

APLICACIÓN MÓVIL PARA EL ACCESO AL SERVICIO DE TRANSPORTE PÚBLICO DE LAS PERSONAS CON DISCAPACIDAD VISUAL EN MANIZALES

Alejandro Idárraga Vélez

Informe final de trabajo de grado presentado como requisito parcial para optar al título de
Especialista en Sistemas de Información Geográfica

Línea de Investigación

Grupo de Investigación y Desarrollo en Informática y Telecomunicaciones

Universidad de Manizales

Facultad de Ciencias e Ingeniería

Especialización en Sistemas de Información Geográfica

Manizales, 2022

Resumen

Entendiendo el contexto actual del transporte en la ciudad de Manizales y la disponibilidad de herramientas para el acceso al mismo por parte de las personas con discapacidad visual, se propone el desarrollo de una app para facilitar el acceso de esta población al servicio público en la ciudad de Manizales.

Este desarrollo se hizo mediante el uso de la herramienta App Inventor, la cual permite el manejo de gran parte de las herramientas actuales de los Smartphone facilitando su integración y adaptabilidad a las necesidades definidas para este proyecto.

Tomando como base una ruta definida, se desarrolló una aplicación que le permite al usuario identificar si se encuentra en la ruta adecuada, además de conocer los servicios que se encuentran disponibles en la ruta que seleccionó.

Se utilizan las herramientas de voz que permiten que la selección de la ruta se haga con una orden verbal y la respuesta del equipo sea un audio, todo dentro de las herramientas de sintetización de voz presentes en los smartphone.

Esta información que se almacena en la web y se toma en tiempo real, permite que con el uso de software de Georeferenciación, se obtengan resultados que sirvan de base para un desarrollo más avanzado que dé una solución más puntual al problema de inclusión social presente en el transporte público en la ciudad de Manizales.

Se espera que con la implementación de la aplicación, se logre dar una solución con los criterios de funcionalidad, facilidad de uso, accesibilidad y una herramienta de ayuda a las personas en condición de discapacidad visual.

Palabras clave: Aplicación, discapacidad visual, smartphone, app, georeferenciación, App Inventor.

Abstract

Understanding the current context of transport in the city of Manizales and the availability of tools for access to it by people with visual disabilities, the development of an app is proposed to facilitate the access of this population to public service in the city of Manizales.

This development was made through the use of the App Inventor tool, which allows the handling of a large part of the current smartphone tools, facilitating their integration and adaptability to the needs defined for this project.

Based on a defined route, an application was developed that allows the user to identify if he is on the right route, in addition to knowing the services that are available on the route he selected.

Voice tools are used that allow the selection of the route to be made with a verbal command and the response of the team is an audio, all within the voice synthesis tools present in smartphones.

This information, which is stored on the web and taken in real time, allows that with the use of Georeferencing software, results are obtained that serve as the basis for a more advanced development that gives a more specific solution to the problem of social inclusion present in public transportation in the city of Manizales.

It is expected that with the implementation of the application, a solution will be achieved with the criteria of functionality, ease of use, accessibility and a tool to help people with visual disabilities.

Keywords: Application, visual impairment, smartphone, app, georeferencing, App Inventor.

Contenido

	Pág.
1. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN Y SU JUSTIFICACIÓN.....	9
1.1 DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA	9
1.2 FORMULACIÓN DEL PROBLEMA	10
1.3 JUSTIFICACIÓN	11
2. OBJETIVOS.....	12
2.1 OBJETIVO GENERAL	12
2.2 OBJETIVOS ESPECÍFICOS	12
3. ANTECEDENTES.....	13
4. REFERENTE NORMATIVO Y LEGAL.....	24
5. REFERENTE TEÓRICO	26
5.1 DISCAPACIDAD VISUAL A NIVEL MUNDIAL Y EN COLOMBIA.	26
5.2 ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD AL ENTORNO	29
5.3 TÉCNICAS PARA EL ACCESO A LA INFORMACIÓN DE LAS PERSONAS CON DISCAPACIDAD VISUAL	32
5.4 ANÁLISIS GEOESPACIAL	34
5.5 SOFTWARE.....	38
6. METODOLOGÍA	51
6.1 DESCRIPCIÓN GENERAL.....	51
6.2 PROCEDIMIENTO.....	55
6.3 TIPO DE ESTUDIO.....	73
7. RESULTADOS	74
8. CONCLUSIONES	75
9. RECOMENDACIONES	78
10. REFERENCIAS.....	79

Lista de figuras

FIGURA 1. ESTRUCTURA DE DATOS DE LA RUTA	39
FIGURA 2. MAPA CONCEPTUAL DEL FUNCIONAMIENTO DE LA APLICACIÓN	54
FIGURA 3. MAPA ARCGIS RUTAS SERVICIO PÚBLICO DEL GEOPORTAL DE LA ALCALDÍA DE MANIZALES	55
FIGURA 4. JSON DESCARGADO DEL GEOPORTAL DE LA ALCADIA DE MANIZALES CON TODAS LAS RUTAS.	56
FIGURA 5. MAPA ARCGIS CON LA DEFINICIÓN DE LA RUTA DE ESTUDIO.	56
FIGURA 6. JSON CON LA DEFINICIÓN DE LA RUTA DE ESTUDIO	57
FIGURA 7. SEGMENTO DE CÓDIGO QUE DEFINE LAS ZONAS DE INFLUENCIA POR NODO.....	59
FIGURA 8. SEGMENTO DE CÓDIGO QUE DEFINE LAS COORDENADAS DE CADA NODO EN LA RUTA.	60
FIGURA 9. SEGMENTO DE CÓDIGO QUE PERMITE LA IDENTIFICACIÓN DEL USUARIO CON RESPECTO A CADA BUS	61
FIGURA 10. PANTALLA PRINCIPAL DE LA APLICACIÓN.....	62
FIGURA 11. BLOQUE SPEECHRECOGNIZER.....	63
FIGURA 12. BLOQUE WEBVIEWER QUE RECIBE LA INFORMACIÓN PARA PROCESAR EN LOS SCRIPTS.	64
FIGURA 13. BLOQUE TEXTTOSPEECH PARA INDICAR EL ERROR DE LA UBICACIÓN DEL USUARIO CON LA RUTA.....	64
FIGURA 14. ESTRUCTURA DE LA BASE DE DATOS EN FIREBASE.....	65
FIGURA 15. BLOQUE DE DEFINICIÓN DE TAG Y CONSULTA LA BASE DE DATOS.....	65
FIGURA 16. BLOQUE DE RESPUESTA A LA CONSULTA A LA BASE DE DATOS.....	66
FIGURA 17. BLOQUE QUE ENTREGA LA RESPUESTA DEL PROCESAMIENTO HECHO CON LOS SCRIPTS.	66
FIGURA 18. BLOQUE QUE ALMACENA EN UNA LISTA LA INFORMACIÓN QUE DEVUELVEN LOS ALGORITMOS.....	67
FIGURA 19. BLOQUE QUE PERMIRE QUE EL CELULAR VIBRE.....	67
FIGURA 20. BLOQUE QUE PERMIRE DEFINE LA SALIDA DE LA APLIACIÓN.....	68
FIGURA 21. SEGMENTO DE CÓDIGO QUE PERMITE DEFINIR LA POSICIÓN Y DISTANCIA RECORRIDA POR CADA BUS.....	69

FIGURA 22. PANTALLA PRINCIPAL APLICACIÓN BUS.	71
FIGURA 23. PANTALLA PRINCIPAL APLICACIÓN BUS.	71
FIGURA 24. BLOQUE DE LISTA QUE ALMACENA LOS DATOS A SER PROCESADOS....	72
FIGURA 25. BLOQUE QUE DEFINE LA INFORMACIÓN QUE SE ENVIA A CADA SCRIPT.	72
FIGURA 26. BLOQUE QUE ENVIA LA INFORMACIÓN DEL BUS Y SERÁ PROCESADA POR LA APLICACIÓN DEL USUARIO	73
FIGURA 27. IMÁGENES DE PRESENTACIÓN	85
FIGURA 27. IMÁGENES DE PRESENTACIÓN	85
FIGURA 29. INICIO DE LA APLICACIÓN.....	86
FIGURA 30. SALIDA DE LA APLICACIÓN	87
FIGURA 31. APLICACIÓN CONDUCTOR	87

Lista de símbolos y abreviaturas

Símbolos con letras latinas

Símbolo	Término	Unidad SI	Definición
$dd_{i,j}$	Distancia entre dos puntos	m	Ec. 1
Lm	Longitud maxima del poligono	m	$Lm = \max (d_{i,j})$
A	Area de un poligono	m^2	$A = \sum_{i=1}^{N-1} At_{i,i+1} $
A_s	area bajo el segmento formado por los vértices i e i + 1	m^2	$At_{i,i+1}$
D_i	Medida que representa la cercanía de la ubicación actual del autobús al nodo i.	1	Ec. 4
u_i	Ubicación del nodo superior y anterior al nodo evaluado	m	(x_{ui}, y_{ui}) y (x_{di}, y_{di})
v_i	Velocidad pronosticada del segmento de ruta i	m/seg	$v_{i=\frac{av_r+bv_{ai}}{a+b}}$
T	Tiempo	seg	Ec. 6

Subíndices**Superíndice Término**

a, b	Factores de ponderación
------	-------------------------

n	Nodos de la ruta
---	------------------

Abreviaturas**Abreviatura Término**

app	Aplicación móvil
-----	------------------

DVS	Discapacidad visual
-----	---------------------

1. Planteamiento del problema de investigación y su justificación

1.1 Descripción del área problemática

La inclusión generalizada de la población en condición de diversidad funcional viene acompañando a la sociedad desde los últimos años. Políticas de inclusión de personas en condición de discapacidad en un entorno social ha generado una sucesión de normas que han impulsado el desarrollo de herramientas de movilidad y diseños arquitectónicos, que consideran desde el momento inicial la diversidad de la población en su conjunto, de manera que las características de accesibilidad quedan generalizadas en el proyecto desde su inicio.

El mercado ha demostrado que no es capaz de propiciar un acceso igualitario a las telecomunicaciones para todas las personas, ahora bien, a pesar de los avances, si particularizamos la población con problemas visuales, es una población que se enfrenta diariamente a dificultades en la selección de su medio de transporte alargando el tiempo de traslado y dificultando aún más su movilidad.

En los últimos años en Colombia se han presentado avances con respecto a la inclusión de las personas con discapacidad. La creación del INCI (Intituto Nacional para ciegos) que se hallaba bajo el control y vigilancia del Ministerio de Salud Pública y con la inspección del Ministerio de Educación Nacional “Decreto Ley 1955 de 1955”, es un punto de partida de la inclusión y el apoyo de las personas con discapacidad visual.

En este sentido se han creado varias leyes y decretos que están encaminados a definir mecanismos de integración social de las personas con discapacidad y dictan parámetros para el diseño y construcción de los elementos del espacio público y su accesibilidad, así como establece la normatividad sobre la accesibilidad a los modos de transporte y la movilización en

estos medios, de la población en general y en especial de todas aquellas personas con discapacidad.

Aunque en el mercado existen herramientas que permiten a la población con discapacidad visual interactuar con el entorno, no se tiene en la ciudad de Manizales, una aplicación que esté asociada al servicio público y que se oriente a las personas con este tipo de discapacidad, los desarrollos construidos, solo incluyen el acceso a las plataformas de manera visual, dificultando el uso a personas discapacitadas y analfabetas, dichas aplicaciones móviles no generan alarmas auditivas o sensoriales que permitan al usuario reconocer rutas y paradas que le permita tener autonomía en su movilidad.

Sumado a lo anterior, al ser servicio de transporte público de Manizales concesionado, no es posible estandarizar un desarrollo que aglutine todas las empresas, debido a políticas internas de cada una de estas. Este limitante podría llevar a que solo algunas rutas puedan integrarse a la aplicación móvil, necesitando entonces que las políticas públicas regionales y nacionales, reglamenten su uso a fin de que la inclusión sea total.

Es indispensable que los trabajos que se desarrollen a futuro cuenten con un desarrollo más estructurado, tomando como base un estudio de las necesidades de la población en condición de discapacidad visual brindándoles la oportunidad de tener independencia y autonomía al movilizarse y además de poder permitirles un uso masivo de la tecnología disponible.

1.2 Formulación del problema

Existe una app que permita el acceso al servicio público de manera específica y segura para las personas con discapacidad visual en la ciudad de Manizales.

1.3 Justificación

El cambio de paradigma en las actitudes y enfoques respecto a la inclusión de la población con discapacidad y la posibilidad que nos brindan las tecnologías de la información permiten el desarrollo de alternativas y soluciones que mejoren su calidad de vida.

Las personas con discapacidad tienen el mismo derecho que las personas sin discapacidad, implementar la igualdad de oportunidades en todos los aspectos, social, religioso que le permitan participar plenamente en todas las actividades de la sociedad.

Como partícipe de la sociedad y en apoyo de la tecnología de la información como un factor fundamental en la disminución de la exclusión social se podrá contribuir a la movilidad de la población en condición de discapacidad visual.

Siendo Manizales el punto de estudio se considerará una ruta que permita definir como se orientaran con la aplicación en función del grado de movilidad de esta población.

El desarrollo de una alternativa que contribuya al beneficio de la población en general tanto discapacitada como no discapacitada requiere de la articulación de varias empresas que estén dispuestas a contribuir a la movilidad y al acceso de la población en general de un servicio más efectivo que entienda el comportamiento humano e identifique patrones sobre su movilización permitiendo a las autoridades de tránsito mejorar los tiempos de transporte público optimizando el servicio.

La app es un prototipo de sistema para la orientación de personas en condición de discapacidad visual, con el fin de promover la inclusión social de este sector de la población en el sistema integrado de transporte público de la ciudad de Manizales.

Se propone el uso de una tecnología móvil ya que el desarrollo tecnológico y el fácil acceso a la tecnología Android, ayudará a que las personas en condición de discapacidad visual obtengan esta aplicación y puedan acceder de forma fácil al servicio de transporte público.

2. Objetivos

2.1 Objetivo general

Desarrollar una aplicación móvil con una ruta de prueba (Villa-Pilar la _Enea) que permita a las personas en situación de discapacidad visual obtener el servicio de transporte correcto en la ciudad de Manizales.

2.2 Objetivos específicos

- Realizar el levantamiento de requerimientos para la aplicación siguiendo el estándar IEEE 830.
- Realizar el análisis y diseño de la aplicación.
- Desarrollar un servicio que consuma la información geográfica asociada a una ruta de prueba de una empresa de servicio público en la ciudad de Manizales.
- Implementar un versión de prueba de la aplicación móvil para Android.

3. Antecedentes

Tomando como base los objetivos específicos del trabajo se hace una relación de los trabajos de investigación que sirven de soporte. Se hace una revisión de la norma IEEE 830 – 1993 para evaluar las prácticas recomendadas para el desarrollo de la aplicación.

Como parte fundamental en el desarrollo de la aplicación es considerar la población objetivo por lo que se retoman varias experiencias en relación con la participación de las personas con discapacidad visual en el entorno de los SIG y las aplicaciones móviles y como es la incidencia del estrés asociado al transporte público de personas con discapacidad Visual.

Posteriormente se hace un seguimiento a trabajos relacionados con el objetivo general encaminados a mejorar la autonomía y la participación de la población con discapacidad visual.

Al tener un panorama más amplio y entender bien el impacto de nuestro desarrollo en la población objetivo procedemos a la búsqueda de una base de datos que nos permita almacenar las rutas que servirán como soporte de la aplicación.

Como parte fundamental de nuestro desarrollo es nuestra población objetivo y como el usar la herramienta App Inventor nos permite la democratización de los desarrollos en un entorno más didáctico y académico.

IEEE 830 Recommended Practice for Software Requirements Specifications (IEEE Std 830-1993)

Se describe el contenido y las características de los requerimientos específicos de software (SRS). Se recomienda el seguimiento de esta norma la cual está dirigida a especificar los requisitos del software que se desarrollarán, pero también se puede aplicar para ayudar en la selección de productos de software internos y comerciales.

Esta práctica describe los enfoques recomendados para la especificación de los requisitos de software.

Describe el contenido y cualidades de una buena especificación de requisitos de software (SRS) y presenta varios esquemas de SRS de muestra.

Se basa en un modelo en el que el resultado del proceso de especificación de requisitos de software es un documento de especificación completo e inequívoco.

Debería ayudar

- a) A los clientes de software para describir con precisión lo que desean obtener.
- b) A los proveedores de software para comprender exactamente lo que quiere el cliente.
- c) A los individuos para lograr los siguientes objetivos:
 - 1) Desarrollar un esquema de especificación de requisitos de software estándar (SRS).
 - 2) Definir el formato y el contenido de sus especificaciones de requisitos de software específicos.
 - 3) Desarrollar elementos de apoyo locales adicionales, como una lista de verificación de calidad y un manual del SRS.

Para los clientes, proveedores y otras personas, un buen SRS debe proporcionar varios beneficios específicos, tales como:

- Establecer la base de acuerdo entre los clientes y los proveedores sobre lo que el software va a hacer.
- La descripción completa de las funciones que debe realizar el software, especificado en el SRS ayudará a los usuarios potenciales a determinar si el software satisface sus necesidades o cómo se debe modificar el software para cumplir con lo solicitado.
- Reducir el esfuerzo de desarrollo.
- La preparación del SRS obliga a los diversos grupos interesados a La organización del cliente debe considerar rigurosamente todos los requisitos

antes de que comience el diseño y reduce el rediseño posterior, la recodificación y la repetición de pruebas. Una revisión cuidadosa de los requisitos en el SRS puede revelar omisiones, malentendidos e inconsistencias al principio del ciclo de desarrollo cuando estos problemas son más fáciles de corregir.

- Proporcionar una base para estimar costos y horarios.
- Proporcione una línea de base para la validación y verificación.

LAS TIC Y LA DISCAPACIDAD VISUAL (José Antonio Muñoz Sevilla, p. 293-308)

Se hace un estudio general de los productos que utilizan las personas con discapacidad visual para el acceso a la información digital en su vida laboral, estudios y actividades de su vida cotidiana.

Partiendo del concepto de calidad de vida en su ámbito más universal y lo que encierra en sí mismo, aspectos relativos a la felicidad y el bienestar, pero no cabe duda de que va mucho más allá influenciando todos los aspectos de la vida como trabajo, salud, ocio, economía, participación social, etc. En definitiva, se concreta en la percepción positiva o negativa que uno tiene sobre sí mismo.

Es así como la limitación de la actividad y la restricción de la participación se puede extender a todos los ámbitos de la vida de estas personas: aprendizaje, comunicación, movilidad, relaciones interpersonales, etc. Y la forma de luchar contra ella es ofreciendo oportunidades de superación de la limitación visual. Una de estas oportunidades no cabe duda de que es el desarrollo de productos y servicios teniendo en cuenta el diseño universal.

Existen en la actualidad diferentes líneas de investigación y desarrollo que estudian la mejor manera de presentar al sujeto, para su movilidad y orientación, información del entorno en el que se encuentra, por medio tanto de códigos sonoros como táctiles, o bien a través de neuroprótesis visuales. Los sistemas de posicionamiento por satélite, que ya se vienen utilizando en los distintos medios de transporte mecánicos (automóviles, naves aéreas y

marítimas, etc.), se presentan ya como una incipiente realidad de dispositivos de ayuda tecnológica a la orientación.

Algunas investigaciones en marcha en tecnología para discapacidad visual:

Proyectos actualmente en desarrollo en telefonía y dispositivos móviles:

PIRAMIDE (Personalizable Interactions with Resources on Ami-Enabled Mobile Dynamic Enviroments)

El objetivo que pretende este proyecto es hacer que los dispositivos móviles se conviertan en una especie de sexto sentido. Concretamente en el caso de personas ciegas

Proyecto VIABLE (Entorno Colaborativo, Tecnologías Web y Movilidad para la Vida Independiente y la Accesibilidad):

Plataforma inteligente basada en el concepto de red social que sea capaz de ofrecer información sobre los posibles “puntos negros” de una ruta introducida, y en caso de encontrarlos y considerar que existe una ruta alternativa que no suponga un exceso innecesario de distancia a recorrer y que no tenga ninguna incidencia, ofrecerla.

Proyecto HAPTIMAP (Interfaces hápticos, Visuales y de Audio para servicios basados en mapas de localización):

HACIENDO ACCESIBLES A LAS PERSONAS CON DISCAPACIDAD VISUAL LOS SISTEMAS DE POSICIONAMIENTO (J. A. Muñoz Sevilla,C. Blocona Santos, 2011)

Se analiza la utilidad y fiabilidad de los Sistemas de Posicionamiento Global (gps) para los usuarios con discapacidad visual, señalando sus deficiencias y las posibles soluciones a los problemas que plantean.

Tomando como base el proyecto HaptiMap (Haptic, Audio and Visual Interfaces for Maps and Location Based Servi-ces), se definen directrices de diseño y herramientas de accesibilidad que sirvan de ayuda a los desarrolladores de las principales aplicaciones y

servicios basados en el posicionamiento global que permitan el acceso a la información geográfica a todas las personas con discapacidad visual.

El objetivo de este proyecto es el desarrollo de directrices de diseño junto con un juego de herramientas de accesibilidad que sirva de ayuda a los desarrolladores de las principales aplicaciones y servicios basados en posicionamiento global (gps), con el fin de que se diseñen mapas digitales en general más accesibles y fáciles de usar (no solo para personas con discapacidad sino para todos los usuarios). Junto a este juego de herramientas también se desarrollan métodos de diseño e instrumentos que apoyen prácticas de desarrollo existentes, de modo que se haga más fácil integrar el conocimiento y los principios de accesibilidad en el proceso de diseño, integrando estos aspectos desde el comienzo.

ARCHITECTURE OF A MOBILE APP RECOMMENDER SYSTEM FOR PEOPLE WITH SPECIAL NEEDS

(Torres, C. María Isabel, Rodríguez, F. María José, Visitación H. María, Samos José, and Espín Vanesa 2012).

Una arquitectura de una nueva plataforma que soporte un sistema de recomendación basado en ontologías sobre las características de las aplicaciones disponibles en tiendas de aplicaciones para personas con necesidades especiales.

Como centro del trabajo utilizan una app m-RECACC la cual permite categorizar diferentes niveles de discapacidad y la información contextual que puede ser categorizada.

El sistema de recomendación considera el perfil definido por el usuario el cual analiza el nivel de discapacidad, las funciones de las aplicaciones, los intereses de los usuarios y con esta información base, recomienda las aplicaciones más adecuadas disponibles. El usuario puede entonces decidir si instala la aplicación recomendada o no.

El proceso de recomendar aplicaciones a los usuarios requiere que un experto en el dominio de aplicaciones estudie y clasifique las aplicaciones en repositorios según diferentes

criterios. Este experto también categoriza diferentes niveles y tipos de discapacidad y la información contextual que puede ser considerada y categorizada. El resultado de este proceso es la base de conocimientos utilizada por la aplicación m-RECACC.

STRESS ASSOCIATED WITH TRANSPORTATION:

A SURVEY OF PERSONS WITH VISUAL IMPAIRMENTS (Adele Crudden, Jennifer L. Cmar, and Michele C. McDonnall, 2017)

Se presentan los diferentes niveles de estrés de la población con discapacidad visual y su relación con el transporte público.

Se identifica como el estrés relacionado con el transporte incide en la población con discapacidad visual en actividades de entretenimiento o de ocio y visitas a familiares y amigos las cuales suelen ser evitadas.

Por medio de encuestas electrónicas se pudo determinar cuáles eran las rutas que por estrés la población con discapacidad visual trata de evitar (Rutas desconocidas, área sin aceras, caminar en lugares desconocidos), también se determina cuáles son los predictores significativos del estrés que les produce el movilizarse en el transporte público.

Sobre los niveles de estrés identificados asociados con diversas habilidades de Orientación y movilidad en las personas con discapacidad visual y su acción en el transporte se recopiló información útil para los proveedores de servicios, administradores, e individuos con discapacidad visual al recibir, planificar o proporcionar Servicios de Orientación y Movilidad acorde a sus necesidades.

Incorporando la tecnología como dispositivos de orientación y formación virtual de los participantes se puede disminuir el estrés y la ansiedad de los consumidores del servicio público, al involucrar a los consumidores finales en el diálogo sobre el acceso al transporte para actividades sociales y la formación en habilidades para el uso de varios métodos de transporte.

MOBILE ASSISTIVE TECHNOLOGIES FOR THE VISUALLY IMPAIRED (Lilit Hakobyan, Jo Lumsden, Dympna O'Sullivan, Hannah Bartlett, (2013)

Se hace una revisión de las investigaciones y la innovación dentro del campo de las tecnologías de asistencia móvil para las personas con discapacidad visual, destacando la necesidad de una colaboración exitosa entre la experiencia clínica, la informática y el dominio que los usuarios adquieran de los beneficios potenciales de dichas tecnologías.

Inicialmente evalúan que investigaciones se han hecho para hacer que los equipos móviles sean más accesibles para personas con discapacidad visual, luego que aplicaciones de asistencia se encuentran integrados en lo equipos móviles.

La importancia del diseño centrado en el usuario es fundamental a la hora de estructurar cualquier desarrollo por lo que cada persona difiere en sus necesidades, capacidades y actitudes.

Es así como fomentando el uso de herramientas de diseño centradas en el usuario y prácticas que incluyen entrevistas, grupos focales, encuestas, pruebas de usabilidad y procesos de diseño participativo será fundamental para el éxito en cualquier desarrollo tecnológico.

DESIGN AND DEVELOPMENT OF A MOBILE APP FOR ACCESSIBLE BEACH TOURISM INFORMATION FOR PEOPLE WITH DISABILITIES (Diego Mayordomo-Martínez, Juan-Carlos Sánchez-Aarnoutse, Juan M. Carrillo-de-Gea, 2019)

Estudio de caso sobre la accesibilidad a playas en la Región de Murcia, España, que es una de las principales zonas turísticas del país y se analizan y exponen en detalle elementos importantes que permiten el uso accesible de las playas.

El resultado de este trabajo es el desarrollo de una app que permite en tiempo real a las personas acceder a información actualizada de los diferentes servicios de las playas.

Es así como se evalúan e aspectos importantes en el desarrollo de este trabajo:

1. Existen equipos y servicios de fácil acceso a las personas con discapacidad.
2. Definir que deben incluir las playas inclusivas y su clasificación según una escala de usabilidad y conservación.
3. Resultados obtenidos del campo de estudio y el desarrollo de la app.

Con este desarrollo se unen a todas aquellas alternativas creadas y concebidas para desempeñar un papel fundamental en el bienestar e integración en la sociedad de las personas con discapacidad.

SISTEMA DE SEGUIMIENTO MEDIANTE MAPA ONLINE PARA LOS NIÑOS QUE VIAJAN EN TRANSPORTE PÚBLICO (Isaac, B. Lázaro, 2018).

Uso de la Aplicación Informática (App Inventor) como herramienta para el aprendizaje de algoritmos.

Se genera un sistema de seguimiento mediante un mapa online para los niños que viajan en transporte público. Para ello se utilizaron los dispositivos “Beacon” que mediante bluetooth y una aplicación para móvil o tableta electrónica nos permiten llevar el seguimiento de los niños que viajan en transporte público.

La aplicación de este trabajo es, como dice el propio título, llevar el seguimiento de los niños cuando van a montar en el transporte público, de este modo los padres tendrán la total tranquilidad de que el niño entra y sale del transporte público y podrán ver el recorrido que el menor ha llevado a cabo.

Una vez elegido el dispositivo para el que se va a hacer la programación, el sistema operativo utilizado para el uso de la programación, el sistema de geoposicionamiento a utilizar y el dispositivo beacon apropiado para la finalidad del trabajo, se lleva a cabo el diseño de la aplicación.

La finalidad es programar una aplicación, que pueda leer coordenadas del punto donde está situado el dispositivo móvil para que así nos proporcione la dirección y se pueda ver situada en un mapa.

A la vez esta aplicación realiza una detección de los dispositivos bluetooth con la finalidad de poder localizar los beacon y de este modo tener localizados a los menores que acceden al autobús. Seguidamente estos datos que se obtienen a través de la aplicación se deben enviar a los padres o tutores de los menores para poder tener un control sobre ellos.

Para el desarrollo de la aplicación se ha optado por utilizar App Inventor que permite desarrollar aplicaciones móviles de nivel medio en la plataforma Android. Para realizar la programación en esta plataforma no es necesaria la instalación de ningún desarrollador ya que es una herramienta Web mantenida por el Massachusetts Institute of Technology y diseñada para personas que no poseen unos grandes conocimientos de programación para dispositivos móviles.

Este desarrollador Web realiza el diseño de la aplicación en 2 capas, que consiste en separar el funcionamiento de la aplicación en 2 capas diferenciadas, diseñador y editor de bloques. Con esto conseguimos que se pueda modificar la aplicación y los datos no se alteren en ningún momento.

Con esta plataforma podemos diseñar la interfaz y programar funcionalidades de forma bastante sencilla, ya que no utiliza un sistema de programación de código escrito, sino utiliza un concepto denominado como programación visual con bloques. Cada bloque o pieza representa un objeto o función a la que se puede engarzar, como en un puzle, otro tipo de elementos, los llamados argumentos.

DEMOCRATIZING COMPUTING WITH APP INVENTOR

(Wolber David, Abelson Harold, Friedman Mark, 2014).

Este artículo se centra en el impacto que tiene la aplicación APP INVENTOR en el desarrollo de aplicaciones móviles de forma fácil conectando bloques juntos en forma de juguete, en lugar de

escribiendo código.

El diseño en la aplicación se alterna entre el diseñado de componentes, la interfaz de usuario y un Editor de bloques para programar el comportamiento.

Se puede probar las aplicación usando su teléfono / tableta a través de Wi-Fi, o con el emulador proporcionado.

App Inventor fue diseñado para obtener de cero a la aplicación en el menor tiempo posible. La característica clave que permiten que son:

- Una arquitectura de componentes de alto nivel que expone una funcionalidad compleja a través de simples y propiedades, eventos y métodos concretos.
- El manejo de eventos está al frente y al centro.
- Una función de arrastrar y soltar basada en bloques
- Lenguaje de programación que es divertido de usar y eso minimiza los errores de sintaxis y tipo.
- Un modo de "desarrollo en vivo" que permite a los usuarios para ver instantáneamente su aplicación a medida que la desarrollan.
- Experimente con bloques y componentes.
- Puede crear aplicaciones reales que se ejecuten en teléfonos y tabletas.
- Componentes de alto nivel.

App Inventor proporciona "desarrollo en vivo", donde los cambios realizados en el Usuario Diseñador de interfaces o editor de bloques se reflejan inmediatamente en los teléfonos, de modo que ve la aplicación y puede interactuar con él y probarlo instantáneamente en cada etapa de su desarrollo.

El desarrollador incluso puede probar aplicaciones parciales conectando algunos bloques juntos, realizando una operación "Do it" y ver cuál es el resultado. Esto todo significa que los usuarios obtienen comentarios inmediatos sobre sus acciones de la manera más directa y de forma relevante posible.

PLATAFORMA Y APLICACIÓN MÓVIL PARA PROPORCIONAR INFORMACIÓN DEL TRANSPORTE PÚBLICO UTILIZANDO UN DISPOSITIVO EMBEBIDO DE BAJO COSTO

Buele Jorge, Salazar L. Franklin, A. Santiago, Aldás R, Abigail, Urrutia U. Pilar (2018)

Desarrollo de un sistema informativo de transporte público urbano, para mejorar la movilidad de los usuarios. Este sistema consta de una pantalla informativa ubicada en la parada, por donde un bus realiza su recorrido.

Adiciona la reproducción de mensajes auditivos, que promueven la inclusión de personas con discapacidad visual y la población analfabeta. Para una mayor interacción con los pasajeros se ha implementado una plataforma donde se puede apreciar la ruta a seguir, horarios, clima y el tiempo pronosticado hasta que otra unidad de transporte llegue a dicho lugar.

Además de una aplicación intuitiva compatible con teléfonos inteligentes que utilicen sistema operativo Android. Para el procesamiento de los datos se ha utilizado la tarjeta embebida Raspberry Pi 3 como unidad central del sistema. Paralelamente se ha incorporado un servidor web, que permite el procesamiento de información y su almacenamiento en una base de datos.

4. Referente normativo y legal

TITULO	DESCRIPCION
Ley 035 de 2020	Presentado en la Cámara de Representantes para la regulación de la escritura Braille.
Proyecto de Ley 138 de 2019	Ratificar el Tratado de Marrakech, para obtener la exención patrimonial en los libros para ciegos, flexibilizando la ley de derechos de autor.
Ley 2052 del año 2020, artículo 30	Establece que las entidades públicas implementarán los sistemas necesarios para que las personas con discapacidad visual podamos realizar trámites, procesos y procedimientos.
Ley 1955 de 2019 (mayo 25)	<p>“Por el cual se expide el Plan Nacional de Desarrollo 2018 – 2022 “Pacto por Colombia, pacto por la equidad” Artículo 3° PACTOS DEL PLAN NACIONAL DE DESARROLLO Numeral 13 Pacto por la inclusión de todas las personas con discapacidad.</p> <p>Alianza por la inclusión y la dignidad de todas las personas con discapacidad.</p>
Resolución 246 del 31 de enero de 2019, la modificación del artículo 25 de la Resolución 583 de 2018	El Ministerio de Salud y Protección Social implementó la certificación de discapacidad y el Registro de Localización y Caracterización de las Personas con Discapacidad (RLCPD).
Ley 1145 de 2007. Art. 2.	El SND Sistema Nacional de Discapacidad. Es el conjunto de orientaciones, normas, actividades, recursos, programas e instituciones que permiten la puesta en marcha de los principios generales de la discapacidad
La Ley 1618 de 2013	<p>Sistematiza el amplísimo desarrollo existente en relación con los derechos de las personas con discapacidad, a partir de la existencia de varios importantes tratados internacionales sobre la materia ratificados por Colombia, los desarrollos legislativos y la jurisprudencia de la Corte Constitucional.</p> <p>Asigna importantes obligaciones, la mayoría de ellas en cabeza de las autoridades, aunque también, algunas a cargo de los particulares, concebidas bajo la figura de las acciones afirmativas y encaminadas al logro de la igualdad real y efectiva entre las personas con discapacidad y los demás ciudadanos. plantea como objeto el ejercicio efectivo de los derechos de las personas con discapacidad, esto es que no solo se consagra la garantía retórica de los derechos como en anteriores normas, sino, que se avanza en el concepto de ejercer o disfrutar en la práctica del derecho por lo que se acude a las medidas de inclusión, acciones afirmativas, ajustes razonables y a la eliminación de la discriminación por razón de discapacidad, para el logro efectivo de esas consagraciones jurídicas.</p>
LEY ESTATUTARIA 1618 DE 2013 (Febrero 27) D.O 48.717, febrero 27 de 2013	El objeto de la presente ley es garantizar y asegurar el ejercicio efectivo de los derechos de las personas con discapacidad, mediante la adopción por razón de discapacidad, en concordancia con la ley 1346 de 2009
Ley 1346 de 2009	Por medio de la cual se aprueba “Convención sobre los Derechos de las Personas con Discapacidad”, adoptada por la Asamblea General de las Naciones Unidas del 13 de diciembre de 2006

Ley 1346 de 2009 Sentencia de constitucionalidad C-293 de 2010	En su artículo 1 consagra, el propósito de promover, proteger y asegurar el goce pleno y en condiciones de igualdad de todos los derechos de las personas con discapacidad.
DECRETO 1538 DE 2005 (mayo 17) Por el cual se reglamenta parcialmente la Ley 361 de 1997	Reglamenta parcialmente la Ley 361 de 1997, por la cual se establecen mecanismos de integración social de las personas con limitación. Dicta parámetros para el diseño y construcción de los elementos del espacio público y sobre la accesibilidad a edificios abiertos al público y estacionamientos
Decreto 1660 de 2003	<p>“Por el cual se reglamenta la accesibilidad a los modos de transporte de la población en general y en especial de las personas con discapacidad”.</p> <p>El presente decreto tiene por objeto fijar la normatividad general que garantice gradualmente la accesibilidad a los modos de transporte y la movilización en ellos de la población en general y en especial de todas aquellas personas con discapacidad.</p>
Ley 361 de 1997	Por la cual se establecen mecanismos de integración social de las personas con limitación y se dictan otras disposiciones.
Decreto Ley 1955 de 1955	Se crea el INCI (Instituto Nacional para ciegos) que se hallaba bajo el control y vigilancia del Ministerio de Salud Pública y con la inspección del Ministerio de Educación Nacional.
ADAAG	Americans with Disabilities Act Accessibility Guidelines. La norma oficial de los Estados Unidos para el transporte e infraestructura accesible. www.access-board.gov
BANCO MUNDIAL	Varias publicaciones del Banco Mundial son relacionadas con el acceso para personas con discapacidad. www.worldbank.org “Pautas de accesibilidad para Sistemas Integrados de Transporte Masivo”
BENTZEN	Diseño Accesible para los ciegos: El sitio www.accessforblind.org es mantenido por Billie Louise Bentzen y Janet M. Barlow. Sitio con información detallada en inglés sobre señales peatonales accesibles y advertencias táctiles.
Naciones Unidas	<p>Las Naciones Unidas, en la Declaración Universal de los Derechos Humanos en 1948 y en los Pactos Internacionales de Derechos Humanos, han reconocido y proclamado que toda persona tiene los derechos y libertades enunciados en esos instrumentos, sin distinción de ninguna índole.</p> <p>Diseño de lugares accesibles: Guía de diseño y lista de comprobación para la eliminación de las barreras arquitectónicas, preparada por APRODDIS (Lima) y publicada por United Nations Enable: www.un.org/esa/socdev/enable/guiadd/aproddis.html.</p>
ICONTEC	Normas del Instituto Colombiano de Normas Técnicas y Certificación. www.icontec.org.co En este sitio encontraremos normas sobre acceso a edificios, espacios públicos e infraestructura peatonal, incluyendo acceso a estaciones.

5. Referente teórico

Para el desarrollo de este trabajo es importante tener claridad frente al panorama actual de la discapacidad visual en el mundo. Se continúa con la revisión de las herramientas o aplicaciones para la adaptabilidad de la población en condición de discapacidad en el entorno.

Posteriormente revisamos algunas técnicas para el acceso a la información de las personas con discapacidad necesarias para el desarrollo de la app.

Posteriormente se presentan los conceptos asociados al análisis geoespacial utilizado para el desarrollo de la app

Finalmente una descripción de los lenguaje y procedimientos utilizados para su desarrollo.

5.1 Discapacidad Visual a nivel mundial y en Colombia.

Un grupo de expertos en pérdida de visión del estudio de la carga global de la enfermedad junto con colaboradores del GDB 2019 sobre ceguera y discapacidad desarrollan una revisión sistemática y un metanálisis de encuestas poblacionales de enfermedades oculares desde enero de 1980 hasta octubre de 2018. Se incluyeron estudios con muestras representativas de la población y con protocolos de prueba de agudeza visual claramente definidos. Se ajustaron modelos jerárquicos para estimar la prevalencia en 2020 (con intervalos de incertidumbre [UI] del 95%) de discapacidad visual leve (que presenta agudeza visual $\geq 6/18$ y $< 6/12$), discapacidad visual moderada y grave ($< 6/18$ a $3/60$) y ceguera ($< 3/60$ o menos de 10° del campo visual alrededor de la fijación central); y discapacidad visual por presbicia no corregida (presentando visión de cerca $< N6$ o $< N8$ a 40 cm donde la agudeza visual a distancia mejor corregida es $\geq 6/12$). Pronosticamos estimaciones de pérdida de visión hasta 2050.

Es así como dichos resultados definen que en 2020, se estima que 43,3 millones (95% UI 37,6-48,4) de personas eran ciegas, de las cuales 23,9 millones (55%; 20,8-26,8) se estimaron mujeres. Se estimó que 295 millones (267-325) de personas tenían discapacidad visual moderada y grave, de las cuales 163 millones (55%; 147-179) eran mujeres; 258 millones (233-285) tienen discapacidad visual leve, de los cuales 142 millones (55%; 128-157) eran mujeres; y 510 millones (371-667) con discapacidad visual por presbicia no corregida, de los cuales 280 millones (55%; 205-365) eran mujeres. A nivel mundial, entre 1990 y 2020, entre los adultos de 50 años o más, la prevalencia estandarizada por edad de ceguera disminuyó en un 28,5% (-29,4 a -27,7) y la prevalencia de deterioro leve de la visión disminuyó ligeramente (-0,3%, -0,8 a -0,2), mientras que la prevalencia de deterioro de la visión moderado y grave aumentó ligeramente (2,5%, 1,9 a 3,2; no se disponía de datos suficientes para calcular esta estadística para el deterioro de la visión por presbicia no corregida). En este período, el número de personas ciegas aumentó en un 50,6% (47,8 a 53,4) y el número con discapacidad visual moderada y grave aumentó en un 91,7% (87,6 a 95,8). Para 2050, predecimos que 61,0 millones (52,9 a 69,3) de personas serán ciegas, 474 millones (428 a 518) tendrán discapacidad visual moderada y grave, 360 millones (322 a 400) tendrán discapacidad visual leve y 866 millones (629 a 1150) tendrán presbicia no corregida (RA Bourne, 2020, pp 3-12)

Dentro del contexto nacional encontramos que en nuestro país según el Departamento Administrativo Nacional de Estadísticas DANE, para el año 1993 existía una población con discapacidad de 2.02 % del total de la población, que equivalía aproximadamente a 723.160 personas. Estos datos se ubicaban por debajo de los estándares internacionales en un 400%, puesto que lo proyectado por la Organización Mundial de la Salud OMS, nos muestra que la población con discapacidad se encuentra

entre un 10% y 12% para Latinoamérica, llegando hasta el 18% para países en conflicto armado.

Ya para el Censo 2005 se registró que el 6.4% de la población tenía discapacidad en general, correspondiendo al 43,2% de la población con discapacidad visual.

A partir del 9 de enero de 2018, el Departamento Administrativo Nacional de Estadística DANE, desarrolló un nuevo Censo de Población y Vivienda, cerca de doce años después de aquel de 2005.

El Censo de Población y Vivienda 2018 fue una de las principales apuestas del Estado colombiano para mejorar el conocimiento sobre su población, así como los hogares y las viviendas. De acuerdo con la información disponible, el Censo 2018 se comenzó a aplicar de manera virtual a partir del 9 de enero de 2018.

Censo 2018 arrojó un resultado de acuerdo a la escala de medición de la discapacidad del WG, en el CNPV 2018, de 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.

De 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 (P. Dusan, 2018).

En este sentido la prevalencia de la ceguera serán una constante por lo que se requiere del trabajo generalizado de la población que garantice que aquellas personas con Discapacidad Visual puedan integrarse al entorno sin barreras físicas ni mentales. Con base en estos preocupantes datos diferentes gobiernos del mundo adoptaron una nueva resolución en la que se comprometen a realizar mayores esfuerzos para hacer de los servicios de atención oftalmológica una parte integral de la cobertura sanitaria universal y abordar el creciente impacto de la pérdida de la visión.

Colombia participó en la 74ª Asamblea Mundial de la Salud de la Organización Mundial de la Salud y se comprometió a “Alcanzar los objetivos aprobados en la Asamblea Mundial de la Salud número 74 y integrar completamente la atención oftalmológica dentro de los servicios nacionales de salud, incluso a nivel de atención primaria de salud y garantizar que la atención oftalmológica se aborden las necesidades de más personas mediante la prevención, la detección temprana, el tratamiento y la rehabilitación. (Vivas, 2021)

5.2 Accesibilidad de las personas con discapacidad al entorno

Es importante identificar como es el tema de la adaptabilidad de las personas con discapacidad visual en el entorno que permita integrar soluciones que hagan del transporte público un servicio más inclusivo.

Nuestro entorno y espacios físicos se han diseñado y pensado de tal forma que las personas se puedan movilizar con la plena utilización de sus sentidos.

El espacio en el cual vivimos la mayor parte de la vida, el entorno urbano, ha sido pensado para que las personas se movilicen de un lado a otro con la plena utilización de la vista y todas sus extremidades.

Pero estas posibilidades se ven limitadas para personas que presentan algún tipo de discapacidad.

Existen metodologías o estándares que promueven construcciones y políticas más inclusivas, las cuales se fundamentan en un diseño universal que permitan la no discriminación de este gran grupo de población y la plena satisfacción de sus derechos como ciudadanos. (Perez L. 2005).

El diseño universal “busca estimular el diseño de productos atractivos y comerciales que sean utilizables por cualquier tipo de persona. Está orientado al diseño de soluciones ligadas a la construcción y los objetos que respondan a las necesidades de una amplia gama de usuarios”.

En este tipo de diseños se busca obtener una mejor calidad de vida para todos los ciudadanos basándose en:

Igualdad de uso: Fácil de usar y adecuado para todas las personas, independiente de sus capacidades y habilidades.

Flexibilidad Se acomoda a una amplia gama y variedad de capacidades individuales.

Alternativas de uso para diestros y zurdos.

Uso simple y funcional Debe ser simple de entender, sin importar la experiencia, conocimiento, idioma o nivel de concentración del individuo.

Información comprensible Comunicar la información necesaria al usuario, aunque éste posea una alteración sensorial. Usar distintas formas de información (gráfica, verbal, táctil).

Tolerancia al error Reducir al mínimo los peligros y consecuencias adversas de acciones accidentales o involuntarias (proteger, aislar o eliminar aquello que sea posible riesgo).

Bajo esfuerzo físico Disminuir la fatiga física. Permite al usuario mantener una posición neutral del cuerpo mientras utiliza el elemento.

Espacio y tamaño para el acercamiento y uso Espacios de tamaños adecuados para la aproximación, alcance, manipulación y uso, sin importar el tamaño, postura o movilidad del individuo (L. Cabrera, 2013) .

Citado por el mismo autor definimos la accesibilidad como el grado en el que todas las personas pueden utilizar un objeto, visitar un lugar o acceder a un servicio, independientemente de sus capacidades técnicas, cognitivas o físicas. Es indispensable e imprescindible, ya que se trata de una condición necesaria para la participación de todas las personas independientemente de las posibles limitaciones funcionales que puedan tener (L. Cabrera, 2013).

Es indispensable que todas las investigaciones y la formulaciones del transporte público se deben hacer con un ángulo más amplio, respondiendo a las complejas necesidades de movilidad de hoy tomando como base que el transporte público se debe concibir para incluir no solo las redes de carreteras y autobuses, sino también las redes espacio-temporales construidas para personas con algún tipo de discapacidad. La accesibilidad nos permite simplemente ir de un lugar a otro, igualmente condiciona la participación de un individuo en la sociedad (T. McCray, 2007).

Las personas con discapacidad visual merecen igualdad dentro del mundo del transporte, mejorando la disponibilidad de transporte y accesibilidad, se garantizará la igualdad de acceso y al concientizar a la población sobre la discapacidad, incluidos los proveedores de transporte, conductores, y el público, haríamos un largo camino hacia la mejora de la movilidad y la independencia de personas con discapacidad visual (B.Gallagher, P.Hart, C. O'brien, M.Stevenson, A. Jackson, 2010).

5.3 Técnicas para el acceso a la información de las personas con discapacidad Visual.

Existen ayudas técnicas electrónicas y/o informáticas para discapacidad visual las cuales se les denomina productos tiflotécnicos han sido diseñados como productos especiales para ser utilizados por los ciegos y sordo-ciegos, ayudas técnicas para personas que poseen un resto funcional de visión se habla de productos para baja visión, que son ayudas ópticas mecánicas o electrónicas que se utilizan para corregir, mejorar o potenciar la capacidad visual de una persona (J. A. M. Sevilla, pp 296).

Se relacionan algunas técnicas electrónicas e informáticas que ayudarán en el desarrollo de la app.

Instrumentos para acceder a la información en una pantalla de Ordenador:

Programas de ampliación de caracteres. Se trata de programas que permiten a las personas con resto visual funcional ver lo que exhibe la pantalla del ordenador gracias a la ampliación de las partes de la imagen seleccionadas modificando sus atributos de color, tamaño, forma, etc permitiendo una lectura cómodo y fácil.

Lectores de pantalla. Se trata de programas informáticos que permiten el acceso al texto presente en la pantalla del ordenador por medio de su presentación en forma de voz permitiendo acceder a la información presente en la pantalla.

El usuario escucha lo que se le presenta en la pantalla, o bien lo lee a través de alguno de los dispositivos de Braille efímero (línea braille), y puede acceder a la información. Quizás el programa lector de pantalla de mayor venta y auge en el mundo, representante de la filosofía integradora, ha sido el JAWS for WINDOWS (JFW) del grupo empresarial (Freedom Scientific (2021)).

Lectores ópticos autónomos: Son aparatos que permiten el reconocimiento de textos escritos en soporte de papel con salida de la información fundamentalmente en voz. En un mismo dispositivo compacto se integran un escáner para capturar la imagen presentada, una placa de ordenador o cualquier otro tipo de circuitería capaz de alojar el software OCR y el Sistema Operativo y la interfaz, el programa de OCR, un teclado para el manejo de la interfaz, y una salida en síntesis de voz de las palabras reconocidas. Estos productos pueden ser una opción válida en la actualidad para personas que no tengan conocimientos informáticos y no deseen adquirirlos, por ejemplo, personas de edad avanzada.

Reproductores Digitales. Son la versión actual de registro, almacenamiento y reproducción de información; siendo cada vez más pequeños y de mayor capacidad.

Teléfonos Inteligentes: La telefonía móvil ha incrementado notablemente las posibilidades para establecer comunicación, independientemente del lugar y del momento, con familiares, amigos, servicios de emergencia y/o médicos, etc. Esta realidad ha permitido que muchas personas con discapacidad disfruten de mayor autonomía, en igualdad de condiciones ante la tecnología.. Se han desarrollado programas que facilitan la información de la pantalla al usuario mediante mensajes de voz o ampliando el texto de la pantalla en el caso de que el usuario tenga un resto visual funcional.

Ayudas electrónicas para la orientación. En la actualidad todos los dispositivos móviles Smartphone cuentan con sistemas de posicionamiento Global (GPS).

Los sistemas de posicionamiento global que se encuentran inmersos en gran número de dispositivos que permitirán a las personas con discapacidad lograr una orientación y movilidad válida, segura e independiente, así como el conocimiento de la capacidad de procesamiento de la información presentada (J. A. M. Sevilla, pp 296).

5.4 Análisis Geoespacial

El análisis geoespacial es una gran familia de tareas como calcular el área y la distancia y unir puntos a polígonos que permiten a las personas ver patrones o relaciones dentro de sus datos. El análisis espacial se utiliza en muchas industrias: para encontrar la cafetería más cercana, calcular el tiempo de viaje, mostrar estadísticas regionales para el uso de servicios públicos, o para analizar cuestiones de accesibilidad y equidad como el acceso a las urnas. Mapbox Documentacion (2021)

Elementos de representación cartográfica

Dentro del desarrollo de la app se utilizan como base de visualización diferentes elementos de representación cartográfica a la cual se la asocian diferentes entidades espaciales.

Símbolos para indicar la presencia de entidades de un modo puntual (Usuario, Transporte).

Lineas que simbolizan entidades, naturales o artificiales, de forma lineal (ruta).

Polígonos representan objetos poligonales utilizado para definir las áreas de influencia de las rutas, el área de influencia del servicio de transporte y el área de influencia de los usuarios.

Técnicas de análisis utilizadas

Las técnicas de análisis que se utilizaron son: Medición de direcciones, distancias y áreas.

Distancia entre dos puntos

La distancia entre los puntos i y j es también la longitud del segmento recto entre los puntos i y j

$$dd_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1).$$

Perímetro de un polígono

Es la longitud de la línea que define el contorno del polígono.

Longitud máxima del polígono

Distancia máxima entre cualquier par de vértices en la línea que define el contorno del perímetro:

$$Lm = \max (d_{i,j}) \quad (2).$$

Area de un polígono

Es el valor absoluto de la suma de todas las áreas de los trapecios bajo los segmentos que forman el polígono

$$A = \left| \sum_{i=1}^{N-1} At_{i,i+1} \right| \quad (3).$$

donde $At_{i,i+1}$ es el area bajo el segmento formado por los vértices i e $i + 1$.

Funciones que se aplicaron a las geometrías

Buffer(entidad,distancia)

Intersección(polígono,entidad)

Diferencia(entidad,entidad)

Datos GPS y Representación de la Ruta

Como desarrollo de trabajo se definió una ruta entre Villa_Pilar y el barrio la Enea mapeada en ARCGIS PRO y exportada a formato JSON para su análisis.

Los datos GPS de la ruta se definen de forma puntual a un rango de 40 m entre puntos.

Predicción de la hora de llegada del autobús en tiempo real con los datos GPS

Es así como siguiendo la metodología descrita (W. Lin, J. Zeng, pp 6), se puede definir un procedimiento para determinar el tiempo de recorrido de un autobús desde un punto definido hasta la posición del usuario.

Los datos GPS son datos puntuales disponibles solo en determinados intervalos de tiempo. El conocimiento del tiempo que un autobús está en A en un mapa de ruta es necesario pero no suficiente para determinar cuándo el autobús llegará a B. Además de los datos de

ubicación del autobús, la información esencial para estimar cuándo llegaría el autobús a B también debe incluir la dirección en la que se mueve un autobús y la distancia real de viaje del autobús, en lugar de la distancia euclidiana, entre A y B (W. Lin, J. Zeng, pp 6).

Definir estos pasos dentro de la metodología propuesta en el artículo.

- 1, Digitalizar la ruta del autobús en un conjunto de nodos (40 m entre nodos) Aunque la metodología habla de 20 m entre nodos por número y eficiencia en el procesamiento de los algoritmos se desarrollo un trabajo con 40 m.
2. Convertir una ruta de bus bidimensional en una ruta de bus unidimensional con representación de nodo de enlace;
3. Construir trayectorias de bus en un diagrama de tiempo-distancia.

El enfoque de este tercer punto se replantea ya que en el artículo se habla de que los datos de ubicación del bus GPS no presentan un periodo de tiempo fijo lo que en nuestro desarrollo se toma con la posición del nodo y el tiempo de llegada a su área de influencia.

El análisis geoespacial del punto de ubicación del bus y el punto de la ruta permite que se tenga un dato mas preciso que con el proceso de interpolación propuesto en el artículo. Teóricamente, la ubicación del autobús siempre debe caer en un nodo o su zona de influencia de una ruta si el autobús está en servicio para esa ruta. Suponga (x_b, y_b) es la ubicación de un autobús. Para determinar si el bus está en el enlace i , usamos la siguiente expresión:

$$D_i = \sqrt{(x_{ui} - x_b)^2 + (y_{ui} - y_b)^2} + \sqrt{(x_{di} - x_b)^2 + (y_{di} - y_b)^2} - \sqrt{(x_{ui} - x_{di})^2 + (y_{ui} - y_{di})^2} \quad (4).$$

D_i = Medida que representa la cercanía de la ubicación actual del autobús al nodo i .

(x_{ui}, y_{ui}) y (x_{di}, y_{di}) Representan la ubicación del nodo superior y anterior al nodo evaluado.

Según esta lógica, el enlace en el que se encuentra actualmente un bus se identifica mediante el punto i encontrando un enlace tal que D_i es el más pequeño.

La relación con el valor D_i más pequeño se considera como el correspondiente enlace i . Una vez que se hace coincidir un enlace, se proyectan las coordenadas GPS en el enlace y la distancia a cada estación aguas abajo, se determinará en consecuencia sobre la base de la longitud de los enlaces individuales a lo largo de la ruta.

La estimación del tiempo de llegada debe considerar varias variables a la hora de hacer su cálculo como son las condiciones del tráfico, la velocidad del bus.

Las velocidades de viaje están agrupadas por tiempo y se actualizan continuamente para reflejar los cambios en la congestión del tráfico. Por lo tanto, se tiene en cuenta en la hora de llegada del autobús.

Antes de poder predecir la hora de llegada de un autobús a una determinada estación aguas abajo, se necesita su velocidad de viaje promedio a esa estación (W. Lin, J. Zeng, pp 6). En el documento de D. Sun, H. Luo, L. Fu, W. Liu, X. Liao, M. Zhaoi. (2007, p.65), propone un método el cual primero estima la velocidad promedio de viaje a los segmentos de ruta aguas abajo inmediatos, denotados por los nodos i , basados en una estimación de velocidad promedio actualizada dinámicamente de la siguiente manera:

$$v_i = \frac{av_r + bv_{ai}}{a+b} \quad (5).$$

Donde

v_i = Velocidad pronosticada del segmento de ruta i

v_r = Velocidad actual del bus derivada de datos GPS

v_{ai} = Velocidad media histórica del segmento de ruta i en el tiempo y periodo actual.

a, b = Factores de ponderación.

$a = S_{if}$ $b = S_{ib}$

$a = S_{if}$ Distancia desde la ubicación del autobús hasta el final del segmento de la ruta i

$b = S_{ib}$ Distancia desde el comienzo de la ruta del segmento i a la ubicación actual del autobús.

$$T = \frac{S_i}{v_i} + t_{i+1} + t_{i+2} + t_{i+3} + \dots + t_d \quad (6).$$

S_i = Distancia desde la ubicación actual del autobús hasta el final de la ruta segmento i

t_{i+n} = Tiempo de viaje de cada segmento, estimado en base al promedio de las velocidades de viaje de segmentos de ruta individuales en el período actual.

n = Número de tramos de ruta antes de llegar a la estación.

t_d = Suma del tiempo de permanencia promedio en cada estación de autobuses, podría estimarse sobre la base de datos históricos. Datos de tiempo de permanencia para diferentes horas del día.

Con este procedimiento obtendremos un modelo de predicción basado en las velocidades promedio de los segmentos de ruta .

5.5 Software

Formato JSON

JavaScript Object Notation (JSON)

Según MSDN, el sitio *web* oficial para desarrolladores de Microsoft, Microsoft Documentacion (2021), JSON es un formato de datos textuales popular que se utiliza para intercambiar datos en aplicaciones web y móviles modernas.

Json es un formato de codificación de datos eficaz que permite intercambios rápidos de cantidades pequeñas de datos entre los exploradores de cliente como Internet Explorer, Google Chrome y servicios web, por lo cual se le considera una gran librería de desarrollo web para la transmisión de datos (M.Castillo, J. A. 2016).

(Ericksson, M., Hallberg, V. 2011) indica en su artículo que JSON fue especificado e introducido por Douglas Crockford en 2001, utilizándolo en su compañía. Crockford no fue el creador pero sí fue el primero en darle una especificación completa.

Ahora bien, si se revisa el artículo “Comparison of JSON and XML Data Interchange Formats”, JSON fue diseñado para ser un lenguaje de intercambio de datos comprensible para los seres humanos, fácil de interpretar y usar por las computadoras. JSON es directamente soportado por el lenguaje de programación JavaScript y es la mejor opción para el intercambio de datos.

Además, se estima que JSON puede intercambiar información más rápido que otras tecnologías, tales como las librerías de XML en navegadores modernos.

La aplicación de este formato la utilizamos al momento de construir la Ruta, como lo podemos ver en la (Figura 1), la cual por manejo y facilidad de información permite un gran número de datos

Como se utilizó Json para codificar la Ruta, posición.

Figura 1.
Estructura de datos de la Ruta

```
"features" : [
  {
    "type" : "Feature",
    "id" : 1,
    "geometry" : {
      "type" : "Point",
      "coordinates" : [
        -75.527359804999833,
        5.0875078340001778
      ]
    },
    "properties" : {
      "OBJECTID" : 1,
      "Ruta" : "La Enea"
    }
  },
  {

```

HTML

MND Web Docs definen HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) como el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras

tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (Red Informática Mundial). HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Las marcas HTML incluyen "elementos" especiales como `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, ``, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>`, ``, ``, `` y muchos otros.

Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla. Por ejemplo, la etiqueta `<title>` se puede escribir como `<Title>`, `<TITLE>` o de cualquier otra forma.

Como base para el desarrollo de los algoritmos e el procesamiento de la posición del Bus con relación al usuario se hicieron manejando HTML.

JavaScript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo,

dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional). Lee más en acerca de JavaScript.

El estándar para JavaScript es ECMAScript. A partir del 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores viejos soportan al menos ECMAScript 3. Desde Junio 17, 2015, ECMA International publico la sexta versión principal de ECMAScript, que oficialmente se llama ECMAScript 2015, y que inicialmente se denominó ECMAScript 6 o ES6. Desde entonces, los estándares ECMAScript están en ciclos de lanzamiento anuales. Esta documentación hace referencia a la última versión preliminar, que actualmente es ECMAScript 2020 (JavaScript Documentacion (2021)).

FireBase

Firestore se considera una plataforma de aplicaciones web. Ayuda a los desarrolladores para construir aplicaciones de alta calidad. Almacena los datos en Formato de notación de objetos JavaScript (JSON), no utiliza consulta para insertar, actualizar, eliminar o agregar datos.

Está el backend de un sistema que se utiliza como base de datos para almacenar y sincronizar datos con nuestra base de datos NoSQL alojada en la nube. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen disponibles cuando la app no tiene conexión (C.Khawas, 2018).

Como servicios disponibles Firebase cuenta con los siguientes:

Firestore Analytics: Proporciona información sobre el uso de la aplicación.

Firestore Cloud Messaging (FCM): Solución multiplataforma para mensajes y notificaciones para Android, Aplicaciones Web y iOS.

Firestore Auth: Servicio que puede autenticar a los usuarios usando solo el código del lado del cliente y es un pago de servicio.

Firestore Storage: Facilita la transferencia de archivos fácil y segura independientemente de la red para las aplicaciones de Firebase.

Firestore Test Lab for Android Proporciona infraestructura basada en la nube para probar aplicaciones Android.

Firestore Crash Reporting: Genera informes detallados de los errores generados en la aplicación.

Firestore Notifications: Habilita notificaciones de usuarios específicos para aplicaciones móviles (Firestore Documentation (2021)).

Para el desarrollo de app se utiliza el servicio Real-time Database.

Firestore Realtime Database es una base de datos NoSQL basada en la nube que sincroniza datos entre todos los clientes en tiempo real y proporciona funcionalidad fuera de línea. Los datos se almacenan en tiempo real base de datos como JSON, y todos los clientes conectados comparten una instancia, recibiendo automáticamente actualizaciones con los datos más recientes (L. Moroney, 2017, p.51)

ArcGIS Pro

Aplicación basada en cintas. Desde la cinta situada en la parte superior de la ventana ArcGIS Pro hay muchos comandos disponibles; las funciones más complejas o especializadas se encuentran en los paneles (ventanas acoplables) que se pueden abrir cuando se necesitan.

Permite recopilar, organizar, administrar, analizar, compartir y distribuir información geográfica.

ArcGIS Pro le permite almacenar varios elementos, tales como mapas, diseños, tablas y gráficos, en un único proyecto y trabajar con ellos a medida que los necesita. La aplicación también responde contextualmente al trabajo del usuario. Las pestañas de la cinta cambian en

función del tipo de elemento con el que se trabaja. (Pagina oficial ARCGIS (Plataforma ArcGIS (sigsa.info)).

Se hace uso de esa herramienta para el mapeo de los puntos que definen la ruta y su posterior generación en formata Json.

Leaflet

Biblioteca JavaScript de código abierto líder para mapas interactivos compatibles con dispositivos móviles.

Leaflet está diseñado teniendo en cuenta *la simplicidad, el rendimiento y la usabilidad*. Funciona de manera eficiente en todas las principales plataformas de escritorio y móviles, se puede ampliar con muchos complementos, tiene una API hermosa, fácil de usar y bien documentada y un código fuente simple y legible (Leaflet Documentation (2021)).

Se hace uso de esta herramienta por la facilidad que brinda para integrar los formatos Json que se construyen como rutas y las herramientas de Geoprocesamiento de TURF JS (A. Morales. 2021).

TURF.JS

Es una biblioteca JavaScript de código abierto que le permite realizar operaciones espaciales en el navegador. Turf le ayuda a analizar, agregar y transformar datos para visualizarlos de nuevas maneras y responder preguntas avanzadas (Turf Documentation (2021)).

Es con el uso de esta herramienta que podemos hacer los diferentes análisis que nos permitirán desarrollar con mayor precisión la aplicación.

turf.point

Crea una función de punto a partir de una posición.

Ejemplo:

```
var point = turf.point([-75.343, 39.984]);
```

turf.polygon

Crea una función de polígono a partir de una matriz de anillos lineales

Ejemplo:

```
var polygon = turf.polygon([[[[-5, 52], [-4, 56], [-2, 51], [-7, 54], [-5, 52]]], { name: 'poly1' }]);
```

turf.buffer

Calcula un búfer para las funciones de entrada para un radio determinado. Las unidades soportadas son millas, kilómetros y grados.

Ejemplo:

```
var point = turf.point([-90.548630, 14.616599]);  
var buffered = turf.buffer(point, 500, {units: 'miles'});
```

turf.booleanPointInPolygon

Toma un punto y un polígono o multipolígono y determina si el punto reside dentro del polígono. El polígono puede ser convexo o cóncavo. La función tiene en cuenta los agujeros.

Ejemplo:

```
var pt = turf.point([-77, 44]);  
var poly = turf.polygon([[  
  [-81, 41],  
  [-81, 47],  
  [-72, 47],  
  [-72, 41],  
  [-81, 41]  
]]);
```

```
turf.booleanPointInPolygon(pt, poly);
```

turf.area

Toma una o más características y devuelve su área en metros cuadrados.

Ejemplo:

```
var polygon = turf.polygon([[[[125, -15], [113, -22], [154, -27], [144, -15], [125, -15]]]]);
```

```
var area = turf.area(polygon);
```

turf.distance

Calcula la distancia entre dos puntos en grados, radianes, millas o kilómetros. Esto utiliza la fórmula de Haversine para dar cuenta de la curvatura global.

```
var from = turf.point([-75.343, 39.984]);
```

```
var to = turf.point([-75.534, 39.123]);
```

```
var options = {units: 'miles'};
```

```
var distance = turf.distance(from, to, options);
```

turf.intersect

Toma dos geometrías poligonales o multipolígonos y encuentra su intersección poligonal. Si no se cruzan, devuelve null.

Ejemplo:

```
var poly1 = turf.polygon([[ [-122.801742, 45.48565],  
[-122.801742, 45.60491],  
[-122.584762, 45.60491],  
[-122.584762, 45.48565],  
[-122.801742, 45.48565]  
]]);
```

```
var poly2 = turf.polygon([[  
  [-122.520217, 45.535693],  
  [-122.64038, 45.553967],  
  [-122.720031, 45.526554],  
  [-122.669906, 45.507309],  
  [-122.723464, 45.446643],  
  [-122.532577, 45.408574],  
  [-122.487258, 45.477466],  
  [-122.520217, 45.535693]  
  ]]);  
var intersection = turf.intersect(poly1, poly2);
```

Esta información se toma de la página oficial de Turf.JS ([Turf.js | Advanced geospatial analysis \(turfjs.org\)](http://turfjs.org)).

APP INVENTOR

App Inventor es una aplicación originalmente desarrollada por Google y mantenida ahora por el Instituto de Tecnología de Massachusetts. Permite que cualquier persona, incluyendo las no familiarizadas con la programación y SDK de Android, pueda crear aplicaciones de Software para Android. Utiliza una interfaz gráfica, muy similar al Scratch y el StarLogo, que permite a los usuarios arrastrar y soltar objetos visuales para crear una aplicación que puede ejecutarse en el sistema Android. Google puso fin al desarrollo el 31 de diciembre de 2011 cediéndole el código al MIT, quién lo ha puesto a disposición de todos.

Se trata de una utilidad Web desarrollada por Google que permite realizar aplicaciones para Android sin escribir código Java, todo de forma visual e intuitiva (uniendo piezas de un puzzle). Permite a cualquiera crear aplicaciones de software para el sistema operativo Android.

Utiliza una interfaz gráfica que permite a los usuarios arrastrar y soltar objetos visuales para crear una aplicación que puede ejecutarse en el sistema Android, que funciona en muchos dispositivos móviles. Todo ello sin usar ni una sola línea de código, de forma intuitiva y gráfica.

El editor de bloques utiliza la biblioteca Open Blocks de Java para la creación de lenguajes de programación visuales. Open Blocks esta distribuida por el Massachusetts Institute of Technology Program 's Scheller para formación de profesorado y deriva de la investigación de la tesis de Ricarose Roque. El profesor Eric Klopfer y Daniel Wendel del Programa Scheller apoyaron la distribución de bloques abiertos bajo la licencia MIT.

La programación de bloques abiertos y visual está estrechamente relacionada con el StarLogo, un proyecto de Klopfer, y Scratch, un proyecto de la MIT. Estos proyectos están formados por teorías del aprendizaje construccionista, que hace hincapié en que la programación puede ser un vehículo para conseguir ideas de gran alcance a través del aprendizaje activo. Como tal, es parte de un movimiento continuo en las computadoras y la educación que se inició con el trabajo de Seymour Papert y el Grupo de Logo del MIT en 1960, y también se ha manifestado con el trabajo de Mitchel Resnick, Lego Mindstorms y StarLogo. El equipo de App Inventor fue dirigido por Hal Abelson y Mark Friedman (Ramírez B. Kryscia. Daviana (2013)).

Para el desarrollo de nuestra app hacemos uso de los siguientes componentes los cuales pueden tener métodos, eventos y propiedades. La mayoría de las propiedades pueden ser cambiadas por las aplicaciones: estas propiedades tienen bloques que puede usar para obtener y establecer los valores. Algunas propiedades no pueden ser cambiadas por las aplicaciones, estas solo tienen bloques que puede usar para obtener los valores, no para establecerlos (App Inventor Documentation (2021)).

Botón

Botón con la capacidad de detectar clics. Se pueden cambiar muchos aspectos de su apariencia, así como si se puede hacer clic en él (). Sus propiedades se pueden cambiar en el Diseñador o en el Editor de bloques.

Casilla de verificación

CheckBox los componentes pueden detectar los toques del usuario y pueden cambiar su estado booleano en respuesta.

Un componente genera un evento cuando el usuario lo pulsa. Hay muchas propiedades que afectan a su apariencia que se pueden establecer en el Diseñador o en el Editor de bloques.CheckBox

Etiqueta

Las etiquetas son componentes utilizados para mostrar texto.

Una etiqueta muestra el texto especificado por la propiedad. Otras propiedades, todas las cuales se pueden establecer en el Diseñador o en el Editor de bloques, controlan la apariencia y la ubicación del texto.Text

TextBox

Los usuarios escriben texto en un componente de cuadro de texto.

Los cuadros de texto se utilizan normalmente con el componente Button, con el usuario haciendo clic en la entrada de texto cuando se completa.Button

Si el texto introducido por el usuario no debe mostrarse, utilice PasswordTextBox en su lugar.

WebViewer

Componente para la visualización de páginas Web.

HomeUrl se puede especificar en el Diseñador o en el Editor de bloques. La vista se puede configurar para que siga los vínculos cuando se pulsan y los usuarios pueden rellenar formularios Web.

Advertencia: Este no es un navegador completo. Por ejemplo, al presionar la tecla Atrás del hardware del teléfono, saldrá de la aplicación, en lugar de retroceder en el historial del navegador.

Puede usar la propiedad `WebViewString` para comunicarse entre la aplicación y el código Javascript que se ejecuta en la página. En la aplicación, obtiene y establece `WebViewString`. En el , se incluye Javascript que hace referencia al objeto, utilizando los métodos

```
WebViewerWebViewerwindow.AppInventor.getWebViewString()
```

```
WebViewerWebViewerwindow.AppInventor.setWebViewString(text)
```

Reloj

Componente no visible que proporciona el instante en el tiempo utilizando el reloj interno del teléfono. Puede activar un temporizador a intervalos establecidos regularmente y realizar cálculos de tiempo, manipulaciones y conversiones.

Mapa

Un contenedor bidimensional que representa los mosaicos del mapa en segundo plano y permite que varios elementos `marker` identifiquen puntos en el mapa. Los mosaicos de mapas son suministrados por los colaboradores de OpenStreetMap y el Servicio Geológico de los Estados Unidos.

GiroscopioSensor

Componente que proporciona datos del sensor del giroscopio del dispositivo.

LocationSensor

Componente no visible que proporciona información de ubicación, incluida la latitud, la longitud, la altitud (si es compatible con el dispositivo), la velocidad (si es compatible con el dispositivo) y la dirección. Esto también puede realizar una "geocodificación", convirtiendo una

dirección dada (no necesariamente la actual) a una latitud (con el método `LatitudeFromAddress`) y una longitud (con el método `LongitudeFromAddress`).

TextToSpeech

El componente habla un texto dado en voz alta. Puede establecer el tono y la velocidad de habla. `TextToSpeech`

SpeechRecognizer

Utilice un componente para escuchar al usuario hablando y convierta el sonido hablado en texto utilizando la función de reconocimiento de voz del dispositivo. `SpeechRecognizer`.

Player

Componente multimedia que reproduce audio y controla la vibración del teléfono. El nombre de un archivo multimedia se especifica en la propiedad `Source`, que se puede establecer en el Diseñador o en el Editor de bloques. El período de tiempo para una vibración se especifica en el Editor de bloques en milisegundos (milésimas de segundo) (App Inventor Documentation (2021)).

6. Metodología

Para la ejecución de la aplicación móvil se estableció una especificación de requisitos de software (ERS), estructurándola bajo las directrices dadas por el estándar IEEE Práctica recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

6.1 Descripción General

Perspectiva del producto.

Será una app que funcione en entornos Android lo que permitirá su uso a un gran grupo de personas que usan este sistema, conectándose a una Base de Datos en línea (Firebase, 20221), permitiendo una mejor y rápida respuesta.

Funcionamiento del producto

Solicitud de forma verbal de la ruta en su dispositivo móvil y después de el postproceso hecho por la aplicación este dará una señal de forma verbal del tiempo de la disponibilidad de la ruta, tiempo, y bus asignado.

Características de los usuarios

Tipo de usuario: Personas con discapacidad visual.

Formación: General

Actividad: Uso de la app para el acceso al servicio público en la ciudad de Manizales

Restricciones

- Interfaz para ser usada con datos móviles
- La respuesta del servidor a la BD puede verse afectado por el tráfico.
- La app es una propuesta de aplicación móvil para el acceso al servicio público. de las personas con discapacidad visual.
- Qué tan accesible es para una persona con discapacidad visual acceder a esta herramienta.

- El costo de los equipos impediría acceder a la herramienta.
- La app deberá ser un diseño sencillo y de fácil uso.
- Normatividad asociada.
- Definición de los casos de uso.
- Existen suficientes aplicaciones para personas en condición de discapacidad visual.

Requisitos específicos

Autenticación del usuario

La aplicación móvil solo podrá ser consultada por un usuario que se ha registrado previamente.

La aplicación permitirá acceder al servicio a partir de una secuencia de comandos de forma verbal.

Requisitos comunes de las interfaces

Interfaces de usuario

La interfaz de usuario consistirá en una aplicación móvil que se active por la voz (Explorar el uso de Google Scene), la interfaz de usuario consistirá en un mínimo de botones los cuales estarán como soporte de las personas con limitaciones visuales no tan severos. La generalidad y uso de la app se hará por medio de comando de voz.

Interfaces de hardware

Equipos móviles Android

Interfaces de Software

Sistemas Android 5.0 Lollipop o superior

Después de definir el marco metodológico y los requisitos funcionales se procede a definir la ruta que se utilizará como base para prueba de la aplicación.

Se hace una consulta bibliográfica de los desarrollos que se han planteado a nivel nacional frente a las diferentes aplicaciones relacionadas con el transporte público encaminadas a mejorar la movilidad de las personas con discapacidad visual. Lo anterior como punto de proyección de la app. (Torres, C. María Isabel, Rodríguez, F. María José, Visitación H. María, Samos José, and Espín Vanesa 2012).

Se conciben dos apps las cuales corresponderán a la que manejará el servicio público y otra que será la interfaz del usuario.

Par el desarrollado se utiliza el software App Inventor, para celulares inteligentes que posean el sistema operativo Android. Según la página oficial de esta herramienta es compatible con todas las versiones de Android. Wolber David, Abelson Harold, Friedman Mark, (2014).

Esta herramienta permite la generación de aplicaciones móviles de forma fácil e integral ya que gran parte de los módulos que se tienen definidos para el desarrollo de la aplicación se encuentran disponibles (módulo reconocimiento de voz, módulo de conversión voz a texto, módulo de manejo de mapas).

Se tomarán diferentes puntos del recorrido de la ruta y estos estarán en la base de datos como puntos guías los cuales estarán en constante validación por el dispositivo que posee el vehículo como el usuario que hizo el requerimiento de este. Estas consultas a la base de datos se harán en tiempo real y permitirán censar la posición del vehículo, compararla con la base de datos y entregar la información al usuario de su posición.

Como sistema de información geográfica se manejará la plataforma de mapas libre Leaflet (Leaflet es la biblioteca javascript líder en código abierto para mapas interactivos compatibles con dispositivos móviles), Leaflet (2021), el cual viene integrada en la plataforma App Inventor.

Se utilizará como base de datos FIREBASE (plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por Google en 2014). Firebase Documentacion (2021)

Aplicación Usuario

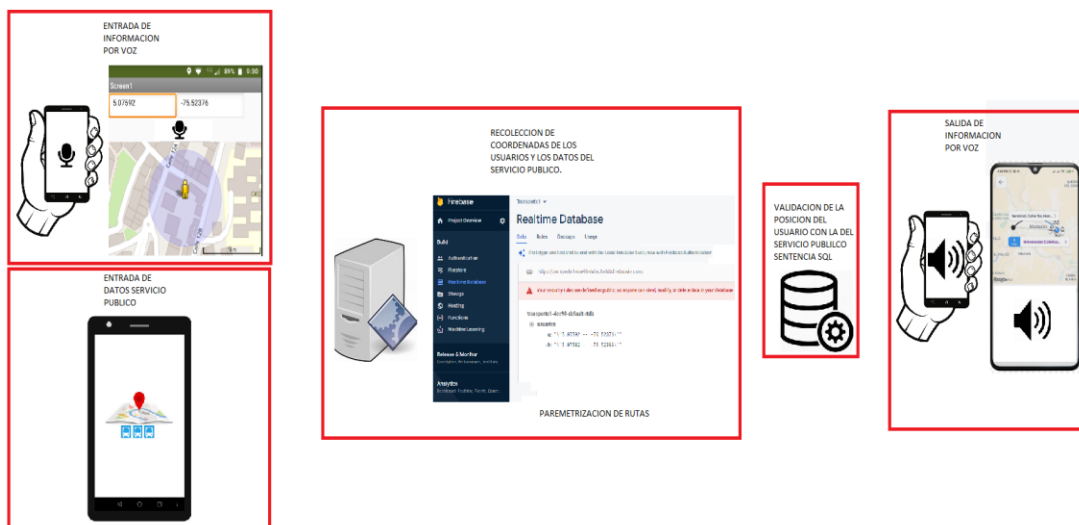
Esta ampliación consta de la definición de la ruta la cual se hará de forma verbal al responder por medio de un comando de reconocimiento de voz.

La aplicación tomará esta orden y la procesara de forma que internamente convierte este comando de voz a texto y con él se define la posición del usuario.

El usuario después de recibir una confirmación de la orden por el impuesta y después de aceptar la app hará la consulta a la base de datos en línea la cual por medio de una sentencia SQL definirá la posición del transporte público y la distancia a la cual se encuentra del usuario. La aplicación generará una señal de audio que orientará al usuario sobre su requerimiento. (Figura 2).

Se finaliza con un prototipo de app.

Figura 2.
Mapa conceptual del funcionamiento de la aplicación



6.2 Procedimiento

Aplicación Servicio Público

Se desarrolló de una aplicación de forma paralela para que el conductor pueda definir la ruta que seguirá.

Este después de registrarse y definir la ruta procede a iniciar su recorrido.

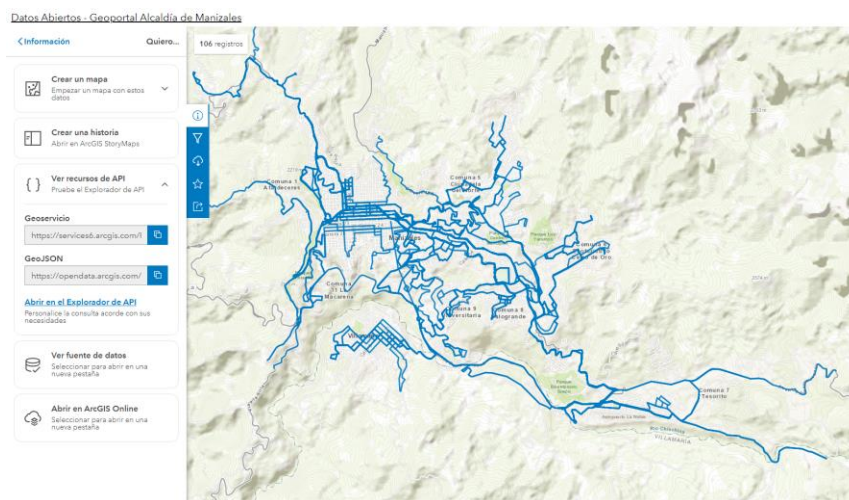
Esta aplicación también se desarrollará en App Inventor y tendrá como base de datos FireBase. Firebase Documentacion (2021)

Para la definición de las rutas se procede a descargar la información presente en la página de la alcaldía de Manizales en su Geoportal datos abiertos.

Se accedió a la página (Datos abiertos – Geoportal Alcaldía de Manizales, Vías y Transporte, 2021), ahí identificamos un mapa en ARCGIS (Figura 3) que nos representa todas las rutas presentes en la ciudad de Manizales.

Se accedió a la pestaña de uso de estos datos en la ventan ver recursos de API procedemos a copiar el enlace presente en el campo GeoJSON el cual al ejecutarlo en cualquier navegador genera un archivo geoJSON.

Figura 3.
Mapa ARCGIS rutas servicio público del geoportal de la alcaldía de Manizales



Es así como se toma la ruta definida dentro del archivo "SENA ENEA - VILLAPILAR" y se hacen unos ajustes de los segmentos de línea que constituyen la ruta los cuales deben estar a un espaciado entre si de 40 m para poder hacer uso de la metodología (W. Lin, J. Zeng, pp 6), para el calculo de tiempo de llegada y distancia entre el bus y el usuario.

Como observamos en la (Figura 5) que la ruta definida para la Enea se hizo transformando los puntos iniciales y finales de cada segmento de línea que compone la ruta filtrada en puntos simétricamente ubicados.

Para poder incluir el archivo json externo en cada uno de los programas html y JavaScript se debe declarar una variable (JavaScript | MDN (mozilla.org)) dentro del archivo generado desde el ARCGIS (Figura 6) que contenga toda la información y después se guarda este archivo en el mismo formato JS el cual será la base de consulta de la aplicación.

Figura 6.
Json con la definición de la ruta de estudio

```

1  var rutas=
2  "type": "FeatureCollection",
3  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
4  "features": [
5  {
6    "type": "Feature",
7    "id": 1,
8    "geometry": {
9      "type": "Point",
10     "coordinates": [
11       -75.527359804999833,
12       5.0875078340001778
13     ]
14   },
15   "properties": {
16     "OBJECTID": 1,
17     "Ruta": "La Enea"
18   }
19 },
20 {
21   "type": "Feature",
22   "id": 2,
23   "geometry": {
24     "type": "Point",
25     "coordinates": [
26       -75.52722916599987,
27       5.0872857480001699
28     ]
29   },

```

Al tener el archivo que sirve de base para la consulta de la ruta se procede a la construcción de los algoritmos que se encargaran de hacer la conexión del usuario la app y los diferentes medios de transporte.

Para poder cargar el archivo JS en el HTML que se usa se deben agregar los siguientes comandos para su uso en el algoritmo.

```
<script src="js/rutas6.js"></script>
```

Y se añade como objeto GeoJson en el javascript que utiliza para el procesamiento:

```
var rutas = L.geoJson(rutas,{});
```

APP USUARIO

Ubicación del usuario en función de la ruta

Para este procedimiento se construye un algoritmo que permite por medio de las funciones definidas en TURF (Librería para realizar análisis geoespacial TURF, Documentacion (2021)) calcular su posición con respecto a la ruta definida previamente.

Al añadir Turf.js al html, se dispone de una variable global Turf desde la que podemos ejecutar cualquiera de sus funciones.

Tomando la posición del usuario (Latitud y Longitud) se construye un objeto geográfico de tipo Point.

```
var point_usuario = turf.point([longitud,latitud]);
```

Para visualizar la información definida en el JS que contiene la ruta de estudio se hace lo siguiente:

```
var rutas = L.geoJson(rutas,{});
```

Como se observa en el código anterior se define una variable ruta que permitirá la conexión del JS con el script.

En la variable rutas tenemos el objeto con toda la información que se definió inicialmente en el JS.

Se realizó una función Buffer al objeto rutas la cual me permite tener una zona de influencia de cada uno de los nodos definidos a lo largo de la ruta.

```
var bufpru = turf.buffer(rutas.toGeoJSON(),25, {units: 'meters'});
```

Se aumenta el tamaño a 25 para que se incluyan entre si los nodos evaluados.

Teniendo definido el objeto bufpru se procede hacer un recorrido de cada uno de estos elementos (zonas de influencia por nodo) previamente generados para identificar cual de estos polígonos cubre el área del punto geográfico del usuario.

Figura 7.
Segmento de código que define las zonas de influencia por nodo.

```
var ru = L.geoJson(bufpru, {
  onEachFeature: function (feature, layer) {
    var polygon = turf.polygon(feature.geometry.coordinates, {});
    var y= turf.booleanPointInPolygon(point_usuario,polygon);
  },
});
```

Para definir esta incidencia del punto sobre cada una de las áreas de influencia se hace con la función de turf punto en polígono (Figura 7).

turf.polygon(feature.geometry.coordinates, {});

Una vez identificado el punto en uno de los polígonos este reporta a la aplicación su posición con respecto a la ruta.

Una vez se cumple esta condición se pasa la información procesada a la aplicación para generar la alerta de ubicación.

Ubicación de los buses en función del usuario

Al cargar el algoritmo de ubicación de buses este tomará de la app la información de la ubicación del usuario suministrada por el celular y la información que descargó de la base de datos implementada en la plataforma FireBase en la nube con la información de los buses.

Igual que el anterior procedimiento se utilizan las herramientas definidas en TURF (Librería para realizar análisis geoespacial TURF, Documentacion (2021)) y así calcular la posición de cada uno de los vehículos con respecto al usuario.

Tomamos la posición del (Latitud y Longitud) y se construye un objeto geográfico de tipo Point.

```
var point_usuario = turf.point([longitud,latitud]);
```

Visualizamos la información definida en el JS (rutas6.js) que contiene la ruta de estudio.

```
var rutas = L.geoJson(rutas,{});
```

Buffer al objeto rutas obteniendo una zona de influencia de cada uno de los nodos definidos a lo largo de la ruta.

```
var bufpru = turf.buffer(rutas.toGeoJSON(),25, {units: 'meters'});
```

Con la información obtenida se procede a definir la ubicación del usuario en la ruta la cual me sirve para compararla con las posiciones de cada bus.

Las posiciones de los buses se establecen en el algoritmo que sirve de soporte a la aplicación del Bus. Esta comunica la posición geográfica del bus (Latitud, Longitud) y su posición en la ruta.

Tomando la información suministrada por la app, latitud y longitud de los usuarios así como las posiciones de los buses se valida cada posición de los buses con la posición del usuario.

Debido a que no se puede tomar una distancia directa entre la posición del usuario y cada bus por lo que la ruta no es una línea recta se define un procedimiento que recorra cada nodo desde la posición del bus a la posición del usuario (W. Lin, J. Zeng, pp 6).

Las posiciones intermedias se toman del objeto previamente calculado (Figura 8).

Figura 8.
Segmento de código que define las coordenadas de cada nodo en la ruta.

```
var rutas = L.geoJson(rutas, {
    onEachFeature: function (feature, layer) {
        coords.push(feature.geometry.coordinates);
    },
});
```

La distancia de cada nodo medido desde la posición del bus y la posición previa se suma hasta que el nodo coincide con la posición del usuario.

```
var distance = turf.distance(point_usuario, point_nodo, options);
```

```
var dist_total=dist_total+distance;
```

Interseccion del usuario con un bus

Esta parte se desarrollo en el mismo algoritmo de ubicación de buses ya que se aprovecha gran parte de la estructura de este para obtener la insterseccion de una de las rutas con el usuario (Figura 8).

Inicialmente se toma la posición del (Latitud y Longitud) del usuario y se construye un objeto geográfico de tipo Point.

```
var point_usuario = turf.point([longitud,latitud]);
```

Figura 9.
Segmento de código que permite la identificación del usuario con respecto a cada bus

```
for (n=5;n<9;n++){
  p1=arra[n];
  var v1=Number(p1);
  var v2=Number(rut_asig);

  var p1 = turf.polygon(bufpru[v1].geometry.coordinates);
  var p2 = turf.polygon(bufpru[v2].geometry.coordinates);

  inter=turf.intersect(p1,p2);

  if (inter != null){
    var q="a" + ",";
  }

  var dist_total=0;
  for (j= v1;j<v2;j++){
    var from = turf.point(coords[j-1]);

    var gg=coords[j];
    var to = turf.point(gg);

    var options = {units: 'meters'};

    var distance = turf.distance(from, to, options);
    var dist_total=dist_total+distance;
  }

  var jl= q+jl+ "," +rut_asig+ "," + v1+ "," + Math.round(dist_total) ;
}
```

Visualizamos la información definida en el JS (rutas6.js) que contiene la ruta de estudio.

```
var rutas = L.geoJson(rutas,{});
```

Buffer al objeto rutas obteniendo una zona de influencia de cada uno de los nodos definidos a lo largo de la ruta.

```
var bufpru = turf.buffer(rutas.toGeoJSON(),25, {units: 'meters'});
```

Teniendo esta información se procedió a definir tanto los puntos de ubicación del usuario como los del bus que me suministra la aplicación y que provienen de la información almacenda en la base de datos.

Como tenemos el objeto bufpru que contiene todos los buffer de los nodos de la ruta y además conocemos las posiciones de los buses podemos hacer por medio de un ciclo una consulta de cada posición del bus con relación a la posición del buffer del usuario y así con la función de Turf de intersección determinar el punto de encuentro de uno de los buses con el usuario.

```
var p1 = turf.polygon(bufpru[posicion_bus[n]].geometry.coordinates);
```

n= iterador para identificar cada bus.

```
var p2 = turf.polygon(bufpru[posicion_usuario].geometry.coordinates);
```

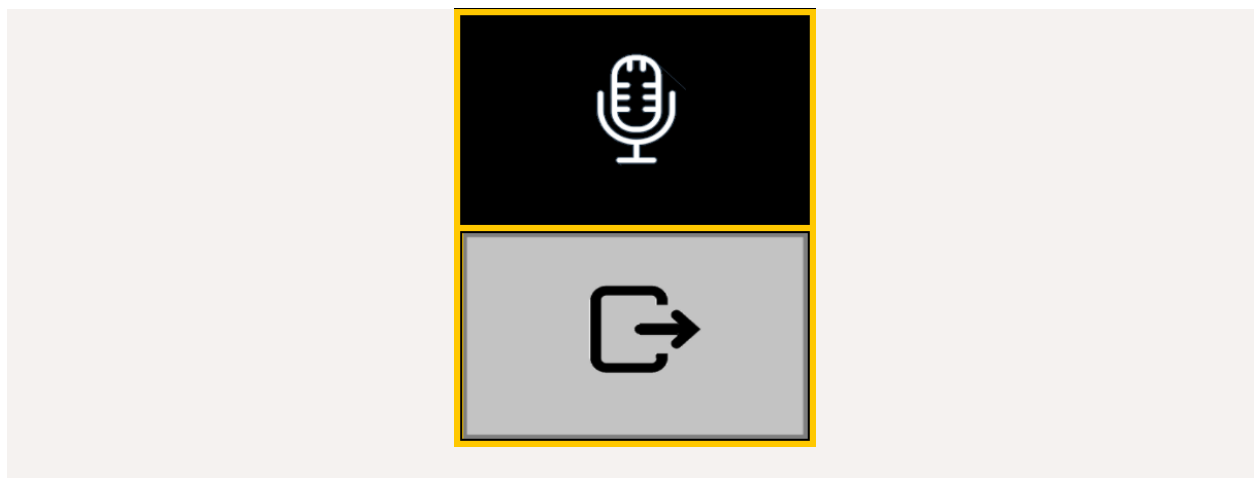
```
inter=turf.intersect(p1,p2);
```

Al tener definido los algoritmos que se encargaran del procesamiento se procedió a la integración con la aplicación.

Para la integración de estos algoritmos con un software de aplicaciones móviles se utilizó el app inventor el cual permite la interactividad y el uso de varias herramientas de los celulares con las herramientas de geoposicionamiento.

La aplicación del usuario consta de 3 Visualizaciones las cuales se representan en 2 introductorias y la final que la aplicación en si y la cual se define en dos botones (Figura 10).

Figura 10.
Pantalla principal de la aplicación.



Boton 1. Consulta de ruta.

Al momento de cargar la imagen de la figura anterior el celular vibrará lo que le permite al usuario saber que ya puede dar la orden.

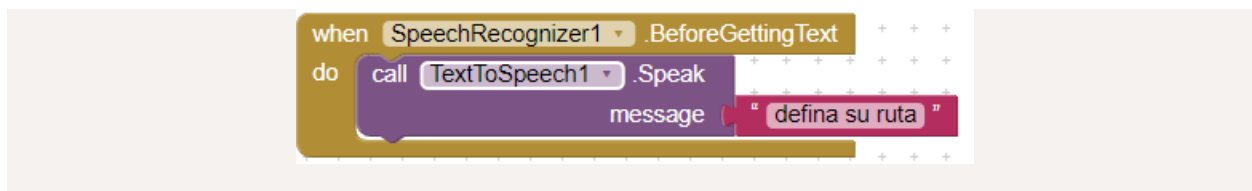
Al habilitarse la pantalla descrita en la figura el usuario podrá reconocer el área que le sirve del celular para hacer su solicitud ya que se habilito el táctil de la mitad hacia arriba de la pantalla para este fin.

Una vez definida esta orden un audio le solicitará al usuario su ruta.

Este procedimiento se hace utilizando la herramienta SpeechRecognizer presente en el app inventor que es un componente que permite escuchar al usuario y convertir su sonido hablado a un texto utilizando la función de reconocimiento de voz del dispositivo (Figura 11).

Se utiliza un evento que después de que SpeechRecognizer haya reconocido la señal transmitida por el usuario entonces se ejecute una vez al final del reconocimiento.

Figura 11.
Bloque SpeechRecognizer.



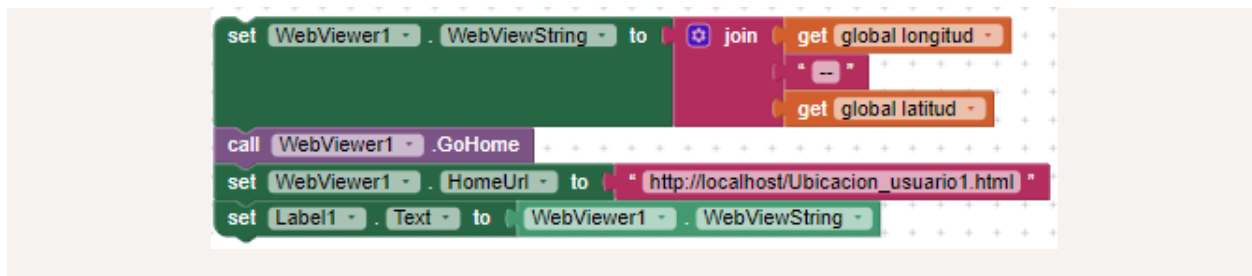
Como la idea era definir un texto antes de que se diera la orden se introduce un TextSpeech componente que transmite un texto en voz alta.

Al activarse el componente SpeechRecognizer se le pregunta al usuario que defina su ruta.

Una vez la ruta es definida la aplicación integra el algoritmo de Ubicación del usuario en función de la ruta por medio del componente WebViewer el cual permite por medio de dos funciones especiales de App Inventor (`AppInventor.getWebViewString()` y `AppInventor.setWebViewString()`) las cuales permiten que las aplicaciones se comuniquen con el javascript que se ejecuta dentro del componente WebViewer.

Teniendo este componente hacemos la conexión con el html y obtenemos la posición del usuario.

Figura 12.
Bloque WebView que recibe la información para procesar en los scripts.



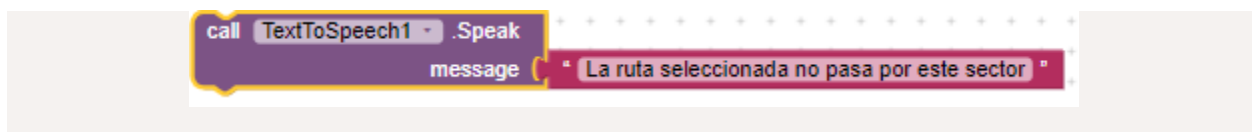
Con la propiedad WebViewString para comunicarse entre la aplicación y el código Javascript que se ejecuta en la página. En esta sentencia definimos la posición del usuario la cual pasara a ser procesada por el algoritmo Ubicación del usuario en función de la ruta.

Un ves obtenida la respuesta que da el procesamiento podemos definir si el usuario se encuentra en la ruta que selecciono o si por el contrario la ruta no pasa por el sector.

Estas sentencias se hacen evaluando si el algoritmo entrego un numero o no el cual corresponde al nodo de ubicación del usuario.

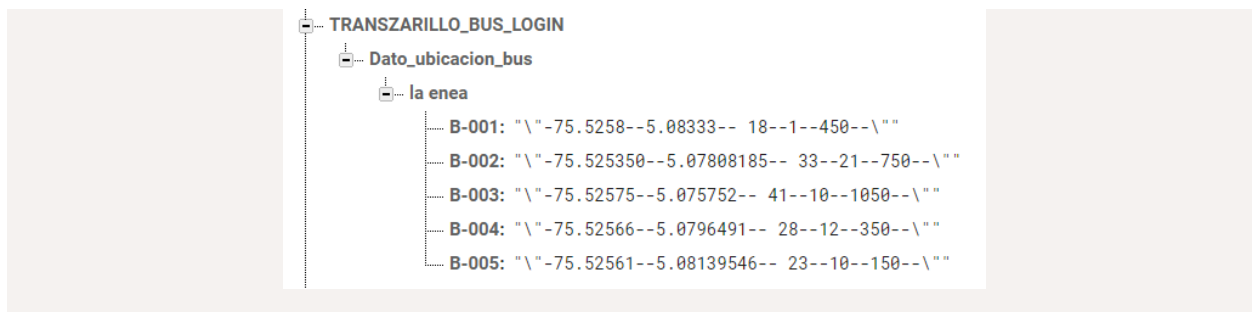
En caso afirmatico continua su procesamiento y consulta a la base de datos, de lo contrario un mensaje de voz indicara que la ruta no pasa por el sector (Figura 13).

Figura 13.
Bloque TextToSpeech para indicar el error de la ubicación del usuario con la ruta.



Paralelamente se esta haciendo una consulta a la base de datos Firebase la cual permite el manejo de una base de datos y en la que almacenamos los datos provenientes del bus y la posición del usuario (Figura 14).

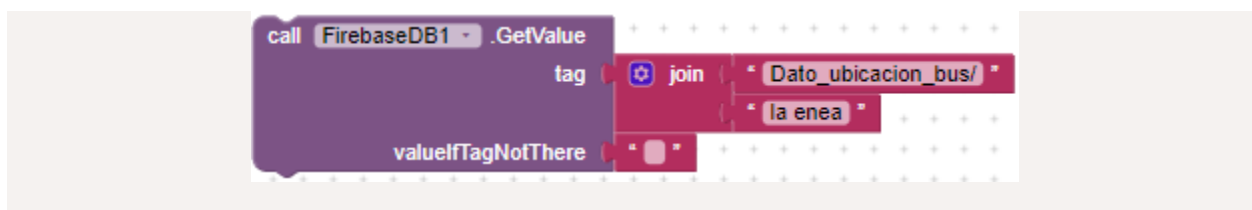
Figura 14.
Estructura de la base de datos en FireBase



Dentro del proceso para la integración de la app y Firebase se siguieron los siguientes pasos.

- Crear una cuenta en Firebase
- Agregar una app (el nombre que se utiliza en App Inventor) en este caso se llama transporte1
- Ver datos de configuración necesarios para App Inventor (Ver la clave en el apartado «Clave de API web»)
- Ver la dirección desde «Realtime Database»
- Configurar las reglas para permitir escribir
- En App Inventor se añade el componente experimental «FirebaseDB» al proyecto
- Configurar las propiedades del componente la misma con la orden inicial dada por el usuario sobre la ruta La Enea (Figura 15).
- Una vez construida la conexión con la base de datos se procede hacer las consultas.

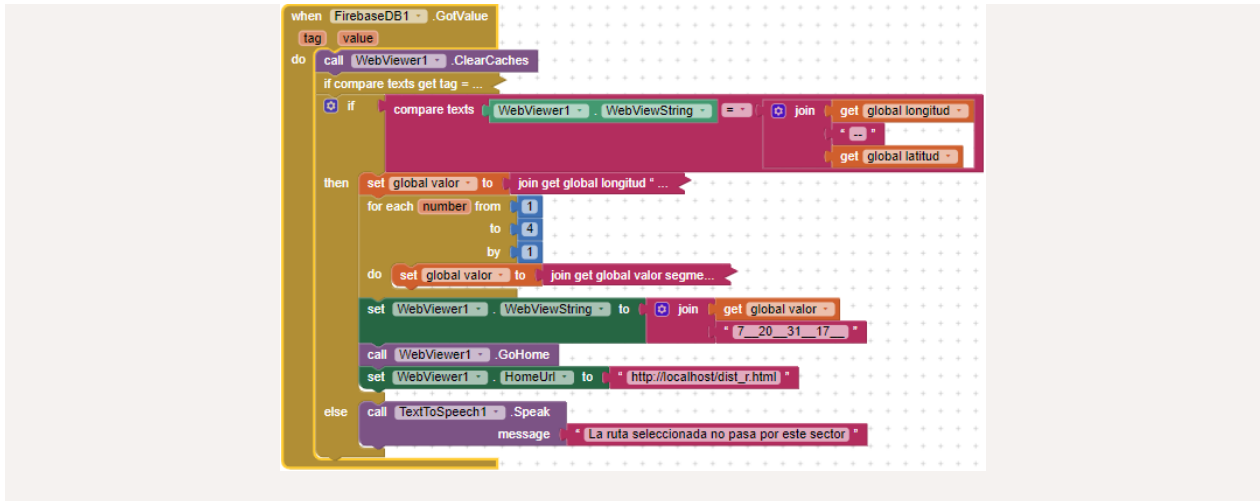
Figura 15.
Bloque de definición de Tag y consulta la base de datos.



Con el bloque anterior la plataforma hace la consulta a la base de datos sobre la ruta la enea.

Una vez procesa esta orden por medio de otro bloque obtenemos la respuesta a la solicitud hecha a la base de datos.

Figura 16.
Bloque de respuesta a la consulta a la base de datos.



Con los datos obtenidos a la consulta a la base de datos los cuales corresponden a las posiciones de los buses definidos en la ruta especificada y con la posición del usuario, se procede a aplicar el algoritmo Ubicación de los buses en función del usuario utilizando el mismo componente WebView (Figura 16).

En su componente WebViewString definimos la información que se entregará al algoritmo.

El algoritmo permite evaluar la intersección del polígono definido al usuario con los polígonos definidos a cada bus y además definir la distancia con cada uno de ellos por lo que la respuesta que se recibe por medio del bloque WebViewStringChange (Figura 17).

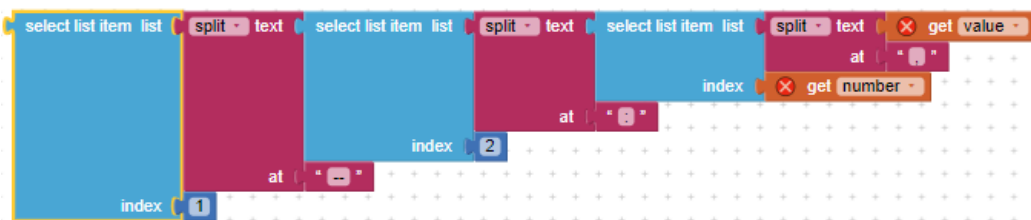
Figura 17.
Bloque que entrega la respuesta del procesamiento hecho con los scripts.



La respuesta que se recibe del algoritmo es una cadena de texto con la siguiente estructura (intersección_bus/usuario+ ruta_usuario+posición_bus+distancia_total_bus/usuario).

Se hace una extracción de cada uno de estos datos y se generan las respectivas alertas.

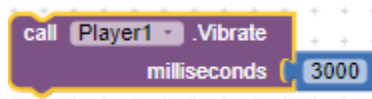
Figura 18.
Bloque que almacena en una lista la información que devuelven los algoritmos



Si al validar la respuesta su primer valor es verdadero que corresponde a la intersección bus/usuario se genera una vibración que le permitirá al usuario saber que su ruta esta cerca.

Esta acción la hacemos utilizando el bloque Player el cual tiene un componente que utiliza la forma de vibración del celular y se programa en milisegundos (Figura 19).

Figura 19.
Bloque que permire que el celular vibre



Las siguientes elementos de la cadena de texto permitirá después de organizar los datos identificar los buses con su distancia y tiempo promedio de llegada al punto de ubicación del usuario.

La información del tiempo de llegada proviene de la información consignada por la app de cada bus y en la que se encuentra la posición del bus (Latitud, Longitud), nodo de ubicación, tiempo promedio y distancia recorrida.

Una vez obtenidos estos datos se organizan de forma que se le transmitan al usuario por medio del comando TextSpeech.

Asi el usuario puede identificar si su orden de ruta es adecuada, si se encuentra en la ruta y de ser positivo recibir un mensaje con los números de los buses, tiempos promedio y distancias al punto de su ubicación.

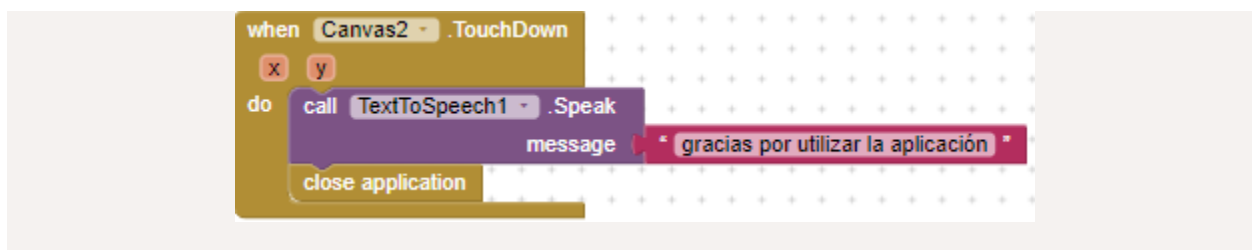
Boton 2. Salir.

Para acceder a este botón solo debe ubicarse en la parte baja del celular y presionar.

Automaticamente se escucha una voz que indica la salida de la aplicación.

Este botón solo cuenta con un comando TextSpeech que transmite el mensaje de salida y una opción de cierre de la app (Figura 20).

Figura 20.
Bloque que permite definir la salida de la aplicación



APP EMPRESA DE TRANSPORTE

Para la aplicación que corresponde a la empresa de transporte se construye el siguiente algoritmo.

Tomando la posición del BUS (Latitud y Longitud) se construye un objeto geográfico de tipo Point.

```
var pt1 = turf.point([longitud,latitud]);
```

Para visualizar la información definida en el JS (ruta6.json), que contiene la ruta de estudio se hace lo siguiente:

```
var rutas = L.geoJson(rutas,{});
```

Como se observa en el código anterior se define una variable ruta que permitirá la conexión del JS con el script.

En la variable rutas tenemos el objeto con toda la información que se definió inicialmente en el JS el cual es igual para la aplicación del usuario.

Dentro del objeto ruta procedemos definir la ubicación geométrica de cada nodo de la ruta y definimos su posición.

```
var pt2 = turf.point(feature.geometry.coordinates);
```

```
var tt=feature.properties.OBJECTID;
```

Una vez identificamos la posición del nodo calculamos la distancia a la que el bus identificado con la variable pt1 se encuentra con respecto a cada nodo.

En caso de que la distancia sea menor a 10 m se asume que el bus se encuentra en la posición del nodo por lo que este pasa a reportarse a la app con la distancia medida entre el punto actual y el anterior (Figura 21).

Figura 21.

Segmento de código que permite definir la posición y distancia recorrida por cada bus

```
var pt1 = turf.point([lo,la]);
var rutas = L.geoJson(rutas,{
  onEachFeature: function (feature, layer) {

    var pt2 = turf.point(feature.geometry.coordinates);
    var tt=feature.properties.OBJECTID;

    var dis = turf.distance(pt1, pt2,{units: 'meters'});
    if (dis <10){
      if (lol !== ""){
        var dis =0;
        var pt3 = turf.point([lol, lal]);
        var options = {units: 'meters'};
        var dis = turf.distance(pt1, pt3, options);
      }
      var j=dis +", "+tt;
      window.AppInventor.setWebViewString(j);
    }
  },
});
```

Intersección del bus con usuarios

Inicialmente se toma la posición (Latitud y Longitud) del Bus y se construye un objeto geográfico de tipo Point.

```
var point_bus = turf.point([longitud,latitud]);
```

Visualizamos la información definida en el JS (rutas6.js) que contiene la ruta de estudio.

```
var rutas = L.geoJson(rutas,{});
```

Buffer al objeto rutas obteniendo una zona de influencia de cada uno de los nodos definidos a lo largo de la ruta.

```
var bufpru = turf.buffer(rutas.toGeoJSON(),25, {units: 'meters'});
```

Teniendo esta información se procedió a definir tanto los puntos de ubicación del usuario como los del bus que me suministra la aplicación y que provienen de la información almacenada en la base de datos.

Como tenemos el objeto `bufpru` que contiene todos los buffer de los nodos de la ruta y además conocemos las posiciones de los usuarios podemos hacer por medio de un ciclo una consulta de cada posición del bus con relación a la posición del buffer de cada usuario y así con la función de Turf de intersección determinar el punto de encuentro de un usuario con uno de los buses.

```
var p1 = turf.polygon(bufpru[posicion_bus[n]].geometry.coordinates);
```

n= iterador para identificar cada bus.

```
var p2 = turf.polygon(bufpru[posicion_usuario].geometry.coordinates);
```

```
inter=turf.intersect(p1,p2);
```

Teniendo ya definido el algoritmo se hizo la aplicación que permite la comunicación del Bus con los usuarios.

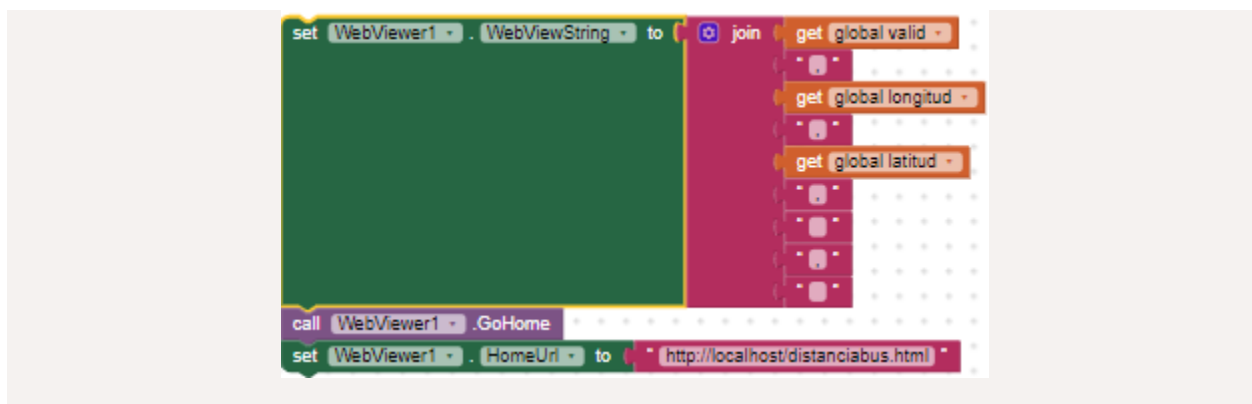
Para la interfaz gráfica se usaron elementos de ingreso de información como la ruta, y el bus los cuales servirán de insumo para el reporte del mismo en la base de datos (Figura 22). Además se presenta un mapa con la ubicación de los usuarios que hicieron la solicitud.

Figura 22.
Pantalla principal aplicación bus.



Al definir los elementos como la ruta y el número del bus la app procede hacer el cálculo de su ubicación utilizando la integración del componente Vieweber con el html generado para el cálculo de posición y distancia (Figura 23).

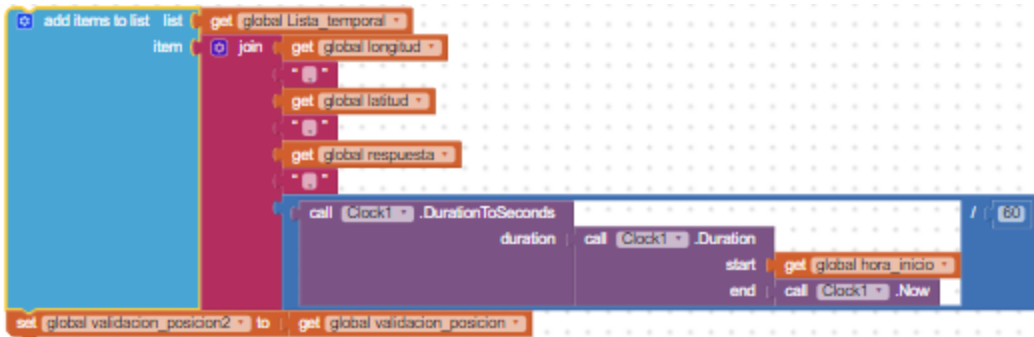
Figura 23.
Pantalla principal aplicación bus.



Inicialmente se calcula la posición del bus (variable pt1) con el nodo cercano (variable pt2) calculando su distancia. En el momento que esta es menor que diez entra a calcular la distancia que hay entre el bus y el nodo anterior.

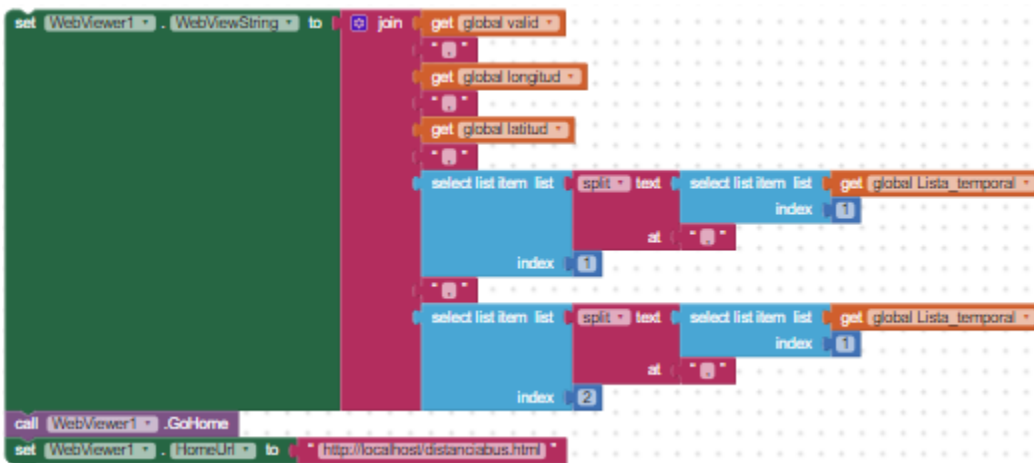
Este resultado se almacena de manera temporal en un arreglo dentro de la aplicación con el tiempo transcurrido entre la hora de inicio del recorrido del bus (Figura 24)

Figura 24.
Bloque de lista que almacena los datos a ser procesados.



Al momento de tener dos posiciones con su nodo definido se procede de nuevo a ejecutar el algoritmo que toma las dos posiciones la variable pt1 que corresponde a la posición actual del bus (latitud-longitud) y variable pt3 que se almacena en el arreglo Lista_temporal y calcula la distancia entre sus puntos (Figura 25).

Figura 25.
Bloque que define la información que se envía a cada script

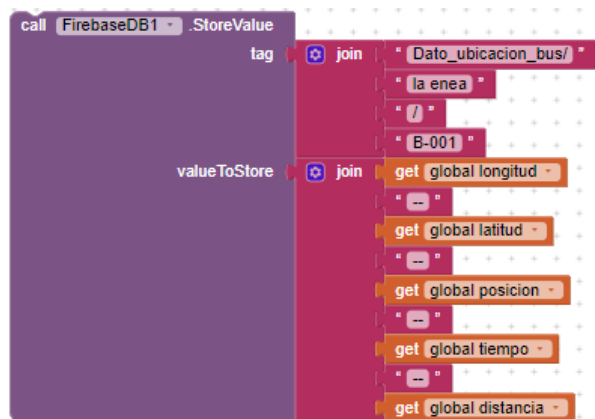


Una vez obtenido este resultado se procede a almacenarlo en otro arreglo para así poder hacer los cálculos de tiempo promedio.

Teniendo ya la información consolidada se procede hacer la carga a la Base de datos que es la misma que se construyó para la App de los usuarios.

Esta información se guarda con el bloque FireDabase.StoreValue el cual permite almacenar datos en la BD.

Figura 26.
Bloque que envia la información del bus y será procesada por la aplicación del usuario



Los datos correspondientes al tag se definen al iniciar el recorrido y corresponde a la ruta (Se selecciona por medio de una lista) y el numero del Bus (Este se ingresa por medio de un cuadro de texto). La información de la ruta (la enea) se deja fija por el ejercicio.

Como se observa (Figura 26) se envia a la base de datos la información relacionada a Numero del bus, ubicación geográfica (latitud, longitud), nodo de ubicación, tiempo transcurrido desde el inicio del recorrido, distancia desde el inicio de la ruta.

6.3 Tipo de estudio

Después de la contextualización de la aplicación y el desarrollo de la misma se define que este estudio es de tipo EXPLORATORIO ya que con el se busca profundizar el estudio de un problema poco abordado como es la movilidad de las personas con discapacidad y se plantea una solución que tomando como soporte las tecnologías de la información permitirá la autonomía y la movilidad de esta población.

7. Resultados

Para la funcionalidad de la aplicación se hicieron pruebas con posiciones simuladas tanto de los usuarios como de los buses con resultados dentro de lo definido.

Es necesario realizar las pruebas de la app y se pueden distribuir en tres fases.

En la primera fase, lo que se hará es probar los escenarios principales de los distintos casos de uso (Ubicarse en la zona, identificar el nodo de ubicación, etc..)

En la segunda fase se probará la app para comprobar si tiene el comportamiento esperado.

Finalmente, la tercera y última fase consistirá en probar algunos escenarios dentro de la ruta para definir la conectividad.

Para que la app funcione correctamente, debe pasar las tres fases de pruebas que se explican anteriormente.

8. Conclusiones

- La aplicación TRANSZARILLO es una app experimental que busca servir de herramienta de ayuda a las personas con discapacidad visual en la identificación del servicio público en la ciudad de Manizales.
- Se hace un recorrido por las investigaciones y la innovación en el campo de las tecnologías de asistencia móvil destinadas a ayudar a personas con discapacidad visual y como la app puede tener un papel relevante en una capacidad perdida.
- Todas las personas ciegas o con deficiencia visual tiene diferentes necesidades específicas de movilidad, orientación y navegación que requieren ser solucionados de varias maneras, el desarrollo de la app busca solucionar en parte esta necesidad.
- Mejorando la disponibilidad del transporte y accesibilidad, brindando capacitación sobre la discapacidad a los proveedores de transporte, conductores y el público en general, se lograría recorrer una gran parte del camino hacia la mejora de la movilidad y la independencia de personas con discapacidad visual.
- El uso de herramientas tecnológicas puede mejorar la accesibilidad de las personas con discapacidad visual al servicio público, independientemente de sus capacidades técnicas, cognitivas o físicas.
- El enfoque de la aplicación puede ser una solución que ofrezca buenos resultados para todas las personas.
- Debido a que se trabaja con el micrófono del celular este puede tener problemas ya que la interfaz de voz requiere un micrófono capaz de registrar la orden del usuario así como aislar el ruido del ambiente.
- A pesar de los avances tecnológicos que se han tenido en las últimas décadas, se pudo evidenciar que muy pocas ciudades en el país, han podido integrar a sus sistemas de

transporte herramientas que permita a los usuarios con discapacidad visual acceder con mayor facilidad al uso de las rutas de la ciudad.

- El desarrollo de este trabajo sirve de base para posteriores investigaciones que ayuden a mejorar las condiciones de movilidad en el transporte público de las personas con discapacidad.
- El desarrollo de nuevas tecnologías en el campo del geoposicionamiento y la adaptabilidad de los equipos a diferentes tipos de discapacidad exigirán a las empresas contar con sistemas que hagan que el transporte público sea mas inclusivo y con un enfoque de igualdad que garanticen la correcta movilidad de toda la población.
- La implementación y fortalecimiento de este tipo de aplicativos permitirá una mayor autonomía y garantizara la seguridad en la movilidad de las personas con discapacidad visual.
- Para una correcta ejecución y manejo de la aplicación se requiere fortalecer las competencias de los conductores de los vehículos y de los usuarios
- La posibilidad de brindar a los usuarios con discapacidad visual un medio de selección segura de su transporte así como el correcto abordaje del mismo es uno de los grandes logros a valorar dentro de las dinámicas de inclusión.
- El manejo de herramientas informáticas tan versátiles como App Inventor la cual funciona a partir de bloques evita la consecución de un gran contenido de código. Esta facilidad permitió el desarrollo y uso de un número importante de herramientas de una forma fácil e intuitiva que ayudaron a la obtención de los objetivos.
- El manejo de sistemas tan versátiles como Java Script, Html y los formatos Json permitieron que el procesamiento de los datos se hiciera de manera mas precisa.

- EL uso de la herramienta Turf Js (Análisis Geoespacial avanzado para navegadores), facilito la evaluación de cada uno de los puntos geográficos y permitió su análisis y validación.
- La disponibilidad de la herramienta Firebase facilita la integración y comunicación entre el usuario y servicio de transporte.

9. Recomendaciones

- Ya que las pruebas se hicieron simuladas y no se realizó en condiciones de ruido elevado se recomienda hacer nuevas pruebas con procedimientos que evalúen la contaminación del ambiente
- Para garantizar que el sistema funciona correctamente es necesario realizar las pruebas de la app con población con discapacidad visual.
- Incluir a diferentes organismos tanto civiles como públicos que ayuden y promuevan este tipo de iniciativas.
- Se recomienda construir la aplicación en una plataforma mas estable como Android Studio que permita un mejor y mas preciso uso de las herramientas del celular.
- Hacer extensivo su uso para potencializar y mejorar la aplicación.
- Continuar con el desarrollo y mejora de la aplicación.
- Permitir el uso masivo de soluciones tecnológicas en el transporte publico que ayuden a las personas en general a mejorar su movilidad.
- Incentivar desde la academia el desarrollo de soluciones tecnológicas encaminadas al bienestar de la población.
- Este tipo de desarrollos debe generar una señal a la academia para que construya los puentes necesarios entre las instituciones del estado los entes privados y toda la comunidad para que su uso sea posible.

10. Referencias

Michelle Andrea Vivas, (2021). Discapacidad visual: 2 mil millones de personas la padece y las cifras aumentarán. Consultor Salud.

Leaflet (2021). Leaflet es la biblioteca javascript líder en código abierto para mapas interactivos compatibles con dispositivos móviles. Leaflet - a JavaScript library for interactive maps (leafletjs.com).

Organización Mundial de la Salud (2021). <http://www.who.int/mediacentre/factsheets/fs282/es/>

Firebase Documentacion (2021). <https://firebase.google.com/docs>.

Microsoft Documentacion (2021). [Work with JSON data - SQL Server | Microsoft Docs](#)

MND Web Docs, Html (2021). <html> - HTML: Lenguaje de etiquetas de hipertexto MDN (mozilla.org)

MND Web Docs, JavaScript (2021).. [JavaScript | MDN \(mozilla.org\)](#)

Plataforma Arcgis (2021). [Plataforma ArcGIS \(sigsa.info\)](#)

TURF Documentacion (2021). Analice datos con turf.js y Mapbox GL JS | Ayuda | Caja de mapas.

AppInventor Documentacion (2021). [Tutorials for MIT App Inventor](#)

Freedom Scientific (2021). <http://www.freedomscientific.com>.

Datos Abiertos - Geoportal Alcaldía de Manizales/Vías y Transporte/Rutas Servicio

Publico/Quiero usar esto/Ver recursos de API/ GeoJSON. Recuperado en Octubre/2020 de. [Rutas Servicio Publico | Rutas Servicio Publico | Datos-Abiertos-Alcaldia-Manizales \(arcgis.com\)](#)

Rupert RA Bourne, (2020). Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the Global Burden of Disease Study. Vision and Eye Research Institute, Anglia Ruskin University, Cambridge.

Diego Mayordomo-Martínez, Juan-Carlos Sánchez-Aarnoutse, Juan M. Carrillo-de-Gea, José A. García-Berná, José Luis Fernández-Alemán and Ginés García-Mateos (2019). Design and Development of a Mobile App for Accessible Beach Tourism Information for People with Disabilities.

Isaac, B. Lázaro, (2018). Sistema de seguimiento mediante mapa online para los niños que viajan en transporte público (Trabajo final de grado). Escuela Técnica Superior de Ingeniería Geodésica, Cartográfica y Topográfica, Valencia.

Buele Jorge, Salazar L. Franklin, A. Santiago, Aldás R, Abigail, Urrutia U. Pilar (2018) Plataforma y aplicación móvil para proporcionar información del transporte público utilizando un dispositivo embebido de bajo costo.

Chunnu Khawas, (2018). Application of Firebase in Android App Development-A Study. Chunnu Khawas Department of Computer Applications Sikkim University Gangtok, India.

Carlos Parra Dussan, (2018). Los ciegos en el Censo 2018. Director General del INCI Instituto Nacional para Ciegos. Inci.gov.co

Laurence Moroney (2017). The Definitive Guide to Firebase Build Android Apps on Google's Mobile Platform.

Adele Crudden, Jennifer L. Cmar, and Michele C. McDonnall (2017). Stress associated with transportation: A survey of persons with visual impairments.

Wolber David, Abelson Harold, Friedman Mark, (2014). Democratizing computing with APP INVENTOR

Torres, C. María Isabel, Rodríguez, F. María José, Visitación H. María, Samos José, and Espín Vanesa (2012). Architecture of a Mobile App Recommender System for People with Special Needs

J. A. Muñoz Sevilla, C. Blocona Santos (2011). Haciendo accesibles a las personas con discapacidad visual los sistemas de posicionamiento. Global (gps): el Proyecto HaptiMap, Making Global Positioning System (gps) accessible for people with visual impairment: the HaptiMap Project

José Antonio Muñoz Sevilla. Las TIC y la discapacidad visual. Centro de Investigación
Desarrollo y Aplicación Tiflotécnica (ONCE-CIDAT)

IEEE (1993). IEEE Std 830-1993 IEEE Recommended Practice for Software Requirements
Specifications.

National Eye Institute (2019). <http://www.nei.nih.gov/health/espanol/bajavision/index.asp>

Mora-Castillo, J. A. (2016). Object serialization/deserialization and data transmission with
JSON: paper review. Revista Tecnología en Marcha, 29(1), 118-125.

Omar David Laverde Cabrera, (2013). Personas con discapacidad visual y su accesibilidad al
entorno urbano. Fundación de Educación Superior INSUTEC, Bogotá, Colombia.

Aurelio Morales (2013) Como crear un mapa web en Leaflet a partir de un GeoJSON.
(mappinggis.com. Aurelio Morales Licenciado en Geografía. Máster en Sistemas de
Información Geográfica. Consultor GIS desde el año 2004.

Kryscia Daviana Ramírez Benavides (2013). App Inventor CI-2657 Robótica M.Sc. Kryscia
Ramírez Benavides. Profesora e Investigadora Universidad de Costa Rica.

Ericksson, M., & Hallberg, V. (2011). Comparison between JSON and YAML for data
serialization: A Case Study 2011. Obtenido desde la página: <http://scholar.google.com>.

Ericksson, M. Hallberg, V. (2011). Comparison between JSON and YAML for data serialization: A Case Study 2011. Obtenido desde la página: <http://scholar.google.com/>

Bláithin A. M. Gallagher, Patricia M. Hart , Colm O'brien, Michael R. Stevenson, Andrew J. Jackson, (2010). Mobility and access to transport issues as experienced by people with vision impairment living in urban and rural Ireland. Department of Research and Projects, National Council for the Blind of Ireland, Dublin, Ireland, Centre for Vision Science, Queen's University, Belfast, UK, Directorate of Ophthalmology, Royal Group of Hospitals, Belfast, UK, Department of Ophthalmology, Mater Misericordiae Hospital, Dublin, Ireland, Institute of Ophthalmology, University College Dublin, Dublin, Ireland, National Council for the Blind of Ireland, Dublin, Ireland, Department of Epidemiology and Public Health, Queen's University, Mulhouse Building, Belfast, UK, and Directorate of Integrated Care, Health and Social Care Board, Belfast, UK

Talia McCray, Nicoe Brais, (2007). Exploring the Role of Transportation in Fostering Social Exclusion: The Use of GIS to Support Qualitative Data. College of Business Administration, University of Rhode Island

MAPBOX Documentacion (2021). mapbox.com

Wei-Hua Lin and Jian Zeng. An experimental study on real time bus arrival time prediction with GPS data. Center for Transportation Research and Department of Civil and Environmental Engineering. Virginia Polytechnic Institute & State University.

Dihua Sun, Hong Luo, Liping Fu, Weining Liu, Xiaoyong Liao, and Min Zhao, (2007). Predicting Bus Arrival Time on the Basis of Global Positioning System Data. Transportation Research Board of the National Academies, Washington.

PEREZ L (2005). "Mejora de la accesibilidad universal a los entornos". CERMI. Madrid.

Decreto 1955 (1955). "Por el cual se disuelve la Federación Nacional de Ciegos y Sordomudos." Decreto 1955 de 1955 - EVA - Función Pública (funcionpublica.gov.co).

Lilit Hakobyan, BSc (Hons), Jo Lumsden, BSc (Hons), PhD, Dympna O'Sullivan, BSc (Hons), PhD, Hannah Bartlett, BSc (Hons) MCOptom, PhD (2013). Mobile assistive technologies for the visually impaired.

A. Anexo: Manual de Usuario

- **App Usuario**

Pantalla de Presentación: Al iniciar la ejecución de la aplicación se presentan dos pantallas que corresponden a créditos a la universidad y el logo y nombre de la aplicación

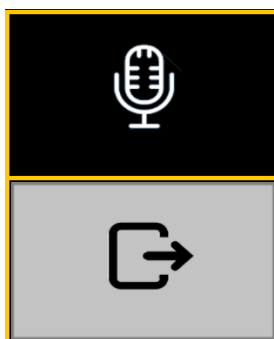
Figura 27.
Imágenes de presentación



Aunque el diseño y la propuesta de la aplicación están hechas para personas con discapacidad visual, se presentan estas dos imágenes como forma de presentarla a la comunidad general.

Pantalla principal: Para dar inicio a la aplicación se presenta la siguiente pantalla, la cual se encuentra distribuida en dos marcos táctiles representados con dos iconos, que permiten la ejecución o salida de la aplicación, brindando un funcionamiento intuitivo para las personas con discapacidad visual.

Figura 28.
Imágenes de presentación



Marco Táctil superior: al presionar esta parte táctil de la pantalla, la aplicación por medio de un comando de voz le solicitará al usuario la ruta requerida “defina su ruta”. En el momento en el que el usuario indica por medio de su voz la ruta que desea, la aplicación procesa esta orden y valida que el usuario se encuentre en la ubicación cuya zona de influencia cubra la ruta deseada. En caso de que la ubicación sea correcta la aplicación mediante un comando de voz, le informará los buses que se encuentran disponibles con su tiempo promedio y distancia. En el momento que haya una intersección entre la zona de influencia del bus y del usuario, la aplicación enviara una señal vibratoria y por medio de un comando de voz indicando que su ruta ha llegado. A continuación la app se cierra.

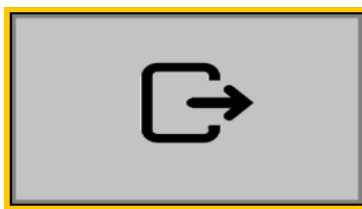
Figura 29.
Inicio de la aplicación



En caso de que el usuario no se encuentre sobre la zona de influencia de la ruta, la aplicación de informará por medio de un comando de voz que su ubicación no es correcta “la ruta no pasa por este sector”.

Marco Táctil inferior: al presionar esta parte táctil de la pantalla, el usuario podrá salirse de la aplicación, la cual le indicará por medio de un comando de voz “Gracias por utilizar la aplicación”.

Figura 30.
Salida de la aplicación



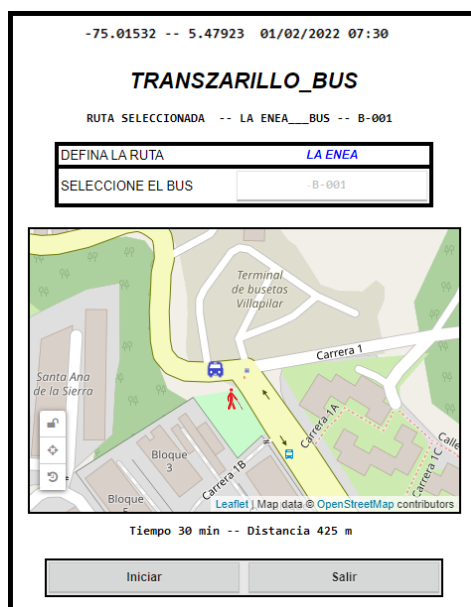
- **App Conductor**

Las pantallas de presentación que dan inicio a la aplicación son las mismas descritas en la App Usuario.

Pantalla principal: Para iniciar la aplicación se presenta la siguiente pantalla, la cual solicita definir la ruta y el número del bus que conduce (Al presionar sobre el icono buses disponibles la ampliación activará un menú con los buses disponibles). Después de validar esta información y dar inicio, la aplicación registrará el tiempo y la distancia recorrida, así como su posición en el mapa.

Figura 31.

Aplicación conductor



Una vez la posición del bus está cercada la posición de un usuario que requiere esta ruta, la aplicación dibuja en el mapa el icono del punto donde se hizo la solicitud indicando que un usuario está cerca.

El conductor puede visualizar el tiempo y la distancia recorrida.

A. Anexo: ALGORITMOS

- **Ubicación del usuario en función de la ruta**

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Turf Leaflet</title>
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.2.0/dist/leaflet.css" />
  <style>
    #map {
      width: cover;
      height: 850px; }
  </style>
</head>
<body>
  <script src="https://unpkg.com/leaflet@1.2.0/dist/leaflet.js"></script>
  <script src='https://npmcdn.com/@turf/turf/turf.min.js'></script>
  <script type="text/javascript" src="rutas6.json"></script>
  <div id="map"> </div>
</script>

  co = window.AppInventor.getWebViewString();
  arra=co.split("--");
  lo=arra[0];
  la=arra[1];
//window.AppInventor.setWebViewString(lo + ", " + la);
  var point_usuario = turf.point([lo,la]);

  var rutas = L.geoJson(rutas,{});
  var bufpru = turf.buffer(rutas.toGeoJSON(),25, {units: 'meters'});

  var ru = L.geoJson(bufpru,{
    onEachFeature: function (feature, layer) {
      var polygon = turf.polygon(feature.geometry.coordinates,
    });
      var y=
      turf.booleanPointInPolygon(point_usuario,polygon);
      var rut_asig =feature.properties.OBJECTID;
      if (y != false){
        var t= "true";
        window.AppInventor.setWebViewString(rut_asig);
      }
    },
  });
</script>
</body>
</html>

```


- **Ubicación de los buses en función del usuario**

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Turf Leaflet</title>
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.2.0/dist/leaflet.css" />
  <style>
    #map {
      width: cover;
      height: 850px; }
  </style>
</head>
<body>
  <script src="https://unpkg.com/leaflet@1.2.0/dist/leaflet.js"></script>
  <script src='https://npmcdn.com/@turf/turf/turf.min.js'></script>
  <script type="text/javascript" src="rutas6.json"></script>

<script>

  var co = window.AppInventor.getWebViewString();

  var arra=co.split(",");
  var lo=arra[0];
  var la=arra[1];

  coords = [];

  var point_usuario = turf.point([lo,la]);

  var rutas = L.geoJson(rutas,{
    onEachFeature: function (feature, layer) {
      coords.push(feature.geometry.coordinates);
    },
  });

  var bufpru = turf.buffer(rutas.toGeoJSON(),30, {units: 'meters'});

  var ru = L.geoJson(bufpru,{
    onEachFeature: function (feature, layer) {

      var polygon = turf.polygon(feature.geometry.coordinates, {});
      var y= turf.booleanPointInPolygon(point_usuario,polygon);

      if (y != false){

        var rut_asig =feature.properties.OBJECTID;
        for (n=2;n<7;n++){
          p1=arra[n];

          var v1=Number(p1);
          var v2=Number(rut_asig);

          if (v1<v2){
            var pp1 =
              turf.polygon(bufpru[v1].geometry.coordinates);
              var p2 =
              turf.polygon(bufpru[v2].geometry.coordinates);

            inter=turf.intersect(pp1,p2);

            if (inter != null){

```

```

        var q="a" + ",";
    }

    var dist_total=0;

    for (j= v1;j<v2;j++){
        var from = turf.point(coords[j-1]);

        var gg=coords[j];
        var to = turf.point(gg);

        var options = {units: 'meters'};

        var distance = turf.distance(from, to, options);
        var dist_total=dist_total+distance;
    }

    var j1= j1 + ", " + v1 + ", " + Math.round(dist_total) ;

    }

    window.AppInventor.setWebViewString(rut_asig+j1);

    }

},

    })

```

- ARCHIVO JSON

```

var rutas={
  "type": "FeatureCollection",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
  "features": [
    {
      "type": "Feature",
      "id": 1,
      "geometry": {
        "type": "Point",
        "coordinates": [
          -75.527359804999833,
          5.0875078340001778
        ]
      },
      "properties": {
        "OBJECTID": 1,
        "Ruta": "La Enea"
      }
    },
    {
      "type": "Feature",
      "id": 2,
      "geometry": {
        "type": "Point",
        "coordinates": [
          -75.52722916599987,
          5.0872857480001699
        ]
      },
      "properties": {
        "OBJECTID": 2,
        "Ruta": "La Enea"
      }
    },
    {
      "type": "Feature",

```

```
"id" : 3,
"geometry" : {
  "type" : "Point",
  "coordinates" : [
    -75.527126042999839,
    5.0871416920001593
  ]
},
"properties" : {
  "OBJECTID" : 3,
  "Ruta" : "La Enea"
}
},
{
  "type" : "Feature",
  "id" : 4,
  "geometry" : {
    "type" : "Point",
    "coordinates" : [
      -75.527016450999838,
      5.0869798210001704
    ]
  },
  "properties" : {
    "OBJECTID" : 4,
    "Ruta" : "La Enea"
  }
},
{
  "type" : "Feature",
  "id" : 5,
  "geometry" : {
    "type" : "Point",
    "coordinates" : [
      -75.526898256999857,
      5.0867855650001843
    ]
  },
  "properties" : {
    "OBJECTID" : 5,
    "Ruta" : "La Enea"
  }
},
{
  "type" : "Feature",
  "id" : 6,
  "geometry" : {
    "type" : "Point",
    "coordinates" : [
      -75.526781583999821,
      5.0865970290001883
    ]
  }
}
```