

**SISTEMA DE ESTANTERÍA INTELIGENTE BASADO EN TECNOLOGÍAS
MÓVILES E IOT PARA PERSONAS CON DISCAPACIDAD VISUAL**

**STEFANNY GONZÁLEZ GELVEZ
YESSICA VIVIANA EUGENIO GARCÍA**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA – UNAB
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
PROYECTO DE GRADO II
BUCARAMANGA, NOVIEMBRE 6 DE 2020**

**SISTEMA DE ESTANTERÍA INTELIGENTE BASADO EN TECNOLOGÍAS
MÓVILES E IOT PARA PERSONAS CON DISCAPACIDAD VISUAL**

**STEFANNY GONZÁLEZ GELVEZ
YESSICA VIVIANA EUGENIO GARCÍA**

**PROYECTO DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE
INGENIERO DE SISTEMAS**

**DIRECTOR:
YAMID GABRIEL GAMBA GONZÁLEZ**

**GRUPO DE INVESTIGACIÓN PRISMA PRESERVACIÓN E INTERCAMBIO
DIGITAL DE INFORMACIÓN Y CONOCIMIENTO
LINEA DE INVESTIGACIÓN EN TECNOLOGÍA Y SOCIEDAD**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA – UNAB
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
PROYECTO DE GRADO II
BUCARAMANGA, NOVIEMBRE 6 DE 2020**

Nota de Aceptación

Firma del jurado

Firma del Jurado

DEDICATORIA

El presente trabajo está dedicado primeramente a Dios, ya que gracias a Él hemos logrado culminar la carrera.

A nuestros padres por habernos forjado como las personas que somos ahora, muchos de nuestros logros se los debemos a ustedes, inclusive este. Nos formaron con reglas y con algunas libertades, pero, a fin de cuentas, nos motivaron constantemente para alcanzar nuestros anhelos y recordarnos cada día que no debemos desfallecer y siempre seguir adelante.

A nuestro director Yamid Gabriel Gamba González, por su asesoría, su apoyo y la aptitud dedicada durante el transcurso de creación de este proyecto, por ser un amigo.

A nuestros hermanos Luisa, Giovanny, Diego y a nuestras familias por su apoyo incondicional, y la motivación durante nuestro proceso de formación académica.

A todos los docentes y compañeros que nos han acompañado durante el proceso de ingeniería, compartiendo sus conocimientos.

¡Gracias a todos los que siempre confiaron en nosotras!

AGRADECIMIENTOS

Primeramente, a Dios por darnos la oportunidad de cumplir una de nuestras metas.

A nuestros padres y familiares por estar presentes brindándonos su apoyo cada instante.

A German por su apoyo y acompañamiento en el transcurso de la carrera.

A la Universidad Autónoma de Bucaramanga por brindarnos la oportunidad de formación.

A la coordinación del programa de Ingeniería de Sistemas por su colaboración.

A los docentes del programa, por sus enseñanzas.

A Freddy por su motivación, su apoyo y su ayuda hasta donde los alcances lo permitían.

A nuestros amigos por tantos momentos.

Al Ingeniero Yamid Gabriel Gamba director del proyecto, por su trabajo incansable, su apoyo y sus valiosos aportes durante las etapas del proyecto.

A toda nuestra familia que siempre confió en nosotras y siempre nos apoyó en todo el camino para ser ingenieras, gracias a Dios y a la vida por permitirnos estar acá y poder decir, ¡Lo logramos!

TABLA DE CONTENIDO

RESUMEN EJECUTIVO	8
1 INTRODUCCIÓN.....	12
2 PLANTEAMIENTO DEL PROBLEMA.....	14
2.1 ÁRBOL PROBLEMA	16
3 JUSTIFICACIÓN.....	17
4 OBJETIVOS.....	20
4.1 OBJETIVO GENERAL.....	20
4.2 OBJETIVOS ESPECÍFICOS	20
5 MARCO REFERENCIAL	21
5.1 MARCO CONCEPTUAL.....	21
5.2 MARCO TEÓRICO.....	24
5.3 MARCO LEGAL.....	57
6 ANTECEDENTES.....	59
7 ESTADO DEL ARTE.....	61
7.1 PROCESO DE BÚSQUEDA	61
8 METODOLOGÍA.....	71
9 DESARROLLO TRABAJO DE GRADO.....	73
9.1 DISEÑO.....	73
9.2 CONSTRUCCIÓN	89
9.3 VALIDACIÓN DE FUNCIONALIDADES.....	115
10 CONCLUSIONES.....	119
11 RECOMENDACIONES	121
12 REFERENCIAS.....	122
13 ANEXOS	127

LISTADO DE TABLAS

Tabla 1. Capas modelo de referencia IoT	26
Tabla 2. Capacidades del modelo de referencia IoT	27
Tabla 3. Tipos de redes móviles.....	32
Tabla 4. Estructura aplicación	36
Tabla 5. Estándares del Wifi.....	39
Tabla 6. Diferencia entre Arduino y Raspberry Pi	42
Tabla 7. Aspectos de Firebase.....	44
Tabla 8. Funciones claves reglas de seguridad de Firebase	47
Tabla 9. Comparativo entre conectividad Wifi vs Bluetooth	50
Tabla 10. Comparativo Kotlin vs Java	53
Tabla 11. Configuración electrodos Sensor MPR121.....	55
Tabla 12. Comparativo sensor MPR03X y MPR121	56
Tabla 13. Análisis tecnologías utilizadas en artículos recuperados.....	70
Tabla 14. Requerimiento funcional 01	74
Tabla 15. Requerimiento funcional 02.....	75
Tabla 16. Requerimiento funcional 03.....	75
Tabla 17. Requerimiento funcional 04.....	75
Tabla 18. Requerimiento funcional 05.....	76
Tabla 19. Requerimiento no funcional 01	77
Tabla 20. Requerimiento no funcional 02.....	77
Tabla 21. Requerimiento no funcional 03.....	77
Tabla 22. Requerimiento no funcional 04.....	77
Tabla 23. Requerimiento no funcional 05.....	78
Tabla 24. Plantilla descripción casos de uso.....	79
Tabla 25. Descripción CU-01	80
Tabla 26. Descripción CU-02	80
Tabla 27. Descripción CU-03	80
Tabla 28. Descripción CU-04	81
Tabla 29. Marcas productos.....	90
Tabla 30. Productos del prototipo.....	92
Tabla 31. Checklist validación funcionamiento de la aplicación móvil.....	118

LISTADO DE FIGURAS

Ilustración 1. Árbol de problema.....	16
Ilustración 2. Usuarios de teléfonos inteligentes en todo el mundo 2016-2021.....	19
Ilustración 3. Modelo de referencia de IoT	25
Ilustración 4. Descripción técnica de IoT	28
Ilustración 5. Tipos de dispositivos y su relación con objetos físicos	29
Ilustración 6. Arquitectura de Androide	32
Ilustración 7. Estructura general de una aplicación	35
Ilustración 8. La alianza Wi-Fi de hoy.....	40
Ilustración 9. Configuración Raspberry Pi	41
Ilustración 10. Aspectos Firebase	43
Ilustración 11. Sintaxis reglas de seguridad en Realtime Database.....	46
Ilustración 12. Sensor MPR121	54
Ilustración 13. Diagrama de Flujo del Proceso de Búsqueda.....	63
Ilustración 14. Diagrama proceso de Búsqueda.....	66
Ilustración 15. Flujograma metodología	71
Ilustración 16. Diagrama casos de uso	78
Ilustración 17. Diagrama de secuencia	82
Ilustración 18. Diagrama de actividades.....	83
Ilustración 19. Mockup aplicación móvil	84
Ilustración 20. Interfaz aplicación móvil – Pantalla principal.....	84
Ilustración 21. Arquitectura conexiones Firebase, Android Studio y Raspberry	85
Ilustración 22. Raspberry Pi 3	87
Ilustración 23. Sensor MPR121	87
Ilustración 24. Cables pinzas de cocodrilo	88
Ilustración 25. Diseño prototipo estantería	89
Ilustración 26. Marcas más consumidas en Colombia	90
Ilustración 27. Programa grabación de audios	92
Ilustración 28. Descarga de audio	93
Ilustración 29. Conversión audio .mp3 a .wav	93
Ilustración 30. Raspberry - Sensor MPR121	94
Ilustración 31. Tarjeta microSD a utilizar	94
Ilustración 32. Formateo de la microSD	95
Ilustración 33. Descarga del S.O Raspbian.....	95
Ilustración 34. Descarga del VNC	96
Ilustración 35. Terminal IP.....	97

Ilustración 36. Activación VNC	97
Ilustración 37. Nueva conexión VNC.....	98
Ilustración 38. Configuración de la IP y conexión.....	98
Ilustración 39. Importar librería pines GPIO y módulos pygame	99
Ilustración 40. Restablecer pin con un estado.....	100
Ilustración 41. Conexión a Firebase	100
Ilustración 42. Bucle de imprimir sonidos	101
Ilustración 43. Librería smbus.....	101
Ilustración 44. Configuración del sensor	102
Ilustración 45. Firebase	103
Ilustración 46. Iniciar proyecto en Firebase.....	103
Ilustración 47. Agregar Firebase a Android Studio	104
Ilustración 48. Llamar paquete	104
Ilustración 49. Complemento Google	105
Ilustración 50. Sincronización de los datos de la aplicación.....	105
Ilustración 51. Conexión al servidor de Firebase.....	106
Ilustración 52. Crear BD en Realtime Database.....	106
Ilustración 53. Reglas de protección de datos.....	107
Ilustración 54. Generar clave privada.....	108
Ilustración 55. Base de Datos "Sonidos"	109
Ilustración 56. Librerías utilizadas	109
Ilustración 57. Arquitectura de la aplicación	110
Ilustración 58. Clases	111
Ilustración 59. Función getFirebaseSoundArroz()	112
Ilustración 60. Función activar sonido	113
Ilustración 61. Función activar audio de bienvenida.....	113
Ilustración 62. Carpeta drawable.....	114
Ilustración 63. Carpeta layouts	114
Ilustración 64. Validar instalación de la aplicación	116
Ilustración 65. Conexión a red.....	116
Ilustración 66. Aplicación iniciada.....	117
Ilustración 67. Validación envío de audio	118
Ilustración 68. Anexo diseño del funcionamiento	128

RESUMEN EJECUTIVO

Introducción: Actualmente la manera como está diseñado el mundo no permite la inclusión de personas con capacidades diferentes, quienes se ven limitadas en su diario vivir, debido a que no encuentran opciones acordes a sus características, por cuanto se hizo necesario generar soluciones que redujeran la desigualdad. Para el caso particular de las personas con discapacidad visual, estas no son la excepción, debido a que sus necesidades básicas tales como la compra de productos de la canasta familiar entre otras que se ven afectadas. Se propuso un sistema que facilitara las necesidades de una persona no vidente mediante un prototipo de un estante inteligente donde por medio del tacto, audio y una aplicación móvil los usuarios pudieran conocer el nombre del producto y sus características.

Objetivo: Construir un sistema de estante inteligente para una experiencia de compras incluyente a personas con discapacidad visual mediante tecnologías IoT y móvil.

Metodología: El sistema de estantería inteligente se desarrolló con base a una metodología de investigación aplicada de tipo experimental, usando programación de dispositivos móviles en Android para la interface de usuario, tecnología Raspberry Pi para la integración del sistema y tecnologías inalámbricas para su interconexión.

Desarrollo: Dado el objetivo principal y la necesidad de facilitar a las personas con discapacidad visual su compra se llevó a cabo una estantería la cual está integrada por un Raspberry pi y un sensor MPR121. Así pues, una aplicación móvil desarrollada utilizando el lenguaje de programación Kotlin, que por medio del IDE de desarrollo Android Studio 4.0.1 da a conocer a la persona la información de cada uno de los productos ubicados en ella. Además, se utilizó Firebase como intermediario entre la estantería y la aplicación móvil.

Conclusión: El proyecto permite la integración de diversas tecnologías en su construcción para brindarle a las personas con discapacidad visual una experiencia diferente a la hora de conocer las características de un producto por medio de un mensaje de audio en su dispositivo móvil. Ello se logra mediante Firebase, que actúa como eje del sistema recibiendo y enviando las solicitudes a la Raspberry, para establecer la interacción con la aplicación móvil mediante una conexión con la misma la URL, actualizando los datos justo en el momento en que se están procesando, ello se logra cuando se cambia el estado del Switch, el cual se refleja inmediatamente en una consulta a Firebase y la ejecución de un script en Python que conecta la Raspberry a dicha URL, obteniendo el valor del estado true para la generación del mensaje de audio

PALABRAS CLAVE: Inclusión, Discapacidad visual, Aplicación Móvil, Tecnología IoT, Conectividad inalámbrica, Compras por app

GRUPO DE INVESTIGACIÓN: PRISMA.

LÍNEA DE INVESTIGACIÓN: Tecnología y sociedad

ABSTRACT

Introduction: Currently the way the world is designed does not allow the inclusion of people with different capacities, who are limited in their daily lives, because they do not find options according to their characteristics, as it became necessary to generate solutions that reduce inequality. In the particular case of people with visual impairment, they are no exception, since their basic needs such as the purchase of products from the family basket, among others, are affected. We proposed a system that would facilitate the needs of a blind person through a prototype of an intelligent shelf where through touch, audio and a mobile application user could know the name of the product and its characteristics.

Objective: To build an intelligent shelf system for an inclusive shopping experience for visually impaired people using IoT and mobile technologies.

Methodology: The intelligent shelf system was developed based on an applied research methodology of experimental type, using mobile device programming in Android Studio for the user interface, Raspberry Pi technology for the integration of the system and wireless technologies for its interconnection.

Development: Given the main objective and the need to make it easier for visually impaired people to buy, a shelf was made up of a Raspberry pi and an MPR121 sensor. Also, a mobile application developed using the programming language Kotlin, which through the development IDE Android Studio 4.0.1 that gives the person the information of each of the products through an audio located in it. In addition, Firebase was used as an intermediary between the shelf and the mobile application for sending audios.

Conclusion: The project allows the integration of various technologies in its construction to provide people with visual impairment a different experience when it

comes to knowing the characteristics of a product through an audio message on their mobile device. This is achieved through Firebase, which acts as the hub of the system receiving and sending requests to the raspberry, to establish interaction with the mobile application through a connection with the same URL, updating the data just as they are being processed, this is achieved when the state of the switch is changed, which is immediately reflected in a query to Firebase and the execution of a script in Python that connects the raspberry to that URL, obtaining the value of the true state for the generation of the audio message.

KEYWORDS: Inclusion, Visual Disability, Mobile Application, IoT Technology, Wireless Connectivity, Shopping by App.

RESEARCH GROUP: PRISM.

RESEARCH LINE: Technology and society

1 INTRODUCCIÓN

En la actualidad los grandes avances tecnológicos han derivado en la importancia del uso y apropiación de las nuevas tecnologías para apoyar el diario vivir de las personas y sus necesidades en múltiples sectores, este va de la mano con la base que definen los objetivos de desarrollo sostenible ODS, como por ejemplo los objetivos 9 industria Innovación e Infraestructura, el 10 Reducción de las desigualdades y 11 Ciudades y Comunidades Sostenibles los cuales se centran entre otras cosas en crear soluciones que den respuesta a las realidades diarias de las personas, buscando favorecer sus condiciones adversas y ayudándole a sostener una vida plena y feliz en comunidad (Desarrollo, 2020).

Las personas con discapacidad visual son una amplia población en el mundo, hecho que se demuestra en las cifras para 2019, donde a nivel mundial se contaba con cerca de 1300 millones de personas que viven con alguna forma de deficiencia visual, y de las cuales 36 millones son ciegos (WHO, 2018a).

“Las personas ciegas o con ceguera son aquellas que no ven nada en absoluto o solamente tienen una ligera percepción de luz pudiendo distinguir levemente entre luz y oscuridad, pero no reconociendo la forma de los objetos” (Once, 2020a).

La población con deficiencia visual en el mundo ha sido generalmente olvidada y en muchos casos discriminada por su discapacidad en la realización de las actividades cotidianas como tomar el bus, ir al lugar de estudios o de trabajo, realizar compras o hacer ejercicio al aire libre entre otras múltiples actividades llevando a que haya grandes brechas de desigualdad (WHO, 2018b).

Esta propuesta se enfocó en las personas con discapacidad visual y en la dificultad que encuentran en el momento de comprar productos en un supermercado. En ella se propuso la construcción de un sistema de estantería teniendo en cuenta cuatro fases: investigación, diseño y desarrollo y validación. Fases en las cuales se diagnosticó el uso de las tecnologías IoT en los entornos de inclusión de las personas con discapacidad visual mediante la búsqueda sistemática de información, posteriormente se identificaron las necesidades de personas ciegas en cuanto a nivel de experiencias de compras y se establecieron los requerimientos funcionales y no funcionales del sistema mediante entrevistas que permitieron diseñar los diferentes subsistemas móviles, interfaz IoT, base de datos mediante técnicas de modelado y diagramas UML y por último la construcción del prototipo funcional a partir del desarrollo de cada uno de los módulos que lo componen con la tecnología Raspberry pi, validando su funcionamiento conforme al diseño. Con el desarrollo del Proyecto de Investigación, se esperó crear una solución tecnológica que brindará un apoyo y fuera de ayuda a las personas con discapacidad visual en su realidad diaria a la hora de hacer sus compras.

2 PLANTEAMIENTO DEL PROBLEMA

Para la Organización Nacional de Ciegos Españoles (ONCE), cuando se habla de ceguera, discapacidad visual grave o deficiencia visual, nos estamos refiriendo a condiciones caracterizadas por una limitación total o muy seria de la función visual. Las personas ciegas o con ceguera son aquellas que no ven nada en su totalidad o solamente tienen una ligera percepción de luz, pero no la forma de los objetos (Once, 2020a).

Según la Organización Mundial de la Salud, es aquella visión menor de 20/400 o 0.05, considerando siempre el mejor ojo y con la mejor corrección. Se considera que existe ceguera legal cuando la visión es menor de 20/200 o 0.1. En el mundo para 2019 se contaba aproximadamente con 1300 millones de personas que viven con alguna forma de deficiencia visual, de las cuales 36 millones son ciegas (WHO, 2018b). A partir de esos datos se puede interpretar que una amplia población puede sentirse discriminada por su discapacidad.

La problemática radica en que el mundo está pensado para las personas videntes, esto conlleva a que se presenten dificultades para las personas con discapacidad visual. Según la propia ONCE, las personas ciegas a la hora de tener que buscar, productos básicos como leche, yogures, galletas, encuentran múltiples inconvenientes desde el momento en que tratan de llegar hasta ellos en un entorno con múltiples pasillos, estantes, y otros clientes con sus canastas o carritos de mercado, haciendo que sea complicado además seleccionar entre la diversidad de ubicaciones y variedad de productos que existen sin contar las marcas, tamaños y formatos de presentación. (Once, 2020b).

Lo anterior hace que una persona ciega deba enfrentarse a la elección de un producto en un gran almacén de supermercado, teniendo en cuenta que tampoco

se encuentran etiquetas en braille que acompañan a los productos o ubicaciones, haciendo que la experiencia de compra sea compleja y frustrante por los obstáculos que encuentran cotidianamente en el proceso de compra de productos en un supermercado de autoservicio (ONCE, 2020).

Según ONCE las personas ciegas y deficientes visuales manifiestan que, como consumidores, también tienen derecho a elegir y comprar en supermercados y grandes establecimientos, en las mismas condiciones que el resto de la ciudadanía, pudiendo conocer la variedad de artículos del mercado, las ofertas de cada momento, así como los ingredientes o la fecha de caducidad del producto que se llevan a casa. Otro factor es la falta de personal y poca disposición del acompañamiento a personas con discapacidad visual en el establecimiento (ONCE, 2020).

La discapacidad visual puede limitar a las personas en la realización de tareas cotidianas y afectar su calidad de vida, así como sus posibilidades de interacción con el mundo circundante debido a que la mayoría de las veces no logran diferenciar un producto del otro y no saben realmente qué están comprando, dependen de alguien o peor aún a no salir para evitar estresarse (GOLÁS, 2019).

Las necesidades anteriormente descritas motivan la pregunta de investigación que da origen a la presente propuesta: *¿De qué manera se pueden usar las tecnologías móviles e IoT para generar soluciones en personas con discapacidad visual mediante un sistema de estantería inteligente?*

2.1 ÁRBOL PROBLEMA

La problemática se llevó a cabo teniendo como soporte un árbol de problemas que permite identificar los factores que afectan a las personas con discapacidad visual y las consecuencias que conlleva la falta de inclusión de personas con limitaciones en un entorno de mercado (ver Ilustración 1).

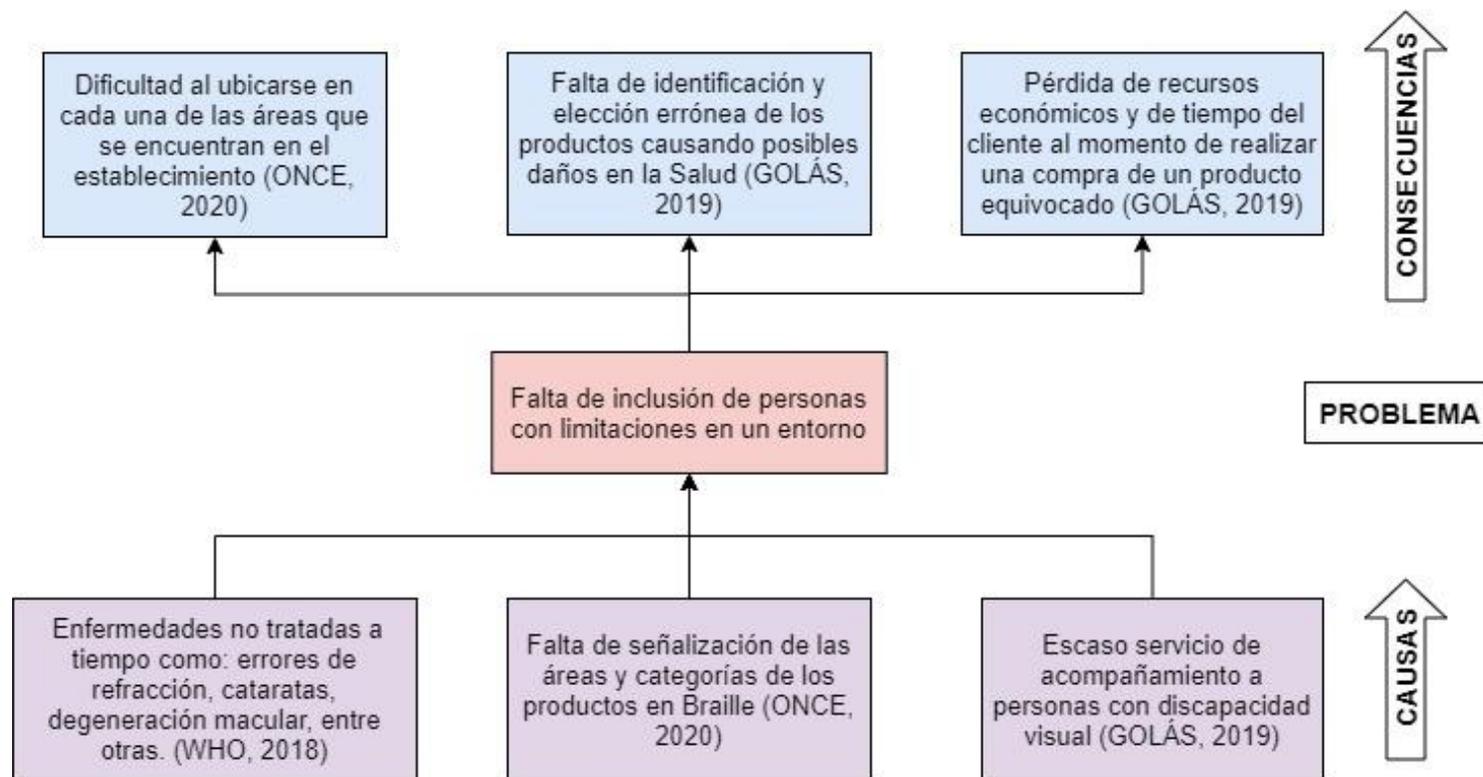


Ilustración 1. Árbol de problema
Fuente: Autores

3 JUSTIFICACIÓN

Las personas con discapacidad visual por su limitación y por las barreras que el entorno les impone, suelen sentirse excluidas; tanto que sus oportunidades de conseguir empleo o realizar compras con facilidad son pocas o nulas. Según la OMS a nivel mundial, alrededor de 1300 millones de personas cuentan con alguna deficiencia de visión ya sea de lejos o cerca y las principales causas de visión deficientes son los errores de refracción que no fueron corregidos y las cataratas; sin embargo, estas varían de un país a otro. En cuanto a la visión de lejos, se cuenta con que 188,5 millones de personas tienen una deficiencia visual moderada, 217 millones tienen una deficiencia visual de moderada a grave y 36 millones de ellas son ciegas (WHO, 2018). Por otra parte, una deficiencia de visión de cerca la padecen 826 millones de personas (Bourne et al., 2017).

Algunos de los problemas por los que se enfrentan los ciegos en su diario vivir es preguntando ¿Cómo realizo mi compra? Debido a que la mayoría de las veces no logran diferenciar un producto del otro y no saben realmente qué están comprando. Actualmente, solo los productos farmacéuticos se ven obligados a tener implantado el etiquetado braille en sus artículos (GOLÁS, 2019).

Del mismo modo, parte del proyecto número 063 que radicó el INCI (Instituto Nacional para Ciegos) en el Senado de la República de Colombia es poner en práctica el sistema braille tanto en cartas de restaurantes, como en empaques de alimentos y en las presentaciones de los medicamentos con el cual busca que el uso de este sistema esté más presente en la cotidiana. Aun así, los distintos espacios de las ciudades deben mejorar las condiciones de accesibilidad para que la población con esta discapacidad pueda acceder a las herramientas lingüísticas que esta le ofrece (Morantes, 2019).

Con el auge de las nuevas tecnologías en todas las facetas cotidianas del día a día ha aumentado la utilización de aplicaciones específicas para personas en situación de discapacidad. En efecto, “el Ministerio TIC creó en el 2013 ConVerTIC, un proyecto de inclusión con el fin de promover la inclusión social, educativa, laboral y cultural a través de uso de las tecnologías para las personas ciegas o con baja visión, mediante la entrega y masificación de los softwares Jaws, Magic y ZoomText” (Ministerio de Tecnologías de la Información y las Comunicaciones, 2014). Estas nuevas tecnologías cada vez abren más campo para la inclusión y son un gran apoyo para las personas con discapacidad.

Estas tecnologías se caracterizan por contar con dispositivos con acceso a internet como teléfonos inteligentes, Tablet. Destacando que el número de usuarios de los teléfonos inteligentes supera los 3.000 millones y se pronostica que seguirá aumentando en los próximos años (Statista, 2020). Según un informe de *Strategy Analytics* se espera que el trabajo móvil a nivel mundial llegué a los 1.87 mil millones en 2022 (Ismail, 2017).

Por lo que se planteó realizar un sistema de estantería inteligente basado en tecnologías móviles e IoT para personas con discapacidad visual en el que se lograra por medio del tacto, audio y una aplicación móvil una mejor interacción persona-producto teniendo como resultado el saber del nombre y las características del producto facilitando así la experiencia de compra.

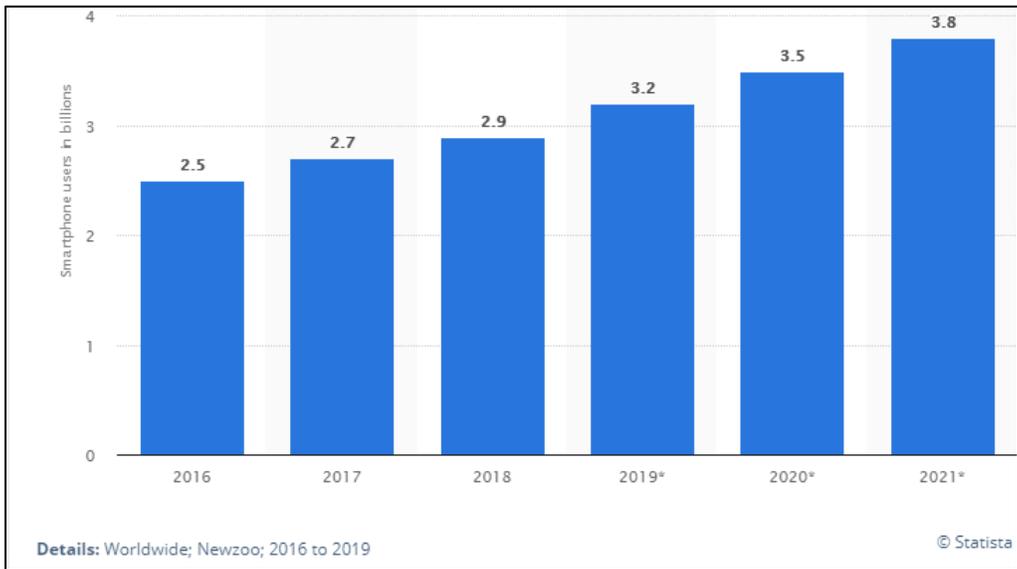


Ilustración 2. Usuarios de teléfonos inteligentes en todo el mundo 2016-2021

Fuente: Tomado de (Statista, 2020)

4 OBJETIVOS

4.1 OBJETIVO GENERAL

Construir un sistema de estante inteligente para una experiencia de compras incluyente a personas con discapacidad visual mediante tecnologías IoT y móvil.

4.2 OBJETIVOS ESPECÍFICOS

- Diseñar el sistema de estante inteligente mediante técnicas de modelado y diagramas UML a partir del establecimiento de los requerimientos funcionales y no funcionales.
- Construir el prototipo funcional a partir del desarrollo de cada uno de los módulos que lo componen.
- Validar el funcionamiento del sistema conforme al diseño.

5 MARCO REFERENCIAL

5.1 MARCO CONCEPTUAL

El marco conceptual del proyecto está desarrollado en torno a los siguientes descriptores: *Internet de las cosas, Experiencia de Usuario, Inclusión, Programación orientada a objetos, Base de datos, Discapacidad visual, Hardware, Software.*

5.1.1 Internet de las cosas

Describe la red de objetos físicos los cuales llevan sensores integrados, software y otras tecnologías, con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet (Oracle, 2019a). Siendo una arquitectura emergente que se basa en la Internet global que facilita el intercambio de bienes y servicios entre redes de la cadena de suministro, teniendo así un impacto importante en la seguridad y privacidad de los actores involucrados (Salazar & Silvestre, 2014).

5.1.2 Experiencia de usuario

La interacción Persona-Producto es un área de usuario que se centra en la interacción que tienen los usuarios con los productos, la experiencia de usuario es aplicable a productos interactivos de manera que estos puedan ser utilizados de manera segura, eficiente, eficaz y satisfactoria. La usabilidad no solo se puede definir como un atributo de calidad sino también como una metodología de diseño y evaluación (Hassan Montero, Yusef. Martin Fernandez, 2013).

5.1.3 Inclusión

“Es un enfoque que responde positivamente a la diversidad de las personas y a las diferencias individuales, entendiendo que la diversidad no es un problema, sino una oportunidad para el enriquecimiento de la sociedad, a través de la activa participación en la vida familiar, en la educación, en el

trabajo y en general en todos los procesos sociales, culturales y en las comunidades” (UNESCO, 2019).

Para obtener un mundo más equitativo y respetuoso frente a las diferencias es importante beneficiar a todas las personas sin importar sus características ni tener que excluir, por este motivo el proyecto lo que busca es la participación de las personas con discapacidad visual valorando cada uno de los aportes que estos puedan realizar a la sociedad, haciendo que se sientan más independientes y con la capacidad de realizar sus compras sin necesidad de estar acompañado y de esta manera acabar con la exclusión que viene de personas con actitudes negativas y que no tienen el reconocimiento necesario para aceptar la diversidad.

5.1.4 Programación orientada a objetos

“La programación Orientada a objetos (POO) se define como un paradigma de la programación, donde se organiza el código en clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones. Es una forma de programar, más cercana a cómo se expresan las cosas en la vida real” (Alvarez, 2019).

5.1.5 Base de datos

“Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos” (Oracle, 2019b).

5.1.6 Discapacidad visual

“Cuando se habla en general de ceguera, discapacidad visual grave o deficiencia visual, se está haciendo referencia a condiciones caracterizadas por una limitación total o muy seria de la función visual. Más específicamente,

las personas con ceguera o ciegas son aquellas que no ven nada en absoluto o solamente tienen una ligera percepción de luz, pueden ser capaces de distinguir entre luz y oscuridad, pero no la forma de los objetos (Once, 2020a).

5.1.7 Hardware

“Se conoce como Hardware a todos los componentes físicos internos de un ordenador, es decir, la parte tangible del equipo, como son: la placa base o placa madre, la CPU o procesador (unidad central de procesamiento), la memoria principal o memoria RAM, el disco duro (HD, SSD), tarjetas gráficas, tarjeta de red, entre otros. También hablamos de Hardware para referirnos a dispositivos periféricos de entrada y salida” (Gestor, 2020a).

5.1.8 Software

“El Software es lo que se conoce como la parte no física, es decir, el soporte lógico de un sistema informático, aquel que hace referencia a uno o más programas de cómputo que incluye datos, reglas e instrucciones para poder comunicarse con el ordenador y que hacen posible su funcionamiento” (Gestor, 2020b).

5.2 MARCO TEÓRICO

5.2.1 INTERNET DE LAS COSAS

Este término hace referencia a variedad de cosas que se conectan a través de Internet para poder compartir datos con otras aplicaciones que también utilicen IoT. Estos dispositivos que cuentan con internet facilitan y mejoran a las personas su forma de trabajar y vivir debido a que parte del Internet de las cosas es empleado en los hogares inteligentes que ajustan automáticamente sus luces, puertas, entre otras.

Según la ITU SECTOR DE NORMALIZACIÓN DE LAS TELECOMUNICACIONES Recomendación UIT-T Y.2060. *“Desde la perspectiva de la normalización técnica, IoT puede concebirse como una infraestructura global de la sociedad de la información, que permite ofrecer servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación (TIC) presentes y futuras”* (Unión Internacional de Telecomunicaciones, 2012).

Su uso es importante en el proyecto dado que el contacto que tiene cada producto y la señal que envía de la estantería a la aplicación móvil se transforma en una señal auditiva por medio de tecnologías móviles para dar facilidad e independencia a las personas ciegas con ayuda de las IoT.

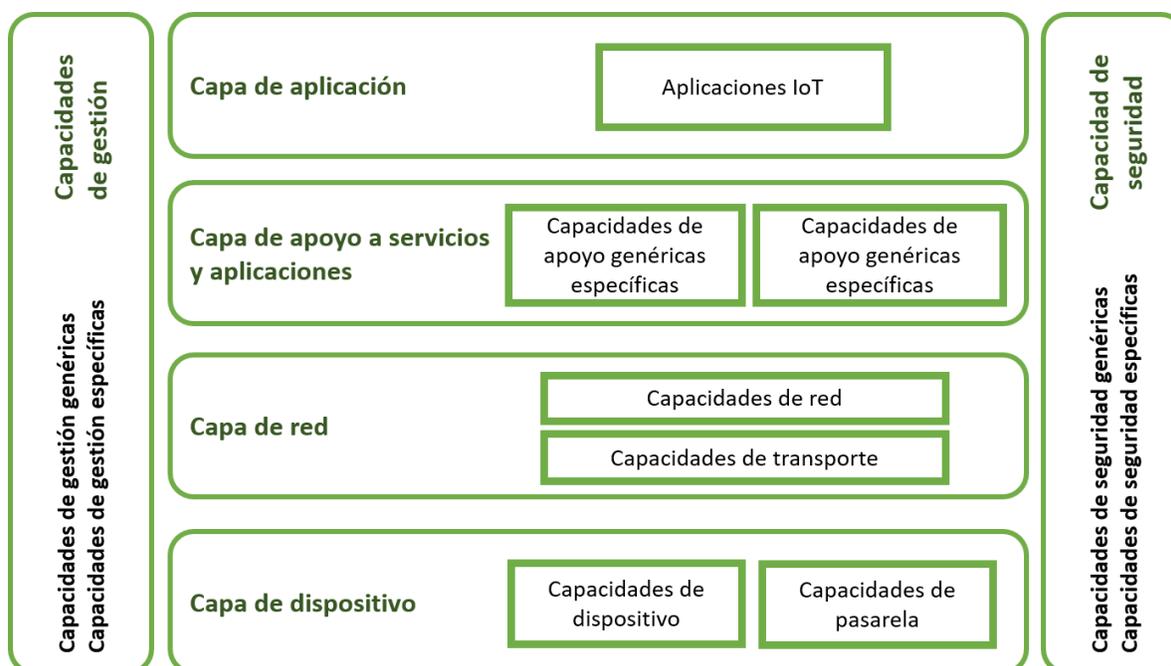


Ilustración 3. Modelo de referencia de IoT
 Fuente: Adaptado de (Unión Internacional de Telecomunicaciones, 2012)

El modelo de referencia de IoT consta de cuatro capas: aplicación, apoyo a servicios y aplicaciones, red y dispositivo; y de dos capacidades: gestión y seguridad que se relacionan directamente con las cuatro capas del modelo. A continuación, en las Tabla 1 y Tabla 2 se hace una descripción de sus características.

CAPAS	DESCRIPCIÓN
APLICACIÓN	Esta capa cuenta con las aplicaciones IoT
SOPORTE DE SERVICIOS Y APLICACIONES	<ul style="list-style-type: none"> • Capacidades de soporte genéricas: Capacidades comunes que puede utilizarlas diferentes aplicaciones IoT, tales como procesamiento o almacenamiento de datos. • Capacidades de soporte específicas: Pueden consistir en diversos grupos de capacidades precisas que ofrecen distintas funciones de apoyo a las diferentes aplicaciones IoT.
RED	<ul style="list-style-type: none"> • Capacidades de red: Ofrecen funciones de control de la conectividad en red, tales como funciones de control de acceso y de recursos de transporte, entre otros. • Capacidades de transporte: Centradas en suministrar conectividad para el transporte de información y datos específicos de servicios y aplicaciones IoT.

CAPAS	DESCRIPCIÓN
DISPOSITIVOS	<p>En dos tipos se efectúa una clasificación lógica de las capacidades de la capa de dispositivo:</p> <ul style="list-style-type: none"> • Capacidades de dispositivo: <ul style="list-style-type: none"> - Interacción directa con la red de comunicaciones - Interacción indirecta con la red de comunicación - Redes ad-hoc - Modo reposo y activo • Capacidades de pasarela: <ul style="list-style-type: none"> - <u>Soporte de interfaces múltiples:</u> <ul style="list-style-type: none"> ○ Capa de dispositivo: soportan dispositivos conectados mediante diferentes tipos de tecnologías alámbricas e inalámbricas, tales como el bus de red de control de zona (CAN), ZigBee, Bluetooth o Wi-Fi. ○ Capa de red: Se pueden comunicar a través de diversas tecnologías, tales como la red telefónica pública conmutada (PSTN), las redes de segunda o tercera generación (2G o 3G), las redes LTE (evolución a largo plazo), Ethernet o las líneas digitales de abonado (DSL). - <u>Conversión de protocolo:</u> <ul style="list-style-type: none"> ○ Cuando las comunicaciones en la capa de dispositivo utilizan protocolos diferentes. ○ Cuando en la comunicación intervienen la capa de dispositivo y la de red y se utilizan protocolos diferentes en cada una.

Tabla 1. Capas modelo de referencia IoT

Fuente: Adaptado de (Unión Internacional de Telecomunicaciones, 2012)

CAPACIDADES	DESCRIPCIÓN
GESTIÓN	<p>Las capacidades de gestión IoT pueden clasificarse en capacidades genéricas y específicas. Las capacidades de gestión genéricas en IoT son:</p> <ul style="list-style-type: none"> • Gestión de dispositivos, como activación y desactivación de dispositivos remotos, entre otros • Gestión de la topología de red local • Gestión del tráfico y la congestión, como la detección de las condiciones de saturación de red y la aplicación de reserva de recursos para los flujos de datos esenciales para la vida o urgentes.

CAPACIDADES	DESCRIPCIÓN
SEGURIDAD	<p>Hay dos tipos de capacidades de seguridad: genéricas y específicas. Las capacidades de seguridad genéricas son independientes de la aplicación y se basa en la autorización, autenticación, confidencialidad de los datos en:</p> <ul style="list-style-type: none"> - <u>En la capa de aplicación:</u> D de aplicación y protección de la integridad, protección de la privacidad, auditorías de seguridad y antivirus - <u>En la capa de red:</u> Datos de señalización y de datos de uso, y protección de la integridad de señalización. - <u>En la capa de dispositivo:</u> Autenticación, autorización, validación de la integridad del dispositivo, control de acceso, confidencialidad de datos y protección de la integridad.

Tabla 2. Capacidades del modelo de referencia IoT
Fuente: Adaptado de (Unión Internacional de Telecomunicaciones, 2012)

5.2.1.1 Descripción Técnica DE IOT

En IoT se presenta una propiedad de correspondencia la cual consiste en representar un objeto físico a nivel de información mediante uno o varios objetos virtuales, sin embargo, esta correspondencia no significa que no puedan existir objetos virtuales que no tienen ningún objeto físico asociado. En IoT los dispositivos que intervienen, además poseen capacidades de comunicación y detección, accionamiento, adquisición, almacenamiento y procesamiento de datos (Unión Internacional de Telecomunicaciones, 2012).

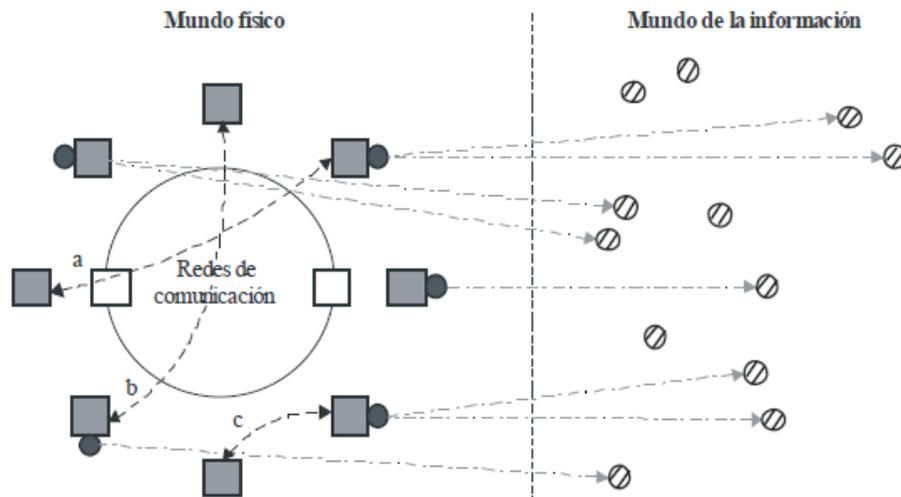




Ilustración 4. Descripción técnica de IoT

Fuente: Tomado de (Unión Internacional de Telecomunicaciones, 2012)

Estos dispositivos lo que hacen es recabar información de distintos tipos y la suministran en redes de información y comunicación para su procesamiento. Algunos de ellos lo que hacen es ejecutar operaciones en función de la información que ha sido recibida (Unión Internacional de Telecomunicaciones, 2012).

“Los dispositivos se comunican con otros dispositivos: a través de la red de comunicaciones por medio de una pasarela (caso a), por medio de una red sin pasarela (caso b) o directamente, esto es, sin utiliza la red de comunicación (caso c). También son posibles otras combinaciones de los casos a y c, y de los casos b y c; por ejemplo, los dispositivos pueden comunicarse con otros utilizando una red local (es decir, una red que ofrece conectividad local entre dispositivos y entre dispositivos y pasarelas, como una red ad-hoc) (caso c) y luego se comunican por medio de la red de comunicación a través de pasarela de red local (caso a)” (Unión Internacional de Telecomunicaciones, 2012).

5.2.1.2 Tipos de dispositivos y su relación con objetos físicos

En la siguiente imagen (ver Ilustración 5) se da a conocer los distintos tipos de dispositivos que se pueden encontrar junto a la relación que tienen con los objetos físicos.

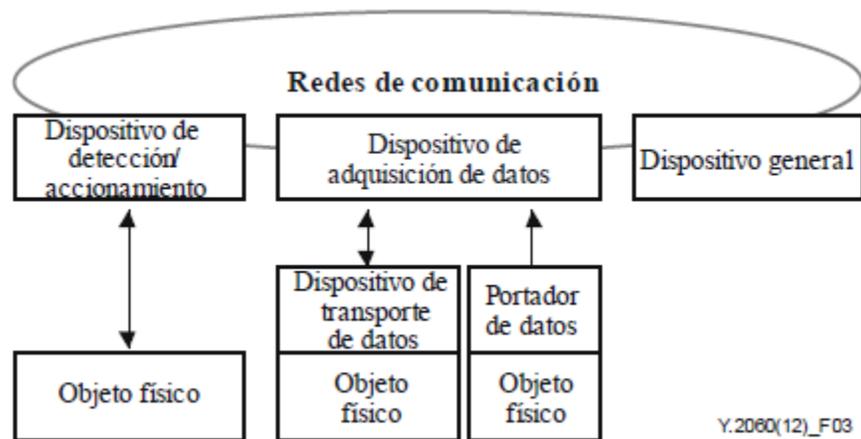


Ilustración 5. Tipos de dispositivos y su relación con objetos físicos
Fuente: Tomado de (UTI, 2012)

Descripción de los dispositivos con relación a objetos físicos (Unión Internacional de Telecomunicaciones, 2012).

- ✓ **Dispositivo de transporte de datos:** Dispositivo que se agrega a un objeto físico para conectar indirectamente el objeto físico con las redes de comunicación.
- ✓ **Dispositivo de adquisición de datos:** Dispositivo de lectura/escritura con capacidad para interactuar con objetos físicos. La interacción puede suceder indirectamente a través de dispositivos de transporte de datos, o directamente a través de dispositivos de transporte de datos unidos a los objetos físicos.
- ✓ **Dispositivo de detección y accionamiento:** Detecta o mide información de su entorno convirtiéndolas en señales electrónicas digitales. Por lo general, los dispositivos de detección y accionamiento forman redes locales que se comunican entre sí utilizando tecnologías de comunicación alámbricas o inalámbricas y utilizan pasarelas para conectarse con las redes de comunicación. También puede convertir señales electrónicas digitales procedentes de las redes de información en operaciones.
- ✓ **Dispositivo genérico:** Cuenta con capacidades de procesamiento y comunicación, este puede comunicarse mediante tecnologías alámbricas e inalámbricas con las redes de comunicación.

5.2.1.3 Desafíos y vulnerabilidades IoT

A continuación, se identifican los desafíos y vulnerabilidades que se pueden encontrar para IoT:

✓ **Ataques de denegación de servicio en Internet of Thing:**

A medida que más dispositivos se mueven hacia IP-enabled, están contribuyendo a un conjunto de cosas que se reclutan fácilmente en plataformas como botnets, que pueden utilizarse para ataques distribuidos. El uso de La distribución de ataques dificulta el rastreo de las fuentes de ataque, al mismo tiempo que hace que es más fácil recargar a los dispositivos y aplicaciones a los que se dirige (Alsaadi & Tubaishat, 2015).

✓ **Espionaje en Internet de las cosas:**

Los atacantes pasivos podrían apuntar a canales de comunicación como Internet, redes locales por cable y redes inalámbricas para recuperar datos del flujo de información. Obviamente, si un intruso interno accede a una infraestructura específica, puede obtener la información que fluye a través de la infraestructura. Si bien existen medidas de seguridad destinadas a proteger los datos y la información, la posibilidad de que un intruso pueda acceder al sistema y secuestrar los datos es real. Otro desafío que enfrentará el desarrollo de IoT es el intercambio de datos (Alsaadi & Tubaishat, 2015).

En el paradigma de IoT, los datos son de suma importancia, aunque el suministro de datos es el resultado de un contrato social entre los clientes y las empresas. Las corporaciones brindan un servicio a precio nominal o gratuito, que intercambiarán por los datos personales del cliente. Estos datos se pueden utilizar en el desarrollo posterior de servicios y productos que satisfagan las necesidades del consumidor, así como también se pueden vender a los comercializadores (Alsaadi & Tubaishat, 2015).

✓ **Captura de nodos en Internet de las cosas:**

Elementos como las farolas y los electrodomésticos están ubicados físicamente en entornos específicos y, en lugar de destruirlos, los atacantes activos pueden intentar extraer la información que contienen. En lugar de las cosas, el atacante activo también podría apuntar a la infraestructura que se utiliza para almacenar información, incluidas las entidades de almacenamiento de datos o el procesamiento de datos. Si, por otro lado, se distribuye la inteligencia real en el IoT, se utilizarán diferentes entidades para crear y procesar la información, lo que significa que los atacantes deben mejorar sus esfuerzos para controlar una cantidad similar de recursos. Sin embargo, se encuentra que la distribución de recursos actúa como un arma de doble filo. Si los atacantes están interesados únicamente en un dato en particular, pueden apuntar a los sistemas que administran la información específica ubicada en entidades centrales. Además, los ataques de captura de nodos son más peligrosos porque hay más lógica integrada en las cosas. Los nodos perceptivos dentro de la capa perceptiva normalmente crean una red ad hoc de distribución dinámica (Alsaadi & Tubaishat, 2015).

✓ **Seguridad física de los sensores:**

Los ataques físicos también podrían destruir físicamente los sensores de los dispositivos IoT o incluso dejarlos inoperables de forma permanente, lo que supondría un claro desafío para las aplicaciones relacionadas con la seguridad. El atacante podría determinar dónde se ubican los sensores en función de las propiedades de las señales recibidas, después de lo cual pueden inhabilitarlo físicamente, destruirlo o robarlo (Alsaadi & Tubaishat, 2015).

5.2.2 TECNOLOGÍAS MÓVILES

Las tecnologías móviles consisten en dispositivos portátiles de comunicación bidireccional, dispositivos informáticos y la tecnología de red que los conecta (IBM, 2020). A continuación, se muestran cuatro tipos de redes móviles:

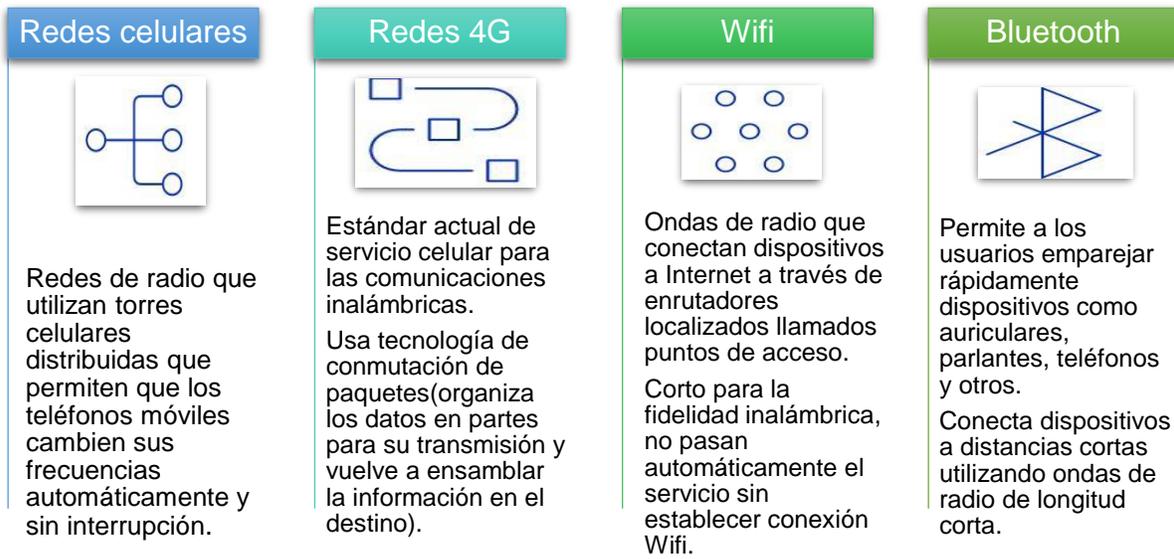


Tabla 3. Tipos de redes móviles
Fuente: Adaptado de (IBM, 2020)

5.2.2.1 Arquitectura Android

Android es una pila de software de código abierto basado en Linux creada para una variedad amplia de dispositivos y factores de forma. A continuación, se muestran los principales componentes de la plataforma Android.

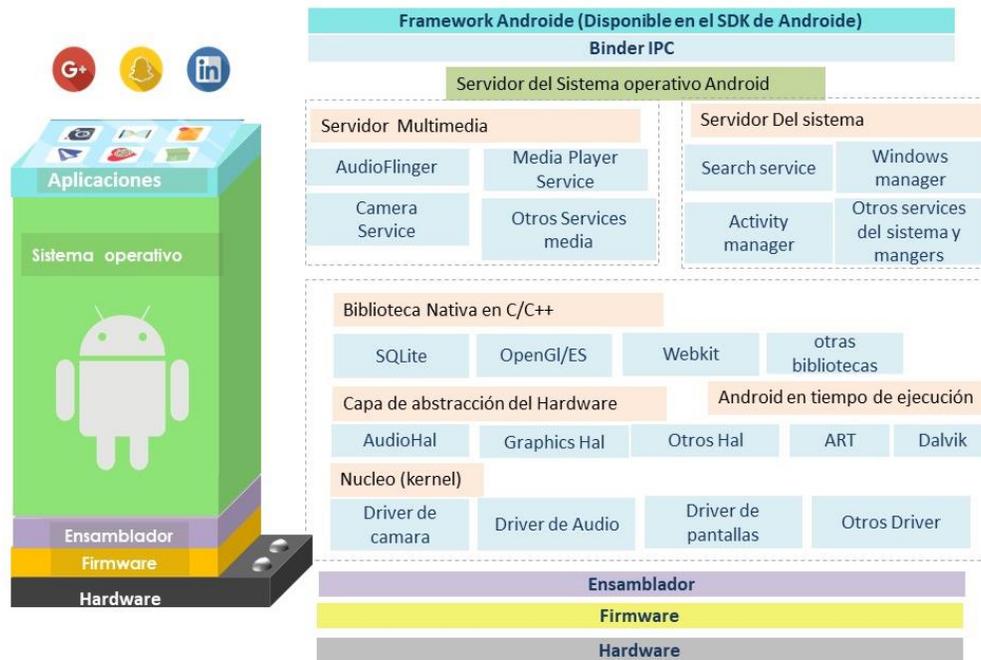


Ilustración 6. Arquitectura de Androido
Fuente: Adaptado de (Invarato, 2018)

A continuación, una descripción de cada nivel de la arquitectura de Android.

- **Aplicaciones:** Se encuentran tanto las apps incluidas por defecto de Android como aquellas que el usuario va añadiendo.
- **Framework de Aplicaciones:** Conjunto de herramientas de desarrollo de cualquier aplicación, estas utilizan el mismo conjunto de API y el mismo "*framework*", representado por:
 - **Activity Manager:** Conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android.
 - **Windows Manager:** Gestiona las ventanas de las aplicaciones y utiliza la librería Surface Manager.
 - **Telephone Manager:** Incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.).
 - **Content Provider:** Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android.
 - **View System:** Proporciona elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, funcionalidades más frecuentes.
 - **Location Manager:** Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
 - **Notification Manager:** Mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wifi disponible, ubicación en un punto determinado.
- **Librerías:** Librerías utilizadas por Android que han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Se puede decir, que estas librerías hacen parte del corazón de Android, junto al núcleo basado en Linux. Entre las librerías más importantes ubicadas aquí, se pueden encontrar:
 - **Librería libc:** Incluye todas las cabeceras y funciones según el estándar del lenguaje C.

- **Librería Surface Manager:** Encargada de componer los diferentes elementos de navegación de pantalla. Gestiona las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.
- **OpenGL/SL y SGL:** Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. OpenGL/SL maneja gráficos en 3D y permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D.
- **Librería Media Libraries:** Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android.
- **FreeType:** Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.
- **Librería SSL:** Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras
- **Librería SQLite:** Creación y gestión de bases de datos relacionales.
- **Librería WebKit:** Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.
- **Tiempo de ejecución de Android:** Al mismo nivel que las librerías de Android se sitúa el entorno de ejecución. Constituidas por las *Core Libraries*, que son librerías con multitud de clases Java y la máquina virtual Dalvik.
- **Núcleo Linux:** Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Lo primero que se debe realizar para que pueda ser utilizado desde Android, al incluir el fabricante un elemento nuevo de hardware, es crear las librerías de control o drivers necesarios dentro de este kernel de Linux embebido en el propio Android.

5.2.2.2 Estructura de una Aplicación

Una aplicación típica en Android se compone de un nivel superior y vistas de detalles y con éstas se conecta si la jerarquía de navegación es profunda y compleja (VideumCorp, 2015).

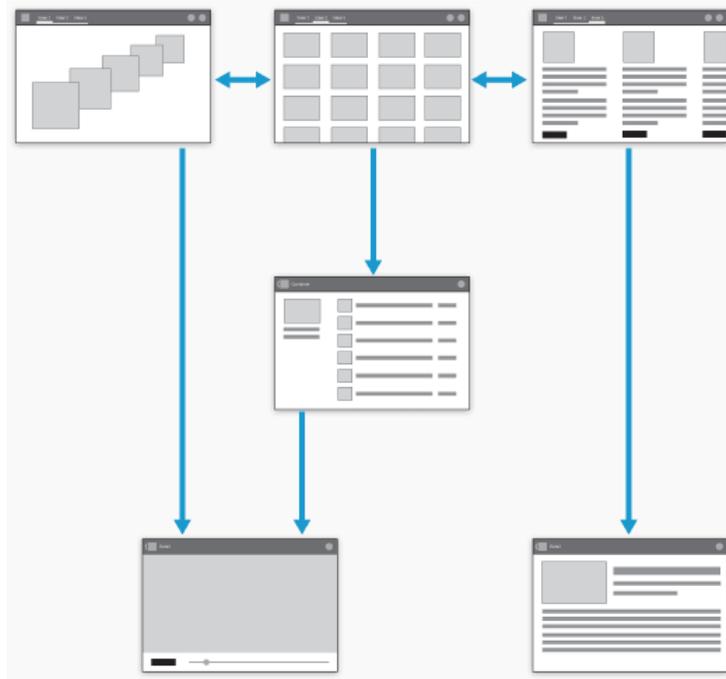


Ilustración 7. Estructura general de una aplicación
Fuente: Tomado de (VideumCorp, 2015)

A continuación, se presenta la descripción de la estructura general de una aplicación:

NIVEL	DESCRIPCIÓN
VISTAS DE CATEGORÍA	Vistas que aparecen en medio de la imagen y que permite incidir más profundamente en los datos de la aplicación.
NIVEL SUPERIOR	La disposición (layout) de la pantalla de inicio es la primera pantalla que las personas van a ver después de abrir la aplicación, así que lo que se busca es ofrecer una experiencia gratificante tanto para usuarios nuevos como los que ya usaban la aplicación.
PRESENTA CONTENIDO	Aquí se presenta las disposiciones que sean visualmente atractivas y apropiadas para el tamaño de pantalla y los tipos de datos, evitando las pantallas que solo se usan para navegar.

NIVEL	DESCRIPCIÓN
VISTA DETALLE – EDICIÓN	<p>Es donde se usa o crean los datos. Son aquellas que están abajo del todo en la imagen.</p> <ul style="list-style-type: none"> • Encontrar la manera de mostrar el contenido útil en la pantalla inicial. • Usar barras de acción para proporcionar una navegación consistente. • Utilizar una navegación horizontal y accesos directos sobre las jerarquías más superficiales. • Usar la opción multi-selección que permita al usuario actuar sobre conjuntos de datos. • Permitir una rápida navegación entre los detalles de los elementos mediante vistas deslizable o desplazamiento entre vistas.

Tabla 4. Estructura aplicación
Fuente: Adaptado de (VideumCorp, 2015)

5.2.3 TECNOLOGÍA WIFI

“Permite que los dispositivos, intercambien información entre sí, estableciendo de esta manera, una red. Wifi es una tecnología de red inalámbrica a través de la cual los dispositivos, como computadoras ya sean portátiles o de escritorio, dispositivos móviles (teléfonos inteligentes y accesorios) y otros equipos (impresoras y videocámaras), pueden interactuar con Internet” (Cisco, 2020).

“Esta conectividad a Internet se logra a través de un router inalámbrico. De este modo, cuando se accede a una red wifi, se conecta a un router inalámbrico que permite que los dispositivos que admiten wifi interactúen con Internet” (Cisco, 2020).

La red inalámbrica está conformada por varios dispositivos donde se intercomunican entre sí, o pueden conectarse con una red exterior como lo es el internet, la mayor ventaja de los dispositivos inalámbricos es que no se necesita de elementos físicos, además es importante tener en cuenta que existen muchos tipos de redes inalámbricas donde la diferencia puede radicar en aspectos como la arquitectura, tecnología, entre otros (Rios, 2011).

Según la UIT-T, el Wifi es una red inalámbrica que funciona gracias al uso de las frecuencias de radio, para la debida comunicación entre computadoras u otros dispositivos de red (Feng, 2017).

Dos componentes de la tecnología inalámbrica son (Rios, 2011):

- **Dispositivos cliente:** Hacen parte aquellos que solicitan la conexión a la red inalámbrica como lo son los ordenadores portátiles, tablets, entre otros.
- **Punto de acceso (en inglés Access Point):** Dispositivo tecnológico usado para conectar los dispositivos clientes entre sí o con el resto de infraestructura cableada de la organización. También es usado como puertas de enlace a otras redes, como Internet (Router).

Así mismo, existen dos tipos de redes inalámbricas (Rios, 2011):

- **Modo Ad Hoc:** Aquellos que no utilizan routers o puntos de acceso, y se comunican de manera directa como, por ejemplo, Bluetooth o WI-FI.
- **Modo infraestructura:** Aquel que necesita un punto de acceso, es decir que necesita routers, por medio del punto de acceso router, se conecta a los diversos puntos de red.

5.2.3.1 Funcionamiento de la tecnología Wifi

El estándar IEEE 802.11 define los protocolos que permiten la comunicación con los dispositivos inalámbricos actuales que admiten wifi, en los que también se incluye routers y puntos de acceso inalámbrico. Los puntos de acceso inalámbrico son compatibles con diferentes estándares IEEE. Ahora bien los estándares funcionan con diversas frecuencias, proporcionan distintos anchos de banda y admiten distintas cantidades de canales (Cisco, 2020).

5.2.3.2 Estándares Wifi

Los estándares Wifi son regulados por el Instituto de Ingenieros Electrónicos y Electricos (IEEE), quienes son los encargados de decidir cómo se nombran y se

manejan los distintos estándares para que funcionen de manera adecuada y global (Commerce, 2019).

De este modo, se obtienen diferentes tipos de Wifi, todos empiezan con los números 802.11 y les sigue una letra, normalmente entre más se acerque a la “z” más rápida será la velocidad de la red. Casi todos funcionan en un radio de 30 metros aproximadamente (Commerce, 2019).

Estas letras que caracterizan cada una de las redes Wifi que existen, indican lo siguiente:

- El alcance que tiene la señal inalámbrica.
- La cantidad de información que se puede enviar a través de la red.
- Si es o no compatible con otros estándares.

Se cuenta con dos variables para la clasificación de los distintos estándares:

- **Velocidad:** Son los datos que puede transmitir la red por segundo (Mbps).
- **Frecuencia:** La radiofrecuencia que tiene la red, puede ser de 2.4 GHz o 5 GHz.

En la siguiente tabla, se da a conocer los estándares que se encuentran para la tecnología wifi:

Nombre	Descripción	Velocidad	Frecuencia
802.11a	No es compatible con redes b o g. Este es uno de los estándares más antiguos, pero aún hoy en día es utilizado por muchos dispositivos.	54 Mbps es el máximo, pero normalmente de 6 a 24 Mbps	5 GHz
802.11b	Compatible con redes g. En realidad, g fue hecho para ser compatible con b para soportar más dispositivos.	11 Mbps	2.4 GHz

Nombre	Descripción	Velocidad	Frecuencia
802.11d	D no es realmente un tipo de red propio. Incluye información adicional como información de puntos de acceso y otra información especificada por las regulaciones de los diferentes países. Normalmente, esto se combina con otras redes como 802.11ad.	N/A	N/A
802.11g	El tipo de red más popular. Su combinación de velocidad y compatibilidad con las versiones anteriores lo convierte en un buen complemento para las redes actuales.	54 Mbps	2.4 GHz
802.11n	El tipo de red más rápido. 100 Mbps es común, aunque velocidades de hasta 600 Mbps son posibles bajo condiciones perfectas. Lo hace usando múltiples frecuencias a la vez y uniéndolo esa velocidad.	100 Mbps	2.4 y 2.5 GHz

Tabla 5. Estándares del Wifi
Fuente: Adaptado de (Commerce, 2019)

Wi-Fi Alliances es una organización sin ánimo de lucro dueña de la marca Wi-Fi, esta corporación se encarga de promover y certificar toda la tecnología y productos concernientes a wifi, haciendo que wifi esté ajustada a los estándares de interconectividad que menciona la norma IEEE 802.11, dicha norma establece los

lineamientos para el funcionamiento de una red de área local inalámbrica (Rios, 2011).

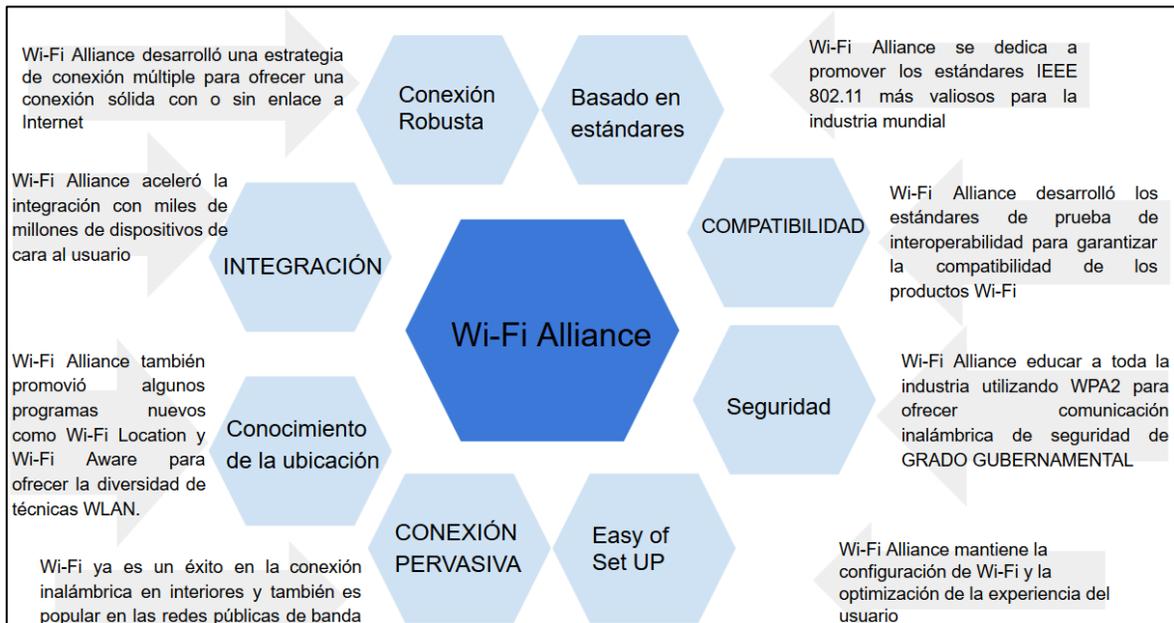


Ilustración 8. La alianza Wi-Fi de hoy
Fuente: Tomado de (Feng, 2017)

5.2.4 ARQUITECTURA RASPBERRY

Raspberry Pi consiste en una arquitectura basada en ARM, que, mediante una placa del tamaño de una tarjeta de crédito, presenta las características propias de una computadora del tipo de placa única o SBC, por sus siglas en inglés (*Single Board Computer*). Las tecnologías ARM de Raspberry Pi Model 2B y 3-series, incluidos 3B, 3B +, 3A +, 3CM y 3CM + permiten la integración con diversos tipos de sistemas operativos como Fedora, Raspbian, Ubuntu y Kali entre otras distribuciones de Linux. (Red Hat, 2020).

5.2.4.1 Especificaciones

La placa Raspberry pi comprende una memoria de programa (RAM), procesador y chip de gráficos, CPU, GPU, puerto Ethernet, pines GPIO, zócalo Xbee, UART, conector de fuente de alimentación. Y varias interfaces para otros dispositivos externos. Así mismo, requiere almacenamiento masivo, para eso se utiliza una tarjeta de memoria flash SD. De modo que la placa Raspberry Pi se iniciará desde

esta tarjeta SD de manera similar a como una PC se inicia en Windows desde su disco duro (Admin, 2020).

A continuación, se muestra una descripción de cada una de las partes para configurar la Raspberry Pi.

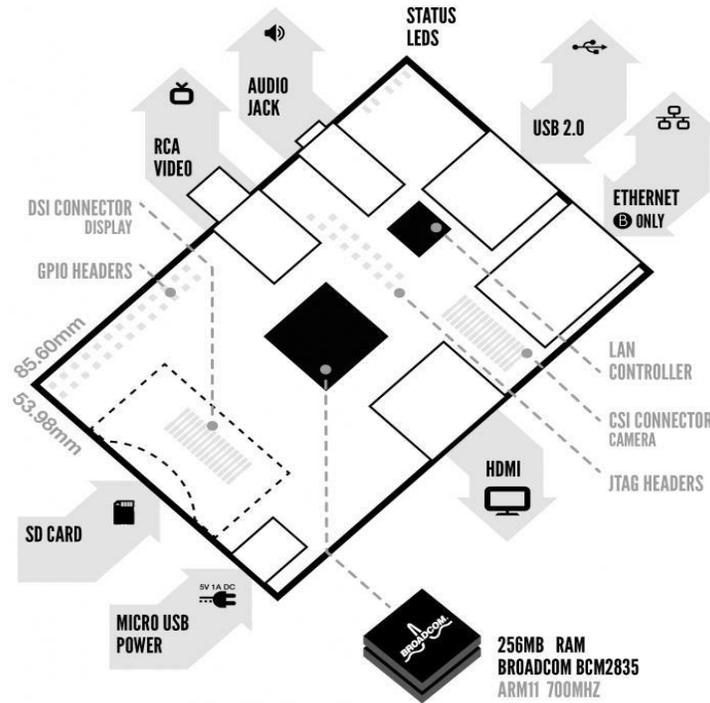


Ilustración 9. Configuración Raspberry Pi
Fuente: Tomado de (Raspberry, 2015)

5.2.4.2 Diferencia entre Raspberry Pi y Arduino

RASPBERRY PI	ARDUINO
Mini controlador	Microcontrolador
Dispone de sistema operativo completo	Puede controlar pequeñas aplicaciones para usos concretos
Cuenta con diferentes conexiones a internet. Un puerto Wifi y una conexión Ethernet	Necesita una placa de expansión para poder conectarse a internet. Esto supone un coste más en el caso de que necesitemos conexión a internet en nuestro proyecto.
Procesador de 1,2 GH y de 64 bits	Se puede conectar muchos componentes y sensores a los conectores de la placa.

RASPBERRY PI	ARDUINO
Salida HDMI para poder conectarle un monitor y cuatro puertos USB para añadirle diferentes periféricos	Incluye un firmware sencillo que permite que nos podamos conectar a ella a través del puerto USB.

Tabla 6. Diferencia entre Arduino y Raspberry Pi

Fuente: Autores, (Lucas, 2019)

En este caso, tuvo importancia en la realización del proyecto debido a que interactúa con toda la parte central haciendo la función de un computador para alcanzar un buen resultado al obtener tanto el audio como los demás componentes.

5.2.4.3 Descripción Wifi Raspberry

La Raspberry Pi 3 utiliza un estándar Wifi de 802.11n. Esto significa que al contar con Wifi ya no necesita depender de una conexión tipo Ethernet para tener acceso a Internet. Así mismo, la incorporación de Bluetooth supone que ahora se puede conectar un teclado o mouse inalámbricos para controlar la Pi 3 y comunicarse de manera inalámbrica con sensores y otros dispositivos (Whitney, 2016).

De tal manera que, la Raspberry Pi es una pequeña tarjeta madre provista de CPU, tarjeta de gráficos, memoria y otros componentes. Al ser conectada a un monitor, un teclado y un mouse, se puede usar como una computadora básica, diseñada para personas con pasatiempos específicos, desarrolladores y estudiantes (Whitney, 2016).

5.2.5 FIREBASE

Firestore nació como una alternativa de integración entre todas las soluciones de Google para que se comuniquen entre ellas a través de un único SDK. De esta forma, lo que se busca por parte de Google es que esta plataforma sea la vía más rápida y fácil de construir una app (Inc, 2020).

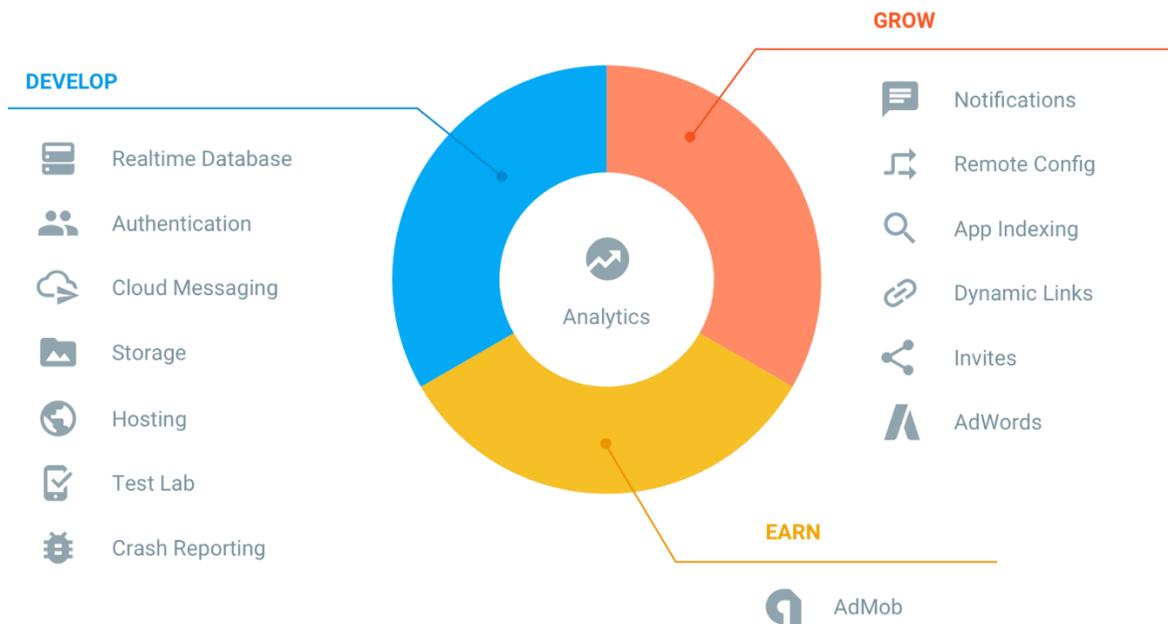


Ilustración 10. Aspectos Firebase
Fuente: Tomado de (Inc, 2020)

Además, Firebase consiste en una plataforma *cross-device* por lo que permite el desarrollo para aplicaciones nativas (Android y iOS) y web. Firebase ayuda, principalmente, en 3 aspectos como son el desarrollo (*develop*), el crecimiento de los usuarios (*grow*) y la monetización de la app (*Earn*).

ASPECTOS	DESCRIPCIÓN
<p style="text-align: center;">DEVELOP DESARROLLO</p> <p>Firestore posee una gran variedad de servicios que ayudan en la creación de las apps de una forma ágil</p>	<ul style="list-style-type: none"> • Bases de datos en tiempo real que permiten almacenar y sincronizar los datos en tiempo real. • Autenticación que gestiona el acceso a toda la infraestructura. • Hosting que suministra un alojamiento estático y seguro para la app. • Almacenamiento construido para desarrolladores que necesitan almacenar y servir contenido como fotos y vídeos generados por los usuarios. • Cloud messaging que permite entregar de manera fiable mensajes y notificaciones sin coste. • Configuración remota que permite cambiar el comportamiento y la apariencia sin una publicación ni actualización de una actualización de la app. • Test Lab para Android que permite testear la app en dispositivos virtuales alojados en un centro de datos de Google.

ASPECTOS	DESCRIPCIÓN
<p style="text-align: center;">GROW CRECIMIENTO DE LOS USUARIOS</p> <p>El crecimiento es un proceso difícil, por este motivo Firebase facilita el crecimiento y la fidelidad de los usuarios mediante varias tecnologías</p>	<ul style="list-style-type: none"> • Crash Reporting que suministra información accionable para ayudar a diagnosticar y solucionar problemas en la app. • Enlaces dinámicos que proporcionan la mejor experiencia en diferentes plataformas y motivan al usuario a instalar la app si todavía no la tienen. • Permite a los usuarios a enviar invitaciones por email o SMS y compartir la app directamente con sus contactos. • AdWords posibilita alcanzar clientes potenciales con campañas online. • Notificaciones que, desde la consola de Firebase, se pueden diseñar y enviar a los usuarios para re-enganchar a los que se han “olvidado” de la app. Más adelante se puede analizar el impacto que han tenido. • Indexación de la app para que los usuarios que van en busca de un contenido relacionado puedan ir directamente a consumir la app.
<p style="text-align: center;">EARN MONETIZACIÓN DE LA APP</p> <p>Existen tecnologías AdMob para monetizar y rentabilizar las apps mediante anuncios atractivos.</p>	<p>Firestore Analytics</p> <ul style="list-style-type: none"> • Ocupa el lugar central en Firebase. • Se trata de una herramienta de analítica gratuita de la que se pueden obtener insights de uso y fidelidad por parte de los usuarios y tomar decisiones en base a datos. • Firebase Analytics proporciona una capacidad ilimitada para generar informes, hasta un total de 500 eventos que pueden ser definidos mediante el SDK de Firebase. • Permite la definición de audiencias personalizadas en base a propiedades de los usuarios, datos de dispositivos o eventos personalizados. Estas audiencias pueden usarse para la configuración de notificaciones u otras funciones de Firebase.

Tabla 7. Aspectos de Firebase
Fuente: Adaptado de (Inc, 2020)

Firestore proporciona funciones como estadísticas, bases de datos, informes de fallas y mensajería, de manera que se pueda ser más eficiente y enfocarse en los usuarios (Inc, 2020)

Firestore *Realtime Database* es una base de datos alojada en la nube donde los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando se compilan las apps multiplataforma con sus SDK de iOS, Android y JavaScript, todos sus clientes comparten una instancia de *Realtime*

Database y reciben actualizaciones automáticamente con los datos más recientes (Inc, 2020).

Las apps de Firebase continúan respondiendo, incluso sin conexión, dado que el SDK de Firebase *Realtime Database* hace que los datos persistan en el disco. Cuando se restablece la conexión, el dispositivo cliente recibe los cambios que faltaban y los sincroniza con el estado actual del servidor. Así mismo, se puede acceder a Firebase *Realtime Database* directamente desde un dispositivo móvil o un navegador web; no se necesita un servidor de aplicaciones. La seguridad y la validación de datos están disponibles a través de las reglas de seguridad de Firebase *Realtime Database*: reglas basadas en expresiones que se ejecutan cuando se leen o se escriben datos (Inc, 2020).

La API de *Realtime Database* está diseñada para permitir solo operaciones que puedan ser ejecutadas rápidamente. Eso permite crear una excelente experiencia de tiempo real que puede servir a millones de usuarios sin afectar su capacidad de respuesta (Inc, 2020).

5.2.5.1 Configuración de las reglas de Realtime Database

Realtime Database proporciona un lenguaje de reglas declarativas que permite definir cómo se deben estructurar los datos, cómo se deben indexar y cuándo se pueden leer y escribir (Firebase community, 2020).

Lenguaje de reglas en cascada que separa la autorización de la validación.

- Las reglas de *Realtime Database* protegen las operaciones de lectura y escritura desde los SDK para dispositivos móviles.
- Transmisión en cascada de reglas de lectura y escritura.
- Debes validar los datos por separado con la regla *validate*.

5.2.5.2 Reglas de seguridad de Realtime Database en Firebase

En el desarrollo de aplicaciones móviles uno de los factores más importantes y complejos es la seguridad. En su mayoría se debe poner en funcionamiento un

servidor que determine la identidad del usuario, es decir, la autenticación y que controle lo que puede hacer cada uno de ellos (Firebase, 2020a).

Es por esto que en las reglas de seguridad que da Firebase quitan la capa intermedia, el servidor, y permiten especificar los permisos basados en rutas de acceso para los clientes que se conectan directamente a los datos. Así mismo, en *Realtime Database*, las reglas de seguridad de Firebase son expresiones similares a las de JavaScript, contenidas en un documento JSON y utilizan la siguiente sintaxis (ver Ilustración 11) (Firebase, 2020a).

```
{
  "rules": {
    "<<path>>": {
      // Allow the request if the condition for each method is true.
      ".read": <<condition>>,
      ".write": <<condition>>,
      ".validate": <<condition>>
    }
  }
}
```

Ilustración 11. Sintaxis reglas de seguridad en Realtime Database
Fuente: (Firebase, 2020a)

De acuerdo a lo anterior, existen tres elementos básicos en la regla (Firebase, 2020a):

- **Ruta:** La ubicación de la base de datos. Lo que dice que duplica la estructura JSON de la base de datos.
- **Solicitud:** Son los métodos que usa la regla para otorgar acceso. Las reglas *read* y *write* otorgan acceso amplio de lectura y escritura, mientras que las reglas *validate* actúan como verificación secundaria para otorgar acceso según los datos entrantes o existentes.
- **Condición:** Condición que permite una solicitud si se evalúa como verdadera

5.2.5.3 Reglas de seguridad de Firebase

Estas reglas de seguridad de Firebase son flexibles y extensibles y se usan para asegurar los datos y definir a que datos pueden acceder los usuarios en

Cloud Firestore, *Firebase Realtime Database* y *Cloud Storage*. Las reglas de Firebase Database se interponen entre los datos y los usuarios maliciosos. De tal manera que se pueden escribir reglas ya sean simples o complejas con el fin de que protejan los datos de la aplicación al nivel de detalle que se requiera (Firebase, 2020c).

En la siguiente tabla se muestran las funciones clave con las que cuenta las reglas de seguridad de Firebase.

FUNCIONES CLAVE

Flexibilidad	Escribe reglas personalizadas que se adapten a la estructura y comportamiento de tu app. Las reglas usan lenguajes que te permiten aprovechar tus propios datos para autorizar el acceso.
Nivel de detalle	Las reglas pueden ser tan amplias o limitadas como necesites.
Seguridad independiente	Debido a que las reglas se definen fuera de tu app (en Firebase console o Firebase CLI), los clientes no son responsables de aplicar la seguridad, los errores no comprometen los datos y los datos siempre se encuentran protegidos.

Tabla 8. Funciones claves reglas de seguridad de Firebase
Fuente: Adaptado de (Firebase, 2020c)

Firebase se ha certificado e implementado las prácticas de más alta calidad para lograr la integridad de la información. Debido a que la seguridad es muy importante para las empresas que siempre quieren saber dónde está su información y su estado (Guillermo, 2018).

5.2.5.4 Conexión de la aplicación móvil a Firebase

Para conectar una aplicación de Android a Firebase se puede realizar mediante una de las siguientes opciones:

- **Opción 1:** Usar el flujo de trabajo de configuración de Firebase *console*, una de las opciones más recomendadas.

En esta parte, agregar Firebase al proyecto implica realizar tareas en Firebase *console* y en el proyecto abierto de Android (por ejemplo, descargar archivos de configuración de Firebase desde la consola y transferirlos al proyecto de Android).

- **Opción 2:** Usar *Firebase Assistant* de Android Studio para esta opción se requiere configuración adicional.

En *Firebase Assistant* se registra la app con un proyecto de Firebase y se agregan los archivos y el código necesarios del servicio al proyecto de Android, todo desde Android Studio.

5.2.5.5 SDK de Firebase Android

El SDK de Firebase para Android es la forma oficial de agregar Firebase a la aplicación de Android, aquí se pueden ver en fuente abierta como serían los SDK para incluir en Android al igual que los SDK de las extensiones de Kotlin (Source, 2020).

Fuente abierta

Incluye los siguientes SDK de Firebase para Android:

- firebase-common
- firebase-database
- firebase-firestore
- firebase-functions
- firebase-inappmessaging-display
- firebase-storage

Extensiones de Kotlin

Los siguientes SDK de Firebase para Android tienen bibliotecas de extensión de Kotlin que permiten escribir código Kotlin más idiomático cuando se utiliza Firebase en la aplicación (Source, 2020).

- firebase-common
- firebase-crashlytics
- firebase-dynamic-links

- firebase-firestore
- firebase-functions
- firebase-inappmessaging
- firebase-inappmessaging-display
- firebase-remote-config
- firebase-storage
- firebase-database

5.2.5.6 Relación entre los proyectos, las apps y los productos de Firebase

Como se puede ver, “un proyecto de Firebase es la entidad de nivel superior de Firebase. En un proyecto se debe registrar las apps web, para iOS o Android a fin de crear apps de Firebase. Después de registrar las apps con Firebase, se puede agregar los SDK de Firebase para la cantidad de productos de Firebase que se quiera, como *Analytics*, *Cloud Firestore*, *Performance Monitoring* o *Remote Config*” (Firebase, 2020b).

5.2.5.7 Normatividad de Firebase

La organización internacional para la estandarización (*International Organization for Standardization*) conocida comúnmente como ISO es una organización no gubernamental a la cual hacen parte varios cuerpos normativos de distintas naciones (Cloud, 2020).

En relación a esto, la familia ISO 27001 es un conjunto de normas que ayudan a las organizaciones a proteger los recursos relacionados con la información, estableciendo los requisitos para implantar un sistema de gestión de seguridad de la información (SGSI), describiendo así buenas prácticas y distintos controles de seguridad para ayudar a gestionar estos riesgos (Cloud, 2020).

Es por esto que Google ha obtenido la certificación ISO 27001 para los sistemas, aplicaciones, personas, tecnología, procesos y centros de datos que utiliza en varios de los productos que ofrece, en el que también hace parte Google Analytics for Firebase, Google Data Studio, entre otros. Siendo así, cuenta con el certificado ISO 27001 de Google LLC del 2019 (Analytics, 2020).

5.2.5.8 Comparativo entre conectividad Wifi y Bluetooth

A continuación, encontramos un cuadro comparativo donde se hace relación a Wifi y Bluetooth identificando unas de sus ventajas, desventajas e inconvenientes de cada una de ellas.

BLUETOOTH	WIFI
Ancho de banda de banda 2,4 GHz	Ancho de banda de bandas 2,4 GHz y 5 GHz
Alcance de 10 metros	Alcance de 100 metros
Consumo de energía más bajo	Consumo de energía más alto
Suele ser mucho más lento al ofrecer menos ancho de banda que Wifi, motivo por el que también la calidad del audio transferido es peor	Más rápido y mejor calidad de audio
Solo se pueden conectar dispositivos de dos en dos, de punto a punto	Permite establecer una red de dispositivos conectados entre sí simultáneamente.
A nivel de seguridad presenta desventajas al no requerir de un proceso de autenticación	<i>Wired Equivalent Privacy (WEP)</i> , en español Privacidad equivalente a cableado, es el primer mecanismo de seguridad de cifrado incluido en el estándar IEEE 802.11 para redes Wireless.

Tabla 9. Comparativo entre conectividad Wifi vs Bluetooth
Fuente: Adaptado de (García, 2020)

5.2.6 KOTLIN

Una de las funciones más importantes de Kotlin es la excelente interoperabilidad que tiene con Java. Debido a que el código Kotlin compila el código de bytes de JVM (*Java Virtual Machine*), el código Kotlin puede llamar directamente al código Java, y viceversa. Eso quiere decir que se puede aprovechar las bibliotecas de Java existentes directamente desde Kotlin. Además, la mayoría de las API de Android están escritas en Java y pueden ser llamadas directamente desde Kotlin (Android Developers, n.d.)

En Kotlin, todo es un objeto en el sentido de que se puede llamar a funciones miembro y propiedades en cualquier variable. Algunos de los tipos pueden tener una representación interna especial, por ejemplo, los números, los caracteres y los valores booleanos se pueden representar como valores primitivos en tiempo de ejecución, pero para el usuario parecen clases ordinarias.

5.2.6.1 Beneficios de Kotlin para el desarrollo de Android

De igual modo, se encuentran beneficios que brinda Kotlin para el desarrollo de Android en los que se pueden encontrar (Kotlin, 2020b):

- **Menos código combinado con mayor legibilidad.** Dedique menos tiempo a escribir su código y a trabajar para comprender el código de los demás.
- **Lenguaje y entorno maduros.** Desde su creación en 2011, Kotlin se ha desarrollado continuamente, no solo como un lenguaje sino como un ecosistema completo con herramientas robustas. Ahora está perfectamente integrado en Android Studio y muchas empresas lo utilizan activamente para desarrollar aplicaciones de Android.
- **Soporte de Kotlin en Android Jetpack y otras bibliotecas.** Las extensiones KTX agregan características del lenguaje Kotlin, como corrutinas, funciones de extensión, lambdas y parámetros con nombre, a las bibliotecas de Android existentes.
- **Interoperabilidad con Java.** Puede utilizar Kotlin junto con el lenguaje de programación Java en sus aplicaciones sin necesidad de migrar todo su código a Kotlin.
- **Soporte para desarrollo multiplataforma.** Puede usar Kotlin para desarrollar no solo Android, sino también iOS, backend y aplicaciones web. Disfrute de los beneficios de compartir el código común entre las plataformas.
- **Código de seguridad.** Menos código y mejor legibilidad conducen a menos errores. El compilador de Kotlin detecta estos errores restantes, lo que hace que el código sea seguro.
- **Fácil aprendizaje.** Kotlin es muy fácil de aprender, especialmente para los desarrolladores de Java.
- **Gran comunidad.** Kotlin tiene un gran apoyo y muchas contribuciones de la comunidad, que está creciendo en todo el mundo. Según Google, más del 60% de las 1000 aplicaciones principales en Play Store usan Kotlin

5.2.6.2 Evolución de Kotlin

Como se ha podido observar Kotlin está diseñado para ser una herramienta pragmática para programadores. Es por esto, que cuando se trata de la evolución del lenguaje, su naturaleza pragmática se refleja en los siguientes principios (Kotlin, 2020a):

- Mantenga el lenguaje moderno a lo largo de los años.
- Manténgase en el circuito de retroalimentación constante con los usuarios.
- Haga que la actualización a nuevas versiones sea cómoda para los usuarios.

5.2.6.3 Comparativo Kotlin vs java

LENGUAJE	PROS	CONTRA
	<ul style="list-style-type: none">• Es más rápida de compilar, liviana y evita que las aplicaciones aumenten de tamaño.• Cualquier fragmento de código escrito en Kotlin es mucho más pequeño en comparación con Java, ya que es menos detallado y menos código significa menos errores.• Kotlin compila el código en un código de bytes que se puede ejecutar en la JVM. Por lo tanto, todas las bibliotecas y marcos creados en Java se pueden mover y ejecutar en un proyecto de Kotlin.• El script de Kotlin se puede utilizar para configurar proyectos en Android Studio para las ayudas de autocompletado, y ayuda a reducir la detección de errores en tiempo de compilación.• Kotlin es seguro contra NullPointerException (el error del billón de dólares).• Kotlin incorpora corrutinas, así como interoperabilidad con Javascript para desarrollo web.	<ul style="list-style-type: none">• Kotlin no es tan popular, por lo que la comunidad de desarrolladores es escasa.• Kotlin coincide con patrones débiles y la legibilidad del código inicial se vuelve un poco difícil de leer y comprender al principio.• Tiene una pequeña comunidad de soporte ya que Kotlin es más joven que Java. Además, tiene menos bibliotecas, publicaciones en blogs y tutoriales.• Kotlin tiene una curva de aprendizaje pronunciada, y cambiar de equipo a Kotlin debido a la sintaxis concisa del lenguaje puede ser un verdadero desafío.• Hay muy pocos desarrolladores de Kotlin disponibles en el mercado. Entonces, encontrar un mentor experimentado es bastante complicado.

LENGUAJE	PROS	CONTRA
	<ul style="list-style-type: none"> • Java es multiplataforma y funciona en prácticamente cualquier dispositivo, servidor o sistema operativo • La codificación de Java es sólida y es imposible que la instrucción de Java corrompa la memoria o comprometa los datos de otras aplicaciones de OS X • Java está orientado a objetos y facilita la creación de aplicaciones modulares y la reutilización de las partes que contribuyen a la solidez. • Está listo para usar y, con Java, obtendrá una gran cantidad de código de terceros listo para usar. • Al comparar el rendimiento de Kotlin vs Java, con otros lenguajes, Java es fácil de usar, compila y hace que la depuración y la implementación sean aún más simples • Java es un lenguaje de código abierto que garantiza la seguridad, ya que muchas de sus bibliotecas son administradas por empresas de confianza como Google, Apache y otras. 	<ul style="list-style-type: none"> • La sintaxis que utiliza Java puede ser un poco complicada o engorrosa en comparación con otros lenguajes como C + o Python. • Con Java, es imposible acceder a ciertos contenidos que pueden ser incompatibles con el dispositivo o equipo que se está utilizando. • No es fácil acceder a las nuevas mejoras de Java en el desarrollo móvil. • Hay momentos en que Java causa problemas con el diseño de la API de Android. • El desarrollo basado en pruebas para Java requiere escribir más código y conlleva un riesgo mucho mayor de errores de programación y errores. • Java es un poco más lento en comparación con otros lenguajes de programación y requiere mucha memoria del sistema

Tabla 10. Comparativo Kotlin vs Java
Fuente: Adaptado de (XenonStack, 2020)

5.2.7 SENSOR MPR121

Este grupo controla el filtrado cuando los datos son línea de base. Este complemento táctil cuenta con 12 sensores táctiles capacitivos. La detección táctil detecta cuando una persona ha tocado uno de los electrodos del sensor. La detección táctil capacitiva se usa para cosas como tabletas y teléfonos reactivos al tacto, así como paneles de control para electrodomésticos. Este módulo permite crear dispositivos electrónicos que pueden reaccionar al contacto humano, con hasta 12 sensores individuales (Industries, 2019).

El Módulo tiene 12 agujeros que se pueden sujetar con cables de pinza de cocodrilo. Conectando un lado del clip al escudo y el otro lado a algo eléctricamente conductor (como metal) o lleno de agua (Industries, 2019).

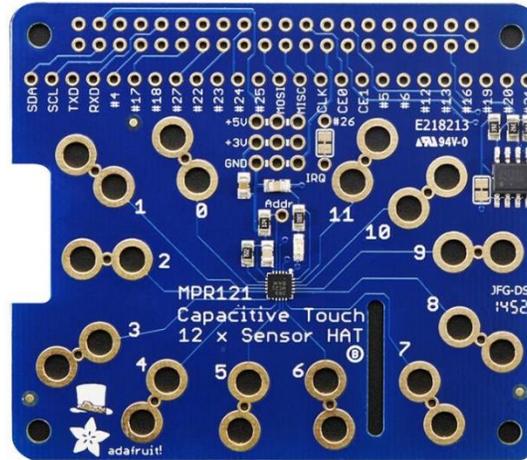


Ilustración 12. Sensor MPR121
Fuente: Tomado de (Industries, 2019)

A continuación, se observa una tabla con la configuración de cada uno de los electrodos que contiene el sensor MPR121.

NÚMERO DEL PIN	NOMBRE DEL PIN	DESCRIPCIÓN
1	IRQ	Pin de interrupción de colector abierto
2	SCL	Reloj 2C
3	SDA	Datos I2C
4	ADDR	Selección de dirección I2C
5	VREG	VREG: conexión de tapa de 0,1 μ F
6	VSS	Suelo
7	REXT	Resistencia externa - 75 k Ω
8	ELE0	Electrodo 0
9	ELE1	Electrodo 1
10	ELE2	Electrodo 2
11	ELE3	Electrodo 3
12	ELE4	Electrodo 4
13	ELE5	Electrodo 5
14	ELE6	Electrodo 6
15	ELE7	Electrodo 7
16	ELE8	Electrodo 8
17	ELE9	Electrodo 9
18	ELE10	Electrodo 10

NÚMERO DEL PIN	NOMBRE DEL PIN	DESCRIPCIÓN
19	ELE11	Electrodo 11
20	+V3	3 voltios
21	+V5	5 voltios

Tabla 11. Configuración electrodos Sensor MPR121
Fuente: Adaptado de (MARTÍN, 2010)

En la siguiente tabla se da a conocer un cuadro comparativo entre dos tipos de sensores táctiles, el MPR03X y el MPR121 con respecto a la característica y el beneficio que ofrecen.

CARACTERÍSTICA	MPR03X	MPR121	BENEFICIO
Detección de capacitancia	x	x	Mide electrodos de 1 pF a 2000 pF desde el mismo dispositivo
Alto rango dinámico		x	Incluso en un solo diseño, los electrodos pueden variar en forma y tamaño
Configuración de electrodo independiente		x	Configuración automática independiente para cada electrodo
Configuración automática		x	Reconfiguración personalizable si la línea de base cae fuera del rango
Reconfiguración automática	x	x	Permite el acondicionamiento de datos simple a través del filtro de promediado
Sistema de filtrado de dos etapas	x	x	La frecuencia de muestreo puede variar entre 1 ms y 128 ms
Frecuencia de muestreo configurable	x	x	Evita el rebote en el reconocimiento táctil al proporcionar histéresis
Detección táctil	x	x	El toque y la liberación se informan mediante interrupción y aserción
Umbral creciente y decreciente	x	x	Mantiene la línea de base filtrando las señales táctiles
IRQ de umbral de toque y liberación		x	Seguimiento de línea de base independiente para cambios de capacitancia positivos y negativos
Sistema de seguimiento de línea de base		x	La velocidad de filtrado se puede reducir para aplicaciones que requieren un tiempo de respuesta rápido y a largo plazo.

CARACTERÍSTICA	MPR03X	MPR121	BENEFICIO
Sistema de seguimiento de línea de base bidireccional		x	Promedio de la línea de base
Retraso del filtro de línea base		x	Las teclas atascadas se detectan y calibran con un sistema personalizable
Seguimiento de línea base de eliminación de clave atascada	x		Detecta la proximidad mediante la combinación de electrodos como un modo configurable independiente
Detección de proximidad		x	Detecta la proximidad mediante la combinación de electrodos como un pseudoelectrodo que permite
Modo de detección de proximidad	x	x	Proximidad y detección táctil
Pseudoelectrodo sensor de proximidad		x	La detección de proximidad se informa mediante la aserción de interrupciones
Detección de proximidad IRQ GPIO / LED		x	Los electrodos pueden ser multiusos como sensor táctil o LED GPIO
Ocho pines de conducción LED compartidos			Los electrodos pueden ser multiusos como sensor táctil o LED GPIO
Extensor GPIO			Si no todos los pines se utilizan como electrodos, el resto pueden ser simples extensores GPIO

Tabla 12. Comparativo sensor MPR03X y MPR121
Fuente: Adaptado de (Smarter, 2014)

5.3 MARCO LEGAL

5.3.1 Ley 1581 de 2012 Protección de Datos

“Mediante la Ley 1581 de 2012 se expidió el Régimen General de Protección de Datos Personales, el cual, de conformidad con su artículo 1, tiene por objeto "desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma”(Comercio & Turismo, 2013).

Esta Ley es importante en el contexto del proyecto ya que para el registro de los usuarios se deben aplicar estrategias que permitan garantizar su cumplimiento para un óptimo manejo de los datos personales de los usuarios.

5.3.2 Ley 1618 de 2013

“Esta Ley establece las disposiciones para garantizar el pleno ejercicio de los derechos de las personas con discapacidad. El objeto de esta ley es garantizar y asegurar el ejercicio efectivo de los derechos de las personas con discapacidad, mediante la adopción de medidas de inclusión, acción afirmativa y de ajustes razonables y eliminando toda forma de discriminación por razón de discapacidad” (D.C., 2013).

Esta Ley es importante en el contexto del proyecto ya que, pretende por medio de diferentes alternativas velar por los derechos de las personas con discapacidad buscando que sean aceptadas en las actividades como cualquier persona teniendo en cuenta todas las medidas de inclusión, lo cual va acorde con la finalidad del proyecto propuesto.

5.3.3 Artículo 13 de la Constitución Política

“Todas las personas nacen libres e iguales ante la ley, recibirán la misma protección y trato y gozarán de los mismos derechos, libertades y oportunidades sin ninguna discriminación. El Estado promoverá las condiciones para que la igualdad sea real y efectiva y adoptará medidas en favor de grupos discriminados o marginados” (Congreso, 1991).

Este artículo de la constitución es importante en el contexto del proyecto ya que el proyecto pretende promover las condiciones de igualdad de las personas ciegas posibilitando el libre acceso a sus compras como a cualquier ciudadano.

5.3.4 Política Pública Nacional de Discapacidad e Inclusión Social

“El objetivo de la Política Pública Nacional de Discapacidad e Inclusión Social - PPDIS, es asegurar el goce pleno de los derechos y el cumplimiento de los deberes de las personas con discapacidad, sus familias y cuidadores para el periodo 2013 – 2022. Dichos derechos deben ser garantizados partiendo de la comprensión de que cada territorio tiene unas características físicas, económicas, culturales y políticas que le son propias y que configuran su particularidad, razón por la cual las acciones de política pública deben adaptarse a dichas particularidades, siempre teniendo presente que el vínculo nación – territorio es el que permite que toda la sociedad colombiana, desde su diversidad, transite por la misma vía para el logro de la inclusión plena de las personas con discapacidad” (Ministerio Colombia, 2020).

Esta Política Pública es importante en el contexto del proyecto ya que busca que las personas ciegas se sientan confiadas y no discriminadas de tal manera que puedan gozar de independencia sin acompañamiento de una persona fomentando la inclusión.

6 ANTECEDENTES

Teniendo como base la búsqueda de información se encontraron algunos proyectos que tienen relación como antecedentes al proyecto que se plantea en esta propuesta.

6.1 Autonomía de personas ciegas en supermercados mediante realidad aumentada

Este proyecto por medio de la solución propuesta busca aumentar de forma efectiva la autonomía de las personas invidentes a la hora de realizar una compra.

Se trata de unas gafas de realidad aumentada, las cuales, por medio del uso de cámaras ubicada en ellas, el usuario podrá captar tanto obstáculos como identificar distintos productos y sus características, códigos de barra o QR. Del mismo modo, para que el usuario pueda orientarse por el supermercado se cree conveniente instalar balizas de proximidad en puntos estratégicos, éstas serán identificadas por las gafas. Así, los clientes que llevan puestas sus gafas sabrán en que pasillo del supermercado se encuentran y de esta manera poder acercarse al pasillo donde se encuentre el producto que desean.

Para procesar toda la información las gafas estarán conectadas al teléfono móvil mediante conexión inalámbrica y por medio de una aplicación específica; el teléfono móvil será el que gestione las comunicaciones con un servidor local. La interfaz que se usa son las gafas de realidad aumentada, estas gafas reproducirán la información de forma auditiva para los usuarios que tengan ceguera (Máster, 2017).

6.2 Generación de paisajes háptico-auditivos para asistencia de personas con discapacidad visual mediante el diseño de un lenguaje de representación sensorial

Este proyecto basa su objetivo general en “Diseñar un prototipo funcional de un dispositivo de asistencia a personas con discapacidad visual que permita la representación del entorno a través de un lenguaje háptico-sonoro para facilitar su movilidad en espacios públicos”.

Trata de la creación de un lenguaje de representación sensorial háptico auditivo que pueda ser implementado en un dispositivo digital para asistir a personas con personas con discapacidad visual. Esto conlleva a diseñar el lenguaje para que la información óptica compleja se pueda transmitir por el canal auditivo y de esta manera sea recopilada a través de Smart Glasses o de las cámaras de los Smartwatches y así sea transmitida al dispositivo de cómputo central donde será analizado el hospedar el lenguaje (Remolina, 2019).

6.3 Diseño e implementación de una aplicación móvil Android para el seguimiento de rutas de transporte urbano en el municipio de Yopal

El trabajo está dirigido a los usuarios de la ciudad de Yopal, departamento de Casanare (Colombia), que toman el servicio de transporte urbano con el objetivo de facilitar la información. Correspondiente a las rutas de los paseos en minibús. El proyecto muestra los resultados de una creación de una aplicación móvil Android para la visualización de las rutas de transporte urbano en la ciudad de Yopal, manejando la metodología XP, que ofrece un marco Configurable a las situaciones del proyecto. La creación de la aplicación se ha realizado con Android Studio el entorno de desarrollo integrado, Google Maps para Android y Firebase para iniciar la base de datos y sesión con la cuenta de Google y facilitar al usuario el proceso de registro e ingreso, lo cual conlleva a una mayor usabilidad (Galindo Pérez & Suárez Vargas, 2017).

7 ESTADO DEL ARTE

La búsqueda de artículos e información para el desarrollo de este documento fue realizada utilizando las siguientes palabras clave: *Blind, Mobile Application, Internet of Things, visual impairment - Aplicación móvil, personas ciegas, discapacidad visual*. La información hallada se encuentra en un rango de fechas entre el 2012 y 2020, en donde fueron seleccionados un total de 8 documentos para el desarrollo del estado del arte del Proyecto de investigación.

7.1 PROCESO DE BÚSQUEDA

Teniendo como punto de partida la pregunta de investigación establecida: *¿De qué manera se pueden usar las tecnologías móviles e IoT para generar soluciones en personas con discapacidad visual mediante un sistema de estantería inteligente?*; y de conformidad con problema planteado: Falta de mecanismos de compra ágiles e incluyentes para personas con discapacidad visual, errores en los productos de compra; se desarrolló la presente metodología de revisión sistemática la cual aporta el soporte científico a la propuesta de proyecto: Sistema de estantería inteligente basado en tecnologías móviles e IoT para personas con discapacidad visual.

Dicha búsqueda sistemática de evidencia disponible se llevó a cabo por las dos proponentes del proyecto, en las bases de datos: ACM, Google Scholar, Scopus, donde se utilizan las palabras claves en inglés y español: ciego, aplicación móvil, internet de las cosas, discapacidad visual - blind, mobile application, internet of things, visual impairment.

A partir de las palabras claves se inició el proceso de búsqueda en las bases de datos propuestas. Una vez se cumple con esta etapa, se aplica un primer filtro de selección a partir de los criterios de inclusión para mantener únicamente las evidencias que cumplan con las siguientes condiciones: La evidencia debe corresponder a documentos completos; El idioma debe ser en inglés y/o español; La fecha de publicación debe ser entre 2012 y 2020; El tipo debe corresponder con

artículos de revistas, y artículos de conferencias. La temática debe estar relacionada con IoT, tecnologías móviles y soluciones tecnológicas a personas con discapacidad.

Una vez depurada la evidencia la siguiente etapa corresponde a un nuevo filtro de refinación a partir de la revisión específica y el análisis minucioso del título, las palabras claves, el resumen y el contenido, excluyendo, además los documentos repetidos, para su utilización y aplicación en el proyecto. En la Ilustración 13 se muestra un diagrama de flujo en detalle de la metodología del proceso de búsqueda, el cual se describe en el siguiente apartado.

PROCEDIMIENTO PARA LA SELECCIÓN DE ESTUDIOS

El proceso inició con el establecimiento de los parámetros de búsqueda y las palabras claves, a partir de los cuales las dos investigadoras se dividieron la búsqueda por temáticas en las bases de datos propuestas:

Paso 1: El proceso de búsqueda inicia escogiendo las cinco bases de datos: ACM, Google Scholar, Scopus.

Paso 2: En cada base de datos se incluyeron las palabras claves (Blind, Mobile App, Internet of things, visually impairment) teniendo en cuenta el conector AND, y que las palabras se encuentren en título y resumen.

Paso 3: Depurar la información filtrándola mediante una revisión específica y exhaustiva según las temáticas, evaluando los resúmenes de cada artículo teniendo en cuenta los criterios establecidos en el protocolo de búsqueda.

Paso 4: Analizar la información de los artículos por medio de una ficha donde se permite desglosar el documento seleccionado. Teniendo en cuenta que puedan ser útiles en la extracción de la información requerida.

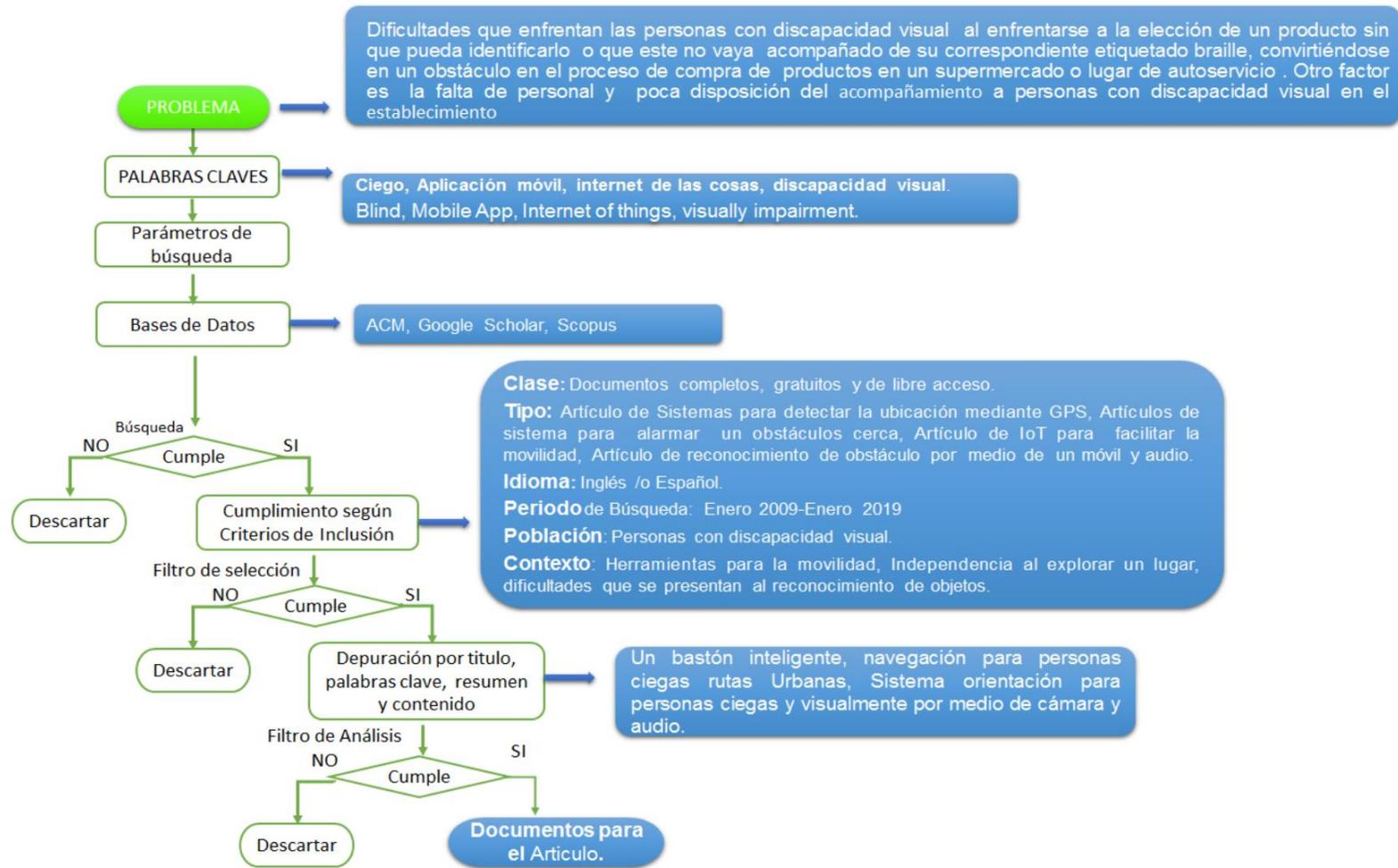


Ilustración 13. Diagrama de Flujo del Proceso de Búsqueda

Fuente: Autoras

PROCESO DE BÚSQUEDA

En la base de datos Scopus, se buscaron los términos: Blind, Mobile App, Visually impairment, usando el operador AND, donde se obtuvieron como resultado inicial 148 documentos. Se refinan los resultados aplicando los filtros correspondientes a los criterios de inclusión: Tipos de (Acceso abierto), Años (2012-2020), Nombre del autor, Área temáticas (Ciencias de la Computación), Tipo de documento (Artículo), Idioma (inglés y español), Etapa de publicación, Título de fuente, Palabra clave, afiliación, Patrocinador de financiación. Luego de aplicar los filtros quedan 42 documentos y en seguida se hace un análisis, seleccionado 4 artículos donde se obtiene la información requerida.

- (1) Amemiya T., Yamashita, J., Hirota, K., & Hirose, M. (2012, 27 marzo). Colorplate: bloques líderes v virtuales para sordo ciegos: un buscador de caminos en tiempo real por interfaz híbrida verbal-no verbal y espacio de etiqueta RFID de alta densidad.
- (2) P., Johnson, V., Strothotte, T., Raab, A., Fritz, S., & Michel, R. (2017). MOBIC: Designing a Travel Aid for Blind and Elderly People.
- (3) Ran, L., Helal, S., & Moore, S. (2012). Drishti: An Integrated Indoor/Outdoor Blind Navigation System and Service.
- (4) Manduchl, R., & Coughlan, J. (2015, 8 noviembre). Functional assessment of a camera phone-based wayfinding system operated by blind and visually impaired users.

En la base de datos Google Scholar, se buscaron los términos: Aplicación móvil, personas ciegas, discapacidad visual, usando el operador AND, donde se obtuvieron como resultado inicial 94 documentos. Se aplicaron los filtros correspondientes a los criterios de inclusión: Cualquier momento (Intervalo específico 2012-2020), Ordenar por relevancia, Cualquier idioma (inglés y español). Luego de aplicar los filtros se obtienen 9 documentos para en seguida hacer una

revisión específica y exhaustiva donde se incluyen 2 artículos de los cuales se obtiene la información requerida.

- (1) (Kher Chaitrali , Kadam, & Yogita, 2015) Un bastón inteligente para los ciegos.
- (2) (Andò, 2017) Accesibilidad móvil: Focalización de cámara para ciegos

En la base de datos ACM, se buscaron las palabras claves: Aplicación móvil, personas ciegas, discapacidad visual, donde se obtiene como resultado inicial 102 documentos. Se Refinen los resultados aplicando los filtros correspondientes a los criterios de inclusión: Nombre, Institución, Autores, Editores, Asesores, Revisores, Publicaciones, Nombres de revistas / revistas, Actas / Nombres de libros, Todas las publicaciones, Tipo de contenido, Formatos (PDF), Premio de papel Editor, Fecha de publicación (Últimos 5 años). Quedando 12 documentos luego de aplicar los filtros se hace un análisis, seleccionado 2 artículos donde se obtiene la información requerida.

- (1) (Bermeo, Bravo, Punin, Ordoñez, & Huerta, 2018). Sistema de detección de obstáculos para mejorar la movilidad de las personas con discapacidad visual
- (2) (Gallagher, Wise, Li, Dempster & Chris 2012) Sistema de posicionamiento en interiores basado en la fusión de sensores para ciegos y deficientes visuales:

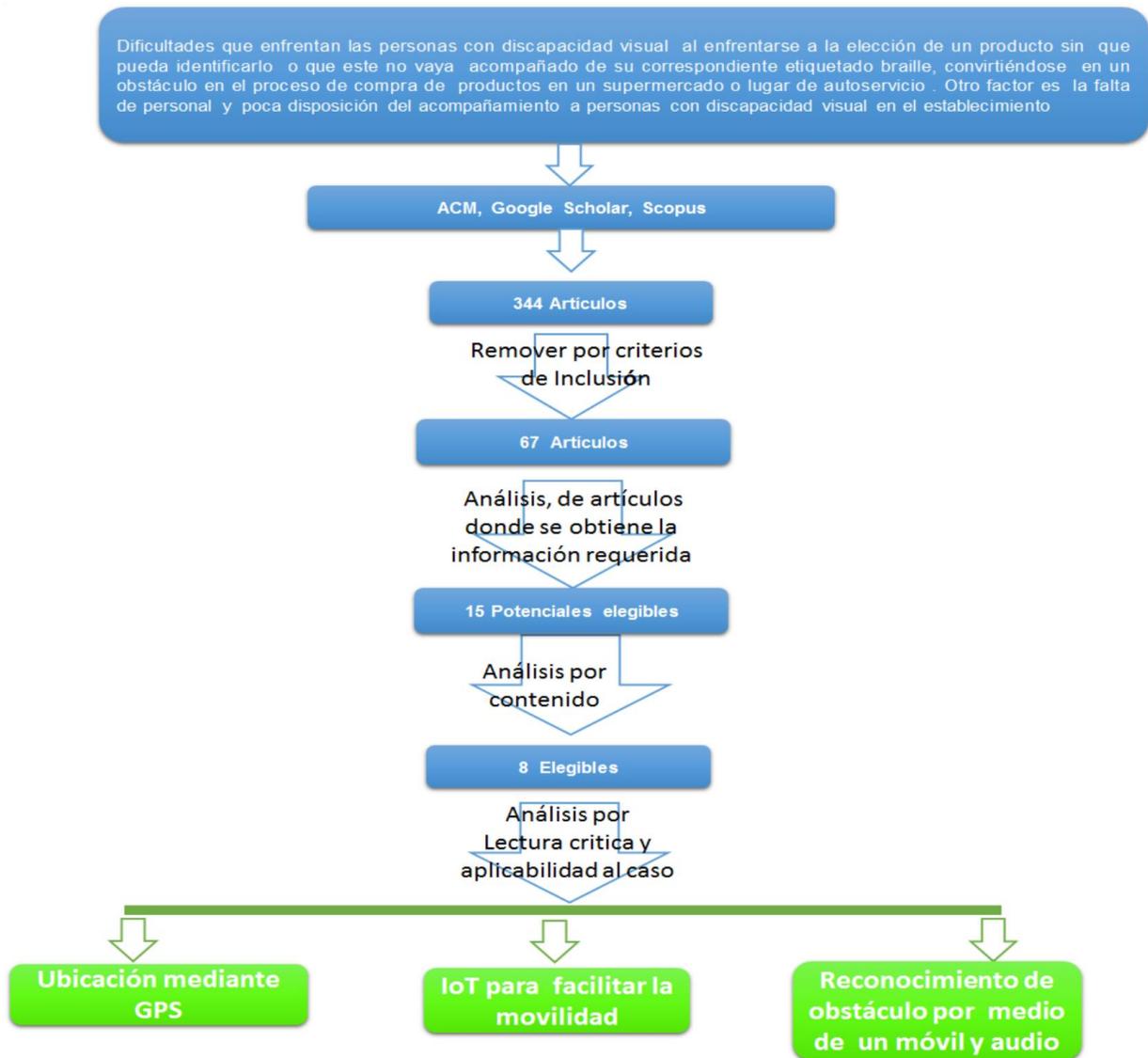


Ilustración 14. Diagrama proceso de Búsqueda

Fuente: Autores

A continuación, exponemos los artículos más relevantes en la investigación.

Un bastón inteligente para los ciegos: En este artículo se muestra un dispositivo inteligente que avisa a la persona ciega sobre la aparición de objetos en función de la distancia entre la persona y el obstáculo. Por medio de salidas de voz, así como con alertas de vibración y actualización reciente de su ubicación utilizando GPS en un servidor al que pueden acceder familiares. Además, implementa la aplicación Android monitoreo del usuario es utilizada para encontrar la ubicación de la persona ciega. Esta aplicación debe ser registrada usando los detalles de autenticación y dirección de la IP del servidor. Esta ubicación se obtiene cada vez que se toca el botón. (Chaitrali, Dabhade, Snehal, Swati & Aarti, 2015).

Una ayuda de navegación para personas ciegas: En este documento se muestra un sistema capaz de proporcionar información a los ciegos sobre las rutas urbanas para caminar y proporcionar información en tiempo real sobre la distancia de los obstáculos colgantes dentro de los 6 m a lo largo de la ruta de viaje por delante del usuario. El sistema sugerido puede detectar el entorno circundante a través de sensores de sonido y enviar comentarios vibrotáctiles al usuario sobre la posición de los obstáculos más cercanos en el rango. (Andò, 2017).

Sistema de detección de obstáculos para mejorar la movilidad de las personas con discapacidad visual: En el artículo proponemos un sistema para detectar obstáculos en el plano frontal para mejorar la movilidad de las personas con discapacidad visual. El sistema propuesto se basa en sensores de ultrasonido y una aplicación Android para procesar y comunicar información útil, inalámbrica, ergonómica y fácil de usar. Los resultados indican que el sistema es capaz de detectar (en promedio) 80% de los obstáculos frontales a la altura de la cara. (Bermeo, Bravo, Punin, Ordoñez, & Huerta, 2018).

Sistema de posicionamiento en interiores basado en la fusión de sensores para ciegos y deficientes visuales: En este documento propone un sistema basado en basada en la localización precisa en tiempo real en grandes espacios. Además de las capacidades básicas de navegación, nuestro sistema también informa al usuario sobre los puntos de interés cercanos y los problemas de accesibilidad (por ejemplo, escaleras adelante). (Gallagher, Wise, Li, Dempster & Chris 2012)

MOBIC: Designing a Travel Aid for Blind and Elderly People: En este documento se muestra el desarrollo de una aplicación proporcionando información relevante para la micro-navegación, como el tipo de superficie y la probabilidad de ciertos obstáculos. la computadora integra un receptor GPS, brújula y telecomunicaciones móviles instalaciones de comunicaciones junto con dispositivos de entrada / salida apropiados por voz a través del auricular. (Johnson, Strothotte, Raab, Fritz & Michel, 2017)

Bloques líderes virtuales para ciegos sordos: un buscador de caminos en tiempo real: En este artículo muestra una Aplicación de realidad aumentada en la movilidad para sordos ciegos que consiste en una interfaz portátil para Finger-Braille, uno de los métodos de comunicación más utilizados entre las personas sordas y ciegas en Japón. consiste en etiquetas de identificación de radiofrecuencia activa (RFID). (Amemiya, Yamashita, Hirota, & Hirose, M. 2012).

Drishti: un sistema y servicio integrados de navegación a ciegas para interiores / exteriores; En este documento muestra una aplicación de una interfaz de comunicación vocal para guiar a los ciegos y ayudarles a viajar en los lugares conocidos y desconocidos ambientes. utiliza el DGPS como su sistema de localización para mantener al usuario como lo más cerca posible de la línea central de las aceras de en el campus y en el centro de la ciudad; proporciona al usuario una ruta óptima por medio de su enrutamiento dinámico y capacidad de redireccionamiento. (Ran, Helal, & Moore, 2012).

Evaluación funcional de una cámara basada en un teléfono sistema orientación para personas ciegas y visualmente usuarios determinados: en este artículo muestra un Sistema de localización de personas ciegas que utiliza un teléfono con cámara para determinar la ubicación del usuario con respecto a los marcadores de color, publicados en lugares de interés (como oficinas), que son detectados automáticamente por el teléfono. Las señales de los marcadores de color están especialmente diseñadas para ser detectadas en tiempo real en entornos desordenados usando un software de visión computarizada que se ejecuta en el teléfono. (Manduchi, & coughlan 2015)

A continuación, en la tabla se presentan las tecnologías, protocolos y dispositivos utilizados en cada uno de los artículos mencionados anteriormente.

CITACIÓN	TITULO	TIPO	TECNOLOGÍAS / PROTOCOLOS / DISPOSITIVOS
(Kher Chaitrali, Kadam, & Yogita, 2015)	Bastón inteligente para ciegos	Artículo	Sensor de infrarrojo, Android Studio, Dispositivos de navegación inteligente, Unidad de PCB, RFID-Identificación por radiofrecuencia
(Andò, 2017)	Una ayuda de navegación para personas ciegas	Artículo	Sensor ultrasónico, Microcontrolador, acelerómetro ADXL213, dispositivo sintetizador de voz
Ran, L., Helal, S., & Moore, S. (2012).	Drishti un sistema y servicio integrados de navegación ciega para interiores exteriores	Artículo	Red inalámbrica, Receptor GPS diferencial, Dispositivo de posicionamiento por ultrasonidos y algoritmos de localización
(Gallagher, Wise, Li, Dempster & Chris 2012)	Sistema de posicionamiento en interiores basado en la fusión de sensores para ciegos y deficientes visuales	Artículo	Sensor Wifi, sensor acelerómetro y magnético flujo sensor, Android Studio APIs
(Bermeo, Bravo, Punin,	Sistema de detección de obstáculos para mejorar la movilidad de	Artículo	Sensor de ultrasonido, Arduino Uno, modulador MK433MHz, Aplicación Android

CITACIÓN	TÍTULO	TIPO	TECNOLOGÍAS / PROTOCOLOS / DISPOSITIVOS
Ordoñez, & Huerta, 2018	las personas con discapacidad visual		
(Amemiya T., Yamashita, J., Hirota, K., & Hirose, M. 2012, 27 marzo)	Color plate: bloques líderes virtuales para sordociegos: un buscador de caminos en tiempo real por interfaz híbrida verbal-no verbal y espacio de etiqueta RFID de alta densidad.	Artículo	Direccional de luz infrarroja (IR), modulada por una señal de voz, WiFi, realidad aumentada
Manduchi, R., & Coughlan, J. (2015, 8 noviembre)	Evaluación funcional de un sistema de localización con cámara y teléfono operado por usuarios ciegos y con problemas de visión	Artículo	Algoritmos para reconocimiento automático y localización, RFID, Talking Signs
P., Johnson, V., Strothotte, T., Raab, A., Fritz, S., & Michel, R. (2017)	MOBIC: Designing a Travel Aid for Blind and Elderly People.	Artículo	Receptor GPS, dispositivo detector de obstáculos, brújula y telecomunicaciones móviles, dispositivos de entrada / salida apropiados por voz a través de auriculares sonidos acústicos
(Máster, 2017)	Autonomía de personas ciegas en supermercados mediante realidad aumentada	Trabajo fin de master	Comandos de voz, Auriculares de conducción ósea, Bitcoins, Aplicación móvil, Base de datos, Cámara HD y de profundidad, Micrófono con cancelación de audio.
(Remolina, 2019)	Generación de paisajes háptico-auditivos para asistencia de personas con discapacidad visual mediante el diseño de un lenguaje de representación sensorial	Proyecto de grado	Raspberry pi, GPS, Sensores de ultrasonido, cámaras, software de visión artificial, análisis de geolocalización, motores, bocinas, auriculares

Tabla 13. Análisis tecnologías utilizadas en artículos recuperados

Fuente: Autores

8 METODOLOGÍA

Para el desarrollo del proyecto de Trabajo de Grado, se definieron cuatro fases principales, relacionadas estrechamente con los objetivos específicos del proyecto. A continuación, son presentadas las actividades desarrolladas asociadas con cada una de las etapas.



Ilustración 15. Flujograma metodología

Fuente: Autores

Fase 1: Elaboración de un estado del arte sobre la aplicación de Internet de las Cosas en escenarios de inclusión junto a la identificación de necesidades sobre los requerimientos funcionales y no funcionales desde un enfoque social y económico para la caracterización de experiencias de compras por medio de encuestas y entrevistas.

Las actividades realizadas en esta fase fueron:

1. Búsqueda de documentos en bases de datos académicas. (ver Ilustración 14)
2. Identificación de aplicaciones de IoT en diferentes escenarios de inclusión para el lograr que todas las personas puedan tener las mismas posibilidades y oportunidades.
3. Síntesis del estado del arte con su respectiva documentación de resultados

4. Determinar los requerimientos funciones y no funcionales para tener en cuenta lo que el sistema va a realizar.

Fase 2: Realización del diseño previo de diagramas de lenguaje de modelado unificado (UML) y estructura de hardware para mayor claridad del funcionamiento.

Las actividades realizadas en esta fase fueron:

5. Diseño de diagramas UML para poder visualizar de una forma más estándar el boceto del sistema.
6. Elaboración del diseño de hardware y la conexión del sistema, teniendo en cuenta la implementación del IoT.

Fase 3: Construcción del prototipo funcional, codificación e interfaces del sistema de estantería para una adecuada representación gráfica de los requisitos ayudada de los diagramas UML mencionados anteriormente por medio del cual se evidenciaron las respectivas interfaces, dispositivos y tecnologías.

Las actividades realizadas en esta fase fueron:

7. Codificación
8. Construcción del prototipo funcional del sistema de la estantería.
9. Construcción del prototipo funcional aplicación móvil.

Fase 4: Validación del sistema de estantería para evaluar la capacidad y el funcionamiento del sistema por medio de encuestas que evaluarán el comportamiento. Teniendo en cuenta un manual de uso para sus instrucciones al público.

Las actividades realizadas en esta fase fueron:

10. Elaboración de manual de usos del sistema de estantería.
11. Obtención de evidencias fotográficas de resultados del sistema.

9 DESARROLLO TRABAJO DE GRADO

9.1 DISEÑO

Para el diseño de la estantería se tuvieron en cuenta los resultados de la encuesta la cual se aplicó en una primera instancia a una muestra de 3 personas para validar si era fácil de entender y de aplicar. Seguido de esto, se pasó a realizar la encuesta final a una muestra de 14 personas en la que se tenían en cuenta varios puntos desde las dificultades que pueden encontrar al momento de la compra, como posibles soluciones viables (Ver **ANEXO 2 – Encuesta**).

9.1.1 Requerimientos

En este apartado del documento se observan los requerimientos del sistema para dar cumplimiento a las actividades que necesita el sistema de estantería inteligente para llevar a cabo el proceso de interacción con el usuario por medio de audio. propuesto en esta investigación.

Respecto de los requerimientos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. Los requerimientos no funcionales son aquellos por medio de los cuales no se refiere directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del mismo: rendimiento, seguridad, disponibilidad.

- **Requerimientos funcionales**

En esta parte, se incluyen todos los requisitos funcionales y serán representados cada uno en una tabla en la que se describe cada uno de los campos que la componen:

- **Identificador:** Campo que corresponde al identificador del requisito. Dicho identificador será único e inconfundible, de tal forma que cada uno de los requisitos sea identificable sin posibilidad de error.
 - Requisito funcional: REQ-F-<Número> El número será un valor de dos cifras que empezará desde el valor 01 y se irá incrementado en una unidad.
- **Descripción:** Campo con la descripción del requisito trabajado.
- **Prioridad:** Este campo indica el grado de prioridad con el que debe de ser resuelto el requisito. Los posibles valores son:
 - **Alta:** El diseño e implementación del requisito es de carácter prioritario.
 - **Media:** El requisito se debe diseñar e implementar con prioridad media.
 - **Baja:** El requisito se debe diseñar e implementar con prioridad baja
- **Precondición:** Campo que corresponde a las condiciones iniciales que se han de cumplir para realizar la actividad
- **Postcondición:** Campo que corresponde al estado del sistema tras realizar el requisito

Tabla 14. Requerimiento funcional 01
Fuente: Autores

REQ-F-01	
Requerimiento - Descripción	Obtener características del producto por medio de la Raspberry: El sistema enviará un mensaje a la aplicación móvil con respecto al producto del cual se requiere las especificaciones mediante el audio
Prioridad	Alta
Precondición	Tener ingresadas las características de los productos disponibles en la estantería para la venta
Postcondición	La aplicación obtendrá la información de las características del producto y así será transmitida al usuario

Tabla 15. Requerimiento funcional 02
Fuente: Autores

REQ-F-02

Requerimiento - Descripción	Enviar audio: El sistema enviará un mensaje por medio de audio al usuario cuando se evidencie que ha tocado alguno de los productos de la estantería y así lo reconocerá
Prioridad	Alta
Precondición	Tener instalada la aplicación móvil
Postcondición	El usuario sabrá las características del producto elegido

Tabla 16. Requerimiento funcional 03
Fuente: Autores

REQ-F-03

Requerimiento - Descripción	Enviar características del producto: Al obtenerse la señal de un producto al ser tocado se enviarán las características por medio del Firebase y así mismo este hará que se envíe el audio por medio de la aplicación móvil para reconocer el producto ubicado en la estantería
Prioridad	Alta
Precondición	Tener la información de que un producto ha sido tocado por el usuario Tener la aplicación móvil activa Conexión a internet
Postcondición	Envío de las características a través del audio

Tabla 17. Requerimiento funcional 04
Fuente: Autores

REQ-F-04

Requerimiento - Descripción	Facilidad al realizar la compra: La persona al momento de realizar podrá encontrar en un mismo estante los productos básicos de su día a día.
Prioridad	Alta
Precondición	Tener la aplicación móvil activa Conexión a internet
Postcondición	Conocer las características del producto a través del audio

Tabla 18. Requerimiento funcional 05
Fuente: Autores

REQ-F-05

Requerimiento - Descripción	Evitar inconvenientes de movilidad: Esto quiere decir que se desea dar soluciones para que las personas con discapacidad visual puedan realizar su compra sin tener problemas debido al espacio físico del establecimiento
Prioridad	Alta
Precondición	Tener la aplicación móvil activa Conexión a internet
Postcondición	Encontrar los productos en un mismo lugar sin necesidad de tener que movilizarse por todo el establecimiento

- **Requerimientos no funcionales**

En esta parte, se incluyen todos los requisitos funcionales y serán representados cada uno en una tabla en la que se describe cada uno de los campos que la componen:

- **Identificador:** Campo que corresponde al identificador del requisito. Dicho identificador será único e inconfundible, de tal forma que cada uno de los requisitos sea identificable sin posibilidad de error.
 - **Requerimiento no funcional:** REQ-NF-<Número> El número será un valor de dos cifras que empezará desde el valor 01 y se irá incrementado en una unidad.
- **Descripción:** Campo con la descripción del requisito trabajado.
- **Prioridad:** Este campo indica el grado de prioridad con el que debe de ser resuelto el requisito. Los posibles valores son:
 - **Alta:** El diseño e implementación del requisito es de carácter prioritario.
 - **Media:** El requisito se debe diseñar e implementar con prioridad media.
 - **Baja:** El requisito se debe diseñar e implementar con prioridad baja

- **Precondición:** Campo que corresponde a las condiciones iniciales que se han de cumplir para realizar la actividad
- **Postcondición:** Campo que corresponde al estado del sistema tras realizar el requisito.

Tabla 19. Requerimiento no funcional 01
Fuente: Autores

REQ-NF-01

Requerimiento	Usabilidad: El sistema debe tener una interfaz que sea intuitiva y que garantice la comprensión total de los datos que están en el sistema para facilidad del usuario
Prioridad	Alta

Tabla 20. Requerimiento no funcional 02
Fuente: Autores

REQ-NF-02

Requerimiento	Disponibilidad: La aplicación debe estar disponible el 100% del tiempo siempre y cuando llegue al establecimiento
Prioridad	Alta

Tabla 21. Requerimiento no funcional 03
Fuente: Autores

REQ-NF-03

Requerimiento	Fiabilidad: La aplicación debe evitar que se cierre inesperadamente siempre que sea posible.
Prioridad	Alta

Tabla 22. Requerimiento no funcional 04
Fuente: Autores

REQ-NF-04

Requerimiento	Portabilidad: La aplicación ha de poder funcionar correctamente en diferentes dispositivos Android desde la versión 5.0 y sus interfaces han de adaptarse a diferentes tamaños de la pantalla.
Prioridad	Alta

Tabla 23. Requerimiento no funcional 05
 Fuente: Autores

REQ-NF-05

Requerimiento	La aplicación será usada por cualquier usuario que la instale sin necesidad de iniciar sesión.
Prioridad	Alta

9.1.2 Diseños UML

Los diagramas UML tienen como finalidad presentar diversas perspectivas del sistema que se está trabajando, a las cuales se les conoce como modelo. Estos diagramas describen y permiten analizar lo que se supone hace el sistema.

9.1.2.1 Casos de uso y descripción

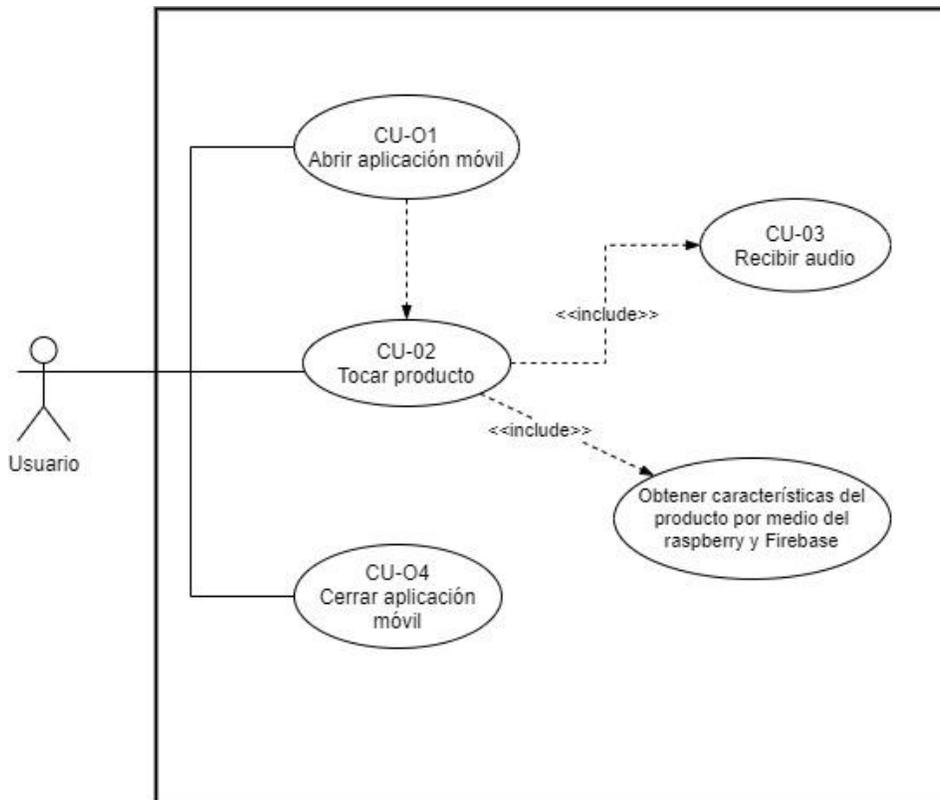


Ilustración 16. Diagrama casos de uso
 Fuente: Autores

9.1.2.1.1 Descripción textual de los casos de uso

Para la descripción de los casos de uso identificados en el diagrama anterior se va a utilizar la siguiente tabla:

Tabla 24. Plantilla descripción casos de uso
Fuente: Autores

Identificador	
Caso de uso	
Actores	
Objetivo	
Precondiciones	
Escenario	
Postcondiciones	
Excepción	

- **Identificador:** Nombre identificativo del caso de uso. Se utilizará la nomenclatura CU-<Número>. El número será un valor de dos cifras que empezará desde el valor 01 y se irá incrementado en una unidad.
- **Caso de uso:** Descripción breve del caso de uso.
- **Actores:** Agentes que interactúan con el caso de uso.
- **Objetivo:** Descripción detallada del caso de uso.
- **Precondiciones:** Condiciones iniciales que se han de cumplir para realizar el caso de uso.
- **Escenario:** Conjunto de pasos que se han de dar para realizar el caso de uso.
- **Postcondiciones:** Estado del sistema tras realizar el caso de uso.
- **Excepción:** Incidencias o alternativas que pueden aparecer al ejecutar el caso de uso. La numeración indicada en cada excepción hace referencia al paso del escenario que ha provocado la incidencia. Para los casos en los que más de un paso provoca la misma excepción se ha asignado valor 0 para la numeración y se ha indicado entre paréntesis los pasos afectados.

Tabla 25. Descripción CU-01
Fuente: Autores

CU - 01

Caso de uso	Abrir aplicación móvil
Actores	Usuario
Objetivo	Que la aplicación móvil quede en modo activo
Precondiciones	Contar con internet en datos móviles
Escenario	<ol style="list-style-type: none"> 1. El cliente debe contar con teléfono inteligente 2. El cliente debe contar con servicio de internet 3. El cliente debe ubicar la aplicación
Postcondiciones	El servicio empieza a funcionar
Excepción	En caso de que la aplicación ya esté iniciada, no sucederá nada

Tabla 26. Descripción CU-02
Fuente: Autores

CU - 02

Caso de uso	Tocar producto
Actores	Usuario
Objetivo	Conocer las características del producto mediante un mensaje de audio
Precondiciones	Contar con servicio de internet CU-01
Escenario	<ol style="list-style-type: none"> 1. El usuario debe contar con teléfono inteligente 2. El usuario debe contar con servicio de internet 3. El usuario debe ubicar la aplicación 4. El usuario debe abrir la aplicación móvil 5. Si el usuario ya abrió la aplicación procede a tocar un producto
Postcondiciones	Conocer las características del producto por medio del audio
Excepción	Que el usuario no toque el producto Que la aplicación este cerrada

Tabla 27. Descripción CU-03
Fuente: Autores

CU - 03

Caso de uso	Recibir audio
Actores	Usuario
Objetivo	Conocer por medio de audio las características del producto elegido
Precondiciones	CU-01 CU-02
Escenario	<ol style="list-style-type: none"> 1. El usuario debe contar con teléfono inteligente 2. El usuario debe contar con servicio de internet 3. El usuario debe ubicar la aplicación

	<ol style="list-style-type: none"> 4. El usuario debe abrir la aplicación móvil 5. Si el usuario ya abrió la aplicación procede a tocar un producto 6. El usuario debe escuchar el audio
Postcondiciones	El usuario tendrá conocimiento de la información correspondiente al producto con el cual hizo contacto por medio del envío de una audio
Excepción	Que el usuario no toque el producto Que el usuario no cuente con servicio de internet

Tabla 28. Descripción CU-04
Fuente: Autores

CU - 04

Caso de uso	Cerrar la aplicación
Actores	Usuario
Objetivo	La aplicación queda en modo apagado
Precondiciones	CU-01 CU-02 CU-03
Escenario	<ol style="list-style-type: none"> 1. El usuario debe contar con teléfono inteligente 2. El usuario debe contar con servicio de internet 3. El usuario debe ubicar la aplicación 4. El usuario debe abrir la aplicación móvil 5. Si el usuario ya abrió la aplicación procede a tocar un producto 6. El usuario debe escuchar el audio
Postcondiciones	El usuario tendrá conocimiento de la información correspondiente al producto con el cual hizo contacto
Excepción	

9.1.2.2 Diagrama de secuencia

Conforme a la caracterización de las funcionalidades y los casos de uso descritos anteriormente, para el diseño de la aplicación se plantea el diagrama de secuencia para definir la relación del usuario y las funcionalidades dentro de la aplicación.

Para el diseño de la aplicación se planteó el diagrama de secuencia que describe básicamente cómo los objetos intercambian mensajes en un orden determinado, definiendo la relación del usuario y las funcionalidades dentro de la aplicación.

DIAGRAMA DE SECUENCIA

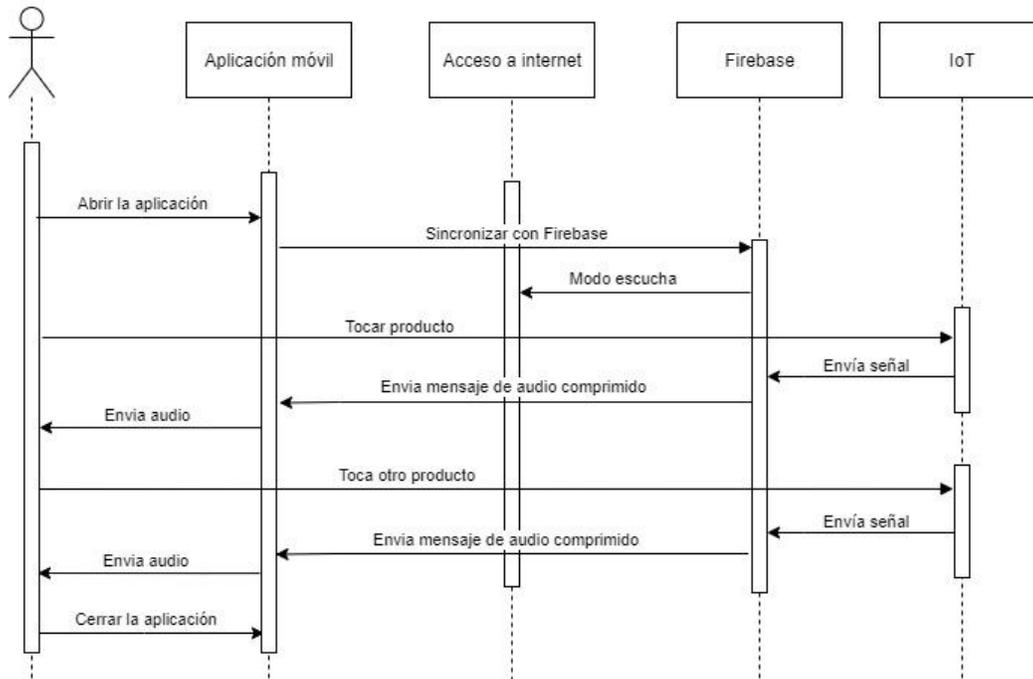


Ilustración 17. Diagrama de secuencia
Fuente: Autores

9.1.2.3 Diagrama de actividades

Para el diseño de las funcionalidades de la estantería junto a la aplicación se generó un diagrama de actividades para definir la interacción de la aplicación al ejecutar los procesos del envío de los audios de cada producto.

DIAGRAMA DE ACTIVIDADES

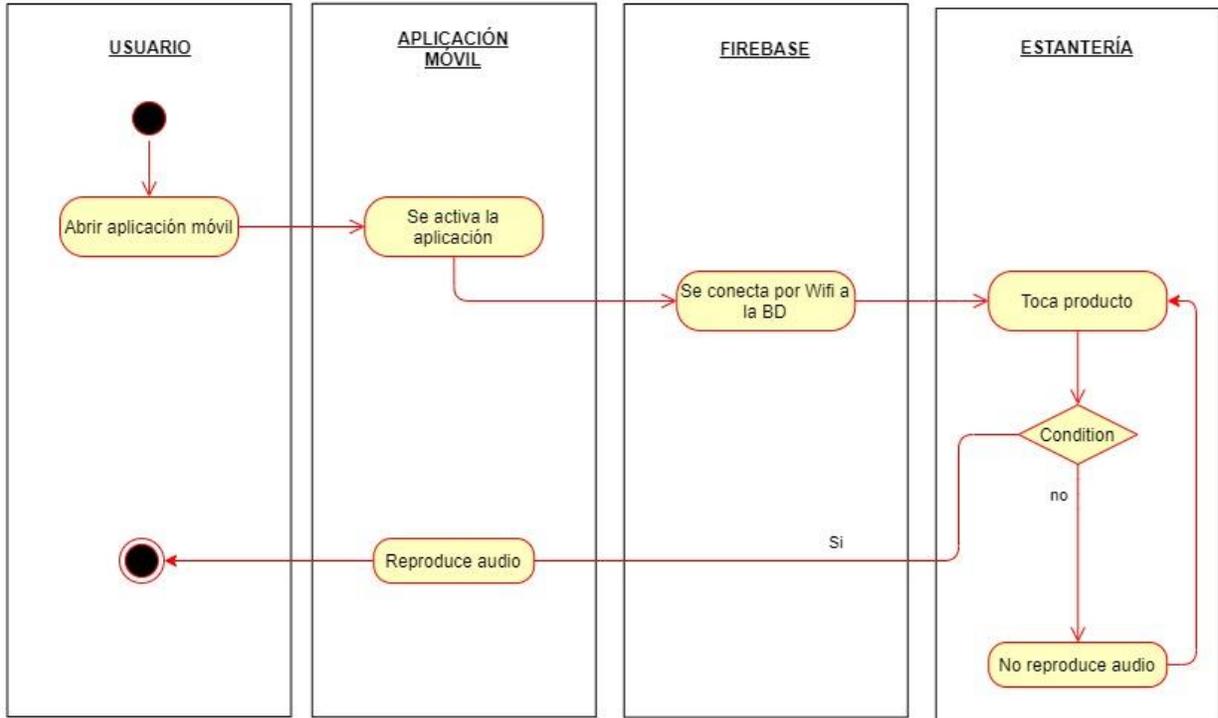


Ilustración 18. Diagrama de actividades
Fuente: Autores

9.1.3 Modelado

9.1.3.1 Diseño de Layouts

Los layouts definen la manera en cómo la aplicación va a mostrar la interfaz gráfica principal y como se visualizaría en la aplicación. La aplicación cuenta únicamente con una vista principal, donde se encuentra una imagen debido a que esta se hizo para funcionar como un interceptor entre la estantería y el usuario para hacerle llegar el audio con las características del producto.

En esta parte de diseño se define el estilo y la forma con la que se presenta la aplicación, en donde se tiene en cuenta los factores como colores, temas e imágenes, pensando en la experiencia de usuario. En la siguiente imagen se muestra un mockup de la pantalla principal del diseño de la aplicación.

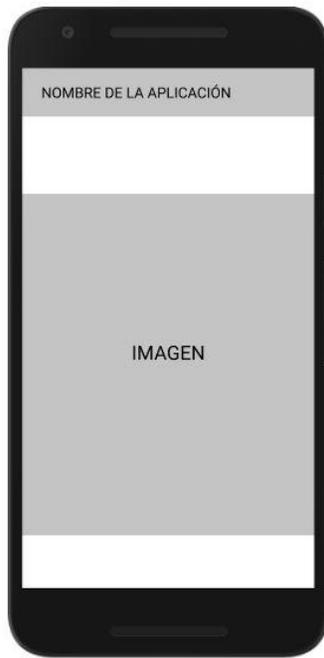


Ilustración 19. Mockup aplicación móvil
Fuente: Autores

A continuación, se muestra la interfaz de la aplicación móvil realizada para la interacción de las personas con discapacidad visual y el estante.



Ilustración 20. Interfaz aplicación móvil – Pantalla principal
Fuente: Autores

9.1.3.2 Modelado de las conexiones

A continuación, se muestra la arquitectura del sistema, aquí se presenta el operativo Raspbian que está altamente optimizado para la Raspberry, conectándose remotamente con el software VNC que permite observar las acciones del ordenador servidor remotamente a través de un ordenador cliente.

Las flechas bidireccionales de la Raspberry pi 3 con Firebase muestran el envío de la información, de igual manera con la aplicación Android le envía una notificación a Firebase.

Al tocar el producto que está conectado por el cable de pinza de caimán al sensor MPR121 que está soldado a la Raspberry se va a cambiar un estado en la base de datos de Firebase en tiempo real, la aplicación está recibiendo la orden para escuchar el audio.

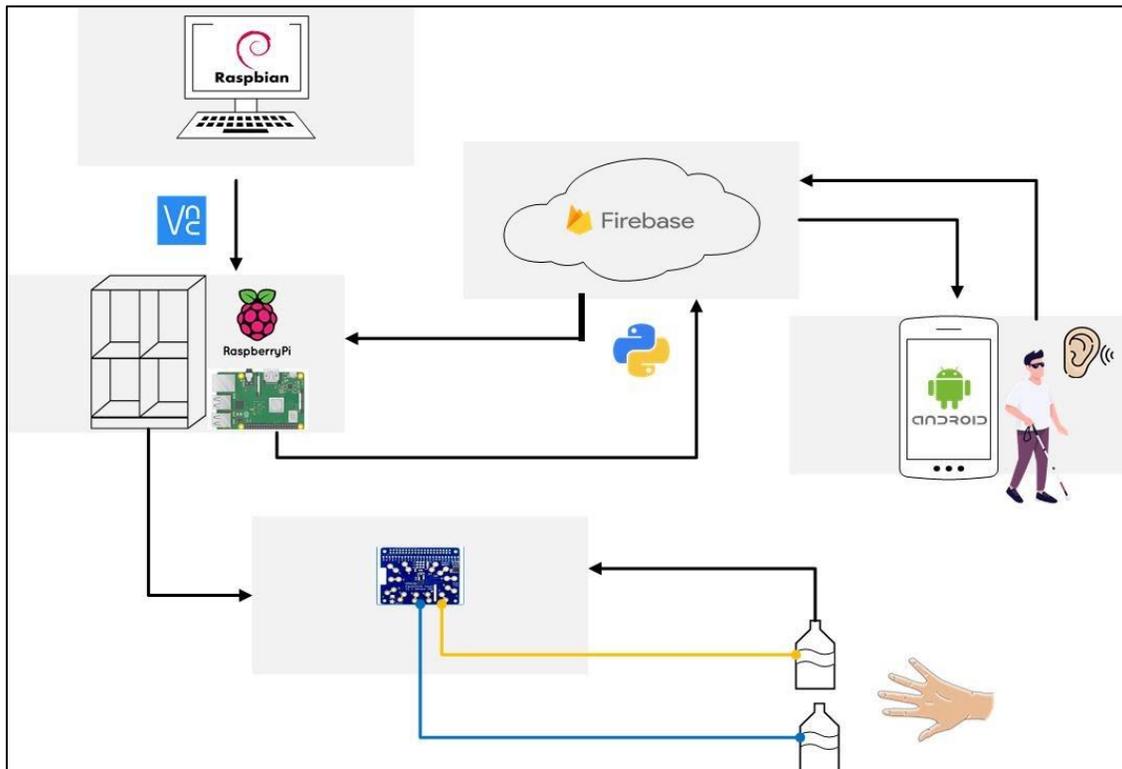


Ilustración 21. Arquitectura conexiones Firebase, Android Studio y Raspberry

Fuente: Autores

9.1.3.3 Modelado de hardware

Estante: Un estante es un instrumento o herramienta que se utiliza para ordenar objetos obteniendo una mejor localización y ubicación a la hora de su utilización. El almacenamiento en estas estanterías consiste en situar los distintos tipos y formas en que se acomodan, registra o se carga el material en la estantería (Gleote, 2014).

Para el diseño a nivel de hardware, se presenta un diseño de la arquitectura teniendo en cuenta aspectos como la conexión de los diferentes dispositivos y módulos que componen la solución, tales como una Raspberry Pi, un sensor MPR121 y cables pinza de cocodrilo.

Teniendo en cuenta esto, en la Ilustración 21 se puede observar la arquitectura del sistema con sus conexiones que permite ver su interacción con cada uno de los dispositivos y como puede ser fácilmente adaptado en diferentes establecimientos.

9.1.3.3.1 Raspberry Pi

Para la elección de la Raspberry Pi se debe a que facilita la conexión ya que permite conectar con un servidor y así comunicarse entre varios dispositivos, en este caso se utilizó el VNC permitiendo por medio de la IP local de la Raspberry conectar al computador portátil.

Por otro lado, para dar alimentación a la Raspberry Pi, se utiliza la batería del portátil con un cable Micro USB que nos permite comunica y conectar para así dar los voltios necesarios para su funcionamiento. En comparación con Arduino la Raspberry tiene cuatro puertos USB para conectar varios dispositivos de comunicación.



Ilustración 22. Raspberry Pi 3
Fuente: Autores

9.1.3.3.2 Sensor MPR121

Es un sensor que permite crear interfaces táctiles y por medio del cual se puede controlar hasta 12 pines. En este caso, se programan únicamente los primeros cuatro pines siendo del 0 al 3 por lo que se realiza un prototipo pequeño. Para su contacto puede ser cualquier objeto conductor como papel metálico, cobre, metal y de esta manera poder activar el pin, de este modo, automáticamente los umbrales del sensor examinan si es Touch o no y a cada pin se le asigna un umbral de activación.

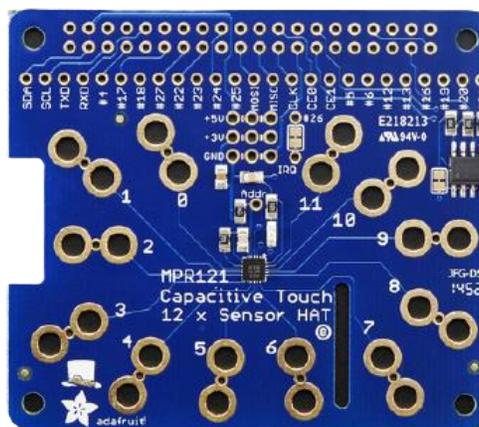


Ilustración 23. Sensor MPR121
Fuente: Autores

9.1.3.3.3 Cables pinza de cocodrilo

Los cables pinza de cocodrilo permiten hacer la conexión táctil, puenteando desde el sensor hasta el producto. Este tipo de cable permite que se pueda transmitir la información fácilmente ya que se asegura con la pinza al sensor sin necesidad de soldar y de esta manera quedando estable la conexión por parte y parte.

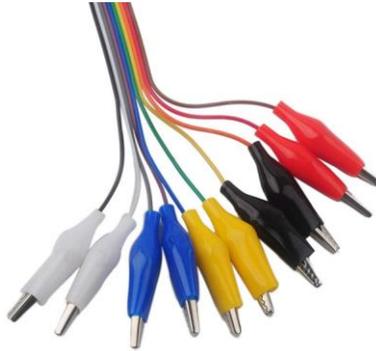


Ilustración 24. Cables pinzas de cocodrilo
Fuente: Autores

9.1.3.4 Diseño físico del prototipo

Para llevar a cabo la realización de la estantería se hizo un diseño a una escala de 1:2 del prototipo; este diseño se compone de: Raspberry Pi, Sensor MPR121, cables pinza de cocodrilo y una estructura en madera con cuadro compartimientos para cada producto.

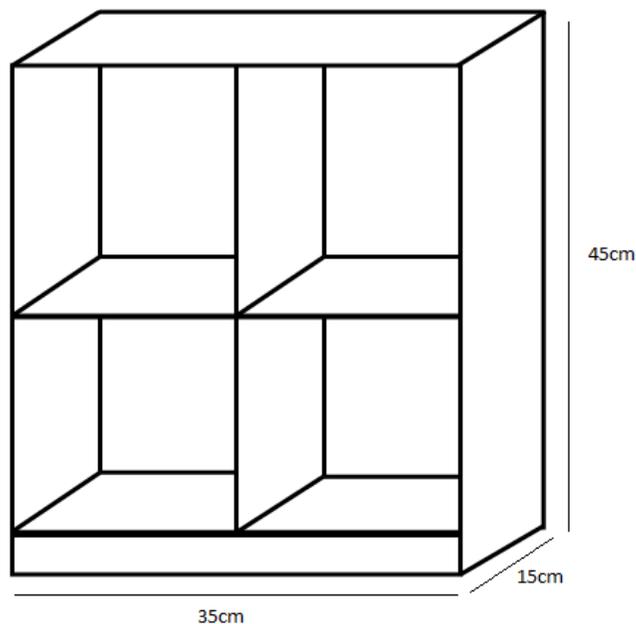


Ilustración 25. Diseño prototipo estantería
Fuente: Autores

Este diseño es la base para la elaboración del prototipo el cual permite realizar las pruebas necesarias con las medidas correspondientes en la vida real.

9.2 CONSTRUCCIÓN

En este apartado, se habla acerca de la construcción del prototipo funcional a partir del desarrollo de cada uno de los módulos que lo componen, dando a conocer la configuración de cada una de las tecnologías utilizadas y su integración.

9.2.1 Elección de los productos

La selección de los productos para la creación del proyecto de la estantería se tuvieron en cuenta de acuerdo a un artículo del Tiempo, en el que informó Mario Arregui, *Country Manager* de Kantar las seis marcas nacionales que se encuentran con mayor penetración en los hogares colombianos (Tiempo, 2020).

De manera que, en la siguiente tabla se da a conocer cada uno de ellos con su porcentaje correspondiente según el estudio realizado.

Tabla 29. Marcas productos
 Fuente: Adaptado de (Tiempo, 2020)

MARCA	PORCENTAJE DE USO
Colgate	96,6 %
Coca Cola	86,2 %
Arroz Diana	86,1 %
Familia	84,8 %
Fruco	79,7 %
Noel	78,6 %
Alquería	78,6 %
Bimbo	78,7 %
Maggi	77,2 %
Doria	76,6 %

Siendo así el porcentaje de las marcas, los cuatro productos elegidos para el proyecto se seleccionaron de las marcas más consumidas en Colombia, tal como se muestra en la imagen.

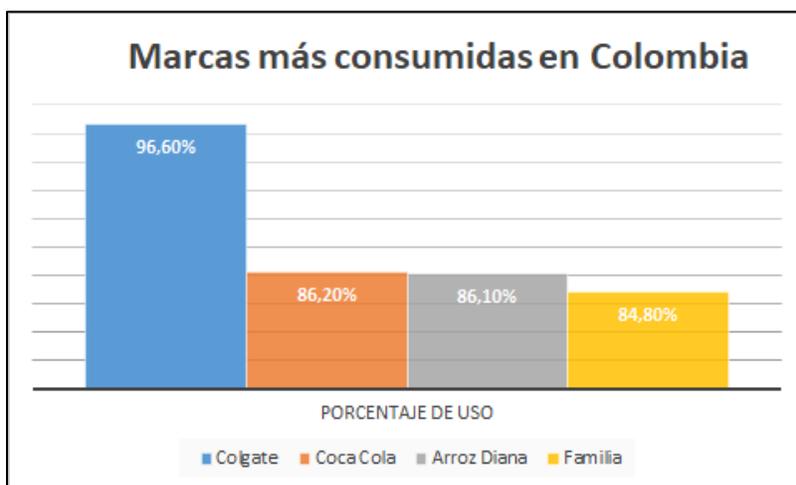


Ilustración 26. Marcas más consumidas en Colombia
 Fuente: Autores

Se puede señalar, que teniendo en cuenta los productos que maneja cada una de estas marcas, se diseñaron cuatro prototipos de los artículos con diferentes marcas pensadas por los autores. A continuación, se muestra una tabla de los prototipos de

los productos junto a la descripción que tiene cada uno de ellos, descripción que se reproduce en la aplicación móvil por medio del audio.

PRODUCTO	DESCRIPCIÓN
	<p>Producto: Crema de dientes Marca: Sonríe Cantidad: 60 ml cm³ Precio sugerido al público: 2600 pesos</p>
	<p>Producto: Gaseosa Marca: Konga Cantidad: 600 ml Precio sugerido al público: 1500 pesos</p>
	<p>Producto: Arroz Marca: Sol Cantidad: 1 kg Precio sugerido al público: 4000 pesos</p>

	<p>Producto: Papel higiénico</p> <p>Marca: Azul</p> <p>Cantidad: Rollo de 50mt</p> <p>Precio sugerido al público: 2900 pesos</p>
---	--

Tabla 30. Productos del prototipo
Fuente: Autores

9.2.1.1 Configuración de la grabación de los audios

Para la elaboración de los audios que se utilizaron para la estantería se usó la página “*texttospeechrobot*”, en la cual se eligió la voz “*Latin American Spanish Sofia V3 (female)*”, una vez seleccionada la voz se escribió el texto de la información correspondiente de cada producto y hecho esto descargarlo en formato mp3.

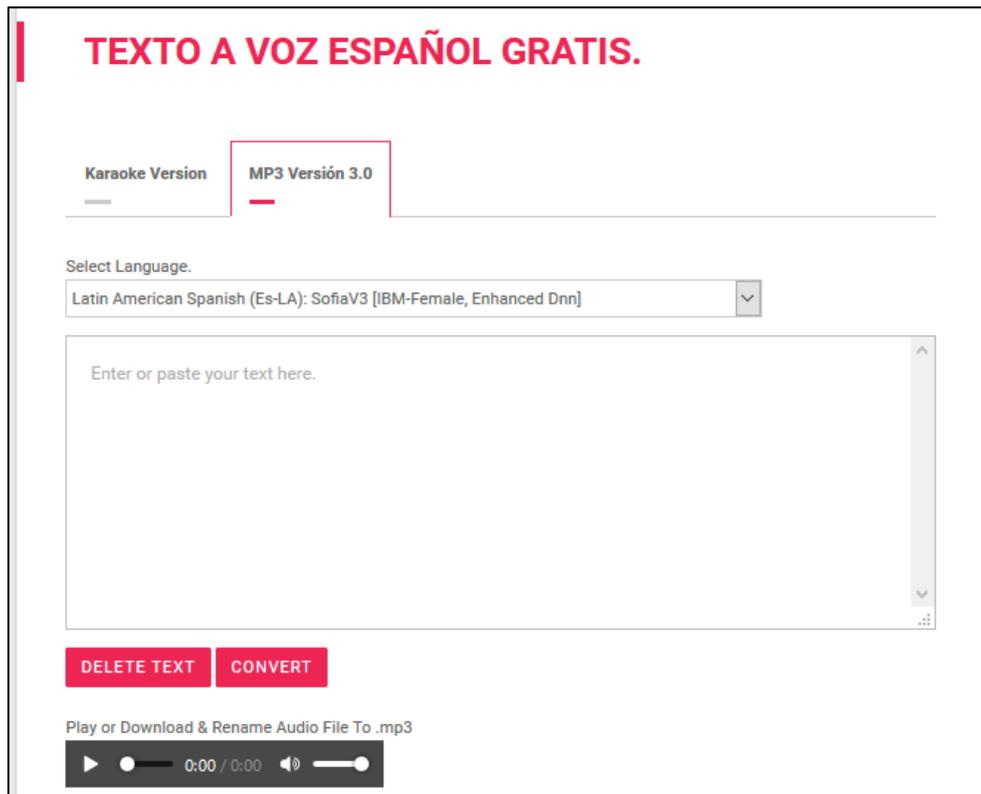


Ilustración 27. Programa grabación de audios

Fuente: Autores

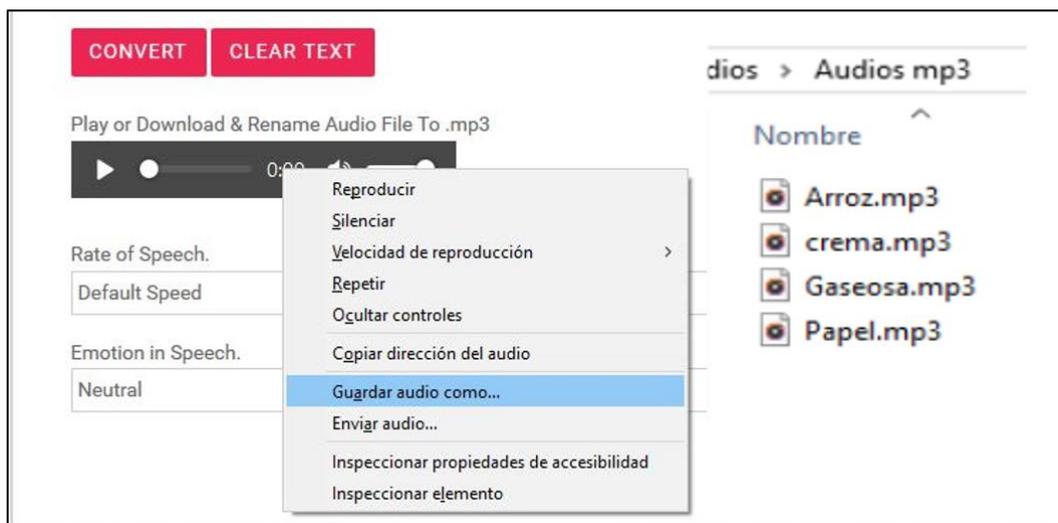


Ilustración 28. Descarga de audio
Fuente: Autores

Luego de esto, para poder reproducir los audios en la Raspberry se tuvo que convertir los audios en formato .wav, para que fueran aceptados, ya descargados se ponen a disposición para la elaboración del proyecto e integrarlos.



Ilustración 29. Conversión audio .mp3 a .wav
Fuente: Autores

9.2.2 Configuración Raspberry Pi 3 y Sensor MPR121

Para iniciar la instalación del sistema operativo se debe soldar los pines para hacer una conexión permanente.

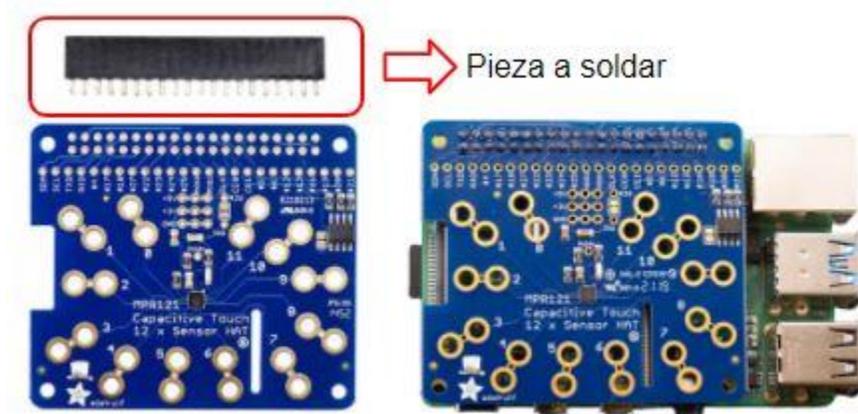


Ilustración 30. Raspberry - Sensor MPR121
Fuente: Autores

La instalación de la imagen de la Raspberry Pi que se utilizó en el proyecto fue el recomendado y el tamaño mínimo de la tarjeta es de 16 GB de clase 4 (velocidad mínima de transferencia de datos) (Raspberrypi, 2019).



Ilustración 31. Tarjeta microSD a utilizar
Fuente: Autores

Para comenzar, en primer lugar, se formateó la tarjeta micro SD en formato Fat32, y para ello se usó el programa *SD Card Formatter*. Como se muestra en la siguiente imagen se elige E:/ que es la Micro SD que se utilizó y ya seleccionada se procede a formatear, una vez realizado el procedimiento aparece un mensaje que muestra la descripción, lo que quiere decir que ya está lista para descargar el sistema operativo.

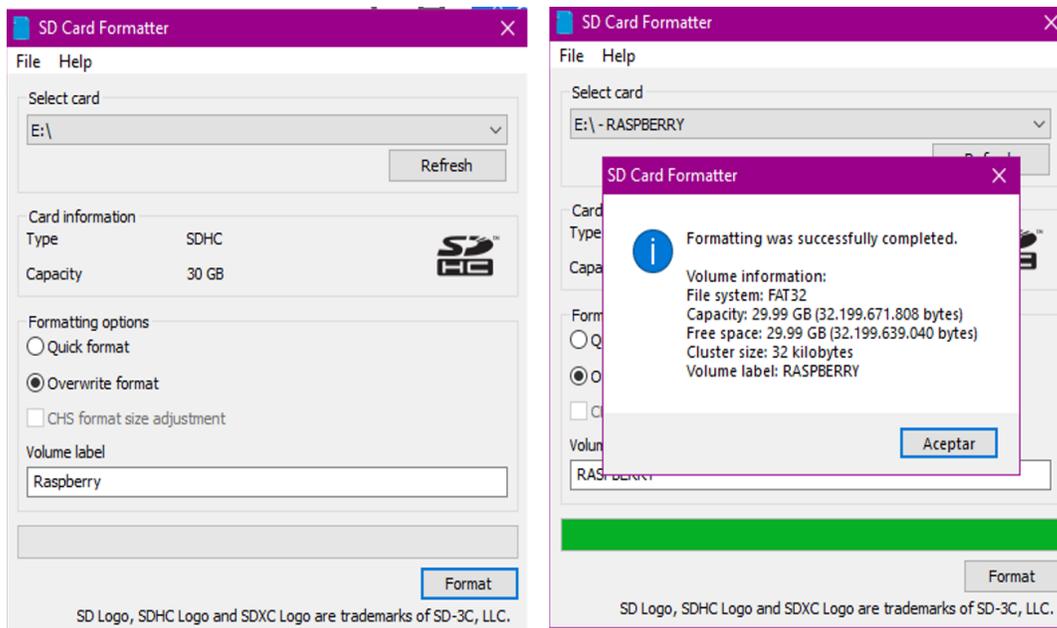


Ilustración 32. Formateo de la microSD
Fuente: Autores

Seguido de esto, en la página oficial de la Raspberry se descarga el sistema operativo recomendado, Raspberry pi os con escritorio y software.

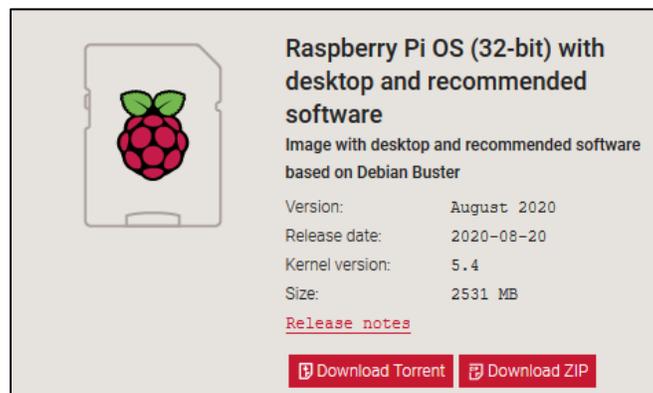


Ilustración 33. Descarga del S.O Raspbian
Fuente: Tomado de Página Oficial Raspberry

Ya instalado se hace la conexión a una pantalla, teclado y mouse para poder ejecutar y hacer la conexión remota. Para facilitar su manejo se trabajó con conexión remota VNC (*Virtual Network Computing*).

9.2.3 Configuración conexión remota VNC

El VNC (*Virtual Network Computing*) es un software que da soporte y acceso remoto sencillo, seguro y listo para el uso para los equipos y dispositivos móviles (RealVNC, 2020). Así mismo, se utiliza para facilitar el manejo y así evitar tener que estar conectado a un teclado, una pantalla y un mouse por aparte.

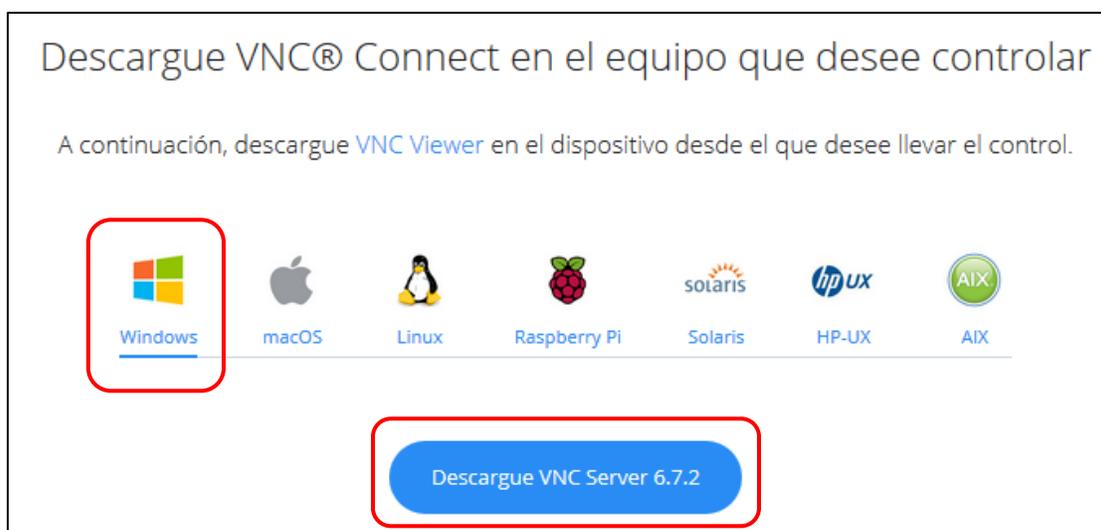


Ilustración 34. Descarga del VNC
Fuente: Autores

Para la conexión remota se tiene en cuenta la IP local, y se hace una activación del VNC (*Virtual Network Computing*).

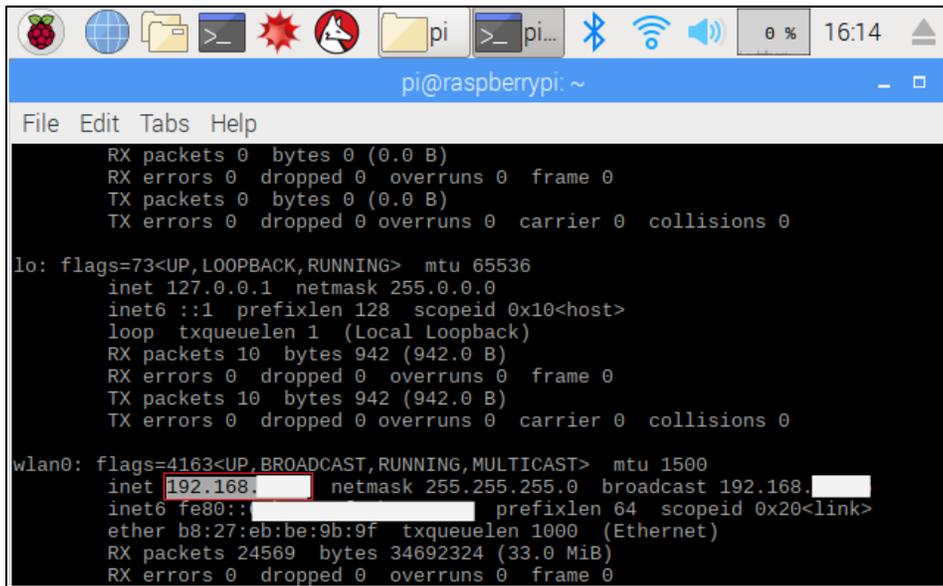


Ilustración 35. Terminal IP
Fuente: Autores

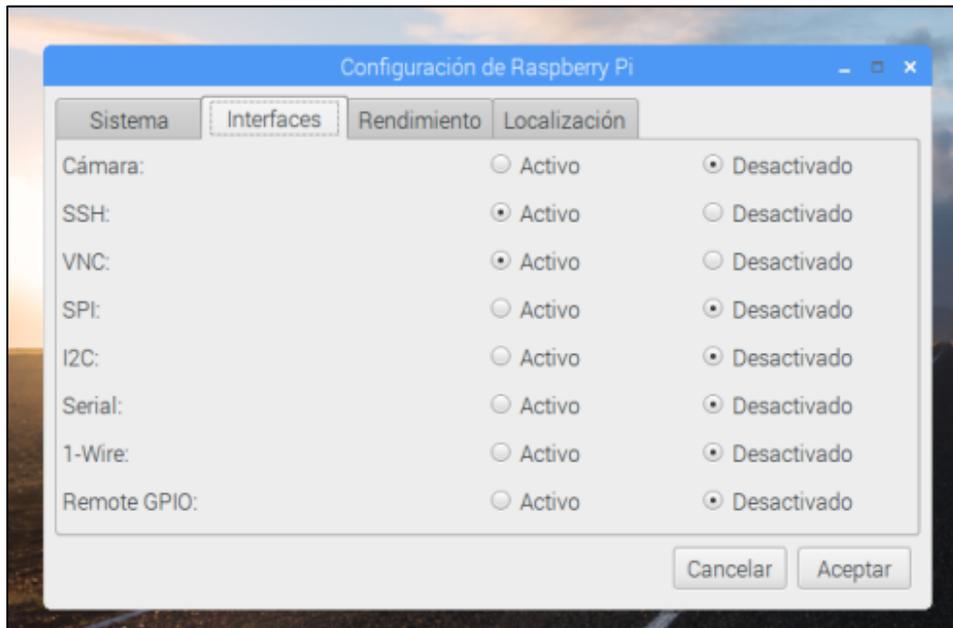


Ilustración 36. Activación VNC
Fuente: Autores

Una vez activado el VNC se procede a crear una nueva conexión en él para poder trabajar sobre ella, tal como se muestra en la imagen.

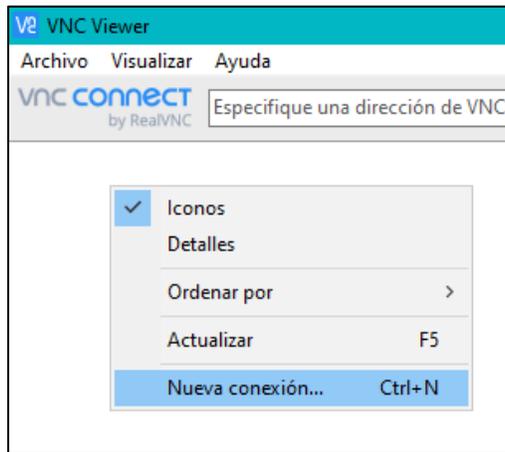


Ilustración 37. Nueva conexión VNC
Fuente: Autores

Luego de crear la conexión VNC se ponen en propiedades la IP de la Raspberry pi en el campo VNC Server, se aceptan cambios y una vez hecho esto se procede a dar clic derecho y conectarse para crear la conexión VNC.

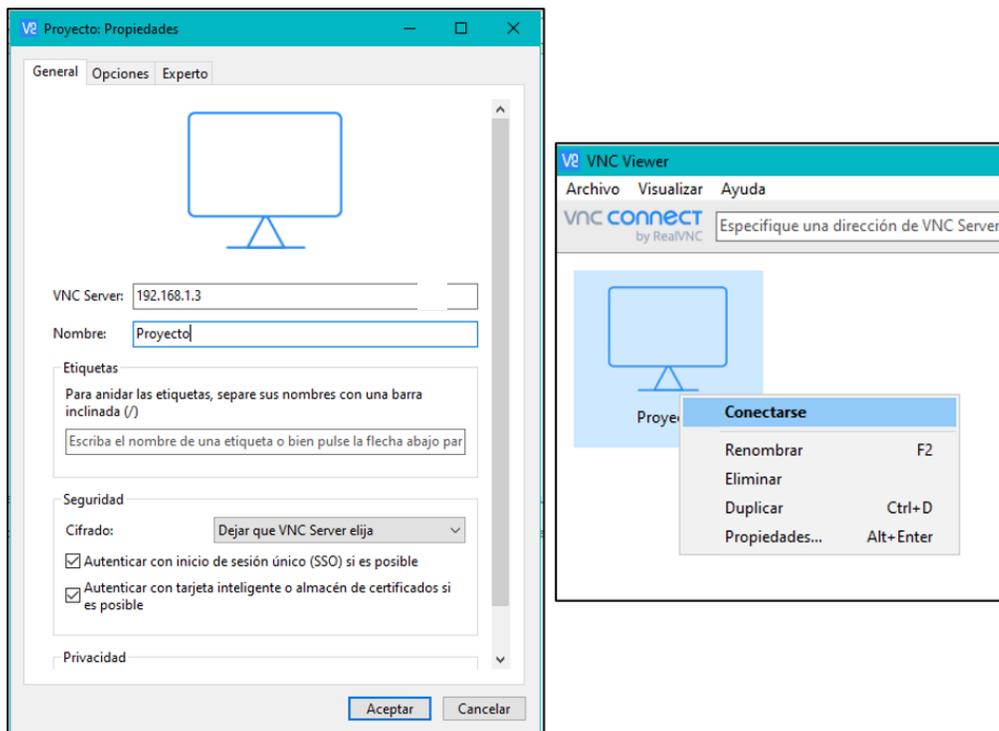


Ilustración 38. Configuración de la IP y conexión
Fuente: Autores

Ya creada la conexión se continúa ingresando el usuario y la contraseña por defecto. El usuario y la contraseña del Raspberry permite proteger diferentes amenazas que pueden llegar a través de Internet y asegurar de que solo se puede conectar a él. Cuando se graba por primera vez a un USB por defecto viene con un usuario y una contraseña configurados y necesarios para poder empezar a utilizarla.

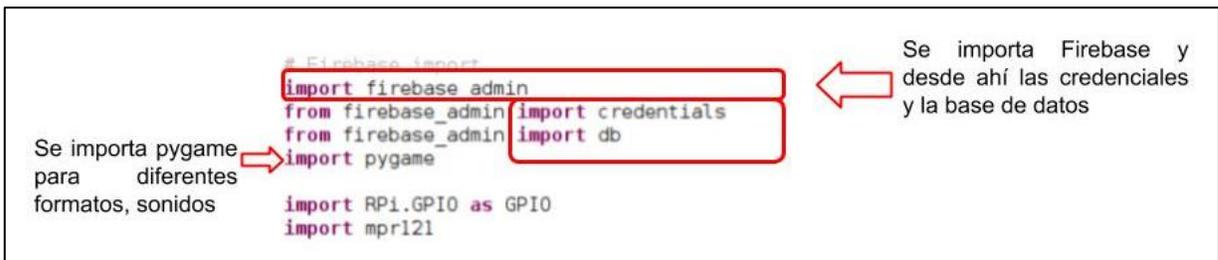
- **Usuario:** pi
- **Contraseña:** raspberry

Para evitar que pongan en peligro nuestro Raspberry Pi se debe cambiar la contraseña, el siguiente paso será ejecutar el comando “**sudo passwd**”, comando para cambiar la contraseña del usuario, este comando pedirá la contraseña actual (defecto) y, después, se introduce la nueva contraseña que se quiere dar al Raspberry Pi dos veces.

9.2.4 Código funcionamiento de Raspberry

Para la configuración de todo lo que tiene que ver con la Raspberry Pi se importan todos los módulos de *pygame* disponibles en el paquete de *pygame*.

También se importa la librería de los pines GPIO, es decir, con esta librería podemos utilizar las funciones implementadas para controlar los pines de nuestra Raspberry Pi. Finalmente, se crea e inicializa una instancia de la clase MPR121.



```
# Firebase import
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import pygame
import RPi.GPIO as GPIO
import mpr121
```

Se importa pygame para diferentes formatos, sonidos →

← Se importa Firebase y desde ahí las credenciales y la base de datos

Ilustración 39. Importar librería pines GPIO y módulos pygame
Fuente: Autores

Una vez definido el comportamiento de un pin en concreto, se procede a establecerlo con un estado.

```
# Use GPIO Interrupt Pin
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.IN)
```

} Pines GPIO

Ilustración 40. Restablecer pin con un estado
Fuente: Autores

En esta parte, se realiza conexión a la base de datos de Firebase, llamando las credenciales y otorgando los privilegios de administrador. De igual manera, se establecen los umbrales de toque y liberación para cada electrodo, es decir establece parte inicial y final. Finalmente, se preestablecen los argumentos de inicialización del mezclador y se inicializa el módulo mezclador.

```
# config firebase DB
# Fetch the service account key JSON file contents
cred = credentials.Certificate('/home/pi/Desktop/proyecto/cred.json')

# Initialize the app with a service account, granting admin privileges
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://raspberrypi-72530.firebaseio.com/'
})

# Ref Firebase
ref = db.reference('sonidos')

# Use mpr121 class for everything else
mpr121.TOU_THRESH = 0x30
mpr121.REL_THRESH = 0x33
mpr121.setup(0x5a)

# User pygame for sounds
pygame.mixer.pre_init(44100, -16, 12, 512)
pygame.init()
```

Se llama las credenciales descargadas del proyecto Firebase que está en la carpeta de la raspberry y se otorga privilegios de administrador

} Se llama a la Base de Datos con nombre sonidos

} Se usa la clase MPR121 para lo demás

} Usuario pygame para los sonidos

Ilustración 41. Conexión a Firebase
Fuente: Autores

En la Ilustración 42, se encuentre el bucle principal el cual es utilizado para imprimir un mensaje cada vez que se toca un pin. Se verifica el último y actual estado de cada pin para ver si fue presionado o liberado, y verifica si cambió de no tocado a tocado, en este caso en Firebase cambia de estado *False* a *True*.

```

touches = [0,0,0,0,0,0]

while True:
    if (GPIO.input(7)): # Interupt pin is high
        pass
    else: # Interupt pin is low
        touchData = mpr121.readData(0x5a)
        for i in range(6):
            if (touchData & (1<<i)):

if (touches[i] == 0):
    print( 'Pin ' + str(i) + ' was just touched')
    if (i == 0):
        print('gaseosa')
        refchild = ref.child('gaseosa')
        refchild.set(True)
    elif (i == 1):
        print('crema')
        refchild = ref.child('crema')
        refchild.set(True)
    elif (i == 2):
        print('papel')
        refchild = ref.child('papel')
        refchild.set(True)
    elif (i == 4):
        print('arroz')
        refchild = ref.child('arroz')
        refchild.set(True)

```

While para cuando se toque el sensor pase a estado verdadero

Si el sensor número 1 se toca se reproducirá el sonido gaseosa y así mismo con cada uno de los sensores

Ilustración 42. Bucle de imprimir sonidos
Fuente: Autores

En esta parte se muestra una librería de Python, librería que permite el acceso SMBus a través de la interfaz I2C / dev en hosts Linux. El núcleo del host debe contar con soporte I2C, soporte de interfaz de dispositivo I2C y un controlador de adaptador de bus. La biblioteca smbus proporciona acceso desde Python a la funcionalidad del driver I2C de Linux.

```

import smbus
bus = smbus.SMBus(1)

```

Ilustración 43. Librería smbus
Fuente: Autores

Así mismo, se registra de cada pin del sensor MPR121 un dispositivo compatible con circuitos integrados (I2C) con un pin de interrupción IRQ. Este pin se activa cada vez que se detecta un toque o una liberación. El dispositivo tiene una dirección

I2C (protocolo síncrono) configurable conectando el pin ADDR a las líneas VSS, VDD, SDA o SCL. Esto da como resultado direcciones I2C de 0x5A, 0x5B, 0x5C y 0x5D.

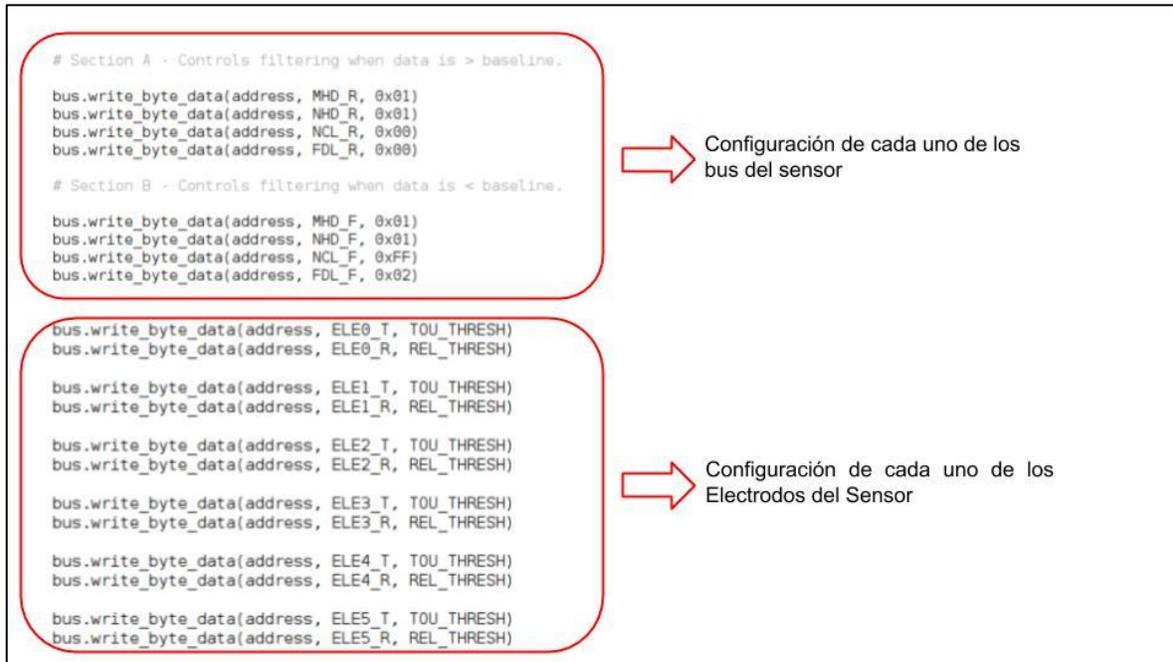


Ilustración 44. Configuración del sensor
Fuente: Autores

9.2.5 Firebase

Firestore se utiliza para la conexión desde la Raspberry Pi a Firestore y de Firestore a Android Studio, cumpliendo la función de ser el medio por el cual se envía la información de un lugar a otro, siendo su base de datos.

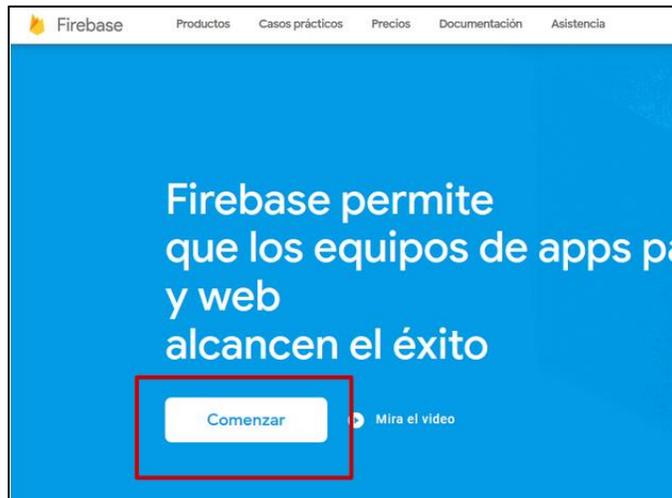


Ilustración 45. Firebase
Fuente: Autores

Una vez comenzado, se procede a crear el proyecto y se coloca el nombre que va a utilizar la base de datos.



Ilustración 46. Iniciar proyecto en Firebase
Fuente: Autores

Para agregar Firebase a la aplicación de Android Studio se debe seleccionar el botón que hace referencia a agregar Firebase a la aplicación de Android, tal como se señala en la Ilustración 47.

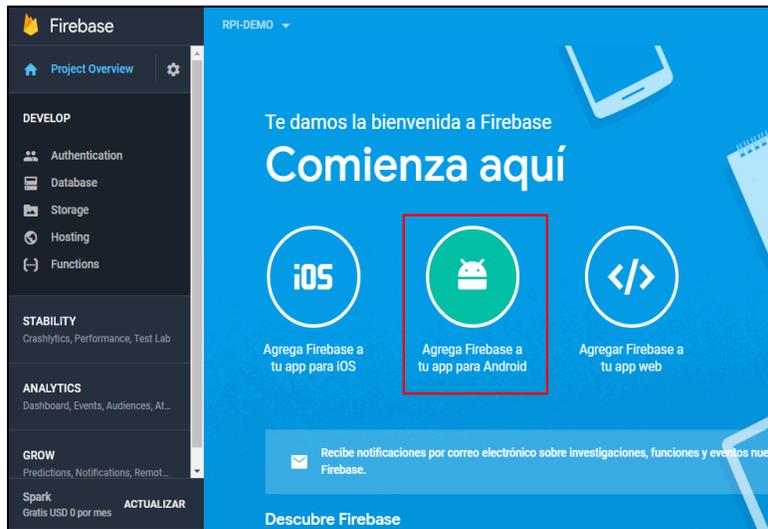


Ilustración 47. Agregar Firebase a Android Studio
Fuente: Autores

Se copia el nombre del paquete de Android Studio para llamar la aplicación a Firebase.



Ilustración 48. Llamar paquete
Fuente: Autores

El complemento de los servicios de Google para *Gradle* carga el archivo *google-services.json* se descarga. Para utilizar dicho complemento, se debe modificar los archivos *build.gradle*.

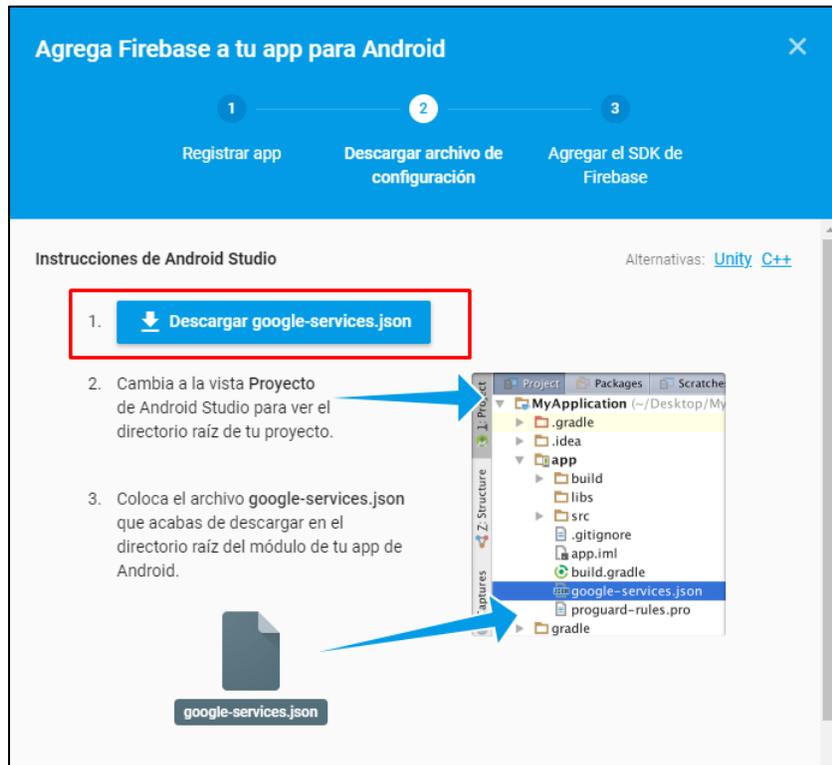


Ilustración 49. Complemento Google
Fuente: Autores

A continuación, se elige el lenguaje de programación que se va trabajar, para este proyecto se selecciona Kotlin, y seguido de esto, se sincroniza los datos de aplicación en milisegundos.

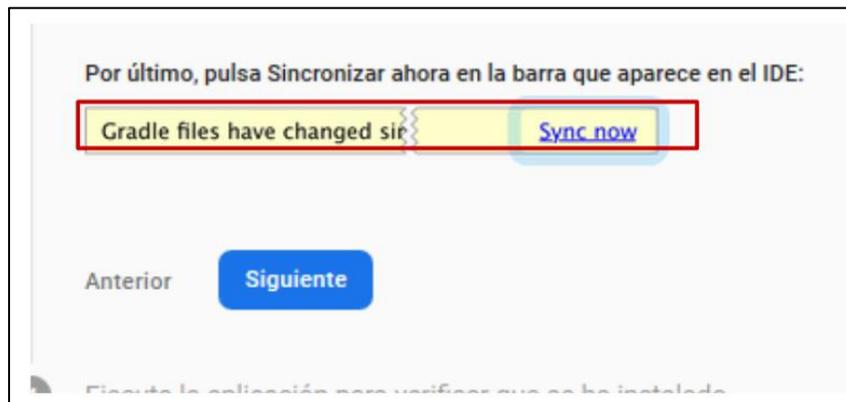


Ilustración 50. Sincronización de los datos de la aplicación
Fuente: Autores

Para comprobar si la aplicación se ha comunicado con los servidores de Firebase se debe mostrar la siguiente imagen.

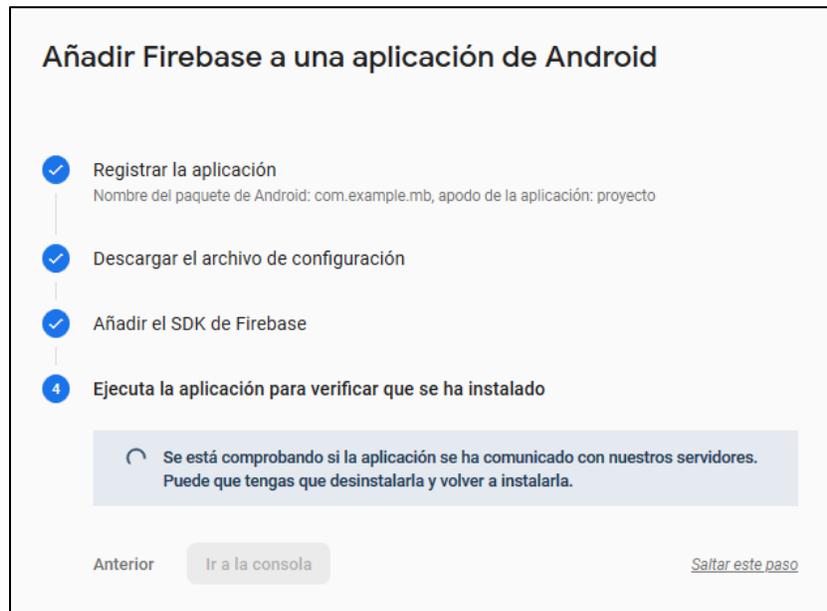


Ilustración 51. Conexión al servidor de Firebase
Fuente: Autores

En la siguiente imagen (ver Ilustración 52) se muestra el inicio de la base de datos en Realtime Database.

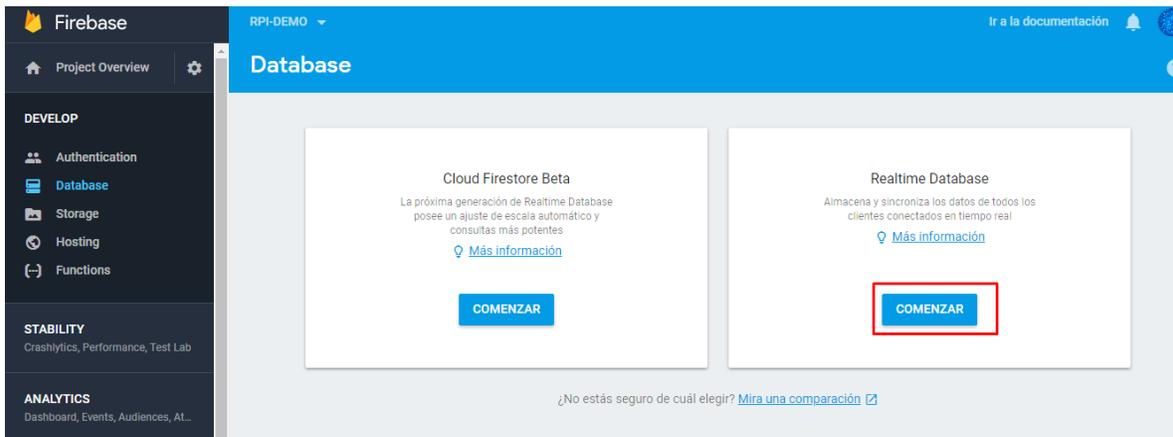


Ilustración 52. Crear BD en Realtime Database
Fuente: Autores

Una vez definida la estructura de datos se crean las reglas para proteger los datos.



Ilustración 53. Reglas de protección de datos
Fuente: Autores

Para la conexión con Python se debe descargar las credenciales de Firebase para generar nueva clave privada como se muestra a continuación.

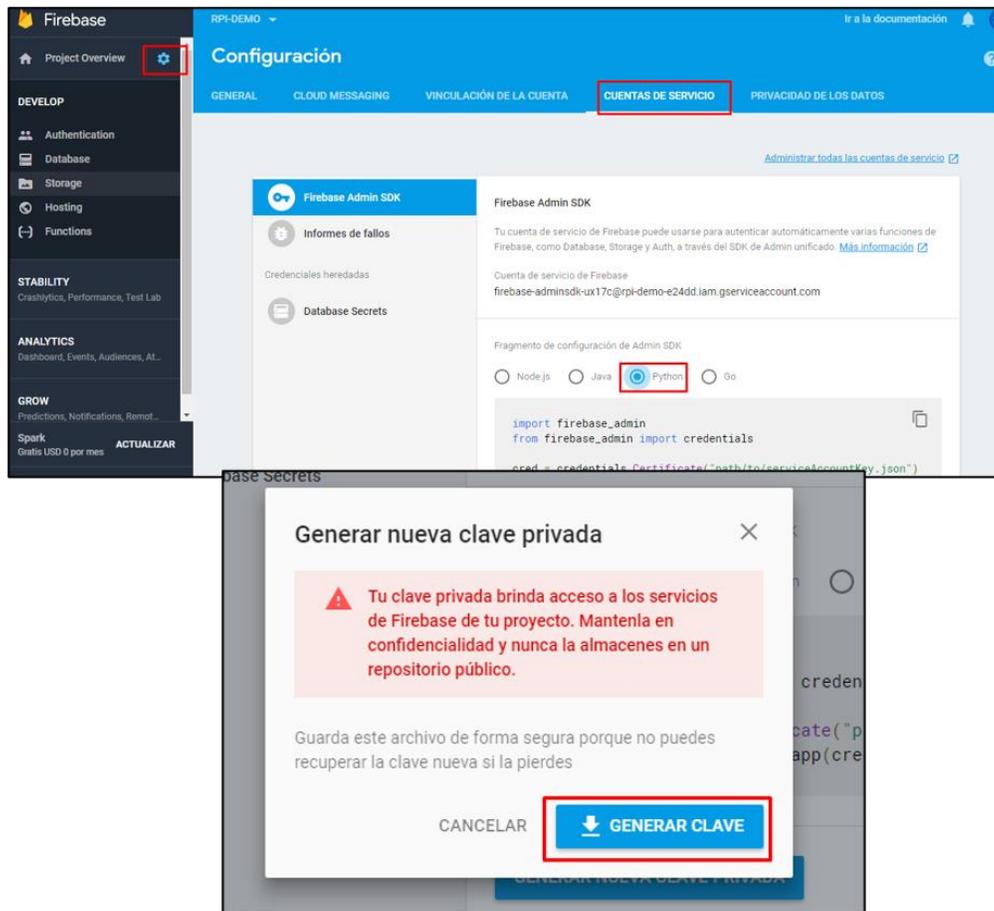


Ilustración 54. Generar clave privada
Fuente: Autores

Realtime Database en Firebase crea el siguiente árbol de nombre “Sonidos” y se despliega los nombres de los diferentes audios de los productos, cuando aparece “False” se debe a que no se ha tocado el producto, una vez se toca cambia su estado a “True”.



Ilustración 55. Base de Datos "Sonidos"
Fuente: Autores

9.2.6 Android Studio

9.2.6.1 Arquitectura

A continuación, se muestra la arquitectura general de la aplicación donde se observa la estructura de la aplicación, la clase usada y la activities que se diseñó.

9.2.6.1.1 Librerías utilizadas

Para realizar el proceso de llamar los audios desde la base de datos de Firebase con el fin de hacerle llegar al usuario su mensaje y optimizar el desarrollo del proyecto se utiliza las siguientes librerías:

```
import android.media.AudioManager
import android.media.MediaPlayer
import android.media.SoundPool
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.TextView
import android.widget.Toast
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
```

Ilustración 56. Librerías utilizadas
Fuente: Autores

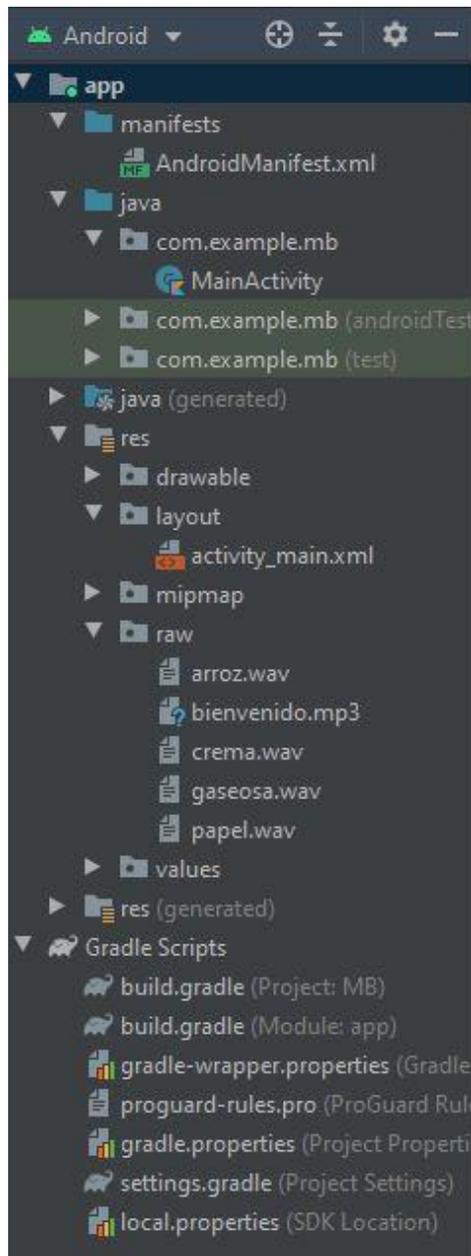


Ilustración 57. Arquitectura de la aplicación
Fuente: Autores

9.2.6.2 Descripción de paquetes

En la parte de manifest la cual es la raíz del proyecto, se encuentra el archivo con el nombre de AndroidManifest.xml que describe la información primordial de la aplicación para las herramientas de creación de Android Studio y el sistema

operativo, este archivo lo deben contener con el mismo nombre todas las aplicaciones móviles.

La carpeta llamada Java contiene el código fuente de la aplicación y clases. Android Studio crea inicialmente el código básico de su activity principal el cual siempre se encuentra en el paquete de java con el nombre de MainActivity.

En la carpeta layout, que se encuentra ubicada dentro de res, se encuentran los archivos relacionados con el diseño de la actividad, en el caso de la aplicación móvil desarrollada se cuenta con el activity_main.xml, el cual representa el diseño de la interfaz principal.

El fichero /app/build.gradle, contiene la versión del SDK necesaria para la compilación del proyecto, esta equivale a la mínima versión de Android que soporta la aplicación, referencias a las librerías externas utilizadas.

9.2.6.2.1 Clases

En la carpeta principal de java se crea el paquete com.example.mb, donde se define las clase que determina las funciones: getFirebaseSoundArroz, getFirebaseSoundCremas, getFirebaseSoundGaseosa, getFirebaseSoundPapel.

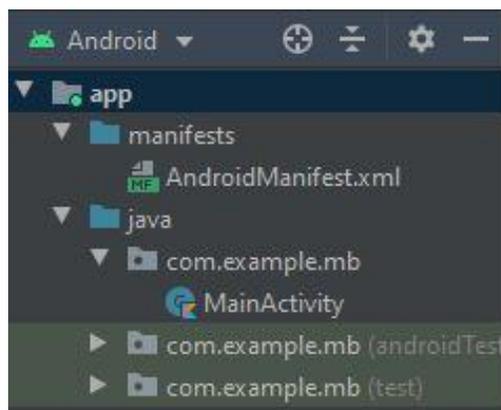


Ilustración 58. Clases
Fuente: Autores

9.2.6.3 Funciones

La aplicación propone cuatro funciones que obtienen de la base de datos el audio de las características del producto ubicado en la estantería para que este después de ser tocado logre reproducir el sonido y así conseguir que la personas con discapacidad visual cuenten con la información necesaria del producto que desea llevar. A continuación, se describe una de las funciones generadas para tal fin.

9.2.6.3.1 Función `getFirebaseSoundArroz ()`

En la clase `MainActivity`, se genera las funciones que obtienen de la base de datos de Firebase el audio que corresponde al producto del estante, en este caso el arroz, una vez reciba la señal de que el producto ha sido tocado su estado cambiará a `true` y se enviará el audio del mismo. De lo contrario, se cancelará.

Al mismo tiempo, se crean otras tres funciones que realizan el mismo procedimiento.

- `getFirebaseSoundCrema()`
- `getFirebaseSoundGaseosa()`
- `getFirebaseSoundPapel()`

```
fun getFirebaseSoundArroz() {
    val refSonidoArroz = refSonidos.child(DbConstants.ARROZ)

    refSonidoArroz.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(dataSnapshot: DataSnapshot) {
            val value = dataSnapshot.value
            Log.d("Main", "Value sonido arroz: $value")
            if (value == true) {
                playSoundArroz()
                textView.text = DbConstants.ARROZ.toUpperCase()
                refSonidoArroz.setValue(false)
            }
        }
        override fun onCancelled(error: DatabaseError) {
            Log.w("Main", "Failed to read value sonido 1.", error.toException())
        }
    })
}
```

Ilustración 59. Función `getFirebaseSoundArroz()`
Fuente: Autores

En la Ilustración 60, se muestra la función usada para activar el audio al cual se hace referencia, si hay un sonido reproduciendo este se detiene para permitir que el otro sonido se reproduzca.

```
fun playSoundArroz() {
    if (mediaPlayer.isPlaying()) {
        mediaPlayer.stop()
        mediaPlayer.reset()
    }
    mediaPlayer = MediaPlayer.create(context = this, R.raw.arroz)
    mediaPlayer?.start()
}
```

Ilustración 60. Función activar sonido
Fuente: Autores

Adicional a esto, se agregar una función aparte pequeña que permite reproducir un audio apenas la persona abre la aplicación.

```
fun playSoundBienvenido() {
    mediaPlayer = MediaPlayer.create(context = this, R.raw.bienvenido)
    mediaPlayer?.start()
}
```

Ilustración 61. Función activar audio de bienvenida
Fuente: Autores

9.2.6.4 Carpeta res

La carpeta res contiene las carpetas de drawable y layout entre otras que son donde se guardan las imágenes que se van a usar y los layout activities los cuales son donde se diseña la aplicación móvil.

9.2.6.4.1 Drawable

Las imágenes usadas durante la creación se almacenan directamente en la carpeta drawable para ser utilizadas por la aplicación móvil.

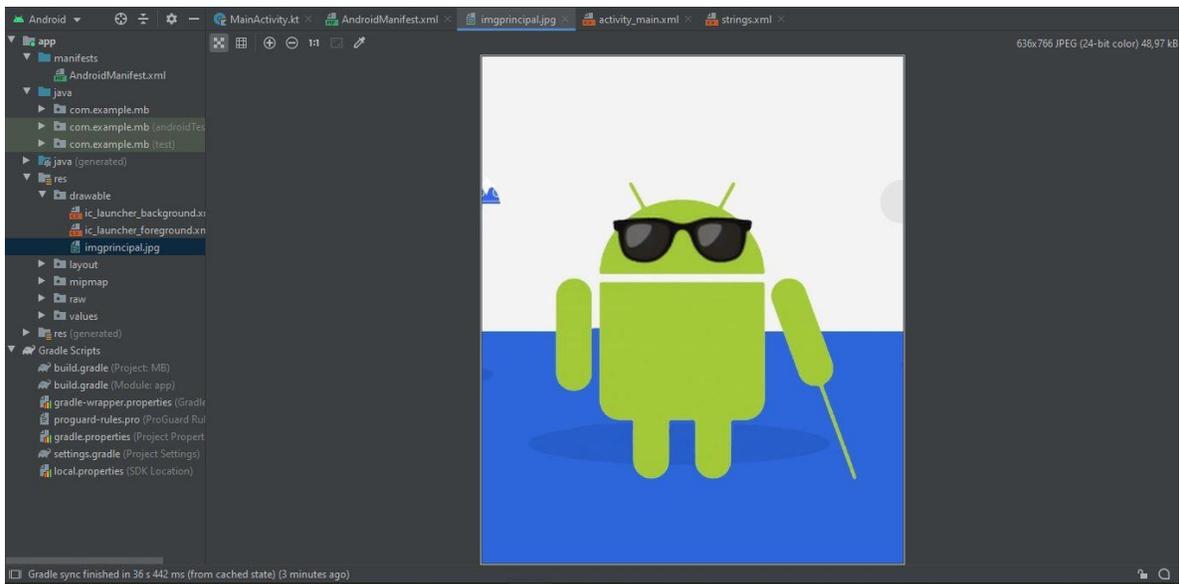


Ilustración 62. Carpeta drawable
Fuente: Autores

9.2.6.4.2 Layouts

En esta carpeta se almacenan los activitiys que corresponden a las diferentes pantallas de la aplicación, en nuestro caso solo es una pantalla.

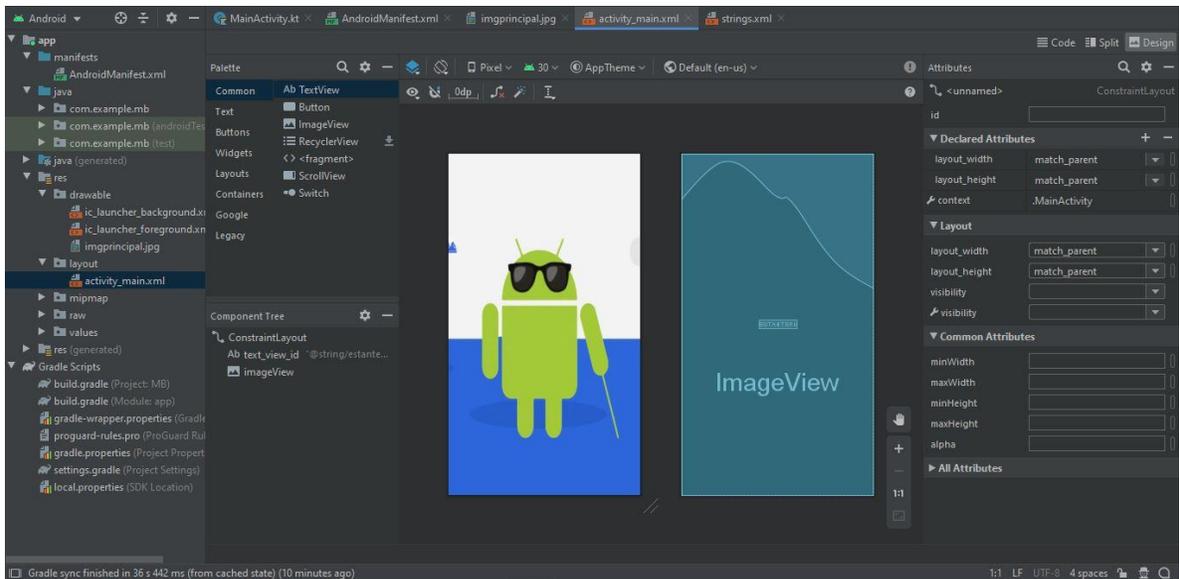


Ilustración 63. Carpeta layouts
Fuente: Autores

9.2.6.4.3 Activitys

9.2.6.4.3.1 activity_main.xml

En el activity main se define la interfaz en la que se despliegan el menú de opciones de la aplicación, en nuestro caso se mostrará únicamente una imagen debido a que como se ha mencionado anteriormente, la función de la aplicación es ser una interceptora entre el usuario y la estantería para permitirle escuchar el audio con las características del producto.

9.3 VALIDACIÓN DE FUNCIONALIDADES

Por último, es conveniente acotar que, una vez finalizado el desarrollo del prototipo final, se procede a validar el funcionamiento del sistema conforme al diseño. A continuación, se hace la descripción de la validación realizada.

9.3.1 Validación de las funcionalidades del prototipo

Para realizar la validación se utilizó un dispositivo móvil marca Motorola C con 4GB de memoria RAM y 34GB de almacenamiento y sistema operativo Android 8.

Lo primero que se validó fue la instalación de la aplicación Estantería.

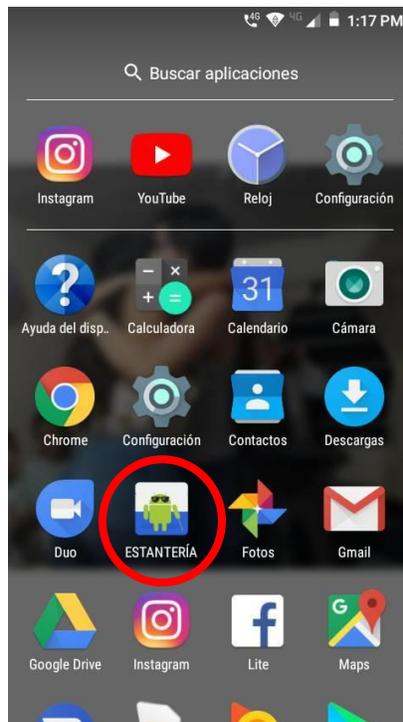


Ilustración 64. Validar instalación de la aplicación
Fuente: Autores

El siguiente paso es conectarse a una red con acceso a internet, ya sea datos móviles o Wifi.

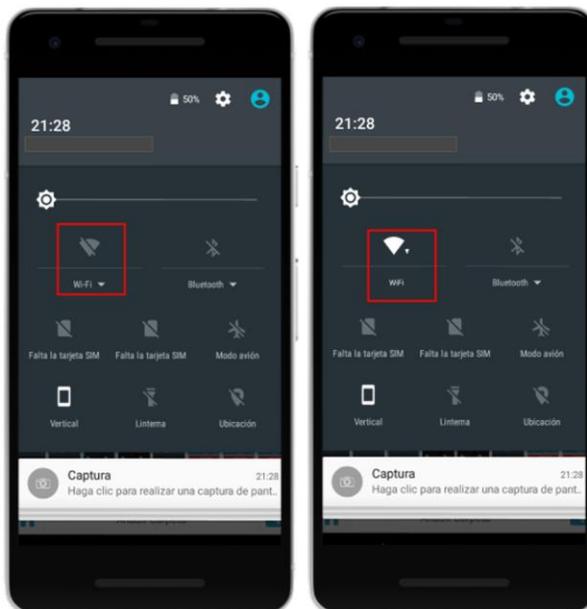


Ilustración 65. Conexión a red

Fuente: Autores

El siguiente paso es abrir la aplicación para realizar la prueba de que se está iniciando de la manera correcta



Ilustración 66. Aplicación iniciada
Fuente: Autores

Una vez en la aplicación se procede a validar el envío de audio de un producto ubicado en la estantería, en caso de haber tocado el producto se muestra un mensaje que dice “se eligió el producto” y seguido de este “se está reproduciendo el audio” para indicar que se está conociendo las características del producto.

```

pi@raspberrypi: ~/Desktop/proyecto
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ cd Desktop
pi@raspberrypi:~/Desktop $ cd proyecto
pi@raspberrypi:~/Desktop/proyecto $ python3 index.py
Pin 2 Se eligio el producto
papel
Pin 2 Se esta reproduciendo el Audio
Pin 1 Se eligio el producto
crema
Pin 1 Se esta reproduciendo el Audio
Pin 0 Se eligio el producto
gaseosa
Pin 0 Se esta reproduciendo el Audio
Pin 4 Se eligio el producto
arroz
Pin 4 Se esta reproduciendo el Audio

```

Ilustración 67. Validación envío de audio
Fuente: Autores

9.3.1.1 Validación del funcionamiento de la aplicación

Se realiza la revisión de funcionamiento de la aplicación de conformidad al diseño validando que la aplicación y sus funcionalidades cumplen correctamente las especificaciones de diseño, y se observa que la aplicación funciona de manera óptima y no tiene problemas en su ejecución. La validación de la aplicación nos permite comprobar que lo que se ha especificado cumple con las acciones que puede ejecutar un usuario con discapacidad visual.

ACCIÓN	RESULTADO
Se puede instalar la aplicación	✓
El usuario abre la aplicación correctamente	✓
El usuario escucha el audio correspondiente al producto tocado	✓

Tabla 31. Checklist validación funcionamiento de la aplicación móvil
Fuente: Autores

10 CONCLUSIONES

1. La utilización de diagramas UML (Lenguaje Unificado de Modelado) permitió entender la interacción entre el usuario y el sistema, así como la forma en que el sistema debe reconocer las peticiones del usuario con el fin de poder conocer sus necesidades y de esta manera desarrollar un buen trabajo.
2. Realtime Database de Firebase logra que se almacenen y sincronicen los datos de la base de datos, lo más importantes es que la sincronización se realiza de forma automática gracias a las funciones y los métodos ya determinados se encargan por sí solos de notificar los cambios y mostrarlos al usuario al momento de realizarlos, en este caso dando a conocer las características actuales del producto.
3. Es importante generar este tipo de proyectos porque por medio de ellos se logra tener en cuenta a las personas que presentan alguna discapacidad y así, entender que al igual que nosotros presentamos dificultades y que son igual de importantes, que por medio de estos proyectos se puede encontrar una posible solución a sus inconvenientes.
4. La definición de requerimientos al comienzo de un proyecto es parte importante, debido a que con ellos se puede tener una mejor claridad de las cosas a las que se enfrentarán durante el desarrollo y de las necesidades que se van a tener en cuenta para el mismo. De este modo, los requerimientos nos permitieron conocer específicamente las necesidades que se pretendían resolver con la realización del prototipo funcional.
5. La elección de Kotlin como lenguaje de programación para el desarrollo de la aplicación Android fue agradable debido a la comodidad y posibilidad que ofrece al desarrollar. Además de haber aprendido y adquirido nuevos conocimientos

tanto del lenguaje de programación como de otras tecnologías durante el desarrollo y que nos aportan positivamente, logrando que durante su desarrollo se ahorra código y fuera amigable trabajar.

6. No fue necesario tener una tarjeta microSD de mayor capacidad a la utilizada en el proyecto para el funcionamiento del sistema, ya que esta permitió la instalación del sistema operativo Raspbian con escritorio y recomendado.
7. En cuanto al contenido del desarrollo del proyecto, la aplicación y el IoT, nos quedamos con la sensación de haber cumplido con lo que debería ser el sistema de estantería en un futuro. Nos quedamos con que todo lo planeado ha sido realizado con la mejor disposición y dando todo de cada una para lograr finalizarlo satisfactoriamente.

11 RECOMENDACIONES

1. Para trabajo futuro se recomienda que la aplicación permita que dos personas a la vez interactúen con la estantería haciendo llegar a cada una de ellas la información del producto que ha tocado sin que se interfiera y no se tenga que esperar el turno.
2. Se recomienda que para la siguiente versión del proyecto la aplicación cuente con la capacidad de brindar al usuario el lugar donde se encuentra ubicada la estantería en el establecimiento.
3. Se recomienda que para próximas versiones la persona pueda realizar un registro y de esta manera obtenga información relevante del establecimiento, ya sea por descuentos y/o eventos.
4. Se recomienda que para el desarrollo de próximas versiones la aplicación logre distinguir por medio del reconocimiento facial a la persona para de esta manera hacer más personalizada la experiencia.

12 REFERENCIAS

- Admin. (2020). *Raspberry Pi Technology, Working and Its Applications*.
WatElectronics.Com. <https://www.watelectronics.com/know-all-about-raspberry-pi-board-technology/>
- Alsaadi, E., & Tubaishat, A. (2015). Internet of Things: Features, Challenges, and Vulnerabilities. *International Journal of Advanced Computer Science and Information Technology (IJACSIT)*, 4(1), 1–13.
- Alvarez, M. A. (2019). *Qué es la programación orientada a objetos*. Desarrollo Web. <https://desarrolloweb.com/articulos/499.php>
- Analytics, G. (2020). *Certificación ISO 27001*. Google Analytics. <https://support.google.com/analytics/answer/3407084?hl=es>
- Android Developers. (n.d.). *Aprende el lenguaje de programación Kotlin*. Android Developers. <https://developer.android.com/kotlin/learn?hl=es>
- Bourne, R. R. A., Flaxman, S. R., Braithwaite, T., Cicinelli, M. V., Das, A., Jonas, J. B., Keefte, J., Kempen, J., Leasher, J., Limburg, H., Naidoo, K., Pesudovs, K., Resnikoff, S., Silvester, A., Stevens, G. A., Tahhan, N., Wong, T., Taylor, H. R., Ackland, P., ... Zheng, Y. (2017). Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet Global Health*, 5(9), e888–e897. [https://doi.org/10.1016/S2214-109X\(17\)30293-0](https://doi.org/10.1016/S2214-109X(17)30293-0)
- Cisco. (2020). *¿Qué es Wi-Fi?* Cisco.Com. https://www.cisco.com/c/es_mx/products/wireless/what-is-wifi.html
- Cloud, G. (2020). *Cumplimiento de ISO/IEC 27001*. GoogleCloud. <https://cloud.google.com/security/compliance/iso-27001?hl=es>
- Comercio, M. D. E., & Turismo, I. Y. (2013). Decreto 1377 de 2012. *Ministerio de Comercio, Industria y Turismo, decreto 13*, 1–11. http://www.mintic.gov.co/portal/604/articles-4274_documento.pdf
- Commerce, M. (2019). *ESTÁNDARES DEL WIFI, TODO LO QUE DEBES SABER*. Mediacommerce. <https://www.mc.net.co/blog/estandares-del-wifi-todo-lo-que-debes-saber/>
- Congreso. (1991). *Constitucion política de colombia 1991 preambulo el pueblo de*

- colombia. 108.
- D.C., S. J. D. de la A. M. de B. (2013). *Ley 1618 de 2013 A Nivel Nacional*. ALCALDÍA MAYOR DE BOGOTÁ D.C.
- Desarrollo, P. de las N. U. para el. (2020). *Objetivos de Desarrollo Sostenible | PNUD*. Programa de Las Naciones Unidas Para El Desarrollo.
- Feng, Z. (2017). *Introduction of Wi-Fi standardization and Interoperability Certification Test*.
- Firebase. (2020a). *Cómo funcionan las reglas de seguridad*. Firebase.
<https://firebase.google.com/docs/rules/rules-behavior?hl=es>
- Firebase. (2020b). *Información sobre los proyectos de Firebase*. Firebase.
<https://firebase.google.com/docs/projects/learn-more>
- Firebase. (2020c). *Reglas de seguridad de Firebase*. Firebase.
<https://firebase.google.com/docs/rules?hl=es>
- Firebase community. (2020). *Firebase Realtime Database*. Firebase Community.
<https://firebase.google.com/docs/database?hl=es>
- Galindo Pérez, P. M., & Suárez Vargas, M. A. (2017). Diseño e implementación de una aplicación móvil android para el seguimiento de rutas de transporte urbano en el municipio de Yopal. *Investigacion e Innovación En Ingenierias*, 5(2), 138–173.
<https://doi.org/10.17081/invinno.5.2.2759>
- García, É. (2020). *Wi-Fi vs. Bluetooth: diferencias, ventajas y desventajas*. Adslzone.Net.
<https://www.adslzone.net/reportajes/wifi/wifi-vs-bluetooth/>
- Gestor, C. (2020a). *Hardware - definición*. Ciset . Centro de Innovación.
<https://www.ciset.es/glosario/451-hardware>
- Gestor, C. (2020b). *Software - Concepto y tipos de software*. Ciset . Centro de Innovación. <https://www.ciset.es/glosario/480-software>
- Gleote, L. M. (2014). *Mantenimiento y medidas de seguridad para estanterías*. Gestipolis. <https://www.gestipolis.com/mantenimiento-y-medidas-de-seguridad-para-estanterias/>
- GOLÁS, E. (2019). *Así son las dificultades de los ciegos para comprar en el supermercado*. SER.
https://cadenaser.com/programa/2019/10/29/ser_consumidor/1572341332_825287.html
- Guillermo, J. (2018). *Firebase, una plataforma usada por millones de aplicaciones -*

- Seguridad, privacidad y open source*. Platzi.
- Hassan Montero, Yusef. Martin Fernandez, F. J. (2013). *La experiencia del usuario no solo usabilidad*. <https://mti2013bgarciabarragan.wordpress.com/2013/12/14/la-experiencia-del-usuario-no-solo-usabilidad/>
- IBM. (2020). *Mobile technology*. Ibm.Com. <https://www.ibm.com/topics/mobile-technology>
- Inc, M. (2020). *Firebase, la infraestructura para crear y lograr el crecimiento de apps*. Merkle Inc. <https://www.merkleinc.com/es/es/blog/firebase-infraestructura-crecimiento-apps>
- Industries, A. (2019). *Adafruit 12 x Capacitive Touch Shield for Arduino - MPR121*. Adafruit.Com. <https://www.adafruit.com/product/2024>
- Invarato, R. (2018). *ART vs DALVIK, arquitectura Android*. Jarroba. <https://jarroba.com/arquitectura-android-art-vs-dalvik/>
- Ismail, N. (2017). *The mobile workforce: the new movement - Information Age*. Information Age. <https://www.information-age.com/mobile-workforce-moving-forward-work-123468825/>
- Kotlin. (2020a). *Kotlin evolution*. Kotlinlang. <https://kotlinlang.org/docs/reference/evolution/kotlin-evolution.html>
- Kotlin. (2020b). *Kotlin for Android - Kotlin Programming Language*. Kotlinlang. <https://kotlinlang.org/docs/reference/android-overview.html>
- Lucas, J. (2019). *Arduino vs Raspberry Pi*. OpenWebinars S.L. <https://openwebinars.net/blog/arduino-vs-raspberry-pi/>
- MARTÍN, A. A. (2010). ESTUDIO E IMPLEMENTACIÓN DE SENSORES DE FUERZA 3D CON APLICACIÓN A MANOS ROBÓTICAS. *Universidad Carlos III de Madrid*, 39–67.
- Máster, T. F. De. (2017). *Autonomía de personas ciegas en supermercados mediante realidad aumentada*.
- Ministerio Colombia. (2020). *Política Pública Nacional de Discapacidad e Inclusión Social*. Ministerio de Salud y Protección Social. <https://www.minsalud.gov.co/proteccionsocial/promocion-social/Discapacidad/Paginas/politica-publica.aspx>
- Ministerio de Tecnologías de la Información y las Comunicaciones. (2014). *ConVerTIC ¿Qué es?* Convertic.Gov.Co. <https://www.convertic.gov.co/641/w3-propertyvalue-36246.html>

- Morantes, C. N. (2019). *El braille, un sistema opacado por la tecnología en Colombia*. RCN Radio. <https://www.rcnradio.com/recomendado-del-editor/el-braille-un-sistema-opacado-por-la-tecnologia-en-colombia>
- Once. (2020a). *Ceguera y deficiencia visual*. ONCE. <https://www.once.es/dejanos-ayudarte/la-discapacidad-visual/concepto-de-ceguera-y-deficiencia-visual>
- Once. (2020b). *Servicios Sociales de la ONCE: inclusión de la discapacidad*. ONCE. <https://www.once.es/servicios-sociales>
- Oracle. (2019a). *¿Qué es Internet of Things (IoT)?* Oracle.Com. <https://www.oracle.com/co/internet-of-things/what-is-iot.html>
- Oracle. (2019b). *Base de datos*. Oracle.Com. <https://www.oracle.com/co/database/what-is-database.html>
- Raspberry. (2015). Quick start 5. *Raspberrypi*, 5–8.
- Raspberrypi. (2019). *SD cards*. Raspberrypi. <https://www.raspberrypi.org/documentation/installation/sd-cards.md>
- RealVNC. (2020). *Soporte y acceso remoto sencillos y seguros*. RealVNC. <https://www.realvnc.com/es/connect/>
- Red Hat, I. (2020). *Arquitecturas / ARM / Raspberry Pi*. Fedoraproject.Org. https://fedoraproject.org/wiki/Architectures/ARM/Raspberry_Pi
- Remolina, L. D. C. (2019). *Generación de paisajes háptico-auditivos para asistencia de personas con discapacidad visual mediante el diseño de un lenguaje de representación sensorial*. Universidad Autónoma de Bucaramanga.
- Rios, D. A. (2011). Seguridad En Redes Wi-Fi. *Seguridad En Redes Telematicas*, 1–29. https://sites.google.com/site/seguridadinformatica97/8-seguridad-en-redes-wifi%0Ahttp://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_ES.pdf
- Salazar, J., & Silvestre, S. (2014). Internet de las cosas. *Universidad Católica*, 1–27.
- Smarter, A. T. (2014). *MPR121*. 1–2.
- Source, F. O. (2020). *Firestore Android SDK*. Firebase Open Source. <https://firebaseopensource.com/projects/firebase/firebase-android-sdk/>
- Statista. (2020). *Usuarios de teléfonos inteligentes en todo el mundo 2016-2021*. S. O'Dea. <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- Tiempo, R. el. (2020). *Estas son las marcas más consumidas por los hogares colombianos*. El Tiempo. <https://www.eltiempo.com/economia/los-productos-que-son->

mas-consumidos-en-los-hogares-colombianos-500208

UNESCO. (2019). *¿Qué es la inclusión?* Incluyeme.Com. <https://www.incluyeme.com/que-es-la-inclusion-2/>

Unión Internacional de Telecomunicaciones. (2012). *Visión general de la Internet de las cosas (ITU-T Y.4000/Y.2060 (06/2012))*. 20. <http://handle.itu.int/11.1002/1000/11559>

VideumCorp. (2015). *Estructura de una app*. Desarrollador Android. <https://desarrollador-android.com/disenopautas-2/estructura-de-una-app/>

Whitney, L. (2016). *La Raspberry Pi 3 llega con Wi-Fi y Bluetooth incorporados*. Cnet.Com. <https://www.cnet.com/es/noticias/la-raspberry-pi-3-llega-con-wi-fi-y-bluetooth-incorporados/>

WHO. (2018a). *Ceguera y discapacidad visual*. Organización Mundial de La Salud. <https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment>

WHO. (2018b). *Discapacidad y salud*. Organización Mundial de La Salud. <https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health>

XenonStack. (2020). *Kotlin vs Java Comparison - Which One is Better?* XenonStack.

13 ANEXOS

ANEXO 1 – Diseño previo de funcionamiento

ANEXO 2 – Encuesta

ANEXO 3 – Resultados de la encuesta

ANEXO 4 – Bosquejo de la arquitectura del sistema

ANEXO 5 – Manual de usuario

ANEXO 6 – Manual de instalación

A. ANEXO 1 – Diseño previo de funcionamiento

A continuación, se presenta un diseño previo del funcionamiento de lo que llegaría a ser el prototipo funcional mostrando la función del usuario con la estantería y la señal del dispositivo móvil.



Ilustración 68. Anexo diseño del funcionamiento
Fuente: Autores

La persona con discapacidad visual tocará el producto deseado y automáticamente en su celular se enviará un mensaje en modo audio diciendo una breve descripción de las características del producto.

B. ANEXO 2 – Encuesta

En el siguiente anexo presentamos la encuesta que se utilizó para abstraer la información como fuentes primarias realizada aproximadamente a una muestra de 14 personas con discapacidad visual.

ENCUESTA A PERSONAS CON DISCAPACIDAD VISUAL

- 1. Genero**
 - a) Masculino**
 - b) Femenino**

- 2. Edad**
 - a) 10 – 20**
 - b) 21 – 30**
 - c) 31 – 40**
 - d) 41 – 50**
 - e) 51 – 60**

- 3. Nivel de estudio**
 - a) Primaria incompleta**
 - b) Primaria**
 - c) Secundaria incompleta**
 - d) Bachillerato**
 - e) Universidad**

- 4. ¿Con qué frecuencia va al supermercado?**
 - a) Frecuentemente**
 - b) Algunas veces**
 - c) Rara vez**

- 5. ¿Ha encontrado en los supermercados, inconvenientes o problemas de espacio físico para su libre movilidad a la hora de hacer compras?**
 - a) Sí**
 - b) No**

- 6. ¿De las siguientes dificultades cuales ha encontrado cuándo realiza sus compras en un supermercado?**
 - a) No encontrar fácilmente el producto**
 - b) Comprar un producto vencido**

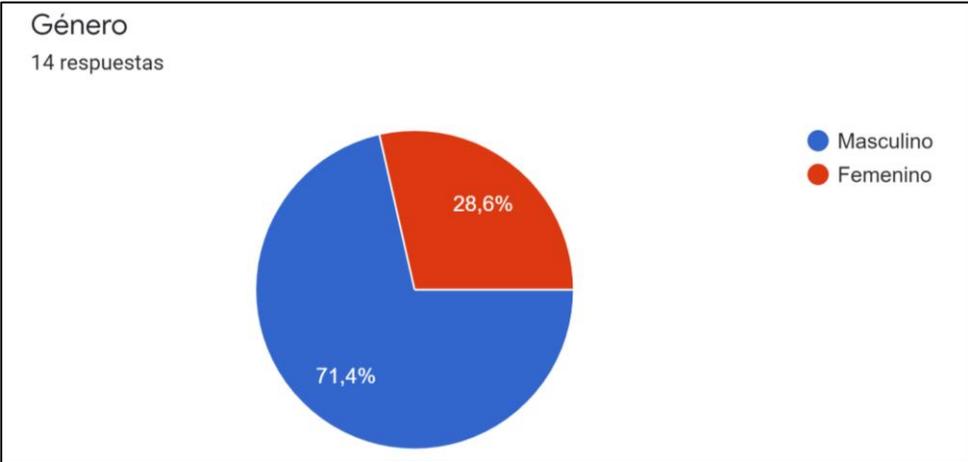
- c) Llevar un producto no deseado
- d) No reconocer el precio del producto
- e) Me toca caminar mucho
- f) No encontrar ayuda
- g) Otro _____

7. ¿De las siguientes ayudas cuales te gustaría encontrar?
- a) Persona guía dentro del establecimiento
 - b) Etiquetado braille
 - c) Sistema que le permita conocer las características del producto elegido
8. ¿Cómo realiza sus compras en un supermercado?
- a) Acompañado
 - b) Por mi mismo
 - c) Otros lo hacen por mí
9. ¿Cómo se guía en el supermercado?
- a) Voy acompañado/a
 - b) Uso bastón
 - c) Perro guía
10. ¿Cuenta con teléfono inteligente con pantalla táctil?
- a) Sí
 - b) No
11. ¿Se le facilita el uso del teléfono inteligente con pantalla táctil?
- a) Poco
 - b) Mucho
 - c) Nada
12. Si encontrara en el supermercado un único lugar en el cual, por medio de un audio, una grabación le orientara y le ayudara a conocer las características del producto a comprar y su valor ¿sería de gran ayuda para su experiencia de compra?
- a) Sí
 - b) No

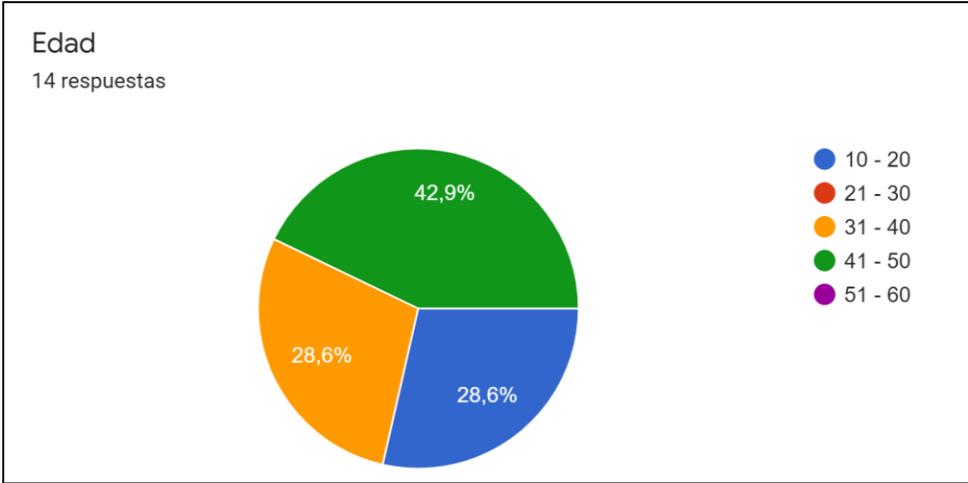
C. ANEXO 3 – Resultados de la encuesta

Como se precisó en el anexo 2, se realizó una encuesta a una muestra de 14 personas con discapacidad visual, para por medio de ellas poder evaluar la viabilidad de nuestro proyecto y así mismo, conocer más certeramente las dificultades e inconvenientes que presentan al momento de dirigirse a realizar sus compras. De acuerdo con esto, se obtuvieron los siguientes resultados.

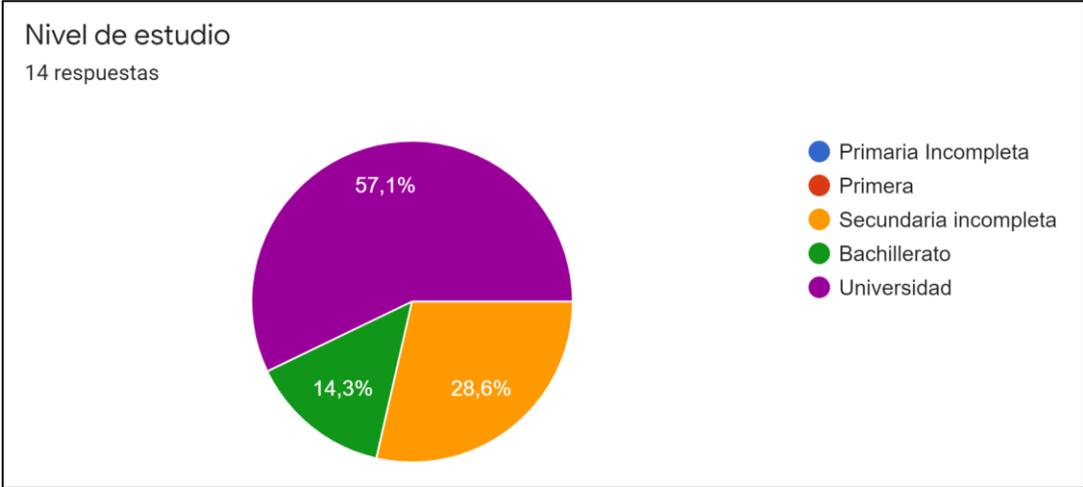
Nuestra muestra fue respondida por un 71.4% en personas masculinas y un 28.6% personas femeninas.



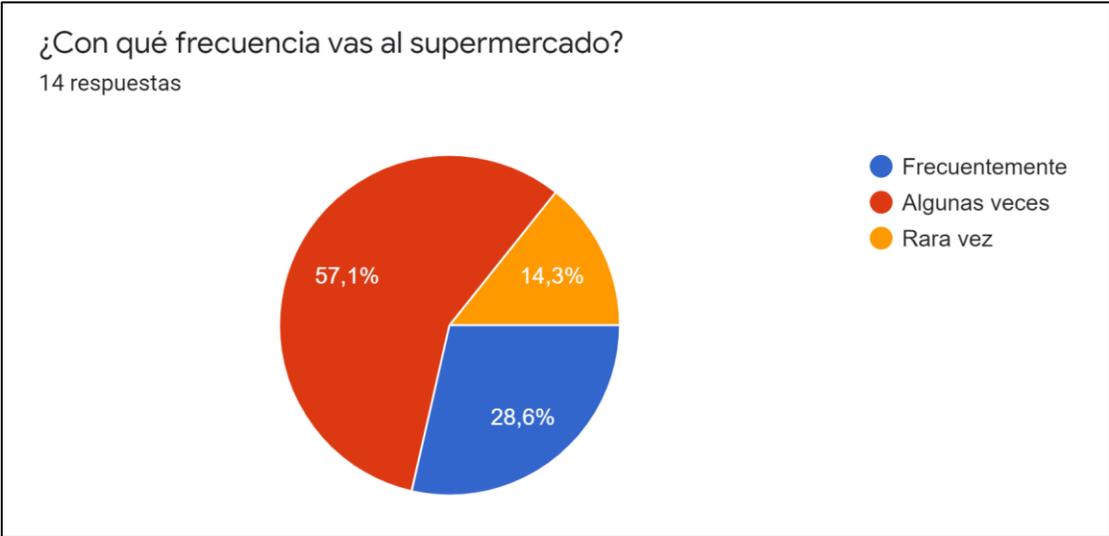
Así mismo, el mayor rango de edad con un 42.9% lo obtuvieron personas entre los 41 – 50 años y con porcentajes iguales a 26.6% personas entre los 10 – 20 y 31 – 40.



Con un 57,1% de las personas que respondieron la encuesta están en un nivel de estudio Universitario, con un 28,6% secundaria incompleta y 14,3% cuenta con bachillerato.



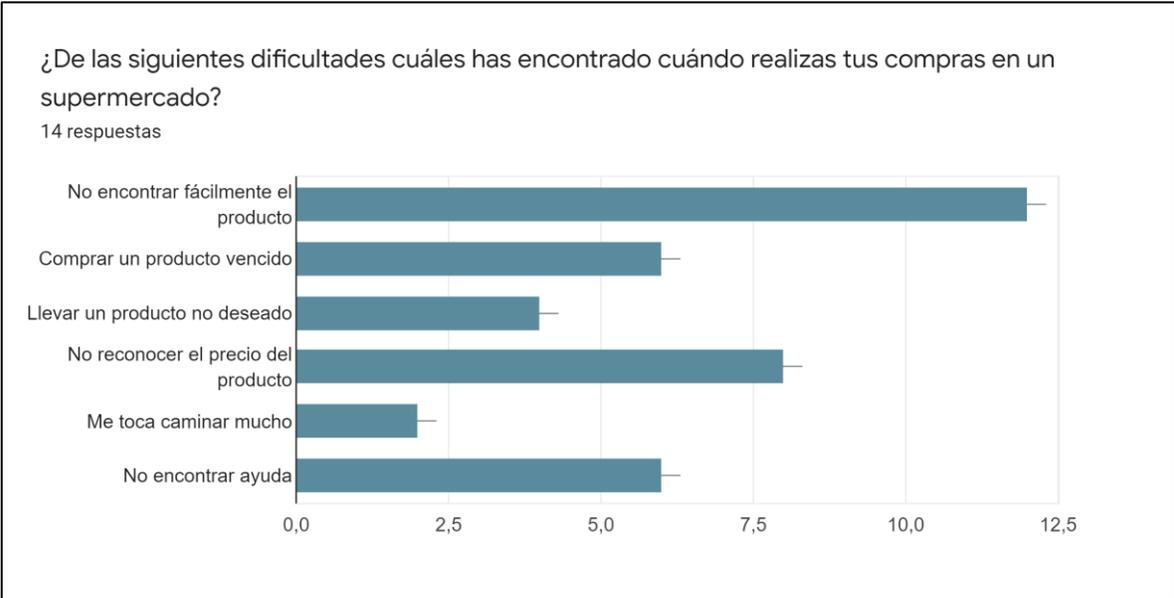
De acuerdo con la frecuencia con la que van al supermercado, las personas dieron respuesta con que un 57,1% de ellas van algunas veces, un 28,6% van frecuentemente y un 14,3% rara vez.



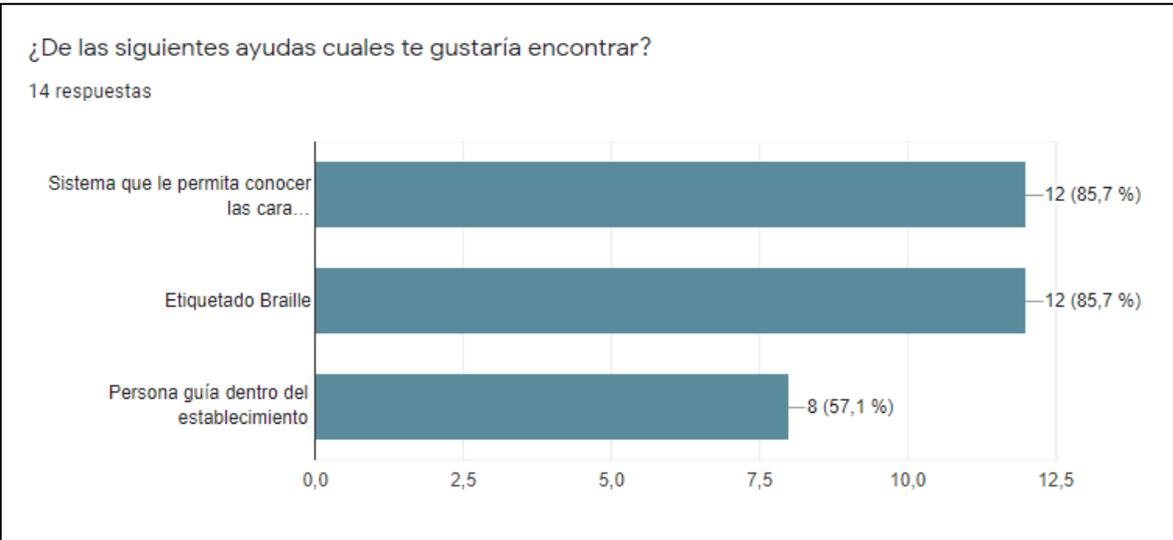
Con un 78,6% de los encuestados se dio respuesta a que si han encontrado en los supermercados, inconvenientes o problemas de espacio físico para su comodidad a la hora de hacer la compra y otro 21,4% no ha encontrado percancez.



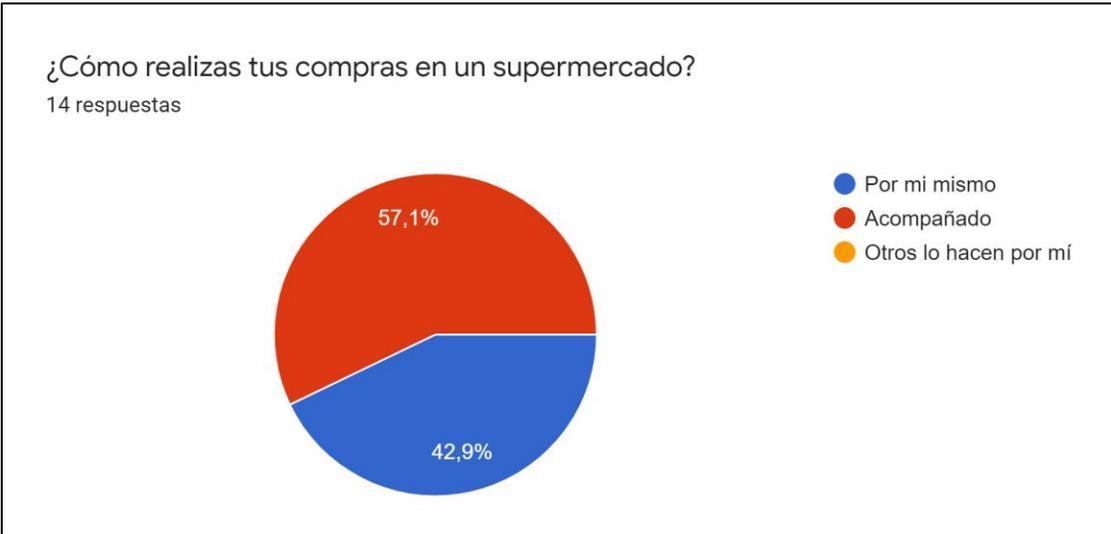
De las dificultades que más han encontrado las personas con discapacidad al momento de hacer la compra han sido no encontrar fácilmente el producto, no reconocer el precio del producto, no encontrar ayuda, comprar un producto vencido y llevar un producto no deseado.



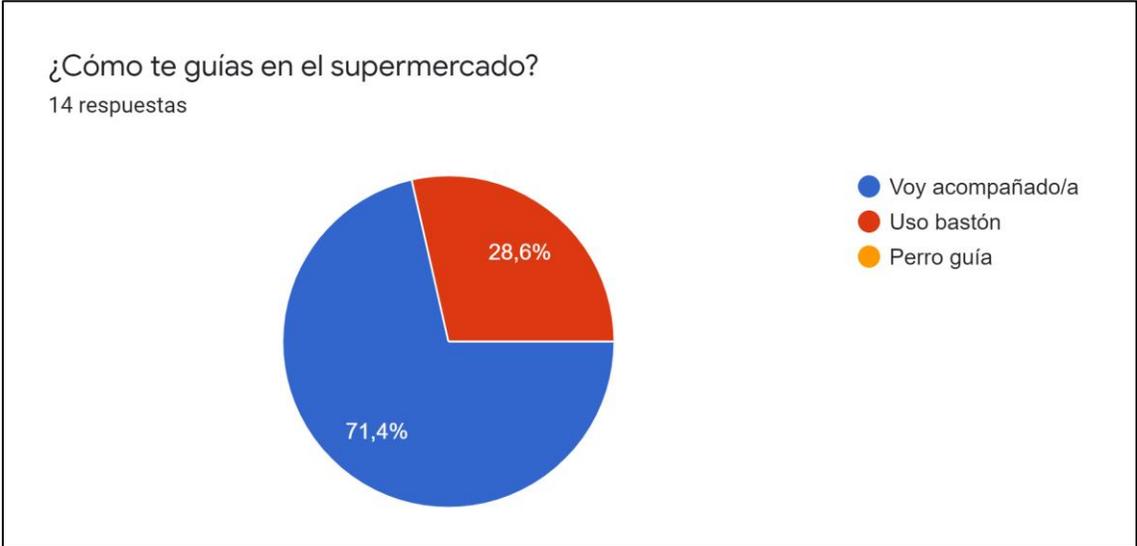
Se planteó la idea de cuales serían esas ayudas que les gustaría encontrar al momento de su visita al supermercado y entre las más votadas con un 85,7% cada una, se encuentra un sistema que le permita conocer las características del producto elegido y el etiquetado braille, dando así el tercer puesto a una persona guía dentro del establecimiento con un porcentaje del 57,1%.



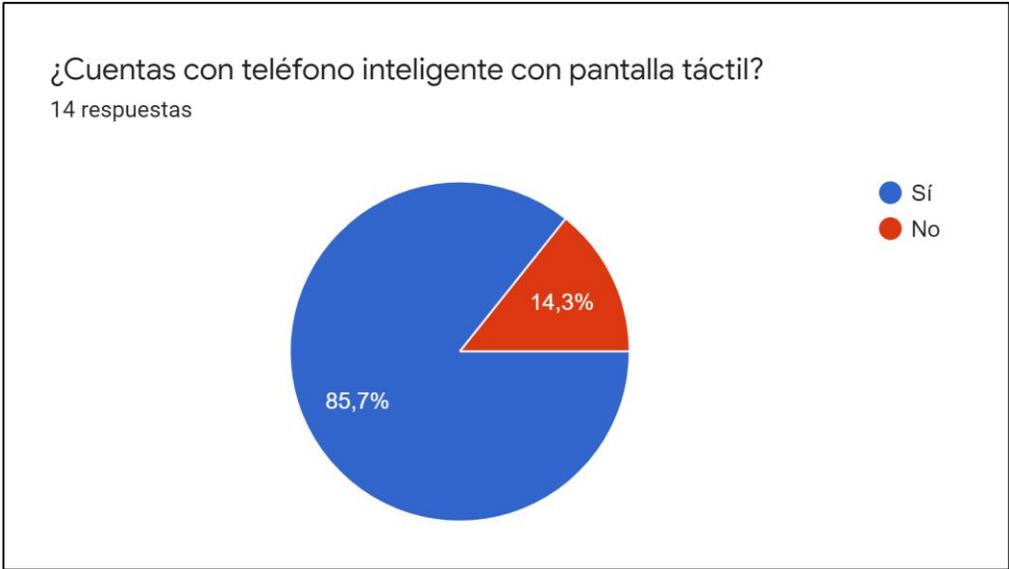
Indagando acerca de como realizan sus compras, nos encontramos con que un 57,1% de las personas van acompañadas y el otro 42,9% va por si solo al supermercado.



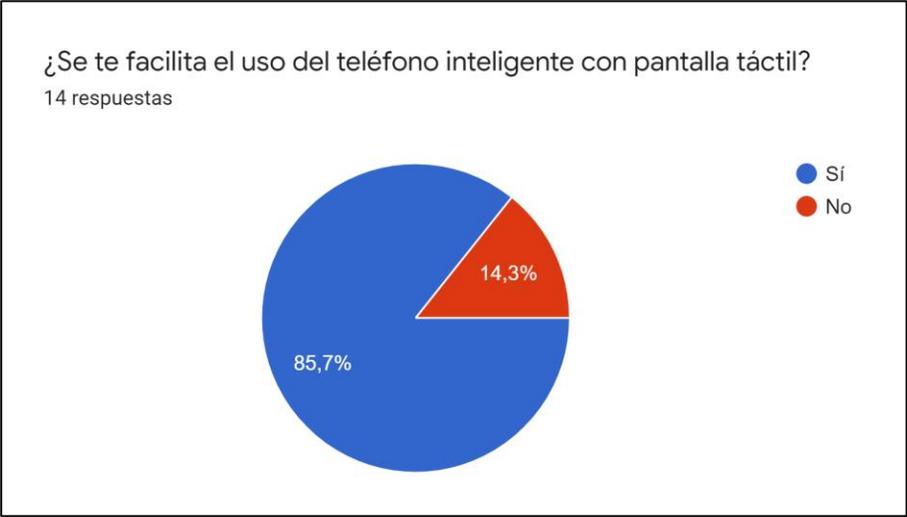
Muy de la mano con la pregunta anterior, está da respuesta con un 71,4% que la manera en la que se guían en el supermercado es porque van acompañados y el otro 28,6% es con ayuda de su bastón.



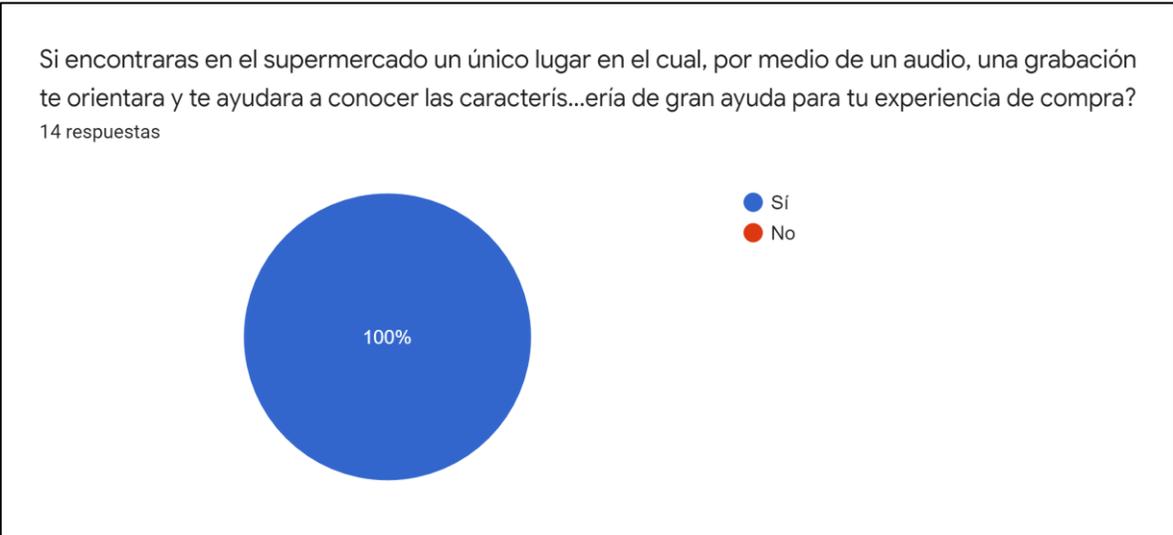
Quiriendo conocer que tanto utilizan las personas con discapacidad visual un teléfono inteligente con pantalla táctil se plantea la pregunta de si contaban con uno de ellos, a lo que una cantidad considerable con un 85,7% respondió que si contaban con él, a lo que un 14,3% no lo hace.



Teniendo como guía la pregunta anterior se preguntaba que tanto se le facilitaba el uso del teléfono móvil con pantalla táctil, obteniendo que a un 85,7% sí se le facilita, contrarrestando con un 14,3% a la que no se le facilita el uso del mismo.

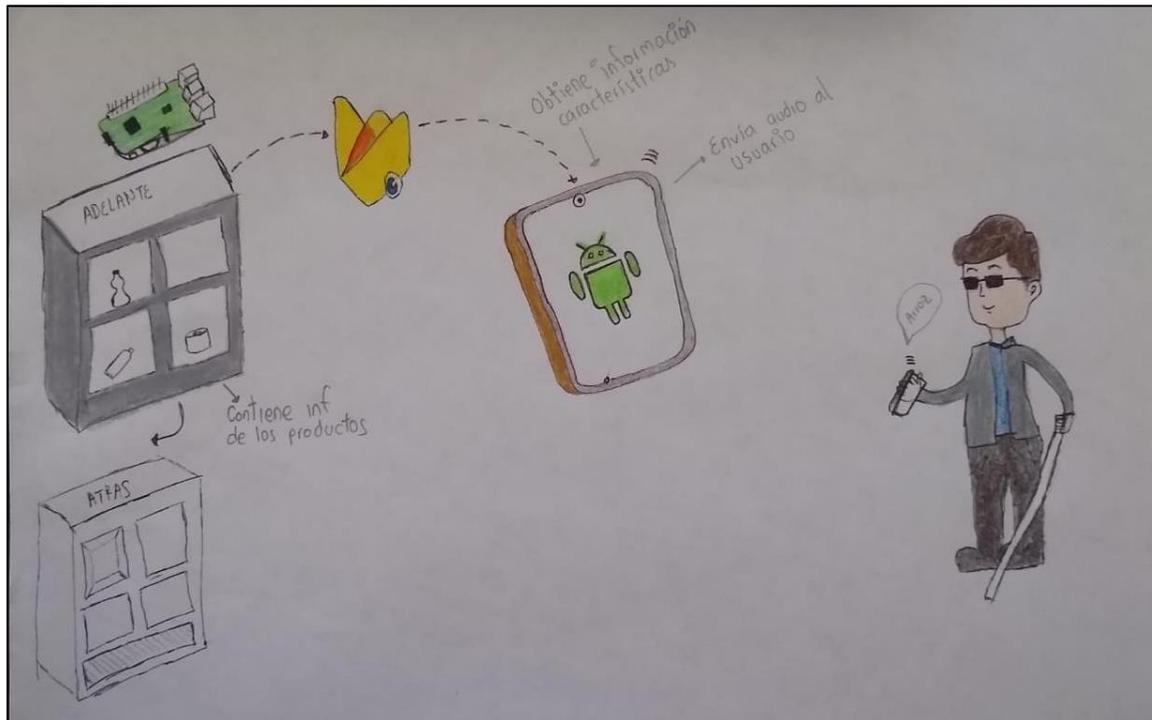


Para finalizar, se expuso que si llegarán a encontrar en el supermercado un único lugar en el cual, en el cual por medio de un audio, una grabación le orientara y le ayudara a conocer las características del producto a comprar y su valor les sería de gran ayuda para su experiencia de compra a lo que con un total del 100% respondieron que dando una respuesta favorable.



D. ANEXO 4 – Bosquejo de la arquitectura del sistema

En este anexo, se presenta una guía del bosquejo que se realizó acerca del funcionamiento del sistema con la interacción que tiene cada parte, IoT y aplicación móvil, para así realizar los diagramas, analizar y aclarar dudas.



MANUAL DE USUARIO

SISTEMA DE ESTANTERÍA INTELIGENTE
BASADO EN TECNOLOGÍAS MÓVILES E IOT
PARA PERSONAS CON DISCAPACIDAD
VISUAL

El presente manual describe los pasos a seguir para uso de la aplicación móvil.

YESSICA VIVIANA EUGENIO GARCIA
STEFANNY GONZALEZ GELVEZ

INGENIERIA DE SISTEMAS – 2020

TABLA DE CONTENIDO

Objetivo:	141
Primer uso	141
Interactuar con la estantería:.....	143

TABLA DE FIGURAS

Figura 1 Descargar APK al dispositivo móvil.....	141
Figura 2 Carpeta de ubicación del APK	142
Figura 3 Instalar APK	142
Figura 4 Aplicación instalada.....	143
Figura 5 Activar WiFi	144
Figura 6 Ir a Aplicaciones.....	144
Figura 7 Aplicación Ejecutada	145
Figura 8 Estantería.....	145
Figura 9 Descripción de productos.....	146

MANUAL DE USUARIO

Objetivo:

Dar a conocer los pasos al usuario inicial para el uso y la instalación de la aplicación móvil estantería en su celular.

Primer uso

Si por primera vez va a interactuar con la aplicación estantería, es recomendable leer los pasos del siguiente manual.

Paso 1: El archivo ejecutable con extensión .apk se descarga, así mismo se debe copiar en el espacio de almacenamiento del dispositivo

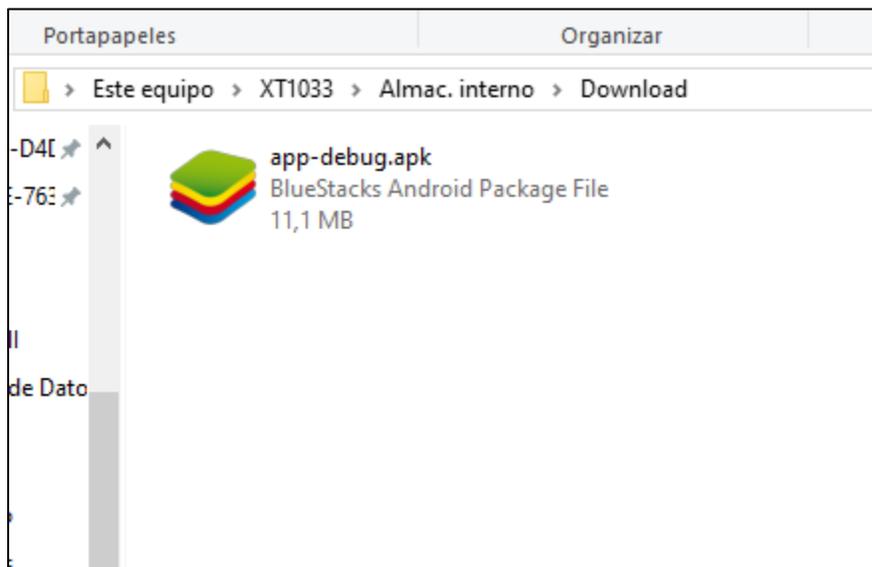


Figura 1 Descargar APK al dispositivo móvil
Fuente: Autores

Paso 2: Al descargar el .apk, se debe ir la carpeta donde se guardó .

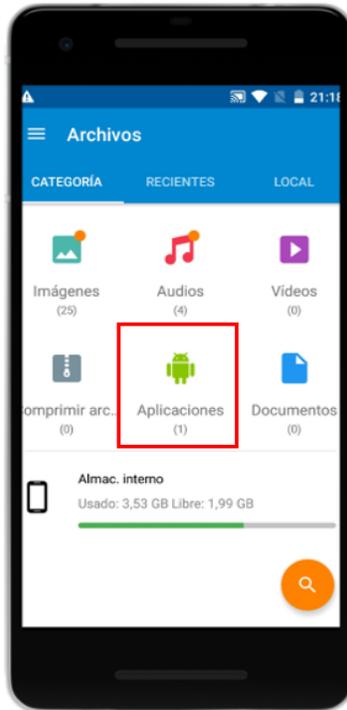


Figura 2 Carpeta de ubicación del APK
Fuente: Autores

Paso 3: Ya ingresado se debe dar instalar.



Figura 3 Instalar APK
Fuente: Autores

Paso 4: En la imagen que se muestra a continuación da a conocer que está instalada la aplicación.

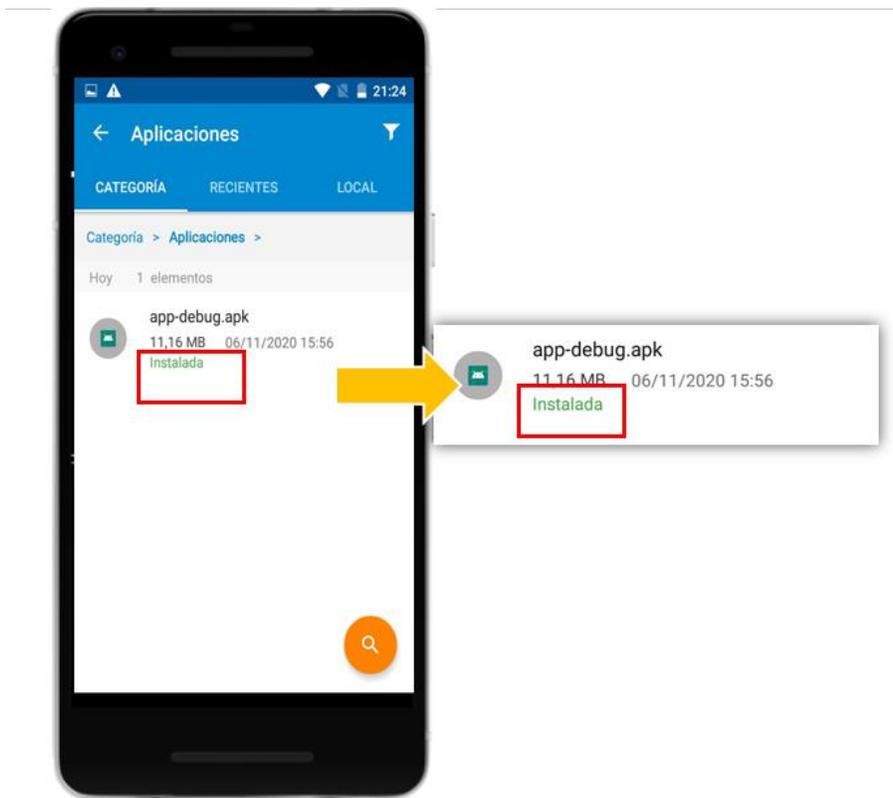


Figura 4 Aplicación instalada
Fuente: Autores

INTERACCIÓN CON LA ESTANTERÍA

Los siguientes pasos son aquellos que se deben realizar cada vez que se va interactuar con la estantería, estando ubicado al frente de ella.

Paso 5: Se debe conectar a internet, en este caso nos conectamos a Wifi como se muestra en la siguiente imagen

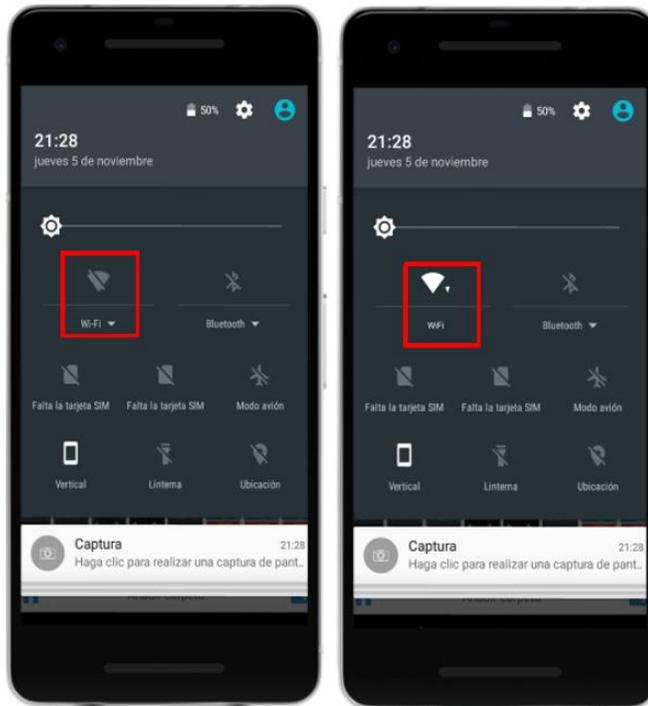


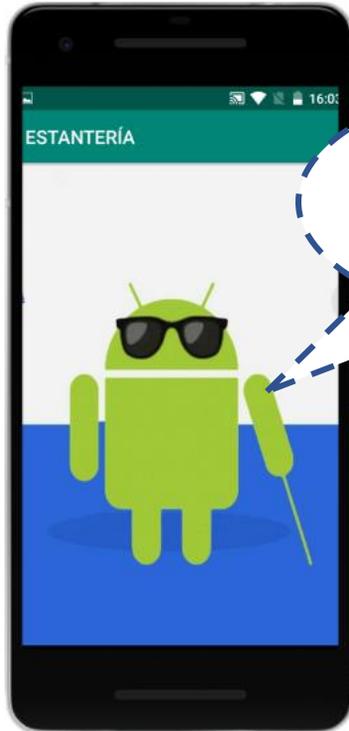
Figura 5 Activar Wifi
Fuente: Autores

Paso 6: Ir a aplicaciones y buscar la aplicación ESTANTERÍA.



Figura 6 Ir a Aplicaciones
Fuente: Autores

Paso 6: En la siguiente imagen se muestra la aplicación ejecutada.



Al abrir la aplicación se reproduce un audio, que dice "Bienvenido disfrute su compra"

Figura 7 Aplicación Ejecutada
Fuente: Autores

Paso 7: Para reproducir los audios de las descripciones de los productos se deben tocar.



Figura 8 Estantería
Fuente: Autores

Paso 8: Una vez hecho el paso número 7, se reproduce el audio en la aplicación dando a conocer la descripción del producto tocado.

A continuación, se muestra un ejemplo de si se llegara a tocar el producto de la imagen anterior.



Producto: Crema de dientes
Marca: Sonríe
Cantidad: 60 ml cm³

Precio sugerido al público: 2600 pesos

Figura 9 Descripción de productos
Fuente: Autores

F. ANEXO 6 – Manual de instalación

MANUAL DE INSTALACIÓN

SISTEMA DE ESTANTERÍA INTELIGENTE
BASADO EN TECNOLOGÍAS MÓVILES E IOT
PARA PERSONAS CON DISCAPACIDAD VISUAL

**El presente manual describe los pasos a seguir para la
instalación de sistema**

YESSICA VIVIANA EUGENIO GARCIA
STEFANNY GONZALEZ GELVEZ

INGENIERIA DE SISTEMAS -2020

TABLA DE CONTENIDO

1	Objetivo.....	150
2	Iniciar sistema de estantería.....	150
3	Instalación VNC.....	150
4	Cambio de contraseña de la Raspberry pi 3.....	153
5	Ejecutar Sistema:.....	153
6	Sistema en Ejecución	153

TABLA DE FIGURAS

Figura 1 Activar VNC.....	150
Figura 2 Descargar VNC	151
Figura 3 IP de la Raspberry.....	151
Figura 4 Nueva conexión	152
Figura 5 Conectar por VNC	152
Figura 6 Ejecutar Sistema	153
Figura 7 Ejecución del Sistema	154

MANUAL DE INSTALACIÓN

Objetivo

Guiar paso a paso a la persona autorizada del manejo del sistema.

Iniciar sistema de estantería

Para la instalación del sistema se deben seguir los siguientes pasos del uno hasta el diez que se explicarán continuación.

Paso 1: Se conecta la Raspberry pi 3 con los dispositivos teclado, mouse, pantalla.

Paso 2: Se debe tener la carpeta llamada proyecto donde se encuentra el sistema

Instalación VNC

Para no conectar varios dispositivos de entrada y salida se descargará el VNC y así tener una conexión remota y facilitar el manejo del sistema.

Paso 3 Una vez iniciado el sistema operativo se da clic en interfaces y para activar se elige la opción VNC.

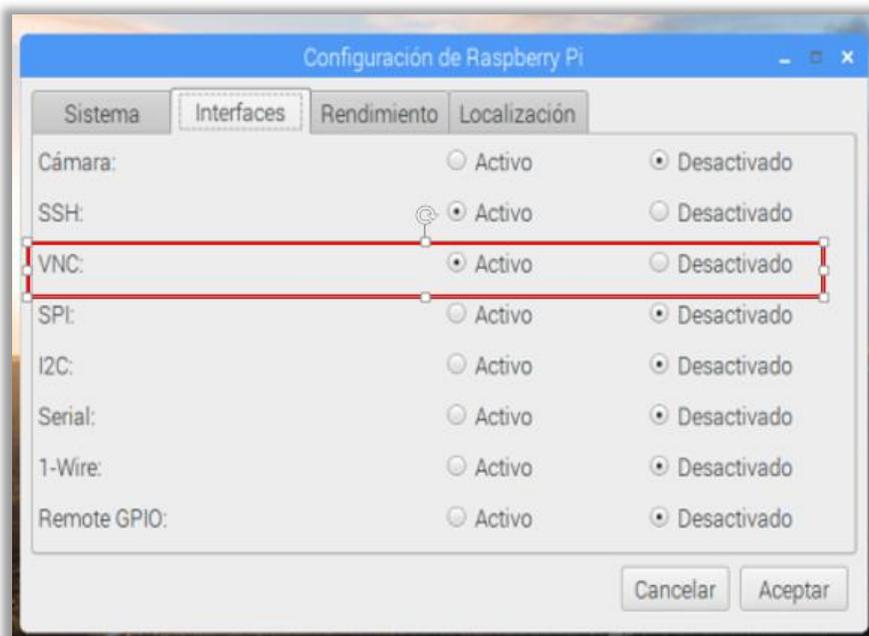


Figura 10 Activar VNC
Fuente: Autores

Paso 4: Se debe entrar a la página oficial del VNC y descargar el archivo con las características permitidas por el computador.

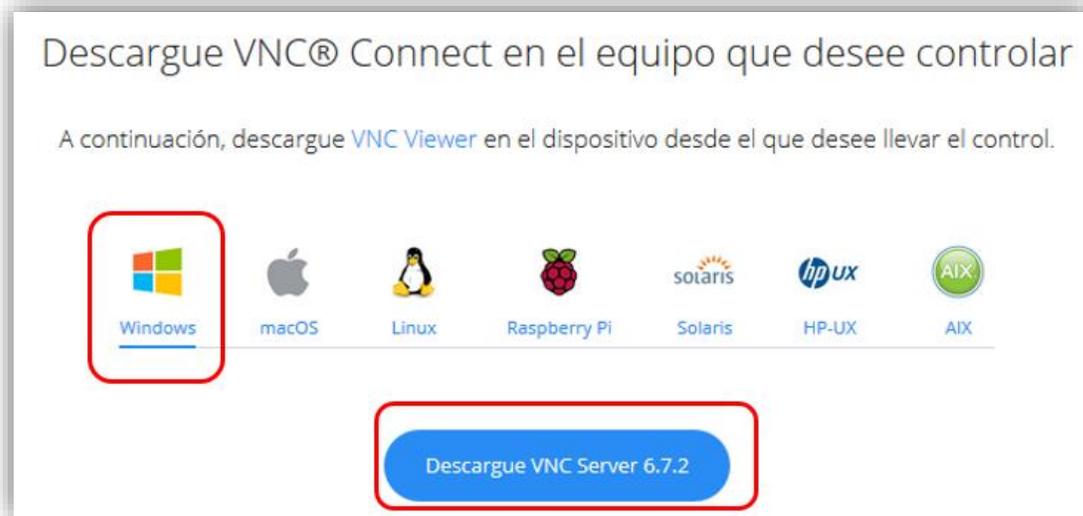


Figura 11 Descargar VNC
Fuente: Autores

Paso 5. Ya instalada la aplicación se debe tener en cuenta la IP local de la Raspberry a continuación se muestra la imagen.

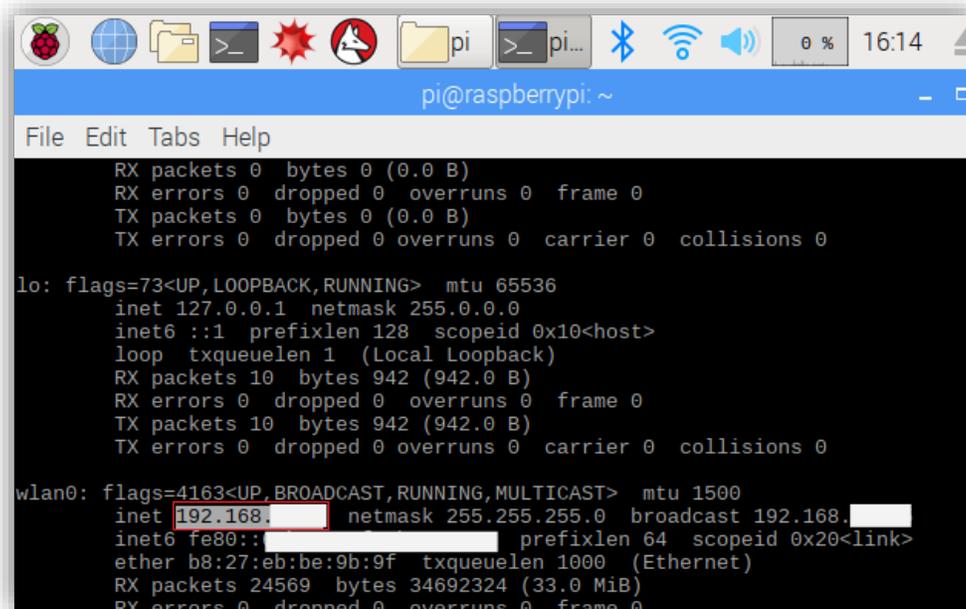


Figura 12 IP de la Raspberry
Fuente: Autores

Paso 6. Cuando se activa el VNC se procede a crear una nueva conexión en él para poder trabajar sobre ella, tal como se muestra en la imagen.

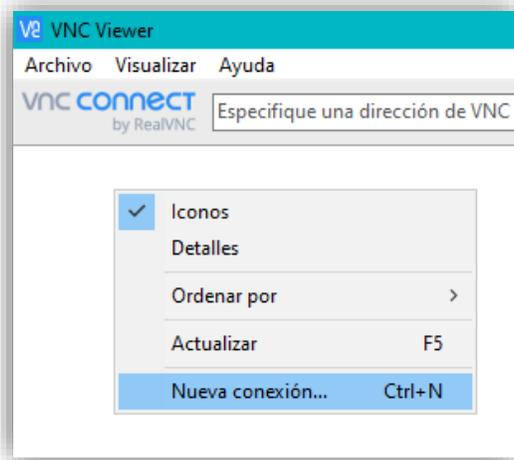


Figura 13 Nueva conexión
Fuente: Autores

Paso 7. Luego de crear la conexión VNC se procede a poner en propiedades la IP de la Raspberry pi en el campo VNC Server, se aceptan cambios y una vez hecho esto se procede a dar clic derecho y conectarse para crear la conexión VNC.

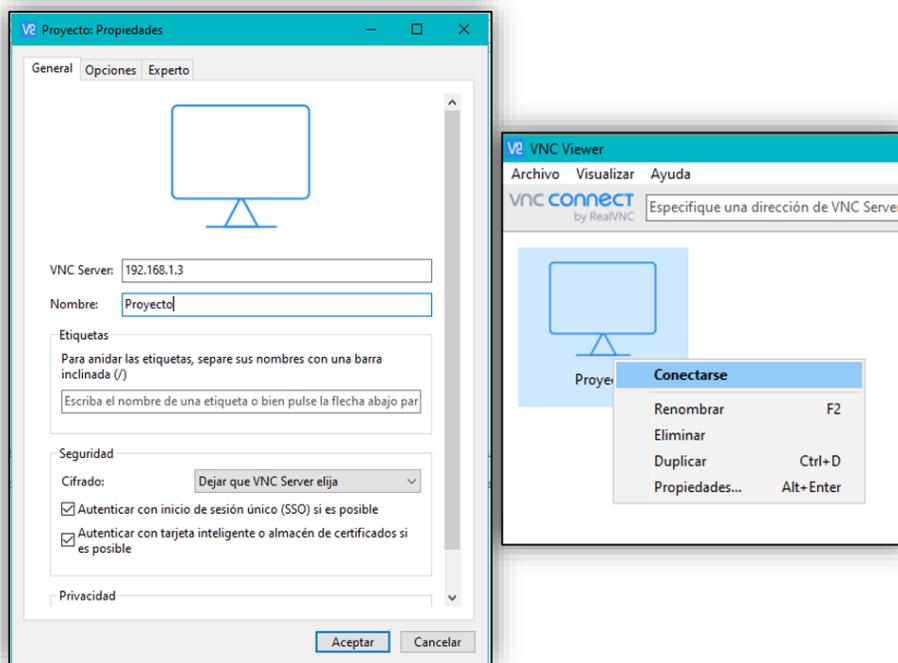


Figura 14 Conectar por VNC
Fuente: Autores

Cambio de contraseña de la Raspberry pi 3

Paso 8: Ya creada la conexión se continúa ingresando el usuario y la contraseña por defecto

- **Usuario:** pi
- **Contraseña:** raspberry

Para evitar que pongan en peligro nuestro sistema se debe cambiar la contraseña, el siguiente paso será ejecutar el comando “**sudo passwd**”, comando para cambiar la contraseña del usuario, este comando pedirá la contraseña actual (defecto) y, después, se introduce la nueva contraseña que se quiere dar al Raspberry Pi.

Ejecutar Sistema

Paso 9: Ya iniciado el sistema operativo en el computador se debe ir a e terminal de Raspberry y escribir los siguientes comandos para ejecutar el sistema que se encuentra en la carpeta llamada proyecto.

```
pi@raspberrypi:~ $ cd Desktop
pi@raspberrypi:~/Desktop $ cd proyecto
pi@raspberrypi:~/Desktop/proyecto $ python3 index.py
```

Figura 15 Ejecutar Sistema
Fuente: Autores

Sistema en Ejecución

Paso 10: Cuando la persona con dicacidad visual toque los diferentes productos de la estantería, en la terminal se va enviar el mensaje de validación que se muestra en imagen.

```
pi@raspberrypi: ~/Desktop/proyecto
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ cd Desktop
pi@raspberrypi:~/Desktop $ cd proyecto
pi@raspberrypi:~/Desktop/proyecto $ python3 index.py
Pin 2 Se eligio el producto
papel
Pin 2 Se esta reproduciendo el Audio
Pin 1 Se eligio el producto
crema
Pin 1 Se esta reproduciendo el Audio
Pin 0 Se eligio el producto
gaseosa
Pin 0 Se esta reproduciendo el Audio
Pin 4 Se eligio el producto
arroz
Pin 4 Se esta reproduciendo el Audio
```

Figura 16 Ejecución del Sistema
Fuente: Autores