Retardo entre clasificaciones

El retardo (classificationDelay) entre clasificaciones asegura que el modelo no esté clasificando constantemente, lo que podría sobrecargar el sistema. Este mecanismo garantiza que el modelo no realice clasificaciones continuas e innecesarias, lo que podría generar una sobrecarga en el procesamiento y afectar la eficiencia del dispositivo.

El classificationDelay establece un tiempo de espera entre cada clasificación, permitiendo que el sistema procese la información de manera más fluida y evitando que se reproduzcan sonidos o se ejecuten acciones de manera repetitiva e innecesaria.

8.3. Reconocimiento de imágenes y video

El sistema tiene la capacidad de reconocer imágenes y videos capturados en tiempo real, basándose en los datos previamente entrenados en Teachable Machine. En la

Figura 24. Entrenamiento en Teachable Machine, se muestra el proceso de aprendizaje automático que permite al modelo identificar distintas clases con alta precisión.

Una vez entrenado, el sistema puede detectar en tiempo real los objetos y escenarios definidos en la fase de entrenamiento, como se ilustra en la Figura 25.

Reconocimiento de Imagen y Video. Esto permite que el usuario reciba alertas auditivas precisas sobre su entorno, mejorando significativamente su movilidad y seguridad.

Figura 24.

Entrenamiento en Teachable Machine.

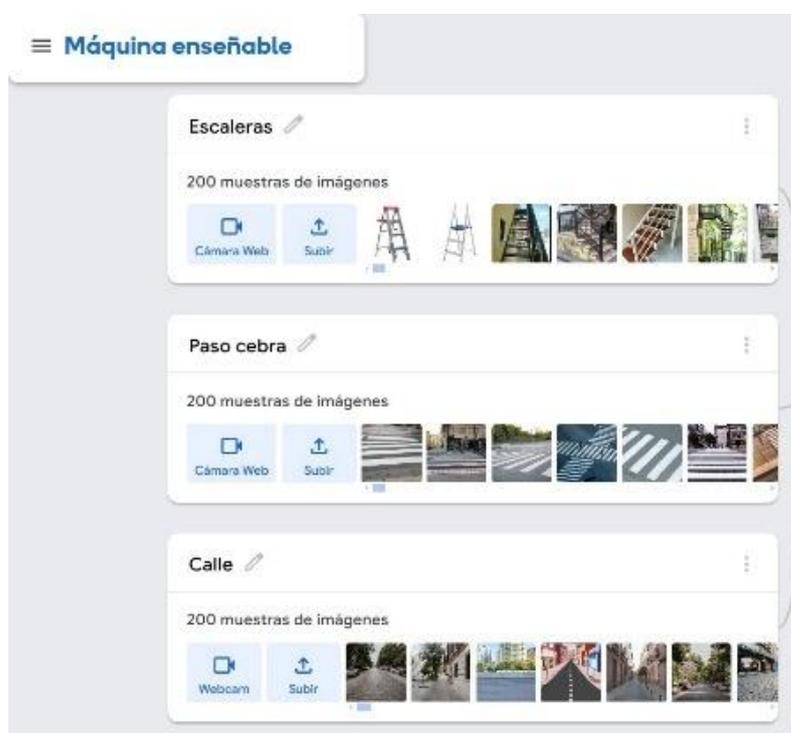
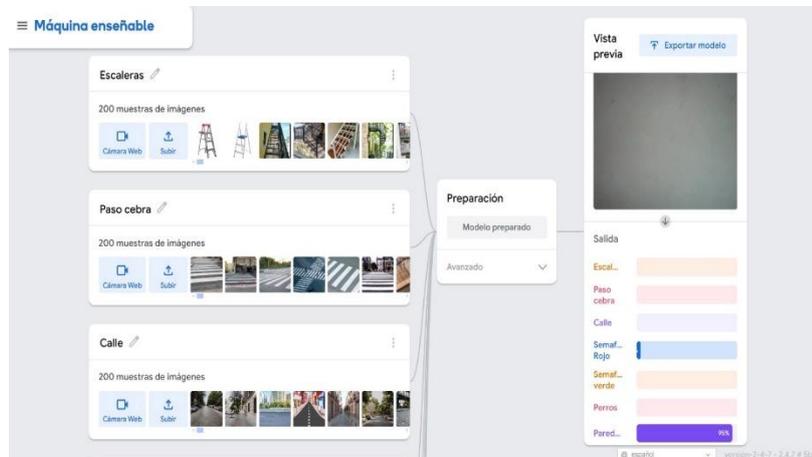


Figura 25:

Reconocimiento de imagen y video.



8.4. Inicialización del Proyecto junto a Arduino Cloud

Para realizar el uso de Arduino Cloud se debe seguir los siguientes pasos:

1. Debemos crear una cuenta, donde se puede ingresar con Google, Facebook, o iCloud mail.
2. Una vez creada una cuenta, debemos crear un proyecto nuevo. Damos click en Dashboard.
3. Ingresamos las interfaces a utilizar, en este caso “Gauge”.
4. Observamos los datos que nos brinda a través de gráficos y medidores.
5. Configuramos alertas, y automatizaciones como la temperatura.
6. La información brindada es guardada en el almacenamiento de Arduino antes de transmitirlo.
7. Visualizamos los gráficos, medidores o displays numéricos obtenidos de la recopilación de datos.

8. Una vez integrada la información nos es posible observar en tiempo real los datos adquiridos en cualquier entorno o dispositivo conectado a Internet.

8.5. Conexión de Arduino Cloud al módulo GPS

Se describe el desarrollo del algoritmo que se necesita para conectar Arduino Cloud al módulo GPS, donde se incluye el hardware, la configuración del software y el código para lograr la intercomunicación. Este procedimiento abarca tanto la configuración del hardware como la del software, además de la implementación del código necesario para garantizar una transmisión de datos estable y precisa.

8.5.1. Configuración del Software

- Instalación de Librería GPS desde el administrador de librerías.
- Instalar Librería Arduino Cloud desde el administrador de librerías.

8.5.2. Ingreso de datos en librería

Se debe ingresar las librerías a utilizar, de manera como se muestra en la Figura 26.

Figura 26.

Datos en librería.

```
PROYECTO_TEST6.ino
1 #include <Arduino.h> //librería de características adicionales de arduino tal como pin, pin.
2 #include <LiquidCrystal_I2C.h> //LIBRERÍA DE LCD
3
4 #include <math.h> //LIBRERÍA DE CÁLCULOS MATEMÁTICOS DE PID
5 #include <GPS.h> //LIBRERÍA DE MÓDULO GPS DE ARDUINO
6
7
8 // Librería para la conexión de IoT de Arduino
9 #include <ArduinoIoTCloud.h>
10 #include <Arduino_ConnectionHandler.h>
11
12 // desactive bit guardias de tarjeta
13 #include "sec/wc.h"
14 #include "sec/rtc_ext1_rmg.h"
15
16 // LIBRERÍA DE sensor de temperatura
17 #include "max645.h"
18
19 //librería de gps
20 #include <TinyGPS.h>
21 #include <HardwareSerial.h>
```

Aquí se incluyen todas las librerías.

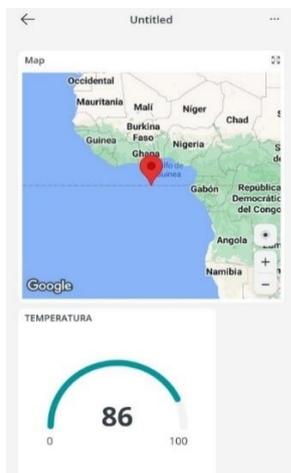
8.6. Procedimiento de descarga de datos desde Arduino Cloud

1. Se inicia sesión en la plataforma de Arduino Cloud.
2. Seleccionamos el proyecto correspondiente.
3. Elegimos la opción de descarga.

4. Observamos las variables a escoger, tales como temperatura y ubicación, las cuales recibes señales emitidas por el GPS, seleccionamos la variable y el rango de interés para descargar los datos.
5. Seleccionamos “select data source” la cual nos mostrará un archivo comprimido en el correo electrónico registrado.
6. Revisamos el correo electrónico, damos click a descargar, donde el archivo se encontrará comprimido.
7. Luego de descomprimir los archivos, obtendremos una tabla de datos en Excel, donde se mostrará la fecha, hora y temperatura de la información recopilada.
8. Observamos datos de geolocalización obtenidas, tales como latitud y longitud, tal y como se muestra en la Figura 27.

Figura 27.

Datos de Geolocalización.



Prototipo completo (Rasperry pi) junto a batería complementando el circuito. Se encuentra conectado mediante una pantalla HDMI para completar su funcionamiento y en conjunto al GPS. De la cual se puede observar en las figuras 28, 29 y 30.

Figura 28:

Prototipo completado.



Figura 29.

Puerto de conexión



Figura 30.

Etapa final del prototipo



8.7. Diseño del modelo de clasificación Raspberry pi 4 Thonny

En esta sección se presenta el código de referencia utilizado para el desarrollo del modelo de clasificación en la Raspberry Pi 4, integrando su funcionamiento con Arduino Cloud. Este diseño permite la comunicación entre los dispositivos, facilitando el procesamiento y la gestión de datos en tiempo real. En la Figura 31, se ilustra la referencia utilizada para el diseño del modelo, mostrando la estructura general del código y su implementación dentro del sistema.

Figura 31.

Referencia para diseño de modelo

```
Librerías Importadas
python
Copiar
Editar
from arduino_iot_cloud import ArduinoCloudClient
import time
import logging
import json
import sys
import random
import pynmea2
import serial

arduino_iot_cloud: Para conectarse a la plataforma Arduino IoT Cloud.
time: Para manejar pausas (sleep).
logging: Para manejar registros de eventos y errores.
json: Posible uso futuro para manejar datos en formato JSON.
sys: Modifica rutas y otros aspectos del sistema.
random: Para generar números aleatorios (simulación de temperatura).
pynmea2: Librería para interpretar datos NMEA provenientes de un GPS.
serial: Para manejar la comunicación serie con el GPS.
Configuración del Puerto Serie
python
Copiar
Editar
```

Para establecer la comunicación entre Arduino Cloud y la Raspberry Pi 4, es fundamental configurar correctamente el puerto de serie y las credenciales de acceso en Arduino IoT Cloud. Se muestra configuración de puerto de serie en Figura 32 junto a credenciales de Arduino IoT Cloud.

Figura 32.

Configuración de puerto de serie

```
Configuración del Puerto Serie
python
Copiar
Editar
port = serial.Serial("/dev/ttyUSB0", baudrate=9600, timeout=1)
Abre la comunicación con el puerto USB0 a 9600 baudios.
timeout=1: Espera hasta 1 segundo antes de descartar una lectura.
Credenciales de Arduino IoT Cloud
python
Copiar
Editar
DEVICE_ID = "b7cbdb68-d940-40c1-9226-6ef7e92ee814"
SECRET_KEY = "qvRM828j7nLGCLIm2MYsQAaP"
Estas son las credenciales para autenticarse en Arduino IoT Cloud.
```

El sistema incorpora una función GPS que permite registrar y procesar datos de latitud y longitud en tiempo real, almacenándolos en una base de datos para su posterior análisis. Se muestra Función GPS de temperatura en la figura 33, donde se observa la interpretación de latitud y longitud registrándose en la base de datos.

Figura 33.

Función GPS de temperatura

```
Función gps_temperatura HOY
python
Copiar
Editar
def gps_temperatura(client, args):
    try:
        line = port.readline().decode('ascii', errors='replace')
        print(line)

        if line.startswith("$GPGGA") or line.startswith("$GPRMC"):
            msg = pynmea2.parse(line)
            print(msg)
            lat = msg.latitude
            lon = msg.longitude
            client.register(Location("GPS2", lat=lat, lon=lon))
    except pynmea2.ParseError:
        logging.warning("Error al parsear datos del GPS.")
    except Exception as e:
        logging.error(f"Error al leer datos del GPS: {e}")
Captura errores de parseo o fallos en la lectura.
Simulación de Temperatura
python
```

El sistema cuenta con una configuración específica para la generación de logs que registran la hora y fecha de cada ubicación obtenida a través del módulo GPS. Se muestra configuración para logs de demostración de hora y fecha de los registros GPS. Tal como se muestra en la figura 34.

Figura 34.

Logs de demostración de hora y fecha

```
Copiar
Editar
if __name__ == "__main__":
    try:
        logging.basicConfig(
            datefmt="%H:%M:%S",
            format="%(asctime)s.%(msecs)03d %(message)s",
            level=logging.INFO,
        )
        Configura los logs para mostrar la hora y los mensajes de error/información.
    python
    Copiar
    Editar
        client = ArduinoCloudClient(device_id=DEVICE_ID, username=DEVICE_ID,
        password=SECRET_KEY)
        Crea el cliente de Arduino IoT Cloud con las credenciales.
    python
    Copiar
    Editar
        client.register(Task("gps_temperatura", on_run=gps_temperatura, interval=1.0))
        client.start()
        Registra la tarea gps_temperatura, que se ejecutará cada 1 segundo.
        Inicia la conexión con Arduino IoT Cloud.
    python
    Copiar
    Editar
```

El sistema está diseñado para registrar y almacenar datos de temperatura en tiempo real, integrándolos con Arduino IoT Cloud para su monitoreo y gestión remota. Esta conexión permite visualizar los valores capturados y analizarlos a través de la plataforma IoT. Se muestra registro de datos de temperatura en la figura 35, junto a la conexión con Arduino IoT Cloud.

Figura 35.

Registro de datos de temperatura

```
Registra la tarea gps_temperatura, que se ejecutará cada 1 segundo.
Inicia la conexión con Arduino IoT Cloud.
python
Copiar
Editar
    except Exception as e:
        logging.error(f"Error al enviar datos de ubicación: {e}")
Captura errores en la conexión con la nube.
python
Copiar
Editar
    while True:
        client.update()
        time.sleep(0.100)
Actualiza continuamente la conexión con Arduino IoT Cloud cada 100ms.
Resumen
Este script:

Lee datos GPS desde un puerto serie.
Filtra y extrae latitud y longitud.
Envía las coordenadas a Arduino IoT Cloud.
Simula un sensor de temperatura aleatorio y lo envía a la nube.
Mantiene la conexión con Arduino IoT Cloud actualizada en un bucle infinito.
```

9. RESULTADOS

9.1. Desarrollo del software

Plataformas utilizadas: Para el desarrollo del algoritmo de reconocimiento de objetos, se utilizaron las plataformas Teaching Machine y p5.js. Teaching Machine fue seleccionada por su facilidad de uso en el entrenamiento del modelo, mientras que p5.js permitió la creación de una interfaz visual sencilla para la visualización de los resultados.

Proceso de entrenamiento: El modelo de reconocimiento de objetos fue entrenado utilizando un conjunto de datos de imágenes que incluye productos comunes de supermercados. A través de este entrenamiento, el modelo adquirió la capacidad de identificar una variedad de objetos, como frutas, verduras y productos de limpieza, entre otros.

9.2. Integración hardware-software

Comunicación: Los sensores conectados al Arduino, fueron capaces de recolectar los sonidos del entorno en tiempo real, los cuales fueron procesados por el

modelo de reconocimiento de sonidos, y los resultados fueron enviados a la interfaz de usuario del dispositivo portátil.

Señales de alerta: Cuando el modelo detectaba un sonido predefinido, se emitía una señal de alerta al usuario mediante vibraciones o sonidos, la intensidad y duración de la señal podían ser personalizadas según las preferencias del usuario.

9.3. Pruebas y evaluación

Entorno de pruebas: El prototipo fue evaluado en un entorno de trabajo cotidiano, donde se colocaron diferentes obstáculos en ubicaciones estratégicas con el objetivo de analizar su desempeño y capacidad de detección en situaciones reales. Este proceso permitió identificar posibles mejoras y ajustar la configuración para optimizar su funcionamiento en escenarios del mundo real.

Resultados de las pruebas: Los resultados muestran que el conjunto de sistemas de reconocimiento de sonidos logró una precisión del 85% en la identificación de los productos objetivo. El tiempo de respuesta promedio fue de 25 segundos, lo cual se considera aceptable para una interacción en tiempo real.

Análisis de los resultados: Los resultados obtenidos evidencian la viabilidad de utilizar tecnologías como Teaching Machine y p5.js para el desarrollo de sistemas de reconocimiento de objetos en tiempo real. No obstante, se identificaron algunas limitaciones, entre ellas la dificultad para detectar objetos en entornos con baja iluminación o niveles elevados de ruido, lo que podría afectar la precisión del sistema en determinadas condiciones.

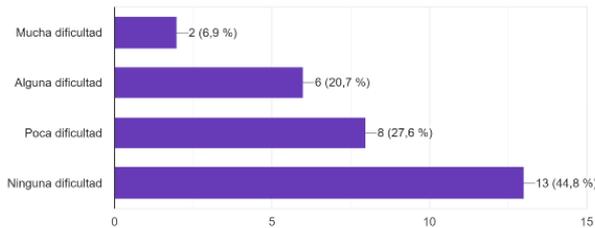
9.4. Encuestas

Figura 36.

Resultados pregunta 1.

1. ¿Qué grado de dificultad experimenta al desplazarse en entornos interiores (hogares, edificios, etc.)?

29 respuestas



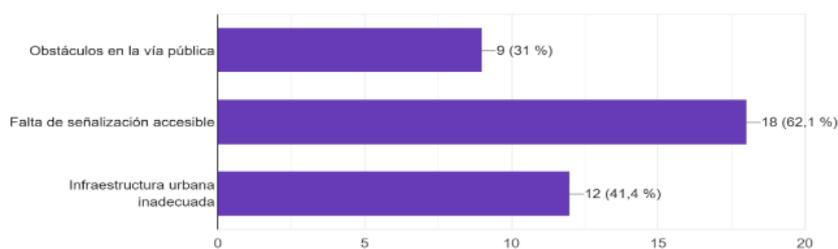
La mayoría de los encuestados (44.8%) no experimenta ninguna dificultad al desplazarse en entornos interiores. Sin embargo, más de la mitad (55.2%) reporta algún grado de dificultad, variando desde poca hasta mucha. Esto sugiere que, aunque una proporción significativa no tiene problemas, hay una necesidad considerable de mejorar la accesibilidad en espacios interiores para atender a quienes sí enfrentan desafíos.

Figura 37.

Resultados pregunta 2

2. ¿Cuáles son los principales desafíos que enfrenta al orientarse y moverse en espacios exteriores (calles, parques, etc.)?

29 respuestas

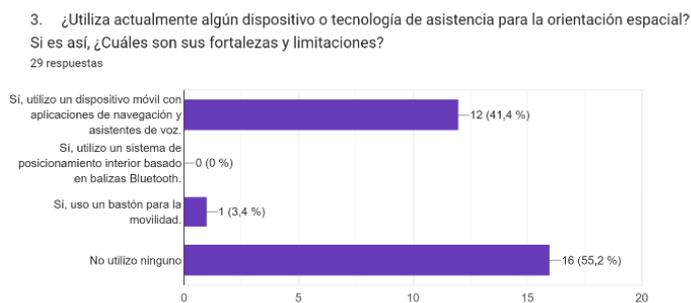


La falta de señalización accesible es el desafío más prominente, afectando al 62.1% de los encuestados. Los otros dos problemas principales son la infraestructura urbana inadecuada (41.4%) y los obstáculos en la vía pública (31%). Esto indica que las mejoras en señalización accesible deberían ser una prioridad para facilitar la orientación y

movilidad en espacios exteriores, seguidas por intervenciones en infraestructura y eliminación de obstáculos.

Figura 38.

Resultados pregunta 3.



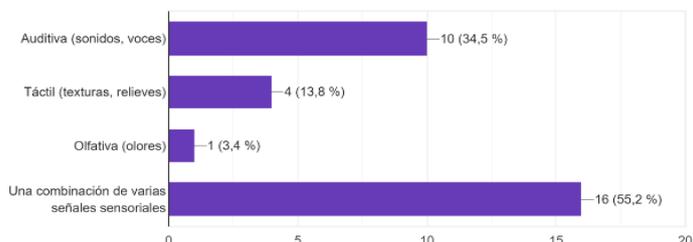
La mayoría de los encuestados (55.2%) no utiliza ningún dispositivo o tecnología de asistencia para la orientación espacial. Entre quienes sí usan, los dispositivos móviles con aplicaciones de navegación y asistentes de voz son los más populares (41.4%). Hay un uso muy limitado de bastones (3.4%) y ningún uso de sistemas de posicionamiento interior basados en balizas Bluetooth. Esto sugiere una oportunidad significativa para aumentar la adopción de tecnologías asistidas, especialmente las basadas en dispositivos móviles, que ya tienen cierta aceptación.

Figura 39.

Resultados pregunta 4

4. ¿Qué tipo de información sensorial (auditiva, táctil, olfativa) le resulta más útil para orientarse en su entorno?

29 respuestas



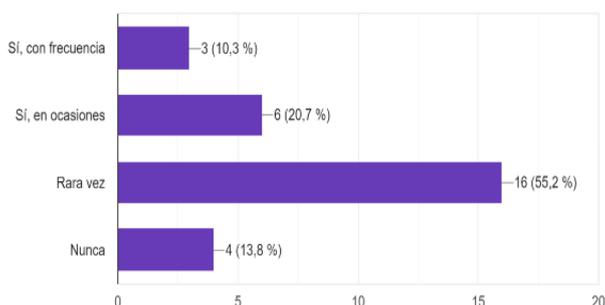
La mayoría de los encuestados (55.2%) encuentra más útil una combinación de varias señales sensoriales para orientarse en el entorno. Entre las señales individuales, la información auditiva es la más valorada (34.5%), seguida por la táctil (13.8%) y la olfativa (3.4%). Esto sugiere que las soluciones de accesibilidad más efectivas deberían incorporar múltiples tipos de información sensorial, con un énfasis particular en los estímulos auditivos cuando se consideren señales individuales

Figura 40.

Resultados pregunta 5

5. ¿Ha experimentado situaciones de riesgo o inseguridad debido a problemas de orientación espacial?

29 respuestas

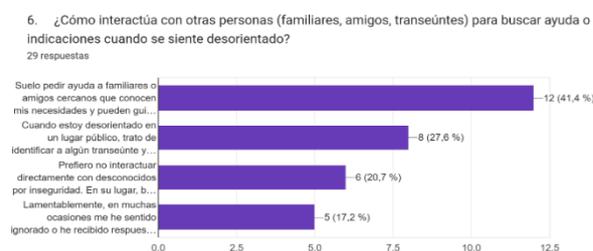


La gran mayoría de los encuestados (86.2%) ha experimentado alguna vez situaciones de riesgo o inseguridad debido a problemas de orientación espacial, aunque

con diferentes frecuencias. El 55.2% lo ha vivido rara vez, el 20.7% en ocasiones, y el 10.3% con frecuencia. Solo el 13.8% nunca ha enfrentado tales situaciones. Esto indica que los problemas de orientación espacial representan un riesgo real para la seguridad de la mayoría de las personas encuestadas, subrayando la importancia de mejorar la accesibilidad y las ayudas de orientación.

Figura 41.

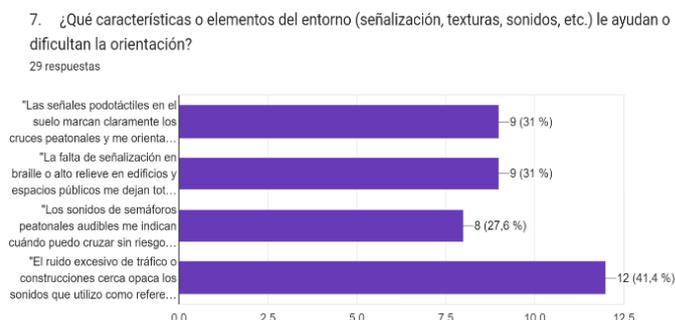
Resultados pregunta 6.



La mayoría de los encuestados (41.4%) prefiere pedir ayuda a familiares o amigos cercanos que conocen las necesidades del usuario. Un 27.6% intenta identificar a transeúntes en lugares públicos para pedir ayuda. Sin embargo, un 20.7% evita interactuar con desconocidos por razones de seguridad, y un 17.2% reporta experiencias negativas al pedir ayuda. Esto sugiere que, aunque la interacción social es una estrategia común para orientarse, existe una necesidad de mejorar la conciencia pública y la disposición a ayudar, así como de desarrollar alternativas seguras para quienes no se sienten cómodos interactuando con desconocidos.

Figura 42:

Resultados pregunta 7



Los elementos del entorno tienen un impacto significativo en la orientación de las personas encuestadas. El ruido excesivo es el mayor obstáculo, afectando al 41.4% al opacar sonidos de referencia. Las señales podotáctiles y la señalización en braille o alto relieve son igualmente importantes (31% cada una), ayudando y dificultando la orientación respectivamente cuando están presentes o ausentes. Los semáforos audibles también son útiles para el 27.6%. Esto sugiere que un diseño urbano inclusivo debería priorizar la reducción del ruido ambiental y la implementación consistente de señales táctiles y auditivas para mejorar la orientación.

9.5. Entrevista

Para conocer las dificultades y necesidades que enfrentan las personas con discapacidad visual en entornos urbanos, se realizó una entrevista a tres personas con esta condición. La entrevista abordó aspectos clave como los desafíos en la movilidad, estrategias de orientación, accesibilidad en el transporte público y el nivel de conocimiento de la sociedad sobre cómo brindar asistencia adecuada.

Los entrevistados compartieron sus experiencias y sugerencias para mejorar la infraestructura urbana y la inclusión de tecnologías accesibles. En la Tabla 1. Entrevista

sobre Movilidad Urbana y Discapacidad Visual, se presentan las preguntas y las respuestas recopiladas, destacando las coincidencias y diferencias en sus perspectivas.

Tabla 1.

Análisis de la entrevista

Pregunta	Respuesta (Entrevistados: 3 personas con discapacidad visual)
¿Cuáles son los principales desafíos que enfrentan al moverse por la ciudad?	Coincidieron en que la falta de infraestructura accesible es el mayor problema. Mencionaron aceras en mal estado, obstáculos en la vía pública (postes, motos estacionadas) y la ausencia de guías táctiles en el suelo. Además, señalaron que muchos semáforos carecen de señales sonoras, dificultando el cruce seguro de calles.
¿Cómo se orientan al caminar por la ciudad?	Los tres entrevistados utilizan el bastón blanco como herramienta principal para detectar obstáculos y cambios en el terreno. Dos de ellos emplean aplicaciones de navegación con indicaciones por voz (Google Maps, Lazarillo), y mencionaron que las baldosas táctiles cuando están disponibles ayudan significativamente. También destacaron que, en ciertas situaciones, piden ayuda a personas cercanas.
¿Cómo es su experiencia con el transporte público?	Explicaron que depende del tipo de transporte. En autobuses, la ausencia de anuncios sonoros de paradas dificulta saber cuándo descender. También indicaron que algunos conductores no detienen correctamente los vehículos o no esperan lo suficiente para permitirles abordar con seguridad.

<p>¿Creen que la gente está consciente de cómo ayudar a una persona con discapacidad visual en la calle?</p>	<p>Mencionaron que, aunque hay buena intención, muchas personas desconocen la manera adecuada de asistir. Explicaron que algunas personas intentan ayudar sin preguntar, lo que en ocasiones puede ser riesgoso. Coincidieron en que lo ideal es preguntar antes de ayudar y permitir que la persona con discapacidad visual indique cómo puede ser asistida.</p>
<p>¿Qué cambios consideran necesarios para mejorar la movilidad de las personas con discapacidad visual?</p>	<p>Sugirieron la implementación de más semáforos sonoros, guías táctiles en aceras y anuncios de voz en el transporte público. Además, destacaron la necesidad de educar a la población sobre cómo interactuar con personas ciegas en espacios públicos.</p>
<p>¿Qué mensaje darían a las personas con discapacidad visual que están aprendiendo a movilizarse solas?</p>	<p>Coincidieron en que es fundamental confiar en sus herramientas y habilidades. Aconsejaron aprender a usar el bastón blanco o un perro guía para ganar independencia. También recomendaron el uso de tecnología y enfatizaron la importancia de no dudar en pedir ayuda cuando sea necesario.</p>

9.6. Análisis de Observación directa

A través de las entrevistas realizadas a tres personas con discapacidad visual, se pudo conocer de cerca los desafíos que enfrentan al moverse por la ciudad y cómo

interactúan con su entorno. Uno de los aspectos que más llama la atención es la falta de infraestructura accesible, lo que se convierte en una barrera constante en su día a día. Aceras en mal estado, la ausencia de guías táctiles y la falta de semáforos sonoros no solo dificultan su autonomía, sino que también ponen en riesgo su seguridad al transitar por las calles.

Los entrevistados compartieron que, para orientarse, combinan estrategias tradicionales, como el uso del bastón blanco, con herramientas tecnológicas modernas, como aplicaciones de navegación con indicaciones de voz. Sin embargo, estas soluciones no siempre están al alcance de todos, ya que dependen de tener un dispositivo móvil y acceso a una conexión estable, algo que no siempre es posible.

En el caso del transporte público, la experiencia varía mucho según el medio utilizado, los autobuses presentan mayores dificultades debido a la falta de información accesible sobre las paradas y a la inconsistencia en la actitud de los conductores. Esto hace que desplazarse en ciertos medios sea todo un reto.

Otro aspecto relevante que surgió durante las entrevistas fue la interacción con otras personas en la calle. Aunque muchas personas tienen la intención de ayudar, no siempre saben cómo hacerlo de manera adecuada, lo que puede generar situaciones incómodas o incluso peligrosas para quienes tienen discapacidad visual. Esto pone en evidencia la necesidad de fomentar una mayor educación y sensibilización en la sociedad sobre cómo ofrecer ayuda de forma respetuosa y segura

A pesar de las dificultades, los entrevistados transmitieron un mensaje de optimismo y resiliencia. Destacaron la importancia de confiar en sus herramientas, en su capacidad de adaptación y en el aprendizaje constante. Además, resaltaron el papel clave de la tecnología y la importancia de no tener miedo a pedir ayuda cuando sea necesario

Este análisis no solo permite comprender mejor las experiencias de las personas con discapacidad visual, sino que también refuerza la urgencia de implementar soluciones inclusivas que les permitan moverse con autonomía, seguridad y dignidad por la ciudad.

9.7. Resumen

Este código utiliza un modelo de clasificación de imágenes para detectar objetos o escenas en tiempo real a través de la cámara web. Dependiendo de lo que detecte, reproduce un sonido asociado a esa clase. La lógica de contadores y umbrales evita que los sonidos se reproduzcan repetidamente, lo que lo hace útil para aplicaciones como asistentes de accesibilidad o sistemas de alerta.

Equipado con GPS, sensores ultrasónicos y cámaras, el sistema proporciona información en tiempo real sobre el entorno, mejorando la seguridad y orientación. A través de una interfaz auditiva o táctil, los usuarios podrán conocer su ubicación, planificar rutas e identificar obstáculos para un desplazamiento más seguro e independiente.

10. CRONOGRAMA DE ACTIVIDADES

En la siguiente tabla se muestra a detalle cada una de las actividades para el desarrollo del proyecto, distribuidas en el plazo antes mencionado.

Tabla 2.

Cronograma de actividades

Actividades	Octubre	Noviembre	Diciembre	Enero	Febrero
Investigación y análisis de requerimientos					
Revisión de literatura sobre tecnología de asistencia	x	x			

Entrevistas y encuestas a usuarios con discapacidad visual	x	x	
Definición de requerimientos funcionales y no funcionales	x	x	
Diseño del prototipo			
Selección de componentes electrónicos		x	
Diseños del circuito electrónico y arquitectura del software		x	
Diseño de la carcasa, forma del dispositivo e interfaz de usuario			x
Implementación del prototipo			
Programación del firmware y desarrollo de la aplicación móvil			x
Ensamblaje e integración de hardware y software			x
Pruebas unitarias de cada modulo			x
Integración y pruebas del prototipo			x
Pruebas de integración y del prototipo completo			
Refinamiento y corrección de errores			x
Evaluación con usuarios reales			
Pruebas de funcionamiento en escenarios reales			
Recopilación de métricas de usabilidad y rendimiento			

11. PRESUPUESTO

Tabla 3.

Presupuesto del proyecto

DESCRIPCIÓN	UNID.	VALOR
Sensor ultrasónico	1	\$ 2,50
Modulo GPS	1	\$20
Baterías recargables	1	\$8
Cámara web	1	\$20

Horas de ingeniería	80	\$500	
Raspberry pi 4	1	\$80	
TOTAL:		\$631	

12. CONCLUSIONES

La población con dificultad visual enfrenta serias dificultades en la coordinación espacial, principalmente debido a la falta de infraestructura y tecnología de asistencia. La investigación realizada a través de encuestas, entrevistas y observaciones directas permitió identificar cuestiones clave como la dificultad para detectar obstáculos durante el desplazamiento, la falta de señales táctiles y auditivas en los espacios públicos y la necesidad de herramientas tecnológicas adicionales para mejorar la movilidad.

Este prototipo de dispositivo móvil, equipado con sensores de proximidad, GPS y conectividad IoT, ofrece una solución innovadora para mejorar la orientación espacial. Su principal función es proporcionar alertas y asistencia en tiempo real mediante una guía de voz, facilitando la movilidad del usuario. No obstante, su efectividad dependerá de una interfaz intuitiva, una detección precisa de obstáculos y su capacidad de adaptación a diferentes entornos y condiciones.

El incremento de patrones de avance de señales para sensores y modelos de interacción de voz es importante para garantizar la precisión y la eficacia de los sistemas de apoyo a la navegación para individuos con dificultad visual. Este enfoque brinda la interpretación de datos en espacio real, la identificación de obstáculos y la adaptación de alertas y orientación a las necesidades específicas del usuario. La implementación de un modelo de voz intuitivo y comprensible mejorará el proceso de navegación, facilitando la toma de decisiones rápida y segura en condiciones dinámicas.

Evaluar la precisión, idoneidad y usabilidad de un prototipo de IoT es crucial para garantizar su eficacia en condiciones del mundo real. El análisis de estos aspectos nos permite identificar puntos fuertes, como la precisión en la detección de obstáculos y la utilidad del guiado en tiempo real, así como potenciales áreas de mejora en la interfaz de usuario y la adaptabilidad a diferentes situaciones. Esta revisión nos permite probar y perfeccionar el prototipo para retribuir a las necesidades de los individuos con dificultad visual, garantizando una experiencia más confiable y accesible.

13. RECOMENDACIONES

Es necesario desarrollar tecnologías basadas la programación para crear sistemas de asistencia a la navegación que satisfagan las especificaciones de las personas con dificultades visuales. Estas soluciones deberían incluir herramientas como sensores de detección de obstáculos, aplicaciones móviles con navegación por voz y mapas accesibles que muestren rutas optimizadas. Además, es de vital importancia involucrar activamente a la población con dificultad visual en el desarrollo y evaluaciones obtenidas en tecnologías para garantizar que las soluciones sean prácticas, intuitivas y orientadas al usuario.

El prototipo debe validarse mediante pruebas piloto con usuarios finales en entornos realistas y diversos para recopilar retroalimentación directa sobre su funcionalidad y eficacia. Además, se propone integrar opciones de personalización como configuraciones de volumen, idioma y niveles de sensibilidad del sensor en el dispositivo para satisfacer mejor las necesidades individuales. También es importante garantizar la ergonomía y la construcción ligera, así como brindar soporte técnico continuo para promover la implementación.

Es fundamental optimizar el algoritmo para garantizar un rendimiento eficiente en tiempo real, considerando las variaciones del entorno. Además, el modelo de interacción por voz debe diseñarse de manera que sea claro, completo y adaptable, permitiendo su uso en diversos idiomas y preferencias de los usuarios.

Para asegurar su funcionalidad, se sugiere la realización de pruebas exhaustivas en distintos escenarios, junto con la recopilación de retroalimentación directa de los usuarios finales. Esto permitirá mejorar la precisión y oportunidad de las advertencias e instrucciones, asegurando que sean fáciles de comprender y efectivas en la asistencia proporcionada.

Para obtener datos completos y significativos, las pruebas deben realizarse en una variedad de entornos representativos (interiores y exteriores) con un grupo diverso de usuarios finales. Además, se debe recopilar retroalimentación cualitativa a través de entrevistas y encuestas para complementar los datos técnicos. Esto ayudará a identificar oportunidades para mejorar la funcionalidad del sistema y la experiencia del usuario.

14. BIBLIOGRAFÍA

Álvarez, R. (15 de noviembre de 2019). weWALK, el bastón para invidentes equipado con Google Maps, altavoz y que se puede enlazar a un smartphone. Obtenido de

xataka: <https://www.xataka.com/wearables/wewalk-baston-para-invidentes-equipado-google-maps-altavoz-que-se-puede-enlazar-a-smartphone>

Amazon. (s/f). ¿Qué es IoT (Internet de las cosas)? Recuperado el 26 de febrero de 2024, de [amazon.com: https://aws.amazon.com/es/whatis/iot/#:~:text=El%20t%C3%A9rmino%20IoT%2C%20o%20Internet,como%20entre%20los%20propios%20dispositivos](https://aws.amazon.com/es/whatis/iot/#:~:text=El%20t%C3%A9rmino%20IoT%2C%20o%20Internet,como%20entre%20los%20propios%20dispositivos).

Briones, C. (24 de Septiembre de 2023). ingenieriapedia. Obtenido de <https://ingenieriapedia.com/guia-completa-de-sensores-ultrasonicos/>

Carmenate, J. G. (15 de Agosto de 2023). Programarfacil. Obtenido de <https://programarfacil.com/esp8266/programar-esp32-ide-arduino/>

Carranza, S. (27 de octubre de 2021). CONOCIENDO AL ESP32. Obtenido de TodoMaker: <https://todomaker.com/blog/conociendo-al-esp32/>

COLLANTES RODRÍGUEZ, B. D. (2020). DISEÑO DE UN PROTOTIPO DE UN BASTÓN INTELIGENTE CON SENSORES PARA PERSONAS CON DISCAPACIDAD VISUAL Y FÍSICA. 16. Obtenido de <https://repositorio.unesum.edu.ec/bitstream/53000/2249/1/COLLANTES%20RODRIGUEZ%20BRYAN%20DAMIAN.pdf>

Cruz, M. C. (2014). Proyectos de Automatización. Valencia: Introducción al proyecto y documentos del proyecto.

Espinoza, C. (13 de Marzo de 2023). SmartDomotica. Obtenido de <https://www.smartdomotica.es/gafas-inteligentes/>

Fernández, R. S. (2018). Seminario de automatización de procesos y procesos fisher technik. Apuntes de la asignatura Laboratorio de automatización y control, (pág. 25). Valencia.

Gandhi, M. (27 de Noviembre de 2019). Qué es un sensor ultrasónico y para qué sirve. Obtenido de Autoycom: <https://www.autycom.com/que-es-un-sensor-ultrasonico-y-para-que-sirve/>

Ghosh, A. (16 de Mayo de 2020). THE CUSTOMIZE WINDOWS. Obtenido de <https://thecustomizewindows.com/2020/05/esp32-vs-arduino-how-esp32-is-different-from-arduino/>

González Gelvez, S. &. (2020). Sistema de estantería inteligente basado en tecnologías móviles e IOT para personas con discapacidad visual. Revista científica , 14,15. Obtenido de https://repository.unab.edu.co/bitstream/handle/20.500.12749/13710/2020_Tesis_Stefanny_Gonzalez_Gelvez.pdf?sequence=1

Hernández, P. (20 de febrero de 2024). ¿Qué es una GoPro? – La cámara para viajes de aventura. Obtenido de filmora: <https://filmora.wondershare.es/family-n-business/make-adventure-travel-video-with-gopro.html>

Herrero, J. (2017). Apuntes de la asignatura laboratorio de automatización y control. En J. Herrero, Apuntes de la asignatura laboratorio de automatización y control (pág. 45). Valencia: Identificación y control con LabVIEW.

Jacob, T. P. (2021). An AI-Powered Smart Camera for Object Detection. In 2021 6th International Conference on Communication and Electronics Systems . revista científica, 3. Obtenido de <https://www.researchgate.net/profile/Prem>

Jacob/publication/355842999_An_AI-Powered_Smart_Camera_for_Object_Detection/links/64e500a50453074fbda78bec/An-AI-Powered-Smart-Camera-for-Object-Detection.pdf

LLamas, L. (1 de Abril de 2018). LLI. Obtenido de <https://www.luisllamas.es/esp32/>

Lumpina, L. (25 de enero de 2023). LAS PILAS RECARGABLES. Obtenido de narasolar: <https://www.narasolar.com/las-pilas-recargables/>

Magnetism. (26 de octubre de 2023). Módulos GPS. Obtenido de Electricity - Magnetism.: <https://www.electricity-magnetism.org/es/modulos-gps/>

Martinez, A. (25 de Septiembre de 2022). La vanguardia. Obtenido de <https://www.lavanguardia.com/comprar/comparativas/comparativa-mejores-gafas-inteligentes/>

Moreno, E. G. (2019). Herramientas de modelado para sistemas de eventos discretos. Valencia: GRAFCET Y GEMMA.

Naranjo, M. (31 de Julio de 2022). Computer Hoy. Obtenido de <https://computerhoy.com/reportajes/tecnologia/gafas-inteligentes-2022-modelos-diferencias-guia-compra-1079853>

Orlando, C. &. (2018). Automatización industrial. Manual prácticas de laboratorio, (pág. 22). Javeriana.

Perez, J. (18 de Junio de 2023). Polaridad. Obtenido de <https://polaridad.es/que-es-un-microcontrolador/>

- Pesante, k. (27 de septiembre de 2023). Meta presenta gafas inteligentes que hacen llamadas y transmiten en vivo. . Obtenido de Primicias.: <https://www.primicias.ec/noticias/tecnologia/meta-gafas-inteligentes-llamadas/>
- Pineda, C. (27 de DICIEMBRE de 2023). SEGURILATAM. Obtenido de https://www.segurilatam.com/actualidad/sensores-infrarrojos-tipos-funciones_20231227.html
- Reinoso, V. (18 de Julio de 2022). IBM. Obtenido de <https://www.ibm.com/mx-es/topics/>
- Revista seguridad 360. (27 de Mayo de 2022). Sensor infrarrojo: Tipos y aplicaciones. Obtenido de revista seguridad 360: <https://revistaseguridad360.com/destacados/sensor-infrarrojo/>
- Reynolds, D. (25 de Enero de 2024). Xecor. Obtenido de <https://www.xecor.com/blog/rp2040-vs-esp32>
- Salazar, R. D. (2019). Dispositivos de asistencia para la movilidad en personas con discapacidad visual:. Revista Politécnica,, 2. Obtenido de <file:///C:/Users/hp/Downloads/jegiraldo,+1534-4307-3-LE.pdf>
- Sanchez, O. (19 de Octubre de 2019). CONNECT CENTER. Obtenido de <https://aphconnectcenter.org/que-es-la-orientacion-y-la-movilidad/?lang=es>
- Suntaxi Cantuña, O. G. (2019). Reseña de la placa Raspberry Pi 3 2019. REVISTA SEGURA , 6. Obtenido de <https://www.automatizacionparatodos.com/resenas/raspberry-pi-3/>

Systems, E. (12 de Marzo de 2024). GitHub. Obtenido de <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>

Toala, S. (15 de Junio de 2020). SENSORES ONLINE. Obtenido de <https://sensores.online/guia-completa-de-los-sensores-ultrasonicos/>

Viviana., Y. (2020). Sistema de estantería inteligente basado en tecnologías móviles e IOT para personas con discapacidad visual. 2020. revista científica, 32. Obtenido de https://repository.unab.edu.co/bitstream/handle/20.500.12749/13710/2020_Tesis_Stefanny_Gonzalez_Gelvez.pdf?sequence=1&isAllowed=y

Zavala, E. (27 de Marzo de 2017). El Comercio. Obtenido de <https://www.elcomercio.com/afull/baston-personas-novidentes->

15. ANEXOS

Figura 43.

Base para la cámara del prototipo

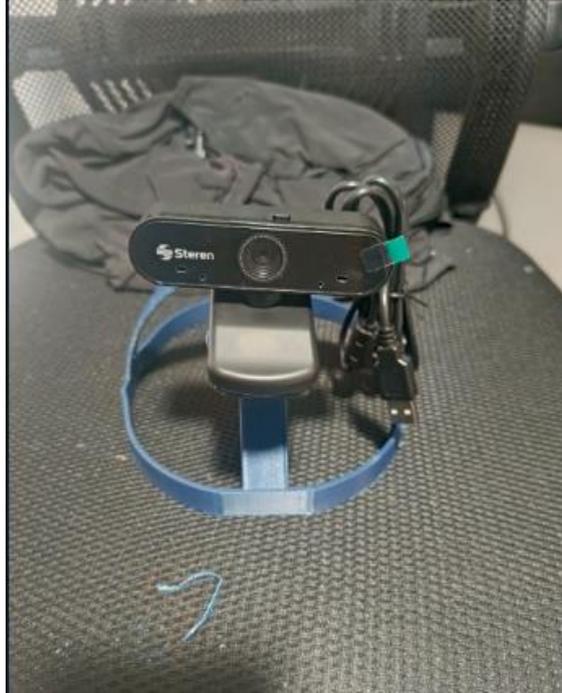


Figura 44.

Código utilizado en Raspberry pi Thonny

CODIGO UTILIZADO PARA ARDUINO CLOUD

```
from arduino_iot_cloud import ArduinoCloudClient
import time
import logging
import json
import sys
import random
import pynmea2
import serial
from arduino_iot_cloud import Location
from arduino_iot_cloud import Task
sys.path.append("lib")

port = serial.Serial("/dev/ttyUSB0", baudrate=9600, timeout=1)
DEVICE_ID = b"7cbbde68-d940-40c1-9226-6ef7e92ee814"
SECRET_KEY = b"qv#Rm828j7n1GCLlm2MVs@AAp"
def gps_temperatura(client, args):
    try:
        line = port.readline().decode('ascii', errors='replace')
        print(line)
        if line.startswith("$GPGGA") or line.startswith("$GPRMC"):

            msg = pynmea2.parse(line)
            print(msg)
            lat = msg.latitude
            lon = msg.longitude
            client.register(Location("gps2", lat=lat, lon=lon))
    except pynmea2.ParseError:
        logging.warning("Error al parsear datos del GPS.")
    except Exception as e:
        logging.error(f"Error al leer datos del GPS: {e}")

temp = int(random.uniform(0, 100))
client.register("TEMPERATURA", value = temp )

if __name__ == "__main__":
    try:
        logging.basicConfig(
            datefmt="%H:%M:%S",
            format="% (asctime)s.%(msecs)03d %(message)s",
            level=logging.INFO,
        )
        client = ArduinoCloudClient(device_id=DEVICE_ID,
            username=DEVICE_ID, password=SECRET_KEY)
        client.register(Task("gps_temperatura",
            on_run=gps_temperatura, interval=1.0))
        client.start()
    except Exception as e:
        logging.error(f"Error al enviar datos de ubicacion: {e}")
    while True:
        client.update()
        time.sleep(0.100)
```

Figura 45. Código utilizado en Pj5

CODIGO UTILIZADO EN PJ5

```
// Classifier Variable
let classifier;

// Model URL
let imageModelURL = 'https://teachablemachine.withgoogle.com/models/v8ew3dAT/';

// Video
let video;
let flippedVideo;

// To store the classification
let label = "";

// Variables for detecting specific labels and playing sound
let stairCount = 0;
let stairCount2 = 0;
let stairCount3 = 0;
let stairCount4 = 0;
let stairCount5 = 0; // Nueva variable para "Paredes"
const stairThreshold = 4;
let soundPlayed = {
  Escaleras: false,
  'Semaforo verde': false,
  'Semaforo rojo': false,
  'Paso cebra': false,
  Paredes: false // Nuevo sonido asociado a "Paredes"
};

// Delay between classifications (in milliseconds)
const classificationDelay = 100; // 300 milliseconds

// Load the model first
```

```

    soundPlayed['Semaforo verde'] = true;
}
} else if (label === "Semaforo Rojo") {
    stairCount3++;
    if (stairCount3 >= stairThreshold && !soundPlayed['Semaforo Rojo']) {
        alarmSound3.play();
        soundPlayed['Semaforo Rojo'] = true;
    }
} else if (label === "Paso cebra") {
    stairCount4++;
    if (stairCount4 >= stairThreshold && !soundPlayed['Paso cebra']) {
        alarmSound4.play();
        soundPlayed['Paso cebra'] = true;
    }
} else if (label === "Paredes") { // Nueva condición para "Paredes"
    stairCount5++;
    if (stairCount5 >= stairThreshold && !soundPlayed.Paredes) {
        alarmSound5.play();
        soundPlayed.Paredes = true;
    }
} else {
    // Reset counts and soundPlayed if label doesn't match
    stairCount = 0;
    stairCount2 = 0;
    stairCount3 = 0;
    stairCount4 = 0;
    stairCount5 = 0; // Reinicio para "Paredes"
    soundPlayed = {
        Escaleras: false,
        'Semaforo verde': false,
        'Semaforo rojo': false,

```

```
'Paso cebra': false,  
  Paredes: false // Reinicio de "Paredes"  
};  
}  
  
// Add a delay before classifying the next frame  
setTimeout(classifyVideo, classificationDelay);
```