

Diseño y creación de un prototipo guía para población con discapacidad visual basado en el uso de
inteligencia artificial llamado blind light

Carlos Andrés Mesa Ordoñez
Cod. 68172526

Marco Giovanni Gómez Barrera
Cod. 62182513

Johan Camilo Herreño López
Cod. 67172519

Corporación Universitaria Unitec

Escuela de Ingeniería / Escuela de Ciencias Económicas y Administrativas

Ingeniería Sistemas / Ingeniería Telecomunicaciones / Administración de Empresas

Bogotá D.C

Noviembre de 2021

Tabla de contenido

1	INTRODUCCIÓN	1
2	PLANTEAMIENTO DEL PROBLEMA	3
3	OBJETIVOS	5
3.1	OBJETIVO GENERAL.....	5
3.2	OBJETIVOS ESPECÍFICOS	5
4	JUSTIFICACIÓN	6
5	DELIMITACIÓN O ALCANCE	8
5.1	PREGUNTAS DE INVESTIGACIÓN	8
6	MARCO TEÓRICO	9
6.1	ANTECEDENTES	9
6.1.1	<i>Prototipos</i>	9
6.1.2	<i>Estudios Previos</i>	11
6.2	DISCAPACIDAD VISUAL.....	12
6.2.1	<i>Niveles de discapacidad visual</i>	12
6.2.2	<i>Causas Principales de la ceguera</i>	13
6.2.3	<i>Algunas enfermedades que causan ceguera</i>	14
6.2.4	<i>Discapacidad visual en Colombia</i>	14
6.3	ACERCA DE ANDROID.....	15
6.3.1	<i>¿Qué es Android?</i>	15
6.3.2	<i>Arquitectura Android</i>	16
6.3.3	<i>Kernel de Linux</i>	16
6.3.4	<i>Capa de abstracción de hardware (HAL)</i>	17
6.3.5	<i>Tiempo de ejecución de Android</i>	17

6.3.6	<i>Bibliotecas C/C++ nativas</i>	18
6.3.7	<i>Apps del sistema</i>	18
6.3.8	<i>SDK Android</i>	18
6.3.9	<i>Librerías y recursos para una aplicación Android</i>	19
6.3.10	<i>Tensorflow</i>	20
6.3.11	<i>Text To Speech</i>	21
6.4	¿QUÉ ES OPEN SOURCE?	21
6.4.1	<i>Licencia BSD</i>	21
6.5	ACERCA DE LA INTELIGENCIA ARTIFICIAL	22
6.5.1	<i>Historia Inteligencia Artificial</i>	22
6.5.2	<i>Tipos Inteligencia Artificial</i>	22
6.5.3	<i>Redes neuronales</i>	23
6.5.4	<i>Arquitectura redes neuronales</i>	26
6.5.5	<i>Tipos de aprendizaje redes neuronales</i>	26
6.6	PROTOTIPO ESPERADO	27
6.6.1	<i>Raspberry PI</i>	27
6.6.2	<i>Arnés</i>	29
7	MARCO METODOLÓGICO	29
7.1	GENERALIDADES	29
7.2	DISEÑO DE LA INVESTIGACIÓN	30
7.3	ENFOQUE DE LA INVESTIGACIÓN	30
7.4	POBLACIÓN	31
7.5	FUENTES DE INFORMACIÓN	32
7.6	DESARROLLO DE LA SOLUCIÓN	32
7.6.1	<i>Listado de actividades</i>	32
7.6.2	<i>Diagramación</i>	33

7.6.3	<i>Casos de uso</i>	34
7.6.4	<i>Diagrama de Clases</i>	38
7.6.5	<i>Lenguaje de programación</i>	38
7.6.6	<i>Selección IDE</i>	38
7.6.7	<i>Investigación librerías</i>	39
7.6.8	<i>Desarrollo módulos</i>	39
7.6.9	<i>Integración de los módulos o servicios</i>	52
7.6.10	<i>Testeo</i>	53
7.7	PLAN DE TRABAJO	53
7.7.1	<i>Matriz DOFA</i>	53
7.7.2	<i>Diagrama de Gantt</i>	54
7.7.3	<i>Plan de trabajo</i>	55
7.8	ANÁLISIS Y DISCUSIÓN DE RESULTADOS	56
7.8.1	<i>Home</i>	56
7.8.2	<i>Detección objetos</i>	57
7.8.3	<i>Ubicación GPS</i>	63
7.8.4	<i>Guía GPS</i>	65
7.8.5	<i>Todo en Uno</i>	68
8	CONCLUSIONES Y RECOMENDACIONES	69
9	LISTA DE REFERENCIA O BIBLIOGRAFÍA	70
11	ANEXOS	76
12	CESIÓN DE DERECHOS	80

Lista de figuras

Figura 1 Prototipo sónico de ayuda a la ceguera en uso	9
Figura 2 Theia-Prototipo de guía robótico y portátil	10
Figura 3 Bastón We Walk	11
Figura 4 T-Mobile G1, así era el primer teléfono Android del mundo	15
Figura 5 Arquitectura de Android	16
Figura 6 Comparativa red neuronal humana versus virtual	23
Figura 7 Estructura red neuronal	25
Figura 8 Tipos de aprendizaje	26
Figura 9 Raspberry pi	27
Figura 10 Arnés	29
Figura 11 Flujo del desarrollo de la aplicación	33
Figura 12 casos de uso	34
Figura 13 Diagrama de clases	38
Figura 14 Android Studio 4.2	38
Figura 15 Configuración API de Google Maps	40
Figura 16 Registro llave API	41
Figura 17 Dependencias	41
Figura 18 Llamado a la app de Google Maps	42
Figura 19 Registro dependencias en build.grade	42
Figura 20 Registro en el MAIN	42
Figura 21 Cargue modelo entrenado	43
Figura 22 Cargue OpenCV	43
Figura 23 Dependencia OpenCV	44

Figura 24 Método onCreate	45
Figura 25 Método onBackPressed	45
Figura 26 Inicialización OpenCV	46
Figura 27 Método onDestroy	46
Figura 28 Interacción objetos	47
Figura 29 Objeto Visión Máquina	47
Figura 30 Cargue modelo	47
Figura 31 OnCameraFrame	47
Figura 32 Bloque de código para reconocimiento de objetos	48
Figura 33 Excepción Try - Catch	49
Figura 34 Código Text To Speech	49
Figura 35 Ciclo FOR	49
Figura 36 Permisos GPS	50
Figura 37 Actualización localización	50
Figura 38 Método setLocation	51
Figura 39 Método adicionales	51
Figura 40 Clase Location	52
Figura 47 Identificación objeto "Señal de PARE"	59

Lista de tablas

Tabla 1 Casos de uso Ingreso a la aplicación	35
Tabla 2 Caso de uso Ubicación GPS	35
Tabla 3 Caso de uso Guía GPS	36
Tabla 4 Casos de uso Texto a voz	36
Tabla 5 Caso de uso Reconocimiento objetos	37

Resumen

Es habitual que los seres humanos se adapten al entorno donde convive, ajustando así los hábitos y la forma de interactuar con los diferentes componentes que lo rodean, sin embargo, hay variables que no se toman en cuenta para integrar a cualquier individuo de la sociedad, por ejemplo, la ciudad de Bogotá no está diseñada para que personas con algún tipo de discapacidad se puedan movilizar con facilidad, a esto se le suma las limitadas herramientas que les sirven de apoyo para su desplazamiento. En la actualidad se ha comenzado a pensar un poco más en esta población, existen diferentes proyectos de estado que desean brindar un mejor estilo de vida a esta comunidad, grandes compañías de software en sus proyectos de elaboración de desarrollo de aplicación incluyen a la población con discapacidad visual como uno de los ítems que se deben revisar previo al desarrollo de una aplicación para no excluirlos y que puedan llegar a disfrutar de los beneficios de la tecnología sin importar el grado de discapacidad.

Dando uso de algunas tecnologías crearemos una herramienta de apoyo para la población con discapacidad visual, a los cuales se les entregará la primera interacción de un prototipo que les permita guiarse a través de audio durante su desplazamiento y haciendo uso de inteligencia artificial se identificara e informara algunos de los obstáculos comunes que se encuentran en un trayecto.

Abstrac

It is common for human beings to adapt to the environment where they live, thus adjusting the habits and the way of interacting with the different components that surround it, however, there are variables that are not taken into account to integrate any individual in society, For example, the city of Bogotá is not designed so that people with some type of disability can move easily, in addition to the limited tools that support them in their displacement. At present, we have begun to think a little more about this population, there are different state projects that want to provide a better lifestyle to this community, large software companies in their application development elaboration projects include the population with visual impairment as one of the items that should be reviewed prior to the development of an application so as not to exclude them and that they can enjoy the benefits of technology regardless of the degree of disability.

Using some technologies, we will create a support tool for the visually impaired population, which will be given the first interaction of a prototype that allows them to be guided through audio during their movement and using artificial intelligence to identify and inform some of the common obstacles encountered on a course.

Glosario

Amaurosis: Pérdida de la visión por problemas en la circulación de sangre hacia la retina, generando ceguera temporal o permanente.

API: Es la vinculación de distintas definiciones y reglas que se emplean para desarrollar y unir el software de las aplicaciones.

ART: Es el android runtime una máquina virtual de android, su entorno de ejecución de aplicaciones por medio de la compilación de códigos.

Glaucoma: Es una enfermedad que se genera en el ojo acusa de un aumento de presión dentro del globo ocular, generando un daño progresivo en la retina del ojo causando la pérdida de la visión.

HAL: Es la capa de abstracción de hardware, que trabaja como una interfaz entre en software y hardware del sistema.

Linux: Es un sistema operativo de código abierto creado para computadora, servidores, dispositivos móviles entre otros, que nos permite el funcionamiento del hardware de los dispositivos.

OnCreate: Se emplea para toda clase de iniciaciones tales como creaciones de interfaz creación de usuarios o la preparación de organizaciones de datos.

onBackPressed: por medio de esta función nos permite devolverá la función o tarea anterior según hayamos programado.

OpenCV: Es una librería libre de visión artificial que tiene como significado open computer vision

Prototipo: Es el inicio de un invento, cosa o idea que tiene como función ser el modelo inicial, con el fin de buscar una mejora continua del mismo.

Retinopatía diabética: Es una enfermedad causada por la diabetes que genera un problema grave en los ojos a causa de la degeneración de los vasos sanguíneos del tejido que es sensible a la luz que se encuentra en la parte interior de los ojos.

Redes neuronales: Es el modelo simplificado en el cual emula el modo en el que el cerebro humano procesa la información por medio de neuronas que están conectadas entre sí y además estas trabajan en conjunto sin que haya una tarea específica para cada una.

Raspberry pi: Es un ordenador que tiene un tamaño de una tarjeta bancaria, que soporta diversos ordenadores que nos permite desarrollar diferentes funciones como procesadores de texto, hojas de cálculo entre otras funciones.

Sistema robótico: Es un conjunto de sistemas conformado por varios robots además de otras fuentes y dispositivos que se unen de una manera integrada para dar un funcionamiento.

Scrum: Son lineamiento de trabajo que se emplean para el desarrollo de una actividad de manera colaborativa y en equipo con el objetivo de obtener mejores resultados.

SDK Android: Es una la unión de diversas herramientas y librerías de desarrollo de software que son indispensables para el diseño de las aplicaciones en android. Ya que esta herramienta permite que el proceso de desarrollo transcurra sin ningún inconveniente.

Tensorflow: Es una librería de códigos abiertos que aprendizaje autónomo a través de rangos de tareas impuestas.

1 Introducción

Según la Organización Mundial Para La Salud (OMS,2020) “en el mundo hay aproximadamente 2.200 millones de personas con alguna discapacidad visual, este tipo de discapacidad se muestra por una disminución total o parcial de la vista”.

Por otro lado, el Instituto Nacional Para Ciegos (INCI,2018) “en Colombia los resultados del último censo realizado por el DANE en el año 2018 arrojó como resultado 1.948,332 casos de personas con algún grado de discapacidad visual que representan al 62,17% del total de discapacitados del país”. Los cuales se ven limitados diariamente en su movilidad debido a las dificultades que se les presenta en sus trayectos a pie, ya que no cuentan con las herramientas tecnológicas necesarias, además de la falta de señalización braille en medio de sus recorridos, dificultando su ubicación, debida a que no es común que los locales comerciales cuenten con estas señalizaciones braille ni tecnología de apoyo para las personas con discapacidad visual.

Según INCI (2018) “esta población para realizar actividades cotidianas como caminar por la calle, desplazarse al lugar de trabajo requieren de habilidades para adquirir técnicas de orientación y movilidad que les permita ejercer su autonomía e independencia”.

El estallido de la industria 4.0 ha traído grandes avances tecnológicos como lo es la inteligencia artificial, dicha tecnología es capaz de adquirir conocimiento según el entrenamiento y configuraciones que se apliquen, definiendo dentro de sus algoritmos instrucciones puede llegar a tomar decisiones de forma autónoma para notificar o realizar algún tipo de acción, un ejemplo de esto es Watson, el cual fue diseñado por IBM haciendo uso de inteligencia artificial, actualmente puede predecir y apoyar la toma de decisiones de las organizaciones que lo adquieren (IBM Watson, s. f.).

Las redes neuronales entrenadas pueden realizar tres tipos de acciones, aprender, donde a través del entrenamiento puede adaptarse a las condiciones que se les estén dando, generalizar, donde pueden llegar a dar respuestas estandarizadas y por último abstraer, donde puede analizar por separado cada uno de los objetos sin importar la cantidad de entradas. (Olabe, X. B, 1998)

Hay dos tipos de entrenamiento aplicables a redes neuronales, el primero corresponde al aprendizaje supervisado, donde se le entrega al modelo las entradas y salidas predeterminadas, logrando así que genere predicciones más acertadas, mientras que en el aprendizaje no supervisado únicamente se le entrega al modelo las características para que realice el análisis y predicciones. (Quesada et al., 1994)

2 Planteamiento del Problema

Las personas con discapacidad visual cuentan con pocas herramientas que los apoyen durante sus desplazamientos, quizá la herramienta más conocida es el bastón, de acuerdo a su color podemos identificar el tipo de discapacidad, blanco con punta roja para personas ciegas, blanco con líneas rojas para personas que adicional a la discapacidad visual presenta sordera y por último el bastón verde, para quienes presentan baja visión. (INCI, s.f)

Sumado a esto, se encuentra que la infraestructura de la ciudad no está diseñada para apoyar el desplazamiento de esta población, por ejemplo, no se cuenta con marcación en los andenes y en las paradas de transporte público, además algunos transeúntes se encargan de terminar de entorpecer los desplazamientos.

Cuando se habla de personas con discapacidad visual en Bogotá, encontramos que dada la infraestructura de la ciudad y de los paradigmas de la comunidad y de los empresarios, son sometidos a la condición de excluidos sociales, esto es otro agravante de su limitación visual. En la actualidad se encuentra que su inclusión se ve truncada dado que las instituciones educativas no cuentan con las herramientas y capacidades para su formación integral. Se encuentra también que es difícil que accedan a servicios de rehabilitación por motivos económicos. Con esto se puede hablar que tienden a entrar en un ciclo de pobreza, al no poder acceder a rehabilitación es más difícil que accedan a la educación, por ende, se dificulta su acceso al empleo y como consecuencia no cuenta con alternativas que le garanticen una vida digna, agravando así su condición de exclusión dada su discapacidad visual. (Bustos Ramírez, 2014).

De acuerdo al Espectador (2008) "las personas con discapacidad insisten en que la ciudad es 'agresiva' con ellos. Aseguran que la gente aún no ha aprendido a convivir con personas diferentes y que

los gobiernos capitalinos no se han preocupado por crear una ciudad más fácil para quienes tienen limitaciones”.

No es común encontrar establecimientos de comercio que en sus fachadas cuenten con algún mecanismo que le informé a las personas con discapacidad visual el nombre del lugar en el que se encuentran, como lo exige la constitución en el Artículo 2. “Las autoridades de la República están instituidas para proteger a todas las personas residentes en Colombia, en su vida, honra, bienes, creencias, y demás derechos y libertades, y para asegurar el cumplimiento de los deberes sociales del Estado y de los particulares.” es por esto que el Congreso de la República de Colombia mediante el Proyecto de ley NO.035 2020 Cámara está tramitando la exigencia de la inclusión del lenguaje braille sobre algunos de los productos alimenticios, plaguicidas de uso domésticos, aseo, medicamentos y servicios turísticos, así como de los sitios de interés de carácter público a las personas en condición de discapacidad visual “ciegas o con baja visión” (Rosario Maria,2020).

3 Objetivos

3.1 Objetivo General

Diseñar una aplicación para controlar el prototipo guía, que con el apoyo de inteligencia artificial le facilite a la población con discapacidad visual el desplazamiento desde la salida sur de la estación de Transmilenio Calle 76 hasta la entrada de la sede F de UNITEC.

3.2 Objetivos Específicos

- Desarrollar una aplicación para dispositivos Android.
- Guiar por medio de voz del dispositivo móvil al usuario desde la salida sur de la estación hasta la entrada de la sede F
- Indicar al usuario la dirección en la que se encuentra al momento de iniciar la aplicación.
- Indicar por medio de voz el objeto detectado.

4 Justificación

En el inicio de este documento se hizo énfasis en los problemas más concurrente a los cuales se enfrentan la población con discapacidad visual, como miembros activos de la Corporación Universitaria UNITEC hemos evidenciado que en el trayecto de la salida sur de la estación “Calle 76” hasta la sede F de la Universidad no se cuenta con variedad de recursos que brinden un apoyo a esta población, adicional a esto a nivel ciudad algunas de las pocas herramientas que se encuentran instaladas están presentando fallas, como por ejemplo, según Vargas (2018) en Bogotá han sido instalados solamente 153 semáforos sonoros, cuando deberían estar ubicados en las 1.340 intersecciones viales con semáforos, sin embargo, tras la visita a varias zonas del centro y las localidades de Teusaquillo y Chapinero, se pudo comprobar que, aunque los dispositivos están instalados no cuentan con sonido o simplemente no funcionan.

En la actualidad existen algunos avances tecnológicos que otorgan apoyo como lo es “Jaws y ZoomText”, este es un software que amplía hasta 16 veces el tamaño de las letras en pantalla y permite variar color y contraste, beneficiando a personas con baja visión o que estén empezando a experimentar problemas visuales por cuestiones de edad. (ZOOMTEXT, s. f.), este software tiene costo, pero según Adriana Pulido la primera colombiana con discapacidad visual en obtener la beca Fulbright y líder de proyectos que facilitan el acceso a la tecnología para personas ciegas durante una entrevista de revista semana el proyecto CONVERTIC del Ministerio de las TIC permite es que las personas con discapacidad visual y las entidades puedan acceder a la licencia de Jaws y ZoomText, programas usados por las personas con baja visión. Es una licencia que en el mercado puede costar unos cuatro millones de pesos y en Colombia es gratuita. Desde que se inició en diciembre de 2013, muchos países se han interesado en replicarlo porque definitivamente ha sido de gran ayuda para muchas personas que en la

vida nunca imaginaron usar un computador. Hoy en día soy beneficiaria de ese proyecto, y espero que siga funcionando. Es un gran avance porque Colombia fue pionera y porque ha logrado resultados importantes en la inclusión en el campo educativo, laboral y cultural. (Echeverri, 2020)

En el mercado hay dispositivos de apoyo como lo son “Envision Glasses” tecnología de Google, el cual es capaz de describir un entorno y llegar a reconocer objetos gracias al apoyo de la inteligencia artificial, el costo puede variar entre los 1700 – 2100 dólares que aproximadamente en pesos colombianos son entre 6.521.200 – 8.055.600, con esto se complica su acceso a personas de escasos recursos. (Envision-glasses, s. f.)

El ser humano se caracteriza por su instinto de supervivencia y su constante curiosidad, es así que se hace necesario que se le brinde distintas herramientas y alternativas que le permitan hacer su vida más fácil o llevaderas, en aras de suplir esta necesidad se desarrollará este prototipo de interacción, teniendo en cuenta que el mercado actual no cuenta con muchas alternativas que puedan apoyar la movilidad de personas con algún tipo de discapacidad visual.

5 Delimitación o alcance

Población con discapacidad visual que tenga la necesidad de dirigirse desde la estación de Transmilenio calle 76 hasta la sede F de la corporación universitaria UNITEC.

Se diseñará una aplicación que haga uso de módulos GPS, cámara, sonido y procesamiento de nuestro prototipo para identificar puntos de referencia en el trayecto, y haciendo uso de la inteligencia artificial logre detectar objetos en su trayecto como se especifican en el siguiente apartado, la aplicación es para teléfonos con sistema operativo Android, pero el código será desarrollado en lenguaje java lo cual facilita su adaptación para ser ejecutado en una tarjeta como Raspberry.

Objetos a detectar:

- Personas
- Vehículos
- Perros
- Señal de PARE
- Semáforo
- Boca de incendio (hidrante)

5.1 Preguntas de investigación

¿Existen desarrollos tecnológicos en Android para reconocer objetos?

¿Qué tan costoso es un perro guía?

¿La población con discapacidad visual usa Smartphone?

¿Existen modelos pre entrenados de inteligencia artificial?

¿El entrenamiento de redes neuronales puede brindar mayor seguridad en el desplazamiento?

6 Marco Teórico

6.1 Antecedentes

Con la evolución de la tecnología en los últimos años se han desarrollado diferentes dispositivos tecnológicos que brindan a personas con discapacidad visual un apoyo en medio de su movilidad facilitando sus desplazamientos cotidianos en medio de sus ciudades. A continuación, alguno de los más destacados.

6.1.1 Prototipos

Prototipo sónico: Este dispositivo tiene como propuesta tecnológica un procesador de estereovisión que mide la diferencia de imágenes capturadas por dos cámaras, calculando la distancia de cada punto captado por las cámaras, por medio de un código de sonido que informa la posición y distancia de los posibles obstáculos (UC3M, 2013).

Figura 1 Prototipo sónico de ayuda a la ceguera en uso



Nota: La imagen representa el prototipo del dispositivo sónico. Tomado de:http://noticiasdelaciencia.com/not/9286/nuevo_prototipo_sonico_de_ayuda_a_la_ceguera/

Theia - prototipo de guía robótico y portátil: Este es un dispositivo portátil tiene como objetivo copiar las funciones de un perro guía, por medio de la programación de rutas seleccionadas evitando accidentes de los usuarios durante el trayecto, aunque por el momento es solo un prototipo su funcionamiento es basado en la combinación de tecnología Lidar con un par de cámaras para “Theia” pueda detectar la imagen tridimensional de su entorno, evitando que el usuario se choque con otras personas entre otras cosas. (Branco, A, 2020).

Figura 2 Theia-Prototipo de guía robótico y portátil



Nota: la imagen representa el modelo del prototipo, tomado de https://www.lespanol.com/omicrono/hardware/20200803/dispositivo-portatil-ciegos-jubilara-perro-guia/509450331_0.html

Bastón We Walk: Este prototipo de bastón tiene como objetivo ayudar en medio del desplazamiento a personas con discapacidad visual en medio de las calles ya que cuenta con tecnología

que le permite por medio de un sensor detectar obstáculos guiando al usuario y evitando tropiezo en medio de sus recorridos diarios (Mourenza, A, 2020).

Figura 3 Bastón We Walk



Nota: la imagen muestra el bastón We Walk y su creador, tomado de https://elpais.com/elpais/2020/01/24/eps/1579868789_604708.html

6.1.2 Estudios Previos

- Diseño e implementación de un aplicativo que reconozca el valor de los billetes para discapacitados visuales usando un smartphone con sistema operativo Android (Peña Suarez & Hurtado Panchi, 2017).
- Módulo asistente de invidentes para el sistema de transporte Transmilenio (Varón & Diaz, 2018)

6.2 Discapacidad Visual

Según A.G.I (2016); la discapacidad visual es el término que se refiere a cualquier tipo de problema visual grave, que afecta la apreciación de imágenes limitando a la persona que la padece. ocasionado por patologías congénitas, accidentes o provocados por algún tipo de virus.

6.2.1 Niveles de discapacidad visual

Deficiencia Visual: es una disminución importante de la vista, pero que le permite al portador ver la luz y orientarse. Estas personas tienen una visión funcional pero reducida y sin contornos definidos lo que los lleva a no tener la visión completa, limitado sus actividades cotidianas. (Puntodis, 2019).

Ceguera: es la pérdida total de la vista, que no se puede corregir, las personas que sufren de esta enfermedad solo recolectan información auditiva y táctil. (Puntodis, 2019).

Ceguera parcial: vista parcial que permite la orientación a la luz además de la percepción de masas, facilitando considerablemente el desplazamiento y la comprensión del mundo exterior. Visión de cerca que es insuficiente para la utilización de la vida escolar o profesional (AGI, 2016).

Amaurosis: corresponde a la pérdida súbita de visión en uno de los dos ojos debido a la falta de circulación de sangre en la retina. Existen dos tipos de amaurosis bilateral que afecta a los dos ojos de la persona y unilateral que afecta solo un ojo (Área oftalmológica avanzada, 2020).

6.2.2 Causas Principales de la ceguera

La OMS (2019) Nos dice que las principales causas del aumento de los casos de deficiencia visual o de pérdida de visión se puede ver relacionado a las cataratas, el tracoma y los errores de refracción. Por otro lado, no se puede pasar por alto las afecciones oculares que no suelen afectar a la visión, como la xeroftalmía y la conjuntivitis, ya que esta son una de las principales razones por las que las personas acuden a los servicios de atención oftalmológica en todos los países.

6.2.3 Algunas enfermedades que causan ceguera

- 6.2.3.1 Cataratas:** las cataratas son una opacidad de la lente natural o cristalino del ojo que se encuentra detrás del iris y la pupila, las cataratas son la causa más común de pérdida de visión en personas mayores de 40 años y es la causa principal de ceguera en el mundo. (All About Vision, 2019)
- 6.2.3.2 Retinopatía diabética:** es una complicación de la diabetes y una de las causas de la ceguera, ocurre cuando la diabetes daña los vasos sanguíneos de la retina que es el tejido sensible a la luz situado en la parte posterior del ojo (National eye institute, 2019).
- 6.2.3.3 Glaucoma:** se produce gradualmente, cuando el ojo no drena el fluido tan bien como debería similar a un drenaje atascado. Como resultado, la presión del ojo aumenta y empieza a dañar el nervio óptico. Este tipo de glaucoma no es doloroso y no causa ningún cambio en la visión al inicio (Boyd, K, 2021).
- 6.2.3.4 Desprendimiento de retina:** es cuando la retina se desprende de la parte posterior del ojo, quitando su funcionamiento. (Boyd,k, 2021)

6.2.4 Discapacidad visual en Colombia

Según el INCE (2018) en el último censo realizado por el DANE en el año 2018 arrojó un resultado de acuerdo a la escala de medición de la discapacidad e 1'948.332 personas con discapacidad visual equivalente al 62.17% de la población con discapacidad en Colombia, de un total de 3'134.036 personas con discapacidad, en general equivalente al 7.1% de la población colombiana. Es decir, que la población con discapacidad subió del 6.4% al 7.1% del total de la población en Colombia; por otro lado,

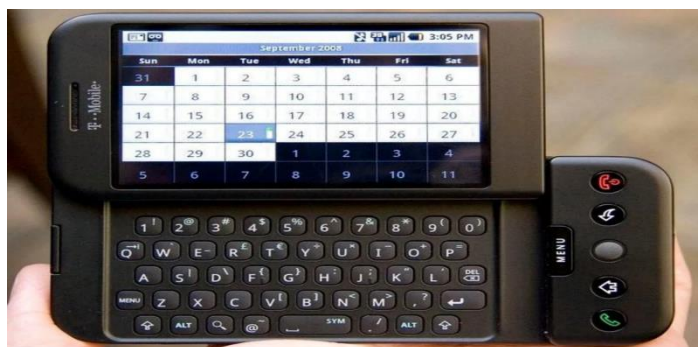
la población con discapacidad visual pasó del 43% de prevalencia de toda la población con alguna discapacidad 6.4%, subiendo al 62.17% en el actual Censo 2018 con una cifra de 1'948.332 personas con discapacidad visual.

6.3 Acerca de Android.

6.3.1 ¿Qué es Android?

Sistema operativo que nace cerca del año 2005, este fue desarrollado por la compañía Android Inc., la cual fue adquirida por Google en el mes de julio de este mismo año. La versión 1.0 fue instalada en 2008, en un smartphone T-Mobile G1, fabricado por en conjunto por HTC y Google. (Android OS, 2012)

Figura 4 T-Mobile G1, así era el primer teléfono Android del mundo



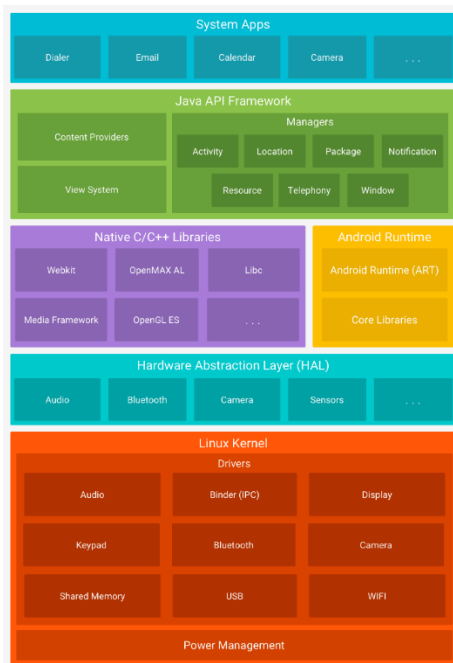
Nota: El grafico representa el primer teléfono con Android, tomado de

<https://andro4all.com/2018/05/t-mobile-g1-primer-android>

6.3.2 *Arquitectura Android*

Android, hace parte del software de código libre, el cual está basado en LINUX, fue diseñado para diferentes dispositivos, en la figura 5, encontraremos los componentes de este. (Developers, mayo 2020)

Figura 5 Arquitectura de Android



Nota: El grafico representa lo que compone el sistema operativo Android, tomado de

<https://developer.android.com/guide/platform?hl=es-419>

6.3.3 *Kernel de Linux*

Hace parte de los componentes base de Android, donde le permite hacer uso de funciones básicas como la generación de subprocesos junto con administración de memoria de bajo nivel. Al hacer

uso del kernel se habilitan opciones primordiales de seguridad, brindándole así la facilidad de desarrollar controles de hardware a los fabricantes de dispositivos. (Developers, 2020)

6.3.4 Capa de abstracción de hardware (HAL)

Esta capa le permite al marco de trabajo de la API de Java de la parte superior trabajar con las capacidades de hardware del dispositivo. La HAL consiste en diferentes módulos que permiten trabajar con los diferentes componentes de hardware, por ejemplo, para nuestro proyecto haremos uso de la cámara. Es así que podemos decir cuando la API realiza el llamado para hacer uso de uno de los compones de Hardware, Android carga el módulo correspondiente a ese componente. (Developers, 2020)

6.3.5 Tiempo de ejecución de Android

En los dispositivos con versiones Android 5.0 en adelante, las aplicaciones ejecutan sus propias instancias del tiempo de ejecución de Android (ART). Estas instancias están diseñadas con el fin de poder trabajar con varias máquinas virtuales en dispositivos de memoria baja haciendo uso de archivos DEX, estos están en formato de código de bytes, con el objetivo de ocupar el mínimo espacio posible de memoria. El ART crea cadenas de herramientas, donde se compilan las fuentes de Java en código DEX.

En Android, se incluyen bibliotecas de ejecución central que brindan parte de la funcionalidad del lenguaje de programación Java. (Developers, 2020)

6.3.6 Bibliotecas C/C++ nativas

Varios componentes base de Android se basan en código nativo que hace uso de bibliotecas en lenguaje C y C++, por ejemplo, los componentes ART y HAL. Android, proporciona la API del marco de trabajo de Java que le permite ver algunas de las bibliotecas nativas de las aplicaciones. Por ejemplo, podemos acceder a OpenGL haciendo uso de la API de OpenGL de Java, este hace parte del marco de trabajo de Android, el cual permite agregar a la aplicación la compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D. (Developers, 2020)

6.3.7 Apps del sistema

Android incluye un conjunto de aplicaciones centrales que permiten el uso de funcionalidades como el correo electrónico, la mensajería de texto, el calendario, la navegación en Internet, los contactos, entre otros. Las aplicaciones que se incluyen en la plataforma no tienen una prioridad entre las aplicaciones que el usuario elige instalar; es por esto, que una aplicación externa se puede llegar a convertir en el navegador web, el sistema de mensajería SMS o el teclado predeterminado del usuario, sin embargo, se presentan algunas excepciones, como la aplicación de ajustes del sistema. (Developers, 2020)

6.3.8 SDK Android

Google desarrolló el SDK de Android este es un paquete de desarrollo de software para la plataforma Android, el cual le permite crear Android de manera sencilla, sin tener conocimiento avanzados de desarrollo. (Vaati, 2020)

6.3.9 *Librerías y recursos para una aplicación Android*

A continuación, se describen las librerías que nos ayudan a desarrollar el tipo de aplicación propuesta

Descripción general del SDK de Maps para Android. Este SDK permite agregar mapas a la aplicación diseñada para Android, incluidas las aplicaciones para dispositivos corporales basados en Android los cuales utilizan datos, mapas y respuestas gestuales de Google Maps. Adicionalmente pueden ofrecer información sobre las ubicaciones del mapa, agregando marcadores, polígonos y superposiciones facilita así la interacción con el usuario. (Developers, s.f).

Servicios de Google Play que contienen un amplio conjunto de API y servicios para Android que apoyaran la compilación de la aplicación, mejorando así la privacidad y seguridad. (Developers, s.f.).

Geocodificador. Clase que permite manejar la codificación geográfica y la codificación geográfica inversa. La codificación geográfica permite transformar una dirección postal o descripción de una ubicación en una coordenada que incluya la latitud y longitud, mientras que la codificación inversa, permite transformar las coordenadas en una dirección parcial. En esta última la información que contiene puede variar, por ejemplo, la dirección de un edificio cercano, o el nombre de la ciudad y su código postal. (Developers, s.f.)

OpenCV. Open Source Computer Vision Library más conocido como OpenCV , es una biblioteca de software de visión artificial y aprendizaje automático de código abierto. OpenCV se creó para proporcionar una infraestructura común para aplicaciones de visión de máquina, con el fin de acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD, OpenCV facilita que las empresas utilicen y modifiquen el código.

La biblioteca contiene alrededor de 2500 algoritmos que ya se encuentran optimizados, incluyen un conjunto de algoritmos de aprendizaje automático y visión por computadora clásicos, al igual que los de última generación. Con estos algoritmos se puede llegar a detectar y reconocer rostros, identificar objetos estándar, clasificar las acciones humanas en videos, rastrear los movimientos de la cámara y objetos en movimiento, permite extraer modelos 3D de objetos, producir nubes de puntos 3D a partir de cámaras estéreo, permite unir imágenes para producir una alta resolución, imagen de una escena completa, busca imágenes similares en una base de datos de imágenes, edición de fotos al eliminar los ojos rojos cuando son tomadas con flash, sigue movimientos oculares, reconocer paisajes y establecer marcadores para superponerlos con realidad aumentada, etc. OpenCV tiene una comunidad más de 47 mil de usuarios, junto con un estimado de descargas superior a 18 millones. (OpenCV)

Otro plus es que esta librería es multiplataforma, facilitando así su ejecución en diferentes sistemas operativos como Windows, MacOS y Linux, este último tiene como ventaja que se puede aplicar en tarjetas como lo son raspberry para creación de micro proyectos, otra de sus ventajas es su compatibilidad con lenguajes de programación como C++, Python, Java y Matlab. Al tratarse de software libre, se puede encontrar diversa documentación y apoyo en la comunidad. (OpenCV, s.f)

6.3.10 **Tensorflow.**

TensorFlow es una plataforma de código abierto de extremo a extremo, cuya función principal es el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad, que permite que los usuarios innoven con el aprendizaje automático y donde los desarrolladores pueden crear e implementen aplicaciones con tecnología de aprendizaje automático de manera sencilla. (TensorFlow, s.f)

6.3.11 Text To Speech.

Text to speech, hace referencia a la síntesis del habla (speech synthesis) o una producción artificial del habla. El sistema utilizado para este propósito se conoce como “speech computer” o “speech synthesizer”. Suele usarse cuando no es posible o no es conveniente leer directamente de una pantalla debido a algún tipo de impedimento o cuando simplemente se quiere mejorar la interfaz de usuario al tener esta tecnología como una opción disponible, usualmente usado en plataformas de e-learning, lectura de noticias mientras el usuario va conduciendo un vehículo, en videojuegos, como asistentes personales como los que se encuentran en los smartphones, etc. (IBM, s.f)

6.4 ¿Qué es Open Source?

Movimiento que surge en la década de 1980, desde ahí ha adquirido una creciente importancia. Este es diferente al software gratis (freeware) aunque es relevante he importante, aunque es de distribución gratuita es de código cerrado. Software de código abierto, es todo el software que permita su utilización para cualquier fin, estos no cuentan con restricción de copias, de acceso al código fuente, ni al estudio de su funcionamiento, permite la adaptación conforme las necesidades de cada usuario, dando la opción de difundir copias a terceros de las modificaciones realizadas. (Meirinhos, 2009).

6.4.1 Licencia BSD

La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Licencia de software libre permisiva como la licencia de OpenSSL o la MIT License, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al

dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. (Licencia BSD, 2015)

6.5 Acerca de la inteligencia Artificial

6.5.1 Historia Inteligencia Artificial

Los primeros acercamientos a la inteligencia artificial datan de los años 40, sin embargo, toma relevancia la publicación “Computing machinery and intelligence” realizada por Alan Turing en la revista Mind, en dicha publicación se plantea la pregunta ¿pueden las máquinas pensar? nos da como introducción a este tipo de planteamientos, donde se busca conocer o entender si las máquinas pueden interactuar con las personas o si solamente pueden llegar a imitarles (Meseguer y López, 2017). En la búsqueda de dar respuesta a esta pregunta, Turing plantea un ejercicio donde una persona interactúa con un entrevistador, en este se busca que la persona identifique si quien da respuesta a sus preguntas es una persona o una máquina, planteando algoritmos complejos (Vidal, 2007). Con este planteamiento comienzan a surgir más trabajos donde se busca dar respuesta a la pregunta planteada por Turing, por ejemplo, se planteó un tablero de ajedrez donde se esperaba que la máquina fuera capaz de resolver un juego de ajedrez evaluando cada uno de los movimientos posibles, en dicho entrenamiento, la máquina debía conocer todas las reglas básicas y adicionalmente todos los movimientos posibles con cada una de las fichas.

6.5.2 Tipos Inteligencia Artificial

Podemos encontrar dos tipos de inteligencia artificial, una débil donde solamente se puede llegar a replicar a modo de espejo la forma en que actual el interlocutor y una fuerte donde esta puede llegar a pensar y procesar información de manera similar a lo que haría un ser humano. También las

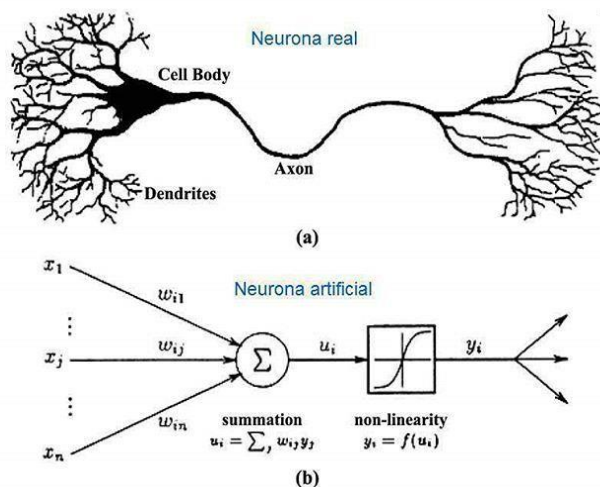
podríamos clasificar de acuerdo al tipo de acciones que esta realice: simbólica, opera con representaciones abstractas basados en lógica matemática, buscando resolver teoremas básicos; conexionista, toma como ejemplo las redes neuronales del cerebro, basadas en entradas y salidas; evolutiva, programación evolutiva que busca la adaptación de los programas a nuevas necesidades; y por último corpórea, el cuerpo da forma a la inteligencia, se tiene en cuenta el hardware al momento de programar (Meseguer y López, 2017).

En la actualidad uno de los ejemplos más avanzados corresponde a Watson, creado por IBM en 2011, el cual es considerado una supercomputadora, capaz de sostener una conversación, con lo cual va aprendiendo y adaptándose a nuevos escenarios, analizando los datos tiene la capacidad predecir información, actualmente es comercializada como una solución integradora para apoyo de las organizaciones, la cual es capaz de adaptarse a las necesidades de dicha organización. Por su parte Apple desarrolló SIRI, mostrando parte de experiencia de aprendizaje automático.

6.5.3 *Redes neuronales*

Al hablar de inteligencia artificial se debe comenzar a hablar de redes neuronales, estas simulan el funcionamiento de una red neuronal humana, con cada una de las ramificaciones y nodos.

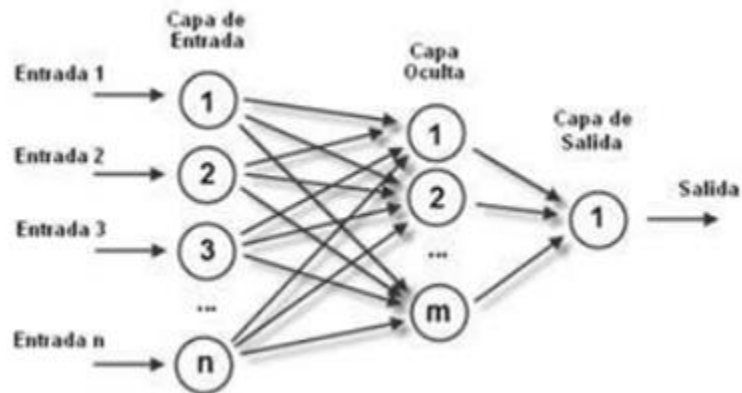
Figura 6 Comparativa red neuronal humana versus virtual



Nota: la imagen representa el comparativo de una neurona biológica con su símil artificial, tomado de <https://pin.it/3SJ5xj2>

Como se puede ver en la figura 6, las redes neuronales cuentan con unas entradas que simulan las dendritas, el razonamiento lógico se realiza en lo que sería el núcleo y la salida de la neurona hace referencia al axón. En las redes neuronales artificiales los núcleos son conocidos como capas donde se van realizando cada uno de los procesos de la información, de acuerdo a la necesidad de entrenamiento se pueden adicionar la cantidad de capas que sean necesarias.

Figura 7 Estructura red neuronal



Nota: la imagen representa la estructura general de una red neuronal con sus distintas divisiones, tomado de <https://sites.google.com/site/lainteligenciaartificialunesum/redes-neuronal>

Entradas, neuronas encargadas de recibir la información del exterior de la red; la capa oculta, se encarga de realizar el procesamiento de la información; la capa de salida corresponde a las neuronas que transfieren la información procesada por la capa oculta y es entregada en la salida.

Como ventaja principal se puede encontrar que las redes neuronales se pueden entrenar de manera paralela con el objetivo de obtener resultados más certeros y específicos, asemejando el funcionamiento del cerebro humano, donde este obtiene conocimiento a través redes neuronales y entre sus conexiones guardan la información, haciendo uso de entradas y salidas, con un módulo de procesamiento (Caicedo y López, 2009)

6.5.4 Arquitectura redes neuronales

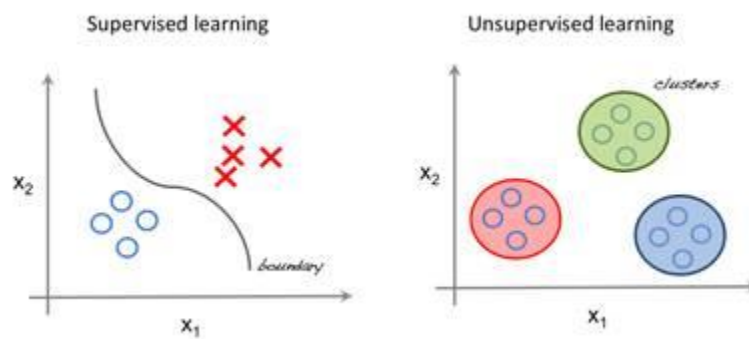
Las redes neuronales se puede clasificar de acuerdo a su arquitectura, hablamos de monocapa donde solamente cuentan con una capa que sirve de transporte, no se evidencia de procesamiento de información; multicapa, donde la información se va transportando por distintas capas hasta lograr procesar al máximo el detalle de la información; feedforward, cuando solamente se encuentra un sentido al flujo de la información que se ha manejado, en este tipo de redes no se manejan ciclos, datan de 1980; por último las redes recurrentes, donde nos permite trabajar distintos ciclos entre sus capas intermedias (Sossa, 2021).

6.5.5 Tipos de aprendizaje redes neuronales

No supervisado: se le entrega a la red neuronal un dato de entrada para ser procesado sin indicarle que tipo de respuesta debe dar, este tipo de aprendizaje no hace uso de patrones.

Supervisado: se le entrega a la red neuronal el dato de entrada y una salida sugerida para que sea comparada con el resultado de su proceso, en este se debe definir un porcentaje de error de entrenamiento para garantizar su éxito (Quesada, Graciani, Bonal, & Díaz-Mata, 1994)

Figura 8 Tipos de aprendizaje



Nota: la imagen representa los tipos de aprendizaje de las redes neuronales, tomado de <https://programmerclick.com/article/34221021205/>

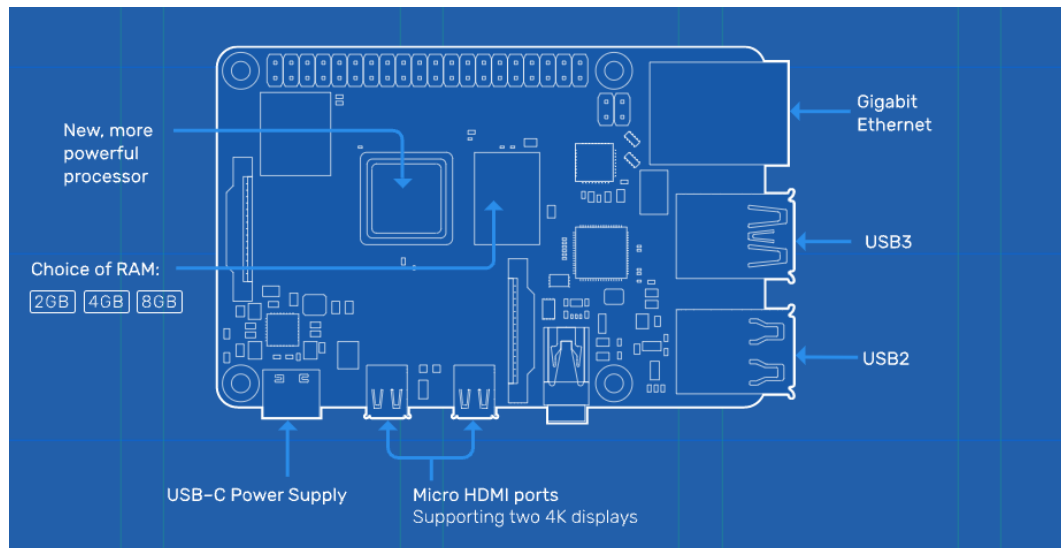
6.6 Prototipo Esperado

6.6.1 *Raspberry PI*

Miniordenador u ordenador compacto, fue lanzado en 2012 por Raspberry Pi Foundation, es usado para desarrollo de prototipos por ser livianos y adaptables a distintos tipos de lenguajes de programación, en nuestro caso es compatible con Java y el desarrollo en Android Studio sobre el que se viene trabajando como solución final (Ltd, s. f.).

En la siguiente figura encontraremos la arquitectura estándar de una Raspberry PI, donde se pueden encontrar los distintos puertos Ethernet, USB, HMDI, RAM y procesador.

Figura 9 Raspberry pi



Nota: En la figura se encuentra detallada la arquitectura de una Raspberry pi, tomado de:

<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>

Accesorios Raspberry Pi: con el fin de brindar una herramienta que le permita al usuario realizar el desplazamiento y servirle de guía, se requiere adicionar a la Raspberry Pi 4 los siguientes elementos:

- Cámara, esta le permitirá visualización a nuestro prototipo, logrando así identificar objetos.
- Módulo GPS, le permitirá identificar las coordenadas de ubicación del usuario, adicional le identificar la ruta trazada.
- Conectores auriculares, le permitirá la salida de audio y la guía a través de voz.

6.6.2 Arnés

Dado que el objetivo principal del prototipo es brindar una herramienta que le facilite el desplazamiento al usuario de Blind Light, se hará uso de un arnés que permita mantener libres las manos de este.

El prototipo se conectará en la parte frontal del usuario, a la altura del pecho, con el fin de poder identificar los elementos que se encuentran al frente de este. Tomando como ejemplo el arnés de las cámaras GoPro que permiten al usuario grabar sin necesidad de ocupar sus manos.

Figura 10 Arnés



Nota: modelo arnés estándar para cámara GoPro, tomado de: <https://acortar.link/02H2yw>

7 Marco Metodológico

7.1 Generalidades

Balestrini (2000) no dice que el marco metodológico se refiere a “el conjunto de procedimientos a seguir con la finalidad de lograr los objetivos de la información de forma válida y con una alta precisión” (p44); En otras palabras, el marco metodológico es la estructura o lineamientos para la

recolección de la información requerida de una forma ordenada que nos permite interpretar los resultados en función del problema planteado de la investigación.

7.2 Diseño de la investigación

Para el desarrollo de este proyecto se usará la investigación aplicada “la cual tiene por objetivo resolver un determinado problema o planteamiento específico, enfocándose en la búsqueda y consolidación del conocimiento para su aplicación y, por ende, para el enriquecimiento del desarrollo cultural y científico.” (Biblioteca DUOC UC, 2018)

7.3 Enfoque de la investigación

El presente proyecto se realizará bajo el planteamiento metodológico del enfoque cualitativo, puesto que esta metodología es la que mejor se adapta a las características y necesidades de la investigación, ya que el enfoque del proyecto es recolectar información de distintas fuentes, que nos permita la elaboración del aplicativo y su mejora continua.

El enfoque cualitativo nos modela un proceso inductivo contextualizado en un ambiente natural, esto se debe a que en la recolección de datos se establece una relación entre los participantes de la investigación donde se busca validar sus experiencias e ideologías en detrimento del uso de un instrumento de medición pre establecido. En este enfoque, las variables no se definen con el objetivo de manipularlas experimentalmente, y con esto nos indica que se analiza una realidad subjetiva, además de tener una investigación cuyo objeto es único por ende de difícil réplica, dado que no cuenta con fundamentos estadísticos. Este enfoque se caracteriza también porque no es necesaria una conceptualización de las preguntas de investigación, por ende, no se presenta una reducción a números de las conclusiones sustraídas de los datos, además busca sobre todo la difusión de la información en

comparación con el enfoque cuantitativo, el cual busca delimitarla. Con el enfoque cualitativo se tiene una gran variedad de ideas e interpretaciones que permite enriquecer el fin de la investigación. El alcance final del estudio cualitativo busca comprender un fenómeno social complejo, más allá de realizar la medición de las variables involucradas, se busca en sí es llegar a entenderlo. (Hernández, 2006)

7.4 Población

Según Tamayo (2012) la población es la totalidad de un fenómeno de estudio, incluye la totalidad de unidades de análisis que integran dicho fenómeno y que debe cuantificarse para un determinado estudio integrando un conjunto N de entidades que participan de una determinada característica, y se le denomina la población por constituir la totalidad del fenómeno adscrito a una investigación

Acorde con las estadísticas del 2018, el país cuenta con 1'948.332 personas que presentan dificultad para ver de lejos, ver de cerca o alrededor, Los datos reflejados permiten inferir que 1'885.773, es decir aquellas personas que realizan las actividades con alguna o mucha dificultad, corresponderían a personas con baja visión y 62'559 a personas ciegas, basados en que no pueden realizar las actividades visuales. (Herrera M 2018)

Según la secretaria distrital de planeación, en Bogotá hay más mujeres que hombres con discapacidad dedicadas a los oficios del hogar. En la ciudad existen 458.088 personas con discapacidad estos se encuentran en 176.150 hogares, esto representa el 6,41% del total de la población de la ciudad. Al validarlas por género encontramos que el 56,6% son mujeres y el 43,4% son hombres. También se encuentra que el 70,8% de las mujeres con discapacidad es menor de 45 años, mientras que los hombres con discapacidad el 62,6% se encuentran en el rango de 45 años o más. Al mirarlo a nivel de

estratos sociales, el 53% de este grupo hace parte de los estratos 1 y 2; sin embargo, las que viven en estrato 2 representan el 43,4%. (Manceras 2020).

El estudio presenta otras cifras de importancia como la identificación de las localidades donde reside un mayor porcentaje de personas con discapacidad en Bogotá, que en su orden son Kennedy con un 14,37%; Engativá con un 12,34%; Suba con 11,67%; y Bosa con un 9,21%; San Cristóbal con un 6,72% y Ciudad Bolívar 6,67%. (Manceras 2020).

7.5 Fuentes de información

- Documentación de la comunidad de desarrolladores.
- Documentación oficial de las librerías y herramientas a usar.
- Desarrollos compartidos al público en repositorios como Github.

7.6 Desarrollo de la solución

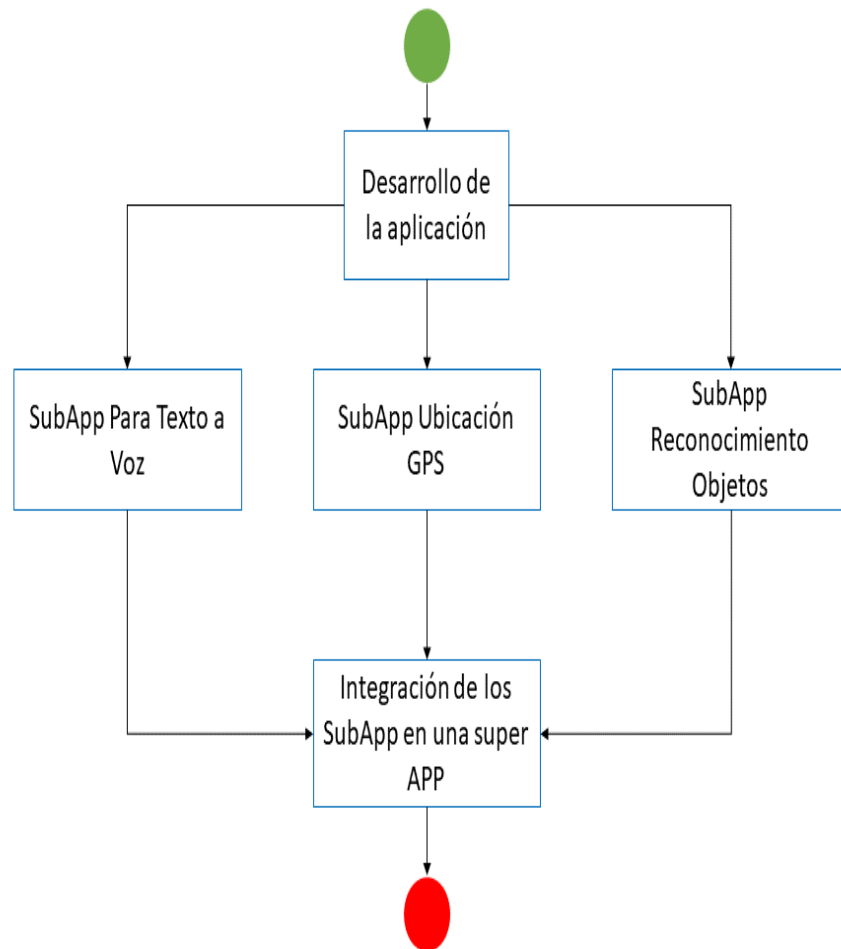
7.6.1 *Listado de actividades*

- Diagramación
- Seleccionar lenguaje de programación.
- Seleccionar IDE para el desarrollo.
- Investigar las librerías a usar.
- Desarrollo de módulos:
 - Reconocimiento objetos
 - Ubicación GPS
 - Texto a voz

- Integración de los módulos o servicios desarrollados.
- Testeo y ajustes de la aplicación.

7.6.2 Diagramación

Figura 11 Flujo del desarrollo de la aplicación



7.6.3 Casos de uso

Figura 12 casos de uso

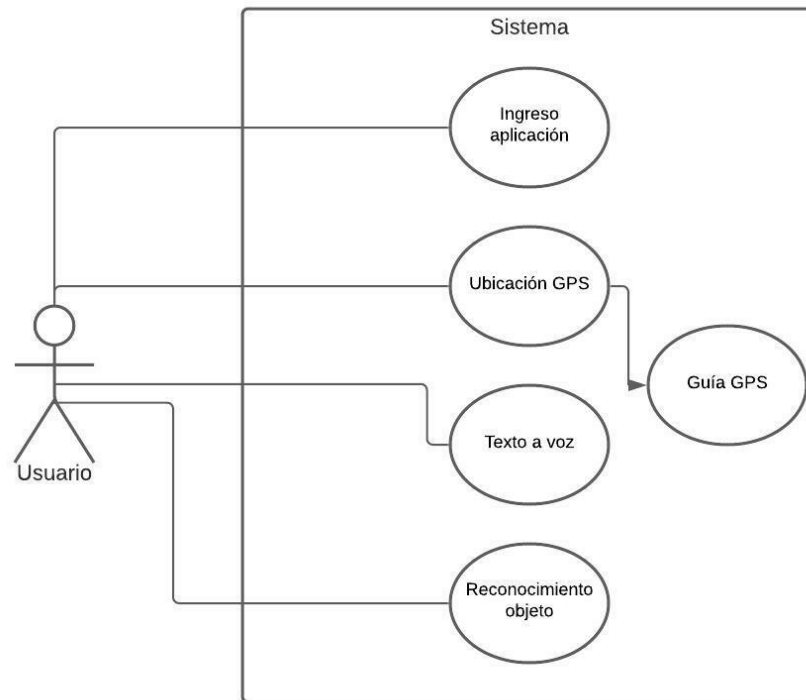


Tabla 1 Casos de uso Ingreso a la aplicación

Caso de uso	Ingreso a la aplicación	
Actor	Usuario	
Dependencia	N/A	
Precondición	Aplicación sin iniciar	
Descripción	La aplicación deberá iniciarse sin requerir ningún tipo de registro o logueo	
Flujo normal	Paso	Actividad
	1	El usuario inicia la aplicación
	2	Aplicación inicia el home o pantalla principal
Flujo alternativo	N/A	
Postcondición	Aplicación debe indicar punto de partida	
Excepciones	Errores de ejecución aplicación	

Tabla 2 Caso de uso Ubicación GPS

Caso de uso	Ubicación GPS	
Actor	Usuario	
Dependencia	Servicio GPS habilitado	
Precondición	Aplicación iniciada	
Descripción	La aplicación deberá indicarle al usuario la dirección en la que se encuentra	
Flujo normal	Paso	Actividad
	1	Usuario ingresa al inicio de la aplicación
	2	Aplicación indica la dirección en la que se encuentra, haciendo uso de las coordenadas entregadas por el GPS

Flujo alternativo	N/A
Postcondición	Aplicación debe indicar la dirección en la que se encuentra
Excepciones	Errores de ejecución del GPS

Tabla 3 Caso de uso Guía GPS

Caso de uso	Guía GPS	
Actor	Usuario	
Dependencia	Servicio GPS habilitado	
Precondición	Punto de partida reconocido por el GPS	
Descripción	La aplicación debe indicarle la ruta por la cual deberá dirigirse a la sede F de UNITEC	
Flujo normal	Paso	Actividad
	1	Aplicación debe reconocer la dirección o punto de partida
	2	Aplicación validar la ruta de posible desplazamiento a pie.
	3	Aplicación debe indicar la ruta de desplazamiento
Flujo alternativo	N/A	
Postcondición	Aplicación debe indicar la ruta de desplazamiento	
Excepciones	Errores de ejecución del GPS	

Tabla 4 Casos de uso Texto a voz

Caso de uso	Texto a voz	
Actor	Usuario	
Dependencia	N/A	
Precondición	Aplicación iniciada	
Descripción	La aplicación a través del dispositivo deberá indicarle a través de audio el punto donde se encuentra, la ruta y objetos que identifique	
Flujo normal	Paso	Actividad
	1	Usuario ingresa al inicio de la aplicación

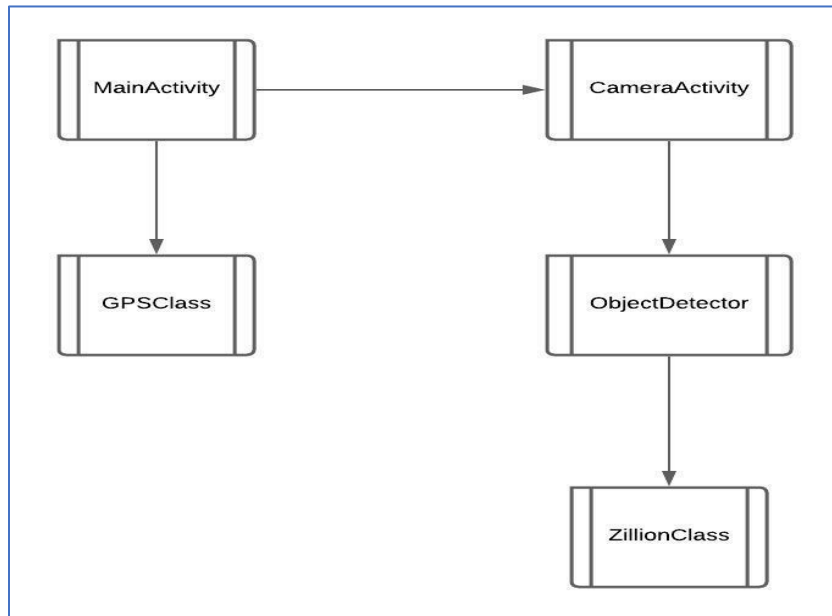
	2	Notificación ubicación actual
	3	Notificación ruta de desplazamiento
Flujo alternativo	N/A	
Postcondición	Aplicación debe indicar ruta a seguir para llegar a la Sede F UNITEC	
Excepciones	Falla de audio del dispositivo o manos libres	

Tabla 5 Caso de uso Reconocimiento objetos

Caso de uso	Reconocimiento objetos	
Actor	Usuario	
Dependencia	Cámara funcional	
Precondición	Aplicación iniciada	
Descripción	La aplicación deberá poder reconocer objetos comunes durante el desplazamiento del usuario	
Flujo normal	Paso	Actividad
	1	Aplicación traza la ruta de desplazamiento
	2	Aplicación identifica objetos
	3	A través del servicio de texto a voz le indicará al usuario los objetos identificados
Flujo alternativo	N/A	
Postcondición	Aplicación debe indicar objeto identificado	
Excepciones	Cámara inhabilitada	

7.6.4 Diagrama de Clases

Figura 13 Diagrama de clases



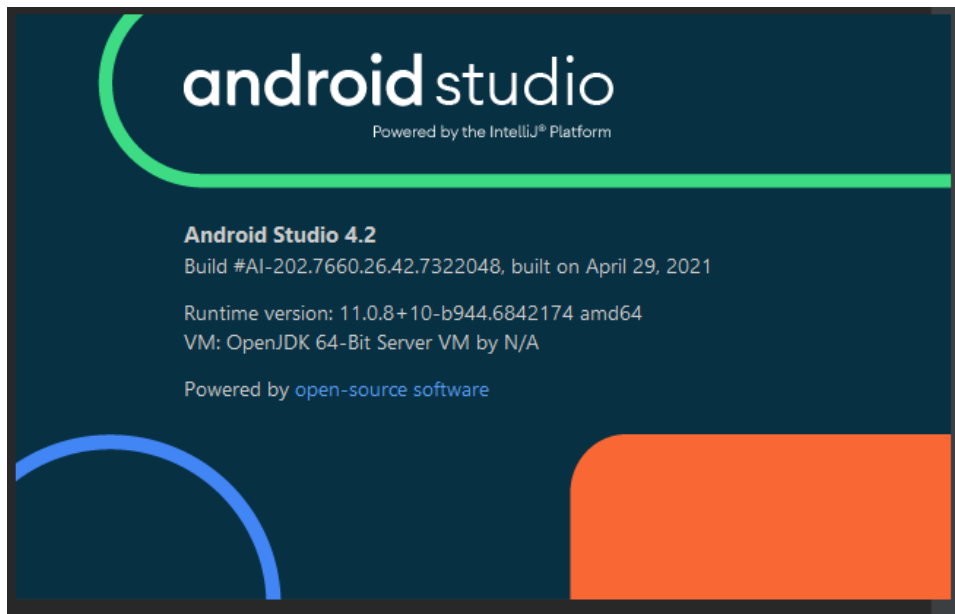
7.6.5 Lenguaje de programación

Se seleccionó el lenguaje de programación JAVA, haciendo uso de programación orientada a objetos, donde se establecen clases y objetos para el desarrollo. Este tipo de lenguaje permite mayor interacción entre sus clases.

7.6.6 Selección IDE

Android Studio 4.2, se elige esta versión dada que es compatible con Open CV, el cual nos permite hacer uso de visión de máquina y/o redes neuronales para la identificación de objetos

Figura 14 Android Studio 4.2



Nota: en la imagen se encuentra la versión seleccionada para el desarrollo de la solución, tomada de: <https://www.xatakandroid.com/programacion-android/android-studio-4-2-renueva-asistente-nuevo-proyecto-te-ayuda-a-actualizar-gradle-otros-cambios>

7.6.7 Investigación librerías

En el marco teórico del presente documento se incluyó la información referente al uso y estructura de las librerías para el desarrollo de nuestra solución.

7.6.8 Desarrollo módulos

Configuración para TextToSpeech

- Clase Zillion, es un apoyo a los objetos que utilizan TextToSpeech para obtener el context de la aplicación, con esto se logra enviar a voz en cualquier método.
- En todas las actividades donde se utilice se debe importar la librería.
android.speech.tts.TextToSpeech.

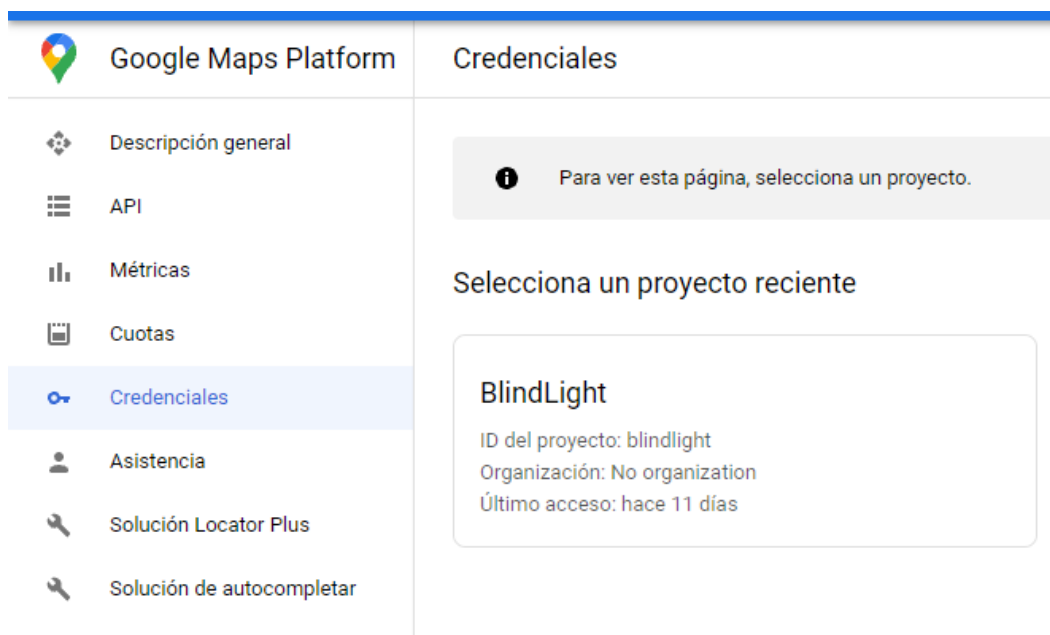
- El objeto se definió como “tts”.
- `Locale.getDefault()`, es la opción que permite que obtenga el lenguaje que tiene configurado el dispositivo.

Configuración para utilizar el API de Google Maps

A continuación, detallamos los pasos que se siguieron para la configuración del API:

Se creó una cuenta en Google Maps Platform para obtener la clave que se usa en el código, dado que sin esto no es posible conectarse a la API

Figura 15 Configuración API de Google Maps



Nota: figura tomada de la configuración de Google Maps

En la siguiente figura se evidencia la configuración de la clave en el archivo `google_maps_api`, que se generó al crear la actividad de mapas.

Figura 16 Registro llave API

```
<resources>
  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">AIzaSyCpmx-XZdDv1wRf-ZJAfdF082biviP...</string>
</resources>
```

Nota: figura tomada del desarrollo de Blind Ligth

Se agregan las dependencias en el archivo `build.gradle`

Figura 17 Dependencias

```
implementation 'com.google.android.gms:play-services-maps:12.0.1'
implementation 'com.google.android.gms:play-services-auth:12.0.1'
implementation 'com.google.android.gms:play-services-location:12.0.1'
implementation 'com.google.android.gms:play-services:12.0.1'
implementation 'com.google.android.gms:play-services-ads:12.0.1'
```

Nota: figura tomada del desarrollo de Blind Ligth

En los términos y condiciones de la plataforma de Google Maps se prohíbe replicar la funcionalidad de navegación de Google Maps, esto se puede ver en el anexo 02, por este motivo, para cumplir esta función la aplicación hace el llamado a la app de Google Maps para obtener las indicaciones para llegar al destino

Figura 18 Llamado a la app de Google Maps

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("google.navigation:q=4.661793355683103,-74.06287213289593&mode=d"));
intent.setPackage("com.google.android.apps.maps");
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
if(intent.resolveActivity(getPackageManager()) != null){
    startActivity(intent);
}
```

Nota: figura tomada del desarrollo de Blind Ligth

Configuración Tensorflow. Se deben agregar las dependencias en el archivo build.grade

Figura 19 Registro dependencias en build.grade

```
implementation 'org.tensorflow:tensorflow-lite-metadata:0.1.0-rc1'
implementation 'org.tensorflow:tensorflow-lite-gpu:2.2.0'
implementation 'org.tensorflow:tensorflow-lite-support:0.1.0'
implementation 'org.tensorflow:tensorflow-lite-task-vision:0.1.0'
implementation 'org.tensorflow:tensorflow-lite-task-text:0.1.0'
```

Nota: figura tomada del desarrollo de Blind Ligth

Se agrega un origen aparte del "main", en este se almacena las diferentes configuraciones de librerías para la arquitectura de la CPU del dispositivo.

Figura 20 Registro en el MAIN

```

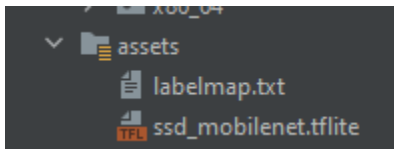
sourceSets {
    main {
        jni {
            srcDirs 'src\\main\\jni', 'src\\main\\jni\\libs'
        }
    }
}

```

Nota: figura tomada del desarrollo de Blind Ligth

Se carga el modelo entrenado en la carpeta "Assets"

Figura 21 Cargue modelo entrenado



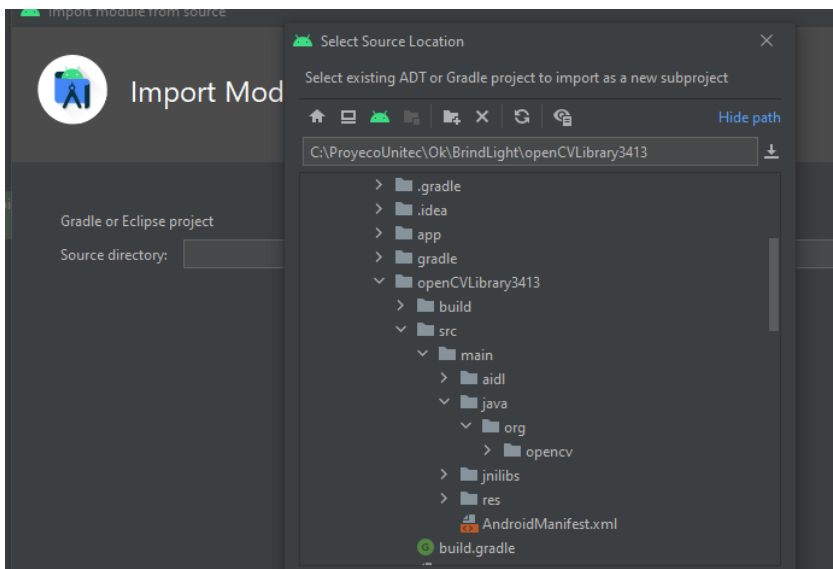
Nota: figura tomada del desarrollo de Blind Ligth

Configuración OpenCV

Por errores al momento de importar el módulo de OpenCV con la última versión de Android

Studio se utilizó la versión 4.2.1 que permitió la carga correcta

Figura 22 Cargue OpenCV



Nota: figura tomada del desarrollo de Blind Ligth

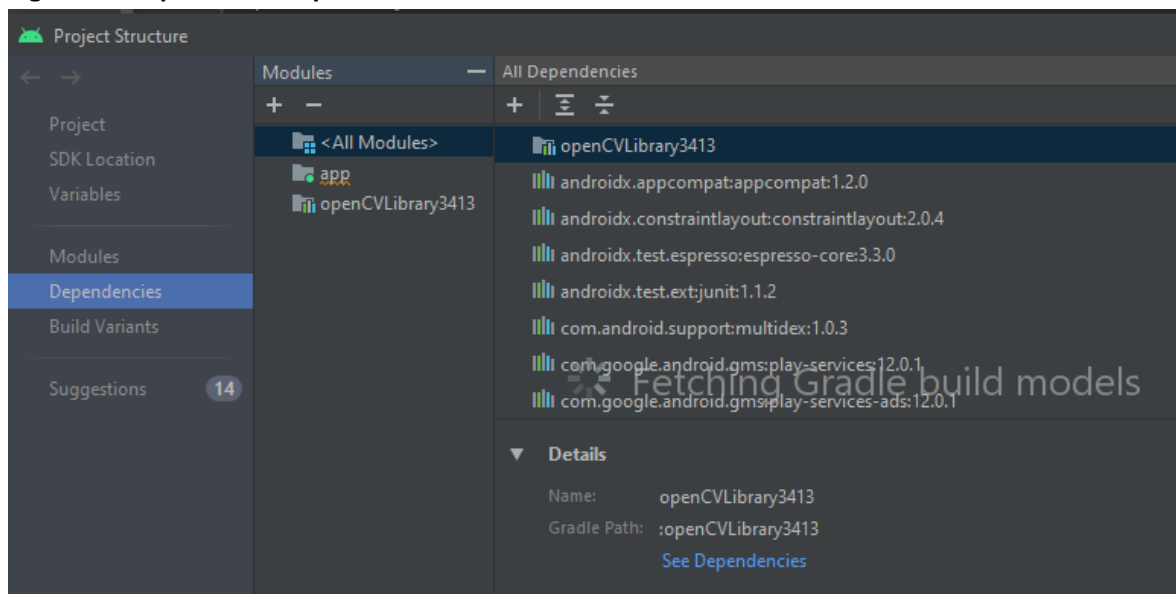
En el archivo build.grade de OpenCV se debe configurar los mismos valores que el de la aplicación, esto cuanto a:

minSdkVersion

targetSdkVersion

Se agrega como dependencia el modulo de OpenCV

Figura 23 Dependencia OpenCV



Nota: figura tomada del desarrollo de Blind Ligth

Clase MainActivity

Es la clase principal de la aplicación, al momento de ejecutar realiza una serie de configuraciones iniciales, está compuesta por 2 métodos.

Método onCreate

Figura 24 Método onCreate

```
@Override  
protected void onCreate(Bundle savedInstanceState) {...}
```

Nota: figura tomada del desarrollo de Blind Ligth

Con este método se realiza lo siguiente:

- Se configura dos botones para permitir la posibilidad de ejecutar una de las dos tareas de la aplicación, sea reconocer objetos o guía GPS, en caso de que no se elija algún botón se ejecutará automáticamente todo el flujo el cual va en el siguiente orden:
- Se inicializa la navegación por GPS de Google Maps.
- Se da una espera de 5 segundos para que se complete la elección de ruta correcta.
- Se inicializa la actividad de reconocer objetos.

Método onBackPressed

Figura 25 Método onBackPressed

```
@Override  
public void onBackPressed() {...}  
}
```

Nota: figura tomada del desarrollo de Blind Ligth

Este método se creó con el objetivo de finalizar todos los recursos de la aplicación al momento de cerrarlo, con esto evitamos uso de recursos como batería, GPS, datos, etc., sin usar la aplicación.

Figura 26 Inicialización OpenCV

```
static {  
    if(OpenCVLoader.initDebug()){  
        Log.d( tag: "MainActivity: ", msg: "OpenCvOk");  
    }  
    else {  
        Log.d( tag: "MainActivity: ", msg: "OpenCvFail");  
    }  
}
```

Nota: figura tomada del desarrollo de Blind Ligth

En la figura 26 se muestra cómo se inicializa de manera estática OpenCV

Método OnDestroy

Al momento de cerrar la actividad, se finalizan los objetos que estaba utilizando la misma con este método onDestroy.

Figura 27 Método onDestroy

```
public void onDestroy(){
```

Nota: figura tomada del desarrollo de Blind Ligth

Clase CameraActivity

Clase que interactúa con la cámara del dispositivo, y genera los objetos con los cuales va interactuar OpenCV.

En la figura 28 se encuentra el objeto de “visión maquina” con el cual OpenCV capta y procesa la imagen.

Figura 28 Interacción objetos

```
mOpenCvCameraView=(CameraBridgeViewBase) findViewById(R.id.frame_Surface);
mOpenCvCameraView.setVisibility(SurfaceView.VISIBLE);
mOpenCvCameraView.setCvCameraViewListener(this);
```

Nota: figura tomada del desarrollo de Blind Ligth

En la figura 29, se evidencia el código con el cual se solicita los permisos para que la aplicación tenga acceso a la cámara.

Figura 29 Objeto Visión Máquina

```
int MY_PERMISSIONS_REQUEST_CAMERA=0;
// Solicitud y validacion de permisos camara
if (ContextCompat.checkSelfPermission( context: CameraActivity.this, Manifest.permission.CAMERA)
    == PackageManager.PERMISSION_DENIED){
    ActivityCompat.requestPermissions( activity: CameraActivity.this, new String[] {Manifest.permission.CAMERA}, MY_PERMIS
}
```

Nota: figura tomada del desarrollo de Blind Ligth

Se carga el modelo obtenido de TensorFlow, este modelo es pre-entrenados y es apto para este proyecto.

Figura 30 Cargue modelo

```
try {
    objectDetectorClass=new objectDetectorClass(getAssets(), modelPath: "ssd_mobilenet.tflite", labelPath: "labelmap.txt", inputSi
    Log.d( tag: "MainActivity", msg: "Modelo cargado");
}
catch (IOException e){
    Log.d( tag: "MainActivity", msg: "Modelo erroneo");
    e.printStackTrace();
}
```

Nota: figura tomada del desarrollo de Blind Ligth

Este método lo provee la librería de OpenCV, es utilizado para procesar imágenes que envía la clase de reconocimiento de objetos.

Figura 31 OnCameraFrame

```

public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame){
    mRgba=inputFrame.rgba();
    mGray=inputFrame.gray();
    Mat out=new Mat();
    out=objectDetectorClass.recognizeImage(mRgba);

    return mRgba;
}

```

Nota: figura tomada del desarrollo de Blind Ligth

Clase ObjectDetectorClass

Esta clase es la que permite reconocimiento de objetos.

En la figura 32 se lee el modelo cargado en la clase CameraActivity , y se establecen unas configuraciones de tema de procesamiento gráfico.

Figura 32 Bloque de código para reconocimiento de objetos

```

objectDetectorClass(AssetManager assetManager,String modelPath, String labelPath,int inputSize) throws IOException{
    INPUT_SIZE=inputSize;
    // use to define gpu or cpu // no. of threads
    Interpreter.Options options=new Interpreter.Options();
    // use to initialize gpu in app
    GpuDelegate gpuDelegate = new GpuDelegate();
    options.addDelegate(gpuDelegate);
    options.setNumThreads(4); // set it according to your phone
    // loading model
    interpreter=new Interpreter(loadModelFile(assetManager,modelPath),options);
    // load labelmap
    labelList=loadLabelList(assetManager,labelPath);
    S_str= "";
}

```

Nota: figura tomada del desarrollo de Blind Ligth

En la figura 33, encontramos un método de OpenCV, al cual le agregamos ese try catch con el fin de dar un tiempo de espera de 1.2 segundos antes de iniciar el procesamiento de imágenes.

Figura 33 Excepción Try - Catch

```
public Mat recognizeImage(Mat mat_image){
    try {
        Thread.sleep( millis: 1200);
    }catch (InterruptedException e) {
        Log.d( tag: "Error", msg: "error durmiendo hilo");
    }
}
```

Nota: figura tomada del desarrollo de Blind Ligth

Text To Speech. En la figura 34, encontramos el uso de Text To Speech para enviar a voz el objeto detectado

Figura 34 Código Text To Speech

```
tts = new TextToSpeech(Zillion.getContext(), status -> {
    if(status != TextToSpeech.ERROR) {
        tts.setLanguage(Locale.getDefault());
        tts.setPitch(1.3f);
        tts.setSpeechRate(1f);
        // tts.speak(SC_str, TextToSpeech.QUEUE_FLUSH, null,null);
        tts.speak( text: "Alfrente hay un"+S_str, TextToSpeech.QUEUE_FLUSH, params: null);
    }
});
}

S_str2=S_str;
```

Nota: figura tomada del desarrollo de Blind Ligth

En el siguiente ciclo FOR, se establece la cantidad de objetos que queremos reconocer, para esta interacción se eligió uno, es decir, sí hay ejemplo un semáforo y un perro, el dispositivo solo detectara uno de los dos.

Figura 35 Ciclo FOR

```
176 for (int i=0;i<1;i++){
```

Nota: figura tomada del desarrollo de Blind Ligth

Clase GPS

Con esta clase obtenemos todo lo referente a geolocalización, se hizo uso de:

- Librería Address, Geocoder, Location de Android.
- Uso del API de Google Maps.
- En la figura 36 se muestra el bloque de código donde se solicitan los permisos de GPS y se establece el nombre del objeto que va almacenar el mapa (R.id.map)

Figura 36 Permisos GPS

```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync( onMapReadyCallback: this);
if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager
.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION,},
requestCode: 1000);
} else {
    locationStart();
}
}
```

Nota: figura tomada del desarrollo de Blind Ligth

En la figura 37, se hace la solicitud de actualizar la localización que nos provee directamente el GPS o nuestra conexión a internet, establecemos un mínimo de 1.5 segundos para renovarla.

Figura 37 Actualización localización

```
assert mLocManager != null;
mLocManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, minTime: 1500, minDistance: 0, Local);
mLocManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTime: 1500, minDistance: 0, Local);
```

Nota: figura tomada del desarrollo de Blind Ligth

Método setLocation. En este método establecemos las coordenadas de la sede F de UNITEC que nos provee Google Maps, se obtiene la dirección en la cual el dispositivo se encuentra, y se envía a voz dos datos, dirección y distancia a la que nos encontramos de la sede F.

Figura 38 Método setLocation

```
    }  
    public void setLocation(Location loc) {  
        Location location2 = new Location( provider: "localizacion 2");  
        location2.setLatitude(4.661793355683103); //latitud  
        location2.setLongitude(-74.06287213289593); //longitud
```

Nota: figura tomada del desarrollo de Blind Ligth

A continuación, incluimos la funcionalidad de algunos métodos adicionales:

- **OnMapReady:** método que provee la librería de OnMapReadyCallback para cargar el mapa de la API de Google.

- **establecerMarcador:** método para ubicar el marcado o más conocido como icono en el mapa de acuerdo a las coordenadas que se le ingrese.

- **updateUbicacion:** método para actualizar la ubicación que se le envía al método establecerMarcador

Figura 39 Método adicionales

```

public void onMapReady(GoogleMap googleMap) {...}
private void establecerMarcador(double latitude, double logintud) {...}

private void updateUbicacion(Location location) {...}

```

Nota: figura tomada del desarrollo de Blind Ligth

Clase Location. Está clase es propia de las librerías de localización de Android que contiene los métodos para dar manejo a la ubicación, se hace llamado a los métodos setLocation, updateUbicación y establecerMarcador para actualizar la ubicación en el mapa.

Figura 40 Clase Location

```

public class Localizacion implements LocationListener {...}

```

Nota: figura tomada del desarrollo de Blind Ligth

7.6.9 Integración de los módulos o servicios

Una vez finalizado el desarrollo por módulos, se realizó la integración de estos, con el fin de entregarle al usuario final la aplicación o solución que le brinde las funcionalidades descritas anteriormente. Se adicionó un Home o Inicio de aplicación, donde se encontrarán cuatro botones, así:

- Detectar Objetos
- Ubicación GPS
- Navegación GPS
- Todo en uno

7.6.10 Testeo

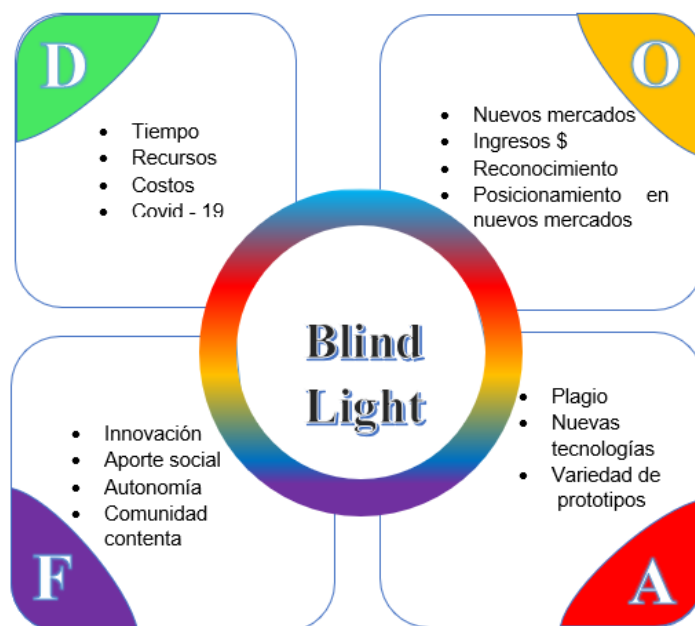
Haciendo uso de la metodología SCRUM, se realizó el desarrollo de la aplicación por módulos garantizando así entregas parciales con ajustes aplicados. Una vez finalizado el desarrollo de cada uno de estos, se realizaron las pruebas funcionales junto con los ajustes a que hubo lugar.

7.7 Plan de trabajo

7.7.1 Matriz DOFA

Por medio de esta herramienta podemos analizar y evaluar la situación actual de una empresa o proyecto que se esté ejecutando. Ya que esta herramienta me permite identificar las debilidades, las oportunidades, fortalezas y amenazas.

Figura 41 Matriz DOFA



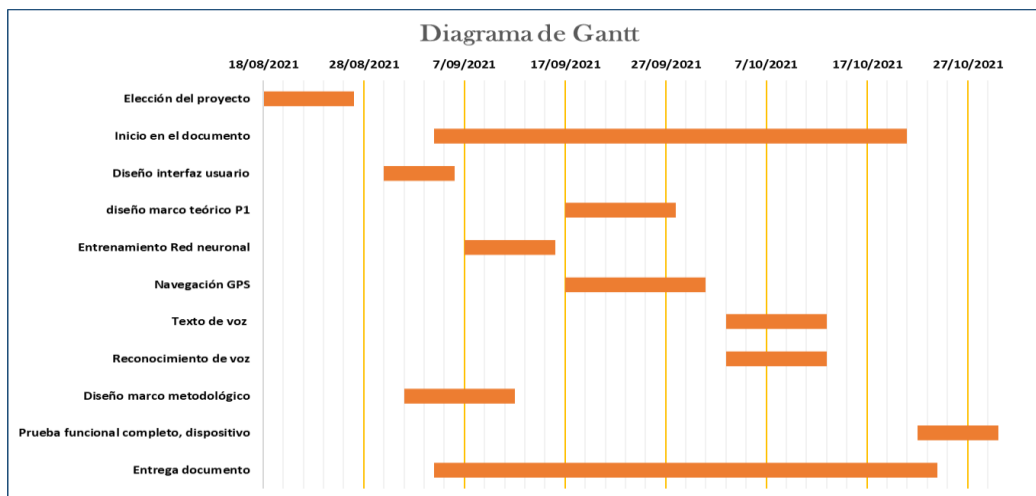
7.7.2 Diagrama de Gantt

Se identificaron las actividades relevantes del proceso, con el fin de definir tiempos estimados para el desarrollo del proyecto, logrando así optimizar los tiempos y definir metas.

Tabla 6 Actividades Diagrama de Gantt

NOMBRE DE LA ACTIVIDAD	FECHA DE INICIO	DURACIÓN EN DÍAS	FECHA DE FIN
Elección del proyecto	18/08/2021	9	27/08/2021
Inicio en el documento	04/09/2021	47	26/10/2021
Diseño interfaz usuario	30/08/2021	7	06/09/2021
diseño marco teórico P1	17/09/2021	11	28/09/2021
Entrenamiento Red neuronal	07/09/2021	9	15/09/2021
Navegación GPS	17/09/2021	14	01/10/2021
Texto de voz	03/10/2021	10	13/10/2021
Reconocimiento de voz	03/10/2021	10	13/10/2021
Diseño marco metodológico	01/09/2021	11	12/10/2021
Prueba funcional completo, dispositivo	22/10/2021	18	10/11/2021
Entrega documento	04/09/2021	50	30/10/2021
Inicio del proyecto	18/08/2021		
Fin del proyecto	30/10/2021		

Figura 42 Diagrama de Gantt



7.7.3 Plan de trabajo

Se dividieron en cuatro fases el desarrollo del proyecto, a continuación, se detallan las actividades planteadas en cada una (ver figura 43) y los resultados correspondientes (ver figura 44) .

Figura 43 Actividades

	Fase 1 Levantamiento de información	Fase 2 Desarrollo P1	Fase 3 Desarrollo P2	Fase 4 Integración y Documentación
Actividades	Lista de actividades	Lista de actividades	Lista de actividades	Lista de actividades
	Planteamiento del problema	Elaboración casos de uso y diagramas	Pruebas unitarias	Desarrollo Home
	Planteamiento de objetivos	Revisión librerías	Revisión modelo de entrenamiento	Pruebas integración
	Planteamiento solución	Desarrollo Ubicación GPS	Desarrollo detección objetos	Generación documento final
	Elección IDE	Pruebas unitarias	Pruebas unitarias	
	Elección Lenguaje	Desarrollo Guía GPS		

Figura 44 Resultados

	Fase 1 Levantamiento de información	Fase 2 Desarrollo P1	Fase 3 Desarrollo P2	Fase 4 Integración y Documentación
Resultados	Lista de resultados	Lista de resultados	Lista de resultados	Lista de resultados
	Problema definido	Casos de uso, flujo aplicación y diagrama de clases	Ajuste módulo	Codificación Home
	Objetivos identificados	Documentación librerías	Ajuste idioma objetos identificados	Ajustes integración
	IDE - Android Studio Versión 4.2	Codificación de la ubicación GPS	Codificación detección objetos	Documento final
	Lenguaje de programación Java	Ajustes módulo	Ajustes módulo	
		Codificación guía GPS		

7.8 Análisis y Discusión de Resultados

Inicialmente se planteó el desarrollo de nuestro prototipo de interacción haciendo uso de un **Raspberry PI** y un arnés para facilitar su uso, sin embargo, en esta fase del proyecto se llegó al punto de codificación de una aplicación para equipos Android que puede ser migrada a la **Raspberry PI**, a continuación, encontraremos algunos ejemplos de los resultados del desarrollo:

7.8.1 Home

En este se integraron los botones que le permite al usuario acceder a las funcionalidades desarrolladas, se adicionó una opción donde encontrará unificadas las funcionalidades

Figura 45 Home



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Ligth

7.8.2 Detección objetos

Al ingresar a este botón se activará la cámara y se podrán identificar objetos que comúnmente nos encontramos durante nuestros desplazamientos, la aplicación a través del audio del dispositivo móvil le indicará a nuestro usuario que objeto está identificando, a continuación, algunos ejemplos:

Figura 46 Identificación objeto "Semáforo"



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Ligth

Figura 417 Identificación objeto "Señal de PARE"



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Lighth

Figura 48 Identificación objeto “Carro o Coche”



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Ligth

Figura 49 Identificación objeto "Persona"



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Ligth

7.8.3 Ubicación GPS

Al ingresar a esta opción, se le indica al usuario de acuerdo a las coordenadas del GPS, la dirección en la que se encuentra junto con la distancia en metros con referencia a la Sede F de UNITEC, punto destino elegido para el desarrollo desde prototipo.

Figura 50 Ubicación GPS

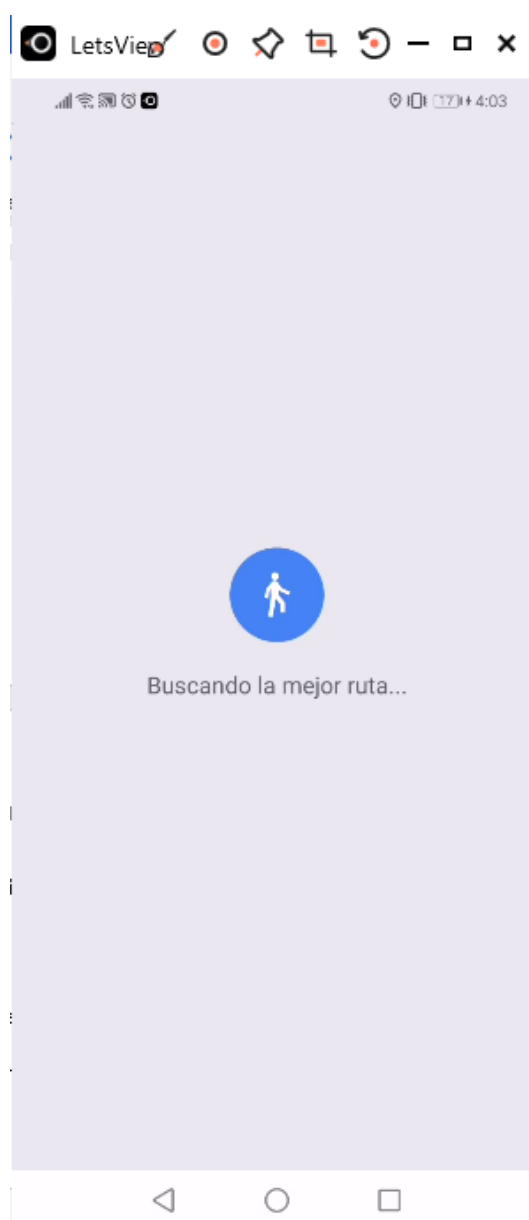


Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Lighth

7.8.4 Guía GPS

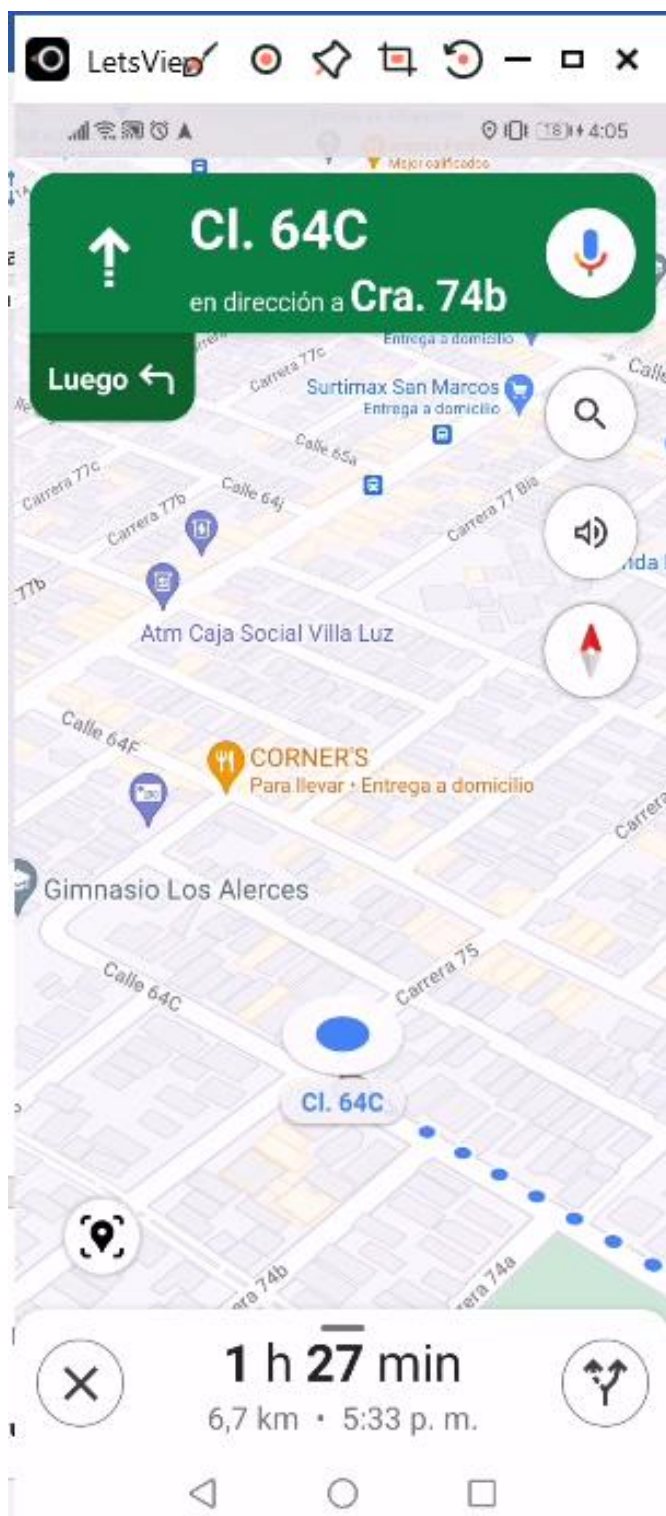
Al ingresar a esta opción, la aplicación buscará la ruta para llegar a la Sede F de UNITEC, posteriormente haciendo uso de Google Maps le indicará por donde debe desplazarse y el tiempo que tardará caminando.

Figura 51 Búsqueda de la ruta



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Ligth

Figura 52 Indicación ruta



Nota: imagen tomada del resultado del desarrollo de nuestra solución Blind Lighth

7.8.5 *Todo en Uno*

Todo en uno. Al ingresar a esta opción, la aplicación de manera automática le brindará todos los servicios sin necesidad de navegar por las demás opciones cargadas.

8 Conclusiones y Recomendaciones

Una vez realizada la revisión de la problemática planteada, proponer un prototipo ideal y desarrollar una primera fase podemos concluir lo siguiente:

- Existen pocas herramientas que le faciliten el desplazamiento a las personas con discapacidad visual sin importar el grado de esta.
- Sumado a lo anterior, la infraestructura de la Ciudad dificulta un poco el desplazamiento, dado que, por ejemplo, no se encuentran marcadas las señales de tránsito, por ende, no son fáciles de identificar para ellos.
- Haciendo uso inteligencia artificial y visión de máquina, podemos apoyar el reconocimiento de objetos para facilitar el desplazamiento de nuestro usuario.
- Para realizar un entrenamiento optimo debemos agregar al nuestro modelo al rededor 300 imágenes del objeto y 200 imágenes de objetos diferentes.
- Haciendo uso de Text To Speech hacemos un poco más accesible los dispositivos móviles a las personas con algún tipo de discapacidad, teniendo en cuenta que este le indicará la información de lo que se encuentra en forma de texto.
- Para una siguiente fase, se sugiere realizar la migración y adaptación del Código a la Raspberry PI, este se encuentra alojado en OneDrive para acceso de personal con correo institucional UNITEC.
(Ver anexo 3)

9 Lista de Referencia o Bibliografía

OMD, (2020). Informe mundial sobre la visión [World report on vision].

<https://apps.who.int/iris/bitstream/handle/10665/331423/9789240000346-spa.pdf>

INCI, (2018, junio 19). Los ciegos en el censo 2018]. <http://www.inci.gov.co/blog/los-ciegos-en-el-censo-2018>

INCI, (2018, octubre 17). Orientación y movilidad de personas con discapacidad visual 2018. <http://www.inci.gov.co/blog/los-ciegos-en-el-censo-2018>

IBM Watson. (s. f.). España | IBM. <https://www.ibm.com/es-es/watson>

Quesada, F. J. G., Graciani, M. A. F., Bonal, M. T. L., & Díaz-Mata, M. A. (1994). Aprendizaje con redes neuronales artificiales. Ensayos: Revista De La Facultad De Educación De Albacete, (9), 169-180.

INCI, (s.f). Apropiándonos del bastón blanco <http://www.inci.gov.co/blog/apropiandonos-del-baston-blanco>

Bustos Ramírez, J. (2014). Bogotá, una ciudad que no ve por sus invidentes. [Comunicación social - Tesis y disertaciones académicas]. <http://hdl.handle.net/10554/18401>.

Espectador, (2008, junio 3). Bogotá, una ciudad difícil para los discapacitados. El Espectador. <https://www.elespectador.com/bogota/bogota-una-ciudad-dificil-para-los-discapacitados-article-17247/>

H.S. María del Rosario Guerra de la Espriella (2020). Por medio de la cual se adopta el uso del sistema braille en empaques de productos alimenticios, médicos y en servicios turísticos, así como en los sitios de carácter público y se dictan otras disposiciones. <https://www.camara.gov.co/sistema-braille-0>

Vargas Castillo, C. (2018). ¿Funcionan los semáforos sonoros en Bogotá? RCN Radio. <https://www.rcnradio.com/bogota/solo-uno-de-cada-diez-semaforos-sonoros-en-bogota-funciona>

ZOOMTEXT. (s. f.). El futuro digital es de todos Mintinc. <https://convertic.gov.co/641/w3-propertyvalue-15340.html>

Echeverri, E. (2020, 19 diciembre). Hay que aprovechar al máximo las tecnologías para generar inclusión. Semana. <https://www.semana.com/contenidos-editoriales/la-ceguera-no-es-una-barrera/articulo/hay-que-aprovechar-al-maximo-las-tecnologias-para-generar-inclusion/202047/>

Universidad Carlos tercero de Madrid (2013). Nuevo prototipo sónico de ayuda a la ceguera. https://portal.uc3m.es/portal/page/portal/actualidad_cientifica/noticias/ayuda_ceguera

Branco, A. (3 de agosto de 2020). Este dispositivo portátil para ciegos jubilara al perro guía, el español https://www.elespanol.com/omicron/hardware/20200803/dispositivo-portatil-ciegos-jubilara-perro-guia/509450331_0.html

Mourenza, A, (26 de enero 2020). El creador del bastón inteligente que guía a los ciegos, el país semanal; https://elpais.com/elpais/2020/01/24/eps/1579868789_604708.html

VARÓN CASTRO, J. F., & DÍAZ CASTAÑEDA, M. S. (2018). MÓDULO ASISTENTE DE INVIDENTES PARA EL SISTEMA DE TRANSPORTE TRANSMILENIO (Profesional). Universidad Piloto de Colombia. <http://polux.unipiloto.edu.co:8080/00004854.pdf>

A.G.I (2016). La discapacidad visual; elkartea; <https://www.agielkartea.org/discapacidad-visual/la-discapacidad-visual.html>

Puntodis (2019), discapacidad visual, https://puntodis.com/featured_item/discapacidad-visual/

OMS (2019) La OMS presenta el primer informe mundial sobre la visión; <https://www.who.int/es/news/item/08-10-2019-who-launches-first-world-report-on-vision>

All about visión, (marzo de 2019) que son las cataratas en los ojos, <https://www.allaboutvision.com/es/condiciones/cataratas.htm>

National eye institute (10 de julio de 2019), La retinopatía diabética;

<https://www.nei.nih.gov/learn-about-eye-health/en-espanol/la-retinopatia-diabetica>

Boyd K, (22 de septiembre 2021) que es glaucoma, American Academy;

<https://www.aaopt.org/salud-ocular/enfermedades/que-es-la-glaucoma>

Boyd K (14 de septiembre 2021). Que es un desprendimiento de retina, American academy;

<https://www.aaopt.org/salud-ocular/enfermedades/desgarramiento-desprendimiento-retina>

AndroidOS. 2012. Historia — Android OS 0.1 documentation.

<https://androidos.readthedocs.io/en/latest/data/historia/>

Arquitectura de la plataforma. (s. f.). <https://developer.android.com/guide/platform?hl=es-419>.

Vaati, E. (2020, 3 julio). Qué es Android SDK y cómo empezar a usarlo. Code Envato Tuplus.

<https://code.tutsplus.com/es/tutorials/the-android-sdk-tutorial--cms-34623>

Meirinhos, M. (2009). El open source en la educación. Congrés Internacional Virtual d'Educació CIVE 2009. <https://core.ac.uk/download/pdf/153406765.pdf>

Licencia BSD. (2015, 20 mayo). I3 Campus.

http://i3campus.co/CONTENIDOS/wikipedia/content/a/licencia_bsd.html

Servicios de Google Play. (s. f.). Developers. <https://developer.android.com/distribute/play-services?hl=es>

Geocodificador. (s. f.). Developers.

<https://developer.android.com/reference/android/location/Geocoder>

OpenCV. (s. f.). OpenCV. <https://opencv.org/>

Tensor Flow (s.f). Por qué TensorFlow . <https://www.tensorflow.org/?hl=es-419>

IBM Watson - Text to speech. (s. f.). IBM. <https://www.ibm.com/co-es/cloud/watson-text-to-speech>

Meseguer González, P. y Ramon López de Mántaras Badia. (2017). Inteligencia artificial. Editorial CSIC Consejo Superior de Investigaciones Científicas. <https://elibro-net.biblioteca.unitec.edu.co/es/lc/corpunitec/titulos/42319>

Vidal, A. E. (2007). Alan Turing y el nacimiento de la inteligencia artificial. Antena de telecomunicación, (167), 45-45.

Andrés Argente, T. D. (2004). Homo Cybersapiens: la inteligencia artificial y la humana. EUNSA. <https://elibro-net.biblioteca.unitec.edu.co/es/lc/corpunitec/titulos/47394>

Palma Méndez, J. T. (2008). Inteligencia artificial: métodos, técnicas y aplicaciones. McGraw-Hill España. <https://elibro-net.biblioteca.unitec.edu.co/es/lc/corpunitec/titulos/50116>

Panezi, A. (Coord.), Pérez Bes, F. (Ed.) y García Mexía, P. (Ed.). (2021). Artificial Intelligence and the Law. Wolters Kluwer España. <https://elibro-net.biblioteca.unitec.edu.co/es/lc/corpunitec/titulos/181956>

Caicedo Bravo, E. F. y López Sotelo, J. A. (2009). Una aproximación práctica a las redes neuronales artificiales. Programa Editorial Universidad del Valle. <https://elibro.net/es/lc/corpunitec/titulos/129183>

Sossa, S. (2021, 23 mayo). Tipos de redes neuronales (Clasificación). Inteligencia-Artificial.dev. <https://inteligencia-artificial.dev/tipos-redes-neuronales/>

Valero Verdú, S. Aplicación de un modelo de red neuronal no supervisado a la clasificación de consumidores eléctricos. San Vicente (Alicante): ECU, 2013. <https://elibro.net/es/ereader/corpunitec/42794?>

Vaati, E. (2020, 3 julio). Qué es Android SDK y cómo empezar a usarlo

Ltd, R. P. (s. f.). 4 Model B specifications –. Raspberry Pi.

<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>

Balestrini, M (2000). Como se elabora el proyecto de investigación. BL consultores asociados Centro de recursos para el aprendizaje y la investigación. 2018. Definición y propósito de la Investigación Aplicada | Biblioteca DUOC UC. <http://www2.duoc.cl/biblioteca/crai/definicion-y-proposito-de-la-investigacion-aplicada>

Hernández. S (2006) Enfoque cualitativo y cuantitativo, según Hernández Sampieri. Portafolio académico; <https://portaprodti.wordpress.com/enfoque-cualitativo-y-cuantitativo-segun-hernandez-sampieri/>

Tamayo, M. (2012) El proceso de la investigación científica. México. Limusa, p.180.

Herrera M (2018 5 de junio). Dificultades para ver según censo 2018, INCI; <http://www.inci.gov.co/blog/dificultad-para-ver-segun-el-censo-2018>

Mancera G (2020, 4 de diciembre), En Bogotá, por cada 100 mil habitantes hay 6.379 personas con discapacidad, GOV.CO; <http://www.sdp.gov.co/noticias/bogota-cada-100-mil-habitantes-hay-6379-personas-discapacidad>

11 Anexos

Anexo 1. AUTORIZACIÓN USO CÓDIGO DETECCIÓN OBJETOS

1/19/21 18:07

Comes: HERRELO LOPEZ JOHAN CAMILO - Outlook

Re: Permission to use Object Detection App**pramod verma <pramodverma360@gmail.com>**

Jue 30/09/2021 11:04 PM

Para: HERRELO LOPEZ JOHAN CAMILO <67172519@unitec.edu.co>

Yes, I approve.

On Fri, Oct 1, 2021 at 5:34 AM HERRELO LOPEZ JOHAN CAMILO <67172519@unitec.edu.co> wrote:

HI

Can you help me please , I need your approval

Johan Camilo Herrefio Lopez
 Código 67172519
 Ingeniería de Telecomunicaciones

De: HERRELO LOPEZ JOHAN CAMILO

Enviado: lunes, 27 de septiembre de 2021 5:52 p. m.

Para: pramodverma360@gmail.com <pramodverma360@gmail.com>**Asunto: Permission to use Object Detection App**

HI PramodVerman

I have seen your tutorials videos , and i think that your code can help me on my project , can you grant me permission for use it ?

I am Telecommunication student of University Unitec from Bogotá Colombia

Regards

Johan Camilo Herrefio Lopez
 Código 67172519
 Ingeniería de Telecomunicaciones

Anexo 2. DOCUMENTACIÓN RESTRICCIÓN USO DE GOOGLE MAPS

3.2.3 Restrictions Against Misusing the Services.

(d) No Re-Creating Google Products or Features. Customer will not use the Services to create a product or service with features that are substantially similar to or that re-create the features of another Google product or service. Customer's product or service must contain substantial, independent value and features beyond the Google products or services. For example, Customer will not: (i) re-distribute the Google Maps Core Services or pass them off as if they were Customer's services; (ii) use the Google Maps Core Services to create a substitute of the Google Maps Core Services, Google Maps, or Google Maps mobile apps, or their features; (iii) use the Google Maps Core Services in a listings or directory service or to create or augment an advertising product; (iv) combine data from the Directions API, Geolocation API, and Maps SDK for Android to create real-time navigation functionality substantially similar to the functionality provided by the Google Maps for Android mobile app.

Referencia tomada de:

Google Maps Platform Terms Of Service |. (2020, 6 mayo). Google Cloud.

<https://cloud.google.com/maps-platform/terms/>

Anexo 3. REPOSITORIO PROYECTO

En el siguiente vinculo se encuentra lo siguiente:

- Código de aplicación versión demo y versión final.
- Apk versión demo y versión final.

Nota: La versión demo es para ejecutar actividades específicas, la versión final ejecuta todo al momento de abrir la app.

https://correounitec-my.sharepoint.com/:f:/g/personal/67172519_unitec_edu_co/Er2ukCJslIPtp2G89UnGDUBHv6qjbP-owU3-vEdXNVsmQ?e=sEFUph

12 CESIÓN DE DERECHOS

Por intermedio del presente documento en mi calidad de autor o titular de los derechos de propiedad intelectual de la obra que adjunto, titulada DISEÑO Y CREACIÓN DE UN PROTOTIPO GUÍA PARA POBLACIÓN CON DISCAPACIDAD VISUAL BASADO EN EL USO DE INTELIGENCIA ARTIFICIAL LLAMADO BLIND LIGHT, autorizo a la Corporación universitaria Unitec para que utilice en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador o titular de la obra objeto del presente documento.

La presente autorización se da sin restricción de tiempo, ni territorio y de manera gratuita. Entiendo que puedo solicitar a la Corporación universitaria Unitec retirar mi obra en cualquier momento tanto de los repositorios como del catálogo si así lo decido.

La presente autorización se otorga de manera no exclusiva, y la misma no implica transferencia de mis derechos patrimoniales en favor de la Corporación universitaria Unitec, por lo que podré utilizar y explotar la obra de la manera que mejor considere. La presente autorización no implica la cesión de los derechos morales y la Corporación universitaria Unitec los reconocerá y velará por el respeto a los mismos.

La presente autorización se hace extensiva no sólo a las facultades y derechos de uso sobre la obra en formato o soporte material, sino también para formato electrónico, y en general para cualquier formato conocido o por conocer. Manifiesto que la obra objeto de la presente autorización es original y la realicé sin violar o usurpar derechos de autor de terceros, por lo tanto, la obra es de mi exclusiva autoría o tengo la titularidad sobre la misma. En caso de presentarse cualquier reclamación o por acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados para todos los efectos la Corporación universitaria Unitec actúa como un tercero de buena fe. La sesión otorgada se ajusta a lo que establece la ley 23 de 1982.

Para constancia de lo expresado anteriormente firmo, como aparece a continuación.

Firma



Nombre Carlos Andrés Mesa Ordoñez
CC. 1.032.364.862

Página 1



Por intermedio del presente documento en mi calidad de autor o titular de los derechos de propiedad intelectual de la obra que adjunto, titulada DISEÑO Y CREACIÓN DE UN PROTOTIPO GUÍA PARA POBLACIÓN CON DISCAPACIDAD VISUAL BASADO EN EL USO DE INTELIGENCIA ARTIFICIAL LLAMADO BLIND LIGHT , autorizo a la Corporación universitaria Unitec para que utilice en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador o titular de la obra objeto del presente documento.

La presente autorización se da sin restricción de tiempo, ni territorio y de manera gratuita. Entiendo que puedo solicitar a la Corporación universitaria Unitec retirar mi obra en cualquier momento tanto de los repositorios como del catálogo si así lo decido.

La presente autorización se otorga de manera no exclusiva, y la misma no implica transferencia de mis derechos patrimoniales en favor de la Corporación universitaria Unitec, por lo que podré utilizar y explotar la obra de la manera que mejor considere. La presente autorización no implica la cesión de los derechos morales y la Corporación universitaria Unitec los reconocerá y velará por el respeto a los mismos.

La presente autorización se hace extensiva no sólo a las facultades y derechos de uso sobre la obra en formato o soporte material, sino también para formato electrónico, y en general para cualquier formato conocido o por conocer. Manifiesto que la obra objeto de la presente autorización es original y la realicé sin violar o usurpar derechos de autor de terceros, por lo tanto, la obra es de mi exclusiva autoría o tengo la titularidad sobre la misma. En caso de presentarse cualquier reclamación o por acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados para todos los efectos la Corporación universitaria Unitec actúa como un tercero de buena fe. La sesión otorgada se ajusta a lo que establece la ley 23 de 1982.

Para constancia de lo expresado anteriormente firmo, como aparece a continuación.

Firma



Nombre Johan Camilo Herreño Lopez
CC. 1073716005

Por intermedio del presente documento en mi calidad de autor o titular de los derechos de propiedad intelectual de la obra que adjunto, titulada **Diseño y relación de un dispositivo guía para población discapacitada visual basado en inteligencia artificial BLIND LIGHT**, autorizo a la Corporación universitaria Unitec para que utilice en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador o titular de la obra objeto del presente documento.

La presente autorización se da sin restricción de tiempo, ni territorio y de manera gratuita. Entiendo que puedo solicitar a la Corporación universitaria Unitec retirar mi obra en cualquier momento tanto de los repositorios como del catálogo si así lo decido.

La presente autorización se otorga de manera no exclusiva, y la misma no implica transferencia de mis derechos patrimoniales en favor de la Corporación universitaria Unitec, por lo que podré utilizar y explotar la obra de la manera que mejor considere. La presente autorización no implica la cesión de los derechos morales y la Corporación universitaria Unitec los reconocerá y velará por el respeto a los mismos.

La presente autorización se hace extensiva no sólo a las facultades y derechos de uso sobre la obra en formato o soporte material, sino también para formato electrónico, y en general para cualquier formato conocido o por conocer. Manifiesto que la obra objeto de la presente autorización es original y la realicé sin violar o usurpar derechos de autor de terceros, por lo tanto, la obra es de mi exclusiva autoría o tengo la titularidad sobre la misma. En caso de presentarse cualquier reclamación o por acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados para todos los efectos la Corporación universitaria Unitec actúa como un tercero de buena fe. La sesión otorgada se ajusta a lo que establece la ley 23 de 1982.

Para constancia de lo expresado anteriormente firmo, como aparece a continuación.

Firma



Nombre Marco Giovanni Gómez Barrera
CC. 1019049833