



Universidad de Tarapacá
Facultad de Ingeniería
Departamento de Ingeniería en
Computación e Informática

Ingeniería@
Computación e Informática

SISTEMA ASISTENCIAL BASADO EN IOT DE RECONOCIMIENTO DE AGENTES EXTERNOS PARA PERSONAS EN CONDICIÓN DE DISCAPACIDAD VISUAL “NFTOUCH”

Autor(es): Vranika Santiago Yovich
Luciano Vera Norambuena
Bastían Vega Devia
Andrew Campos Seguel

Asignatura: Proyecto II
Profesor: Diego Aracena Pizarro

ARICA, 28 DE NOVIEMBRE 2023

HISTORIAL DE CAMBIOS

Fecha	Versión	Descripción	Autor(es)
07/09/2023	1.0	Versión preliminar del Informe I (planificación)	Andrew Campos Vranika Santiago Luciano Vera Bastián Vega
23/09/2023	1.1	Revisión y modificación del plan	Andrew Campos Vranika Santiago
27/10/2023	1.2	Correcciones formulación del plan	Andrew Campos Vranika Santiago Luciano Vera Bastián Vega
13/11/2023	1.3	Versión preliminar del Informe II (avance)	Vranika Santiago
17/11/2023	1.4	Añadidura de entregables (requerimientos, casos de uso, mockups, arquitectura)	Andrew Campos Vranika Santiago Luciano Vera Bastián Vega
23/11/2023	1.5	Revisión y modificación del avance	Andrew Campos Vranika Santiago
12/12/2023	1.6	Corrección de casos de uso	Andrew Campos
19/12/2023	1.7	Incorporación de plan de integración, modelo de implementación, módulos implementación y reporte de revisión.	Andrew Campos Vranika Santiago Luciano Vera Bastián Vega
19/12/2023	1.8	Resultados de pruebas realiza.	Vranika Santiago Bastián Vega
02/01/2024	1.9	Revisión y modificación final	Andrew Campos Vranika Santiago Luciano Vera Bastián Vega

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS	5
ÍNDICE DE ILUSTRACIONES	5
RESUMEN	6
ABSTRACT	6
1. PANORAMA GENERAL	7
1.1. <i>INTRODUCCIÓN</i>	7
1.2. <i>ESCENARIO DEL PROBLEMA</i>	8
1.3. <i>ESCENARIO DE LA SOLUCIÓN</i>	8
1.4. <i>PROPÓSITO</i>	9
1.5. <i>ALCANCE</i>	9
1.6. <i>OBJETIVOS</i>	9
1.6.1. <i>Objetivo General</i>	9
1.6.2. <i>Objetivos Específicos</i>	9
1.7. <i>SUPOSICIONES</i>	10
1.8. <i>RESTRICCIONES</i>	10
1.9. <i>ENTREGABLES</i>	10
2. ORGANIZACIÓN DEL PERSONAL	11
2.1. <i>PERSONAL Y ENTIDADES INTERNAS</i>	11
2.2. <i>ROLES Y RESPONSABILIDADES</i>	11
2.3. <i>MECANISMOS DE COMUNICACIÓN</i>	12
3. PLANIFICACIÓN DE LOS PROCESOS DE GESTIÓN	13
3.1. <i>PLANIFICACIÓN INICIAL DEL PROYECTO</i>	13
3.1.1. <i>Planificación de estimaciones</i>	13
3.1.1.1. <i>Costos de Hardware</i>	13
3.1.1.2. <i>Costos de Software</i>	13
3.1.2. <i>Planificación de Recursos Humanos</i>	14
3.1.2.1. <i>Costos de RRHH</i>	14
3.1.3. <i>Planificación de costos totales</i>	16
3.2. <i>LISTA DE ACTIVIDADES (CARTA GANTT)</i>	17
3.2.1. <i>Actividades de trabajo</i>	17
3.2.2. <i>Asignación de tiempo</i>	20
3.3. <i>PLANIFICACIÓN DE LA GESTIÓN DE RIESGOS</i>	21
4. PLANIFICACIÓN DE LOS PROCESOS TÉCNICOS	24
4.1. <i>MODELO DE PROCESO</i>	24
4.1.1. <i>Especificación de Requerimientos</i>	24
4.1.2. <i>Descripción de la Arquitectura</i>	25
4.1.3. <i>Diseño de Interface de Usuario</i>	26
4.1.4. <i>Caso de Uso General</i>	28
4.1.5. <i>Casos de Uso de Sistema</i>	29
4.1.5.1. <i>Leer NFC</i>	29
4.1.5.2. <i>Ver detalles de objetos</i>	30
4.1.5.3. <i>Registrar objetos</i>	31
4.1.5.4. <i>Listar objetos</i>	32

4.1.5.5. Eliminar objeto	34
4.1.5.6. Transcribir voz a texto.....	35
4.1.5.7. Reproducir texto a audio.....	36
4.1.6. Diagrama de clases.....	37
4.2. <i>HERRAMIENTAS</i>	38
4.3. <i>TÉCNICA UTILIZADA</i>	39
5. IMPLEMENTACIÓN.....	40
5.1. <i>PLAN DE INTEGRACIÓN</i>	40
5.2. <i>MODELO DE IMPLEMENTACIÓN</i>	41
5.3. <i>MÓDULOS IMPLEMENTADOS</i>	41
5.3.1. Conexión Base de Datos con Aplicación Móvil	41
5.3.2. Identificar tarjetas NFC (Lectura).....	42
5.3.3. Identificar tarjetas NFC (Escritura)	43
5.4. <i>REPORTE DE REVISIÓN</i>	44
5.4.1. Prueba Número Uno	44
5.4.1.1. Descripción.....	44
5.4.1.2. Resultados obtenidos	44
5.4.1.3. Conclusiones.....	44
5.4.2. Prueba Número Dos	44
5.4.2.1. Descripción.....	44
5.4.2.2. Resultados obtenidos	44
5.4.2.3. Conclusiones.....	44
5.4.3. Prueba Número Tres	45
5.4.3.1. Descripción.....	45
5.4.3.2. Resultados obtenidos	45
5.4.3.3. Conclusiones.....	45
6. PROBLEMAS ENCONTRADOS Y SOLUCIONES PROPUESTAS	46
7. CONCLUSIONES.....	47
8. TRABAJO FUTURO	48
REFERENCIAS	49

ÍNDICE DE TABLAS

TABLA 1 ROLES Y RESPONSABILIDADES.....	11
TABLA 2 COSTOS DE HARDWARE.....	13
TABLA 3 COSTOS DE SOFTWARE.....	13
TABLA 4 COSTOS DE RRHH.....	15
TABLA 5 COSTOS TOTALES.....	16
TABLA 6 ACTIVIDADES DE TRABAJO.....	19
TABLA 7 TIPOS DE RIESGOS.....	21
TABLA 8 GESTIÓN DE RIESGOS.....	23
TABLA 9 CUS LEER NFC.....	29
TABLA 10 CUS VER DETALLE DE OBJETOS.....	30
TABLA 11 CUS REGISTRAR OBJETOS.....	31
TABLA 12 CUS LISTAR OBJETOS.....	32
TABLA 13 CUS ELIMINAR OBJETO.....	34
TABLA 14 CUS TRANSCRIBIR VOZ A TEXTO.....	35
TABLA 15 CUS REPRODUCIR TEXTO A VOZ.....	36

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 ESCENARIO DEL PROBLEMA.....	8
ILUSTRACIÓN 2 ESCENARIO DE LA SOLUCIÓN.....	8
ILUSTRACIÓN 3 CARTA GANTT.....	20
ILUSTRACIÓN 4 DESCRIPCIÓN DE ARQUITECTURA.....	25
ILUSTRACIÓN 5 VISTA PRINCIPAL.....	26
ILUSTRACIÓN 6 VISTA REGISTRAR OBJETO.....	26
ILUSTRACIÓN 7 VISTA VER DETALLE.....	27
ILUSTRACIÓN 8 CASO DE USO GENERAL.....	28
ILUSTRACIÓN 9 DIAGRAMA DE SECUENCIA - LEER NFC.....	29
ILUSTRACIÓN 10 DIAGRAMA DE SECUENCIA - VER DETALLE DE OBJETOS.....	30
ILUSTRACIÓN 11 DIAGRAMA DE SECUENCIA - REGISTRAR OBJETOS.....	31
ILUSTRACIÓN 12 DIAGRAMA DE SECUENCIA - LISTAR OBJETOS.....	33
ILUSTRACIÓN 13 DIAGRAMA DE SECUENCIA - ELIMINAR OBJETO.....	34
ILUSTRACIÓN 14 DIAGRAMA DE SECUENCIA - TRANSCRIBIR VOZ A TEXTO.....	35
ILUSTRACIÓN 15 DIAGRAMA DE SECUENCIA - REPRODUCIR TEXTO A VOZ.....	36
ILUSTRACIÓN 16 DIAGRAMA DE CLASES.....	37
ILUSTRACIÓN 17 API SERVICE.....	41
ILUSTRACIÓN 18 BASE DE DATOS CREADA.....	41
ILUSTRACIÓN 19 LECTURA NFC.....	42
ILUSTRACIÓN 20 ESCRITURA NFC.....	43

RESUMEN

Las personas con discapacidad visual severa representan un grupo de población vulnerable, debido a que a menudo utilizan la memoria táctil en su vida diaria para identificar y diferenciar objetos. Sin embargo, esta estrategia puede ser peligrosa en situaciones como el manejo de medicamentos. Para mejorar la autonomía y la confianza de estas personas al interactuar con su entorno, se ha propuesto el desarrollo de una aplicación móvil de fácil uso que emplea tecnología IoT y Raspberry-Pi. Dicha aplicación permite escanear chips NFC. Estos chips contendrán información detallada sobre los objetos escaneados y serán reproducidas por medio del dispositivo móvil a través de una función de lectura de texto a voz. Esta solución tiene como objetivo proporcionar una alternativa segura y efectiva para la identificación de objetos y promover la independencia de las personas con discapacidad visual severa.

PALABRAS CLAVES: *Discapacidad visual, aplicación móvil, tecnología IoT, Raspberry-Pi, tarjetas NFC.*

ABSTRACT

People with severe visual impairment represent a vulnerable population group, as they often rely on tactile memory in their daily lives to identify and differentiate objects. However, this strategy can be dangerous in situations such as medication handling. To improve the autonomy and confidence of these people when interacting with their environment, the development of a user-friendly mobile application using IoT and Raspberry-Pi technology has been proposed. This application allows users to scan NFC chips. These chips will contain detailed information about the scanned objects, which will be reproduced through the mobile device through a text-to-speech function. This solution aims to provide a safe and effective alternative for object identification and to promote the independence of people with severe visual impairments.

KEYWORDS: *Visual impairment, mobile application, IoT technology, Raspberry-Pi, NFC chips.*

1. PANORAMA GENERAL

1.1. INTRODUCCIÓN

Internet de las Cosas, también conocido como IoT por sus siglas en inglés, se caracteriza por ser una evolución tecnológica que permite la interacción y comunicación [1] entre objetos y dispositivos a través de la conexión a Internet.

No obstante, no cualquier objeto que esté conectado de manera inalámbrica a Internet es considerado IoT. De acuerdo con uno de los principales productores mundiales de software (SAP) *IoT se refiere más a cosas u objetos que estén equipados con sensores, software y tecnología que tengan la capacidad de transmitir y recibir datos* [2]. Estos sensores recopilan datos del entorno del objeto que luego son procesados por el software y enviados a través de Internet.

En este contexto tecnológico, surge la siguiente interrogante: ¿cómo se puede aprovechar el IoT para mejorar la calidad de vida de las personas? Una de las posibles respuestas es que el IoT tiene la funcionalidad de ayudar a incrementar la independencia y autonomía de las personas. Un grupo que se vería beneficiado, especialmente cuando ingresan agentes externos en el entorno hogareño, sería el de los no videntes.

Según la Organización Mundial de la Salud (OMS), hay alrededor de 45 millones de personas ciegas en el mundo. En Chile, en tanto, aproximadamente 850 mil personas tienen algún tipo de discapacidad visual y alrededor de 84 mil tienen ceguera total [3]. Es decir, la ceguera afecta a un total del 4% de la población chilena. Al pertenecer a una comunidad reducida, las grandes empresas y productores no tienen como prioridad atender a sus necesidades específicas.

Este proyecto, denominado “NFTouch”, nace para facilitar las tareas diarias de las personas con discapacidad visual. Además, potenciará su autonomía con el uso de herramientas tecnológicas como chips NFC y Raspberry-Pi. La aplicación está dirigida a personas que formen parte de la comunidad de invidentes y que dispongan de un dispositivo móvil. Es así como, sensores integrados en objetos que puedan ser utilizados en su vida diaria podrían proporcionar información relevante sobre el entorno en el que conviven. El escaneo de chip NFC (Near Field Communication) en objetos, como en medicamentos o prendas de vestir.

1.2. ESCENARIO DEL PROBLEMA



Ilustración 1 Escenario del Problema.

Las personas que padecen una discapacidad visual enfrentan desafíos para reconocer agentes externos que ingresan a sus hogares y a su vez, los cambios que son producidos por estos mismos. Esto es debido a las formas, tamaños y texturas similares que pueden tener. Por ejemplo, en el caso de los fármacos, el individuo podría tomar un medicamento con un envase parecido al que inicialmente deseaba consumir, causándole efectos no deseados. Por consecuencia, se dificulta la realización de tareas cotidianas reduciendo su autonomía e independencia.

1.3. ESCENARIO DE LA SOLUCIÓN

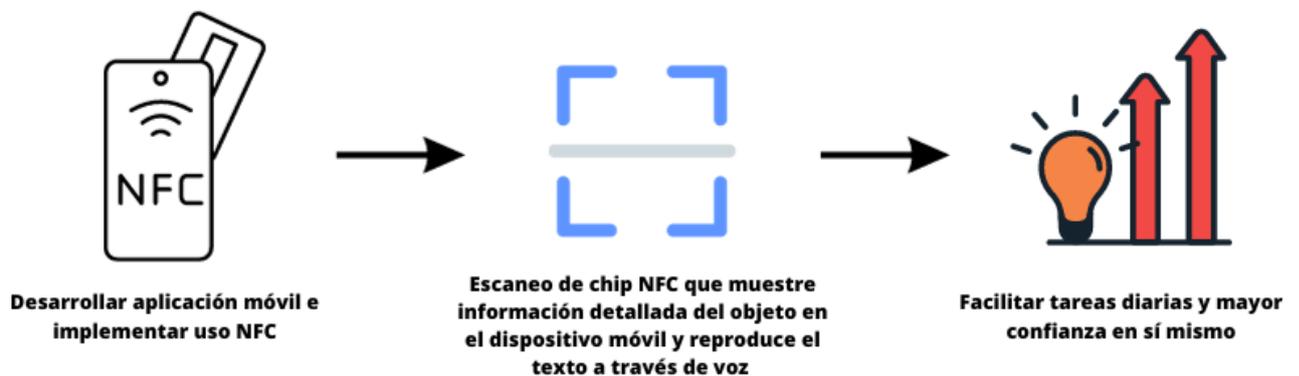


Ilustración 2 Escenario de la Solución.

Para solucionar el problema establecido en el punto anterior ([1.2. Escenario del Problema](#)) se ha propuesto desarrollar una aplicación móvil. Su función principal sería la de escanear el chip NFC, para luego mostrar la información detallada del objeto. Esta información se agregaría mediante la voz y se guardaría en una Raspberry-Pi. Primero, la Raspberry-Pi organizará la información recibida en una estructura de datos en donde cada instancia de dicha estructura contendrá un identificador único. Luego, el chip NFC almacenará el identificador previamente mencionado. Al momento de escanear el chip NFC, este llevará hacia la instancia específica con el identificador único que tenga en la Raspberry-Pi y retornará toda la información almacenada en la estructura de datos utilizada. Cabe agregar que la información sería reproducida a través de voz por el micrófono del dispositivo, para así proveer de una mayor

independencia al invidente al momento de identificar objetos que le supongan mayor dificultad de reconocimiento.

1.4. PROPÓSITO

El propósito será mejorar la calidad de vida de las personas con discapacidad visual al facilitarles el reconocimiento y detección de objetos, además se les brinda una mayor independencia.

1.5. ALCANCE

El alcance del proyecto "NFTouch" se centra en diseñar e implementar una aplicación móvil orientada a personas con discapacidad visual severa para mejorar su autonomía. La aplicación permitirá a los usuarios escanear chips NFC con su dispositivo móvil y acceder a información detallada sobre objetos previamente introducidos en la base de datos. Esta información se reproducirá a través de voz utilizando la función de lectura de texto a voz del dispositivo.

1.6. OBJETIVOS

1.6.1. Objetivo General

El objetivo general es desarrollar una aplicación móvil asistencial basada en IoT para personas con discapacidad visual que les permita reconocer y detectar agentes externos en el entorno hogareño.

1.6.2. Objetivos Específicos

- i. Investigar las necesidades de las personas con discapacidad visual en cuanto a funciones que necesitan acceder en sus dispositivos móviles.
- ii. Comprender los desafíos que enfrentan las personas con discapacidad visual al utilizar dispositivos móviles.
- iii. Desarrollar un prototipo de la aplicación móvil que sea cómoda y de fácil uso utilizando Ionic como framework.
- iv. Evaluar el prototipo de la aplicación móvil con personas con discapacidad visual para obtener comentarios sobre la funcionalidad, facilidad de uso y accesibilidad.

1.7. SUPOSICIONES

- i. Se asume que los usuarios que utilicen la aplicación móvil padecen de alguna discapacidad visual.
- ii. Se asume que los usuarios tienen acceso a un dispositivo móvil (smartphone) que cuente con un sistema operativo Android.
- iii. Se asume que los usuarios tienen acceso a Internet.
- iv. Se asume que los dispositivos móviles que los usuarios utilizarán son compatibles con el escáner NFC.
- v. Se asume que por ser un dispositivo con el sistema operativo Android, el usuario tendrá activada la función de Google Talk-Back.

1.8. RESTRICCIONES

Existe un conjunto de limitaciones para que el desarrollo de NFTouch sea exitoso, las cuales se pueden observar en la siguiente lista:

- **Raspberry-Pi:** conocimientos del funcionamiento de Raspberry-Pi.
- **Temática:** debe ser apto para personas con discapacidad móvil o física.
- **Control:** el manejo de la aplicación debe ser vía smartphone.
- **Maqueta:** diseñar y construir una maqueta con materiales reciclables (en su mayoría).
- **Sistema operativo:** debe ser Android.

1.9. ENTREGABLES

- Informe y presentación de la planificación del proyecto.
- Informe y presentación del avance del proyecto.
- Informe y presentación final del proyecto.
- Maqueta.
- Bitácora semanal.
- Wiki y manual de usuario.
- Producto final.

2. ORGANIZACIÓN DEL PERSONAL

2.1. PERSONAL Y ENTIDADES INTERNAS

A cada integrante del grupo se le designó una responsabilidad, donde estarán a cargo de cumplir en el tiempo estimado. Cabe recalcar que por mucho que existan encargados para cada eje, todos los integrantes deben velar por la realización tanto de sus tareas como la de sus compañeros.

- **Jefe de grupo:** es el líder del equipo, responsable de la supervisión, dirección y toma de decisiones estratégicas.
- **Analista:** este rol implica entender los requisitos del negocio, analizarlos y convertirlos en especificaciones técnicas para el equipo de desarrollo. También puede implicar pruebas y validación para asegurar que el producto final cumpla con los requisitos.
- **Desarrollador Backend:** se encarga de la lógica del negocio, las operaciones de la base de datos, la implementación del servidor y la integración con el frontend.
- **Desarrollador Frontend:** se encarga de la interfaz de usuario y la experiencia del usuario. Trabaja en el diseño y la interactividad de la aplicación móvil.

2.2. ROLES Y RESPONSABILIDADES

Rol	Responsable	Involucrados
Jefe de proyecto	Vranika Santiago	Vranika Santiago y Andrew Campos
Analista	Andrew Campos	Vranika Santiago, Luciano Vera, Bastián Vega y Andrew Campos
Desarrollador Backend	Bastián Vega	Andrew Campos y Bastián Vega
Desarrollador Frontend	Luciano Vera	Luciano Vera, Vranika Santiago y Bastián Vega

Tabla 1 Roles y responsabilidades.

2.3. MECANISMOS DE COMUNICACIÓN

Para garantizar una buena organización y comunicación, se han establecido las siguientes herramientas y plataformas:

- **WhatsApp:** Se utilizará para programar y anunciar horarios de reuniones, llevar un registro de la asistencia diaria de los integrantes y compartir archivos. Se estableció un horario pertinente para compartir mensajes, el cual es de lunes a viernes desde las 8:00 hasta las 18:00.
- **Discord:** Se utilizará como un espacio de discusión y toma de decisiones para las propuestas de mejora del proyecto.
- **Correos electrónicos:**
 - Vranika Santiago Yovich – vranika.santiago.yovich@alumnos.uta.cl
 - Luciano Vera Norambuena – luciano.vera.norambuena@alumnos.uta.cl
 - Bastián Vega Devia – bastian.vega.devia@alumnos.uta.cl
 - Andrew Campos Seguel – andrew.campos.seguel@alumnos.uta.cl

3. PLANIFICACIÓN DE LOS PROCESOS DE GESTIÓN

3.1. PLANIFICACIÓN INICIAL DEL PROYECTO

En la parte de hardware, los productos a utilizar fueron los siguientes:

- Arriendo de equipos
- Samsung Galaxy A53
- Raspberry-Pi3 model B
- Tarjeta NFC NTAG215

Mientras que para la parte de software:

- Licencia Microsoft Office

3.1.1. Planificación de estimaciones

A continuación, se especificarán los costos estimados (en CLP) tanto para hardware como para software. Además, de los valores de cada rol.

3.1.1.1. Costos de Hardware

Arriendos de equipo				
Nombre producto	Cantidad	Arriendo por mes	Tiempo en meses	Costo total
Equipos personales	4	\$80.000	4	\$320.000
Hardware general				
Nombre producto	Costo unidad	Cantidad	Costo total	
Samsung Galaxy A53	\$400.000	1	\$400.000	
Raspberry-Pi3 model B	\$60.000	1	\$60.000	
Tarjeta NFC NTAG215	\$1.590	10	\$15.000	
			Total	\$796.000

Tabla 2 Costos de Hardware.

3.1.1.2. Costos de Software

Nombre producto	Costo unidad	Cantidad	Costo total
Licencia Microsoft Office	\$5306	4	\$21.224
Total por 4 meses			\$84.897

Tabla 3 Costos de Software.

3.1.2. Planificación de Recursos Humanos

3.1.2.1. Costos de RRHH

Puntos para tomar en consideración:

- La fecha final estimada del proyecto es el 6 de diciembre del año 2023, es decir, en total son 16 semanas de trabajo (equivalentes a 4 meses). Considerando, que por semana son 6 horas de trabajo en clases y una estimación de 6 horas de trabajo autónomo, se obtiene un total de 192 horas totales.
- El valor hora para cada rol es un promedio de la información encontrada en Internet. Hay que destacar que cada responsable gana un 10% extra (*).
- Para calcular el valor total mensual de cada encargado, se debe multiplicar el valor hora por las horas trabajadas mensualmente, que en este caso son 48 (se dividen las horas totales por los meses trabajados). En caso de que sea responsable, se le agrega el 10% nombrado anteriormente. Para más detalles véase [tabla 4: Costos de RRHH](#).

Trabajador	Rol	Valor por hora	Horas mensuales totales (48)	Costo total por mes
Vranika Santiago	Jefe proyecto ¹ (*)	\$10.154	8	\$81.232
	Analista ²	\$5.538	7	\$38.766
	Frontend ³	\$8.000	33	\$264.000
Luciano Vera	Analista	\$5.538	7	\$38.766
	Frontend (*)	\$8.800	41	\$360.800
Andrew Campos	Analista (*)	\$6.092	7	\$42.644
	Sub-jefe proyecto	\$9.231	8	\$73.848
	Backend ⁴	\$8.615	33	\$284.295
Bastían Vega	Analista	\$5.538	7	\$38.766
	Frontend	\$8.000	17	\$136.000
	Backend (*)	\$9.476	24	\$227.424
Total por 1 mes				\$1.586.541
Total por 4 meses				\$6.346.164

Tabla 4 Costos de RRHH.

¹ Talent (s.f.). Salario para Jefe De Proyecto en Chile - Salario Medio. Accedido el 24 de septiembre de 2023.
<https://cl.talent.com/salary?job=Jefe+de+proyecto>

² Talent (s.f.). Salario para Analista Programador en Chile - Salario Medio. Accedido el 24 de septiembre de 2023.
<https://cl.talent.com/salary?job=Analista+programador>

³ Talent (s.f.). Salario para Desarrollador Frontend en Chile - Salario Medio. Accedido el 24 de septiembre de 2023.
<https://cl.talent.com/salary?job=Desarrollador+frontend>

⁴ Talent (s.f.). Salario para Desarrollador Backend en Chile - Salario Medio. Accedido el 24 de septiembre de 2023.
<https://cl.talent.com/salary?job=Desarrollador+backend>

3.1.3. Planificación de costos totales

Tipo coste	Costo
Hardware	\$796.000
Software	\$84.896
Recursos humanos + holgura (30%)	\$8.250.013
Costo total del proyecto	\$9.130.909

Tabla 5 Costos totales.

Por lo tanto, la estimación del proyecto durante 16 semanas (o 4 meses) de trabajo tendrá un presupuesto total de \$9.130.909

3.2. LISTA DE ACTIVIDADES (CARTA GANTT)

3.2.1. Actividades de trabajo

Actividad	Descripción	Responsable
Bitácoras	Registro de todas las actividades que se desarrollan semanalmente.	Vranika Santiago
Wiki	Se capturan y comparten ideas e información del proyecto.	Vranika Santiago
Organización	Designación de la actividad que estará encargado cada integrante.	Vranika Santiago
Establecer problemática	Identificar y definir es el problema que el proyecto busca abordar o resolver.	Vranika Santiago
Analizar diversas soluciones	Evaluación de diversas alternativas para resolver la problemática identificada.	Vranika Santiago
Presentación idea	Presentación acerca de la problemática escogida y su respectiva solución.	Luciano Vera
Diseñar maqueta	Realizar bosquejo, decidir los materiales reciclables a utilizar y justificación de la maqueta.	Luciano Vera
Construcción maqueta	Armado de maqueta con materiales reciclables (en su mayoría).	Luciano Vera
Presentación maqueta	Se presenta el desarrollo, la justificación, escenarios experimentales y su resultado.	Bastián Vega
Informe I	Crear informe para la parte I: formulación del proyecto.	Andrew Campos
Análisis de framework	Analizar lenguajes de programación para utilizar en la aplicación e investigar los distintos framework que son útiles	Bastián Vega

	para el desarrollo de aplicaciones Android.	
Estimar costos	Calcular el presupuesto total del proyecto.	Bastián Vega
Presentación I	Elaborar presentación de la formulación del proyecto y luego exponerlo.	Luciano Vera
Realizar Casos de Uso	Realizar diagramas de casos de uso de sistema y casos de uso de interfaz.	Andrew Campos
Creación de mockups	Material que simula el producto final tanto en lo estético como en lo funcional.	Luciano Vera
Definir técnica de prueba	Mecanismo que se utiliza para encontrar defectos a través de pruebas.	Vranika Santiago
Establecer requerimientos	Definir requerimientos no funcionales y funcionales.	Bastián Vega
Establecer arquitectura	Diagrama para visualizar de mejor manera la arquitectura de los sistemas utilizados.	Bastián Vega
Informe II	Redacción del informe del avance del proyecto.	Vranika Santiago
Presentación II	Creación del material de apoyo sobre el avance del proyecto.	Luciano Vera
Diseñar fronted	Creación de la interfaz de usuario y la experiencia del usuario.	Luciano Vera
Diseñar backend	Proporcionar los servicios y las funcionalidades necesarias para que la aplicación móvil funcione correctamente.	Andrew Campos

Codificar reconocimiento de voz a texto	-	Vranika Santiago
Depuración de códigos	Proceso en el cual se identificarán y corregirán errores en el algoritmo.	Bastián Vega
Pruebas de funcionamiento	Chequear que la aplicación móvil este cumpliendo su propósito.	Bastián Vega
Redactar manual de usuario	Creación de material para facilitar el uso de la interfaz y robot.	Andrew Campos
Confeccionar póster	Apoyo visual sobre las características del proyecto.	Vranika Santiago
Informe III	Redacción del informe final.	Andrew Campos
Presentación III	Creación del material para la presentación final.	Luciano Vera

Tabla 6 Actividades de trabajo.

3.2.2. Asignación de tiempo

La Carta Gantt fue creada con el propósito de planificar y gestionar de manera más eficaz las actividades a lo largo del semestre. Además, se puede registrar el tiempo dedicado para cada tarea y compararlo con el tiempo estimado.

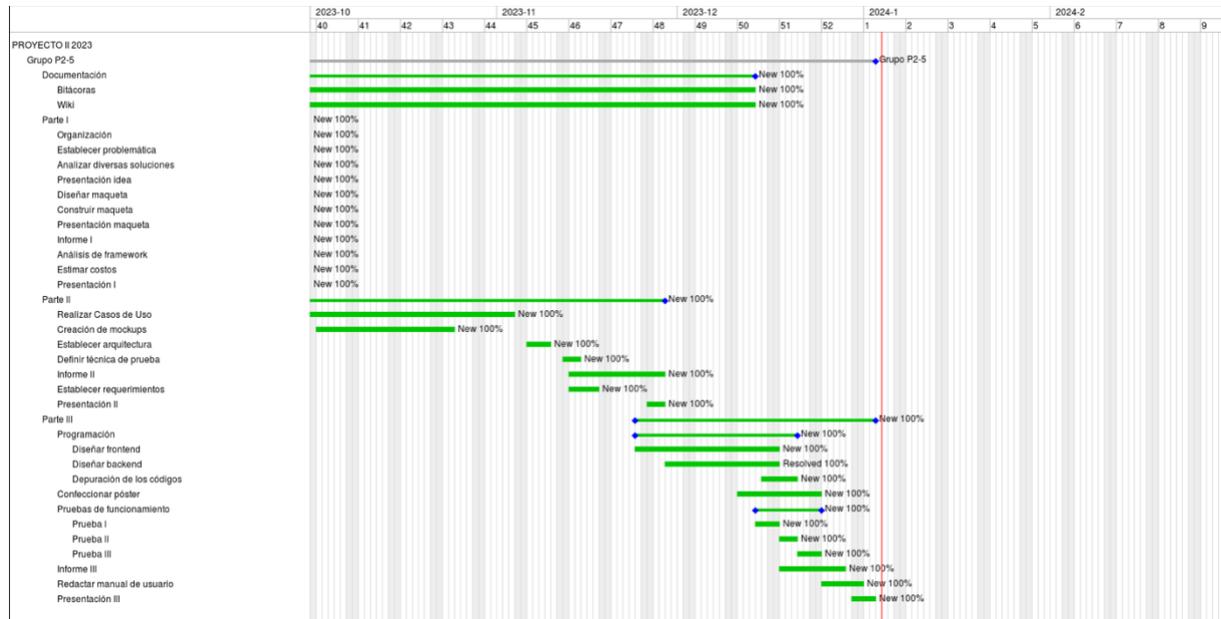


Ilustración 3 Carta Gantt.

El proyecto se dividirá en 3 fases, las cuales son:

- **Parte I:** planificación del proyecto, que está comprendida del 21/08/2023 hasta el 25/09/2023.
- **Parte II:** ejecución del proyecto, que está comprendida del 27/09/2023 hasta el 28/11/2023.
- **Parte III:** cierre del proyecto, que está comprendida del 29/11/2023 hasta el 03/01/2024.

3.3. PLANIFICACIÓN DE LA GESTIÓN DE RIESGOS

Para la gestión de riesgos, primeramente, se categorizaron los tipos de riesgos y sus factores:

Tipo de Riesgo	
Tecnológico	Entrega retrasada o problemas relacionados con el hardware o de la ayuda del software, muchos problemas tecnológicos reportados.
Humano	Baja moral del personal, mala salud, malas relaciones entre los miembros del equipo, disponibilidad de empleo.
Herramientas	Rechazo de los miembros del equipo para utilizar herramientas, quejas acerca de las herramientas CASE, peticiones de estaciones de trabajo más potentes.
Requerimientos	Peticiones de muchos cambios en los requerimientos, quejas del cliente.
Organizacional	Chismorreos organizacionales, falta de acciones por el administrador principal.
Estimación	Fracaso en el cumplimiento de los tiempos acordados y en la eliminación de defectos reportados.

Tabla 7 Tipos de Riesgos.

Mientras que para los riesgos latentes que se podrían presentar en el proyecto, se clasificaron en 4 niveles de impacto:

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

Riesgo	Tipo	Probabilidad de Ocurrencia	Nivel de Impacto	Acción Remedial
Falta de conocimiento y comprensión de la metodología de trabajo por parte del cliente	Herramientas	40%	Catastrófico	Entregar documentación temática y precisa sobre la metodología en cascada.
Presupuesto afectado por el entorno económico	Estimación	20%	Catastrófico	Solicitar una reunión con el cliente en la que se discuta la situación económica del proyecto y se llegue a un acuerdo en cuanto a la necesidad de un mayor apoyo financiero.
Daño o pérdida en la tarjeta SD	Tecnológico	25%	Catastrófico	Comprar otra tarjeta SD y respaldar la información regularmente.
Detenimientos de las actividades por recesos o paros	Humano	5%	Crítico	Reprogramación de actividades.
Quejas del cliente	Requerimientos	30%	Catastrófico	Trabajar con el cliente para conocer su descontento y encontrar una solución.
Falta de comunicación entre las áreas de trabajo	Organizacional	20%	Crítico	Realizar reuniones periódicas para mantener al equipo informado sobre las actualizaciones en la carta Gantt y la asignación de nuevas tareas.

Atraso en actividades de la Carta Gantt	Estimación	10%	Crítico	Agrupar las tareas atrasadas y darle prioridad por su relevancia en el desarrollo de nuevas tareas.
Enfermedades del personal	Humano	10%	Marginal	Un involucrado se encargará temporalmente de las tareas del personal faltante.
Daños físicos en la Raspberry-Pi3	Tecnológico	20%	Catastrófico	Reemplazar Raspberry-Pi3
Pérdida de avances	Tecnológico	10%	Catastrófico	Realizar respaldos frecuentemente del avance del desarrollo del software.

Tabla 8 Gestión de Riesgos.

4. PLANIFICACIÓN DE LOS PROCESOS TÉCNICOS

4.1. MODELO DE PROCESO

4.1.1. Especificación de Requerimientos

Requerimientos funcionales

1. El sistema recibe y muestra la información de un chip NFC.
2. El sistema está comunicado a la base de datos (Raspberry).
3. El sistema es capaz de realizar operaciones DML en la base de datos.
4. El sistema es capaz de asignar y retornar una ID a cada instancia automáticamente.
5. El sistema es capaz de reproducir mensajes de texto a voz.

Requerimientos no funcionales

1. La aplicación debe estar programada en JavaScript ES12.
2. La aplicación debe ser hecha con el framework Ionic.
3. La base de datos debe ser programada con Python 3.12.0.
4. El sistema operativo de la Raspberry debe ser Raspbian.
5. La aplicación debe reconocer el texto correctamente para reproducirlo en voz.
6. La aplicación debe ser instalada en un dispositivo Android.
7. La aplicación debe ser de fácil acceso y manipulación para el usuario.
8. El dispositivo para la aplicación debe tener activada la opción de Google Talk-Back.
9. La base de datos debe ser hecha con SQL.
10. La base de datos debe estar disponible en una red local.

4.1.2. Descripción de la Arquitectura

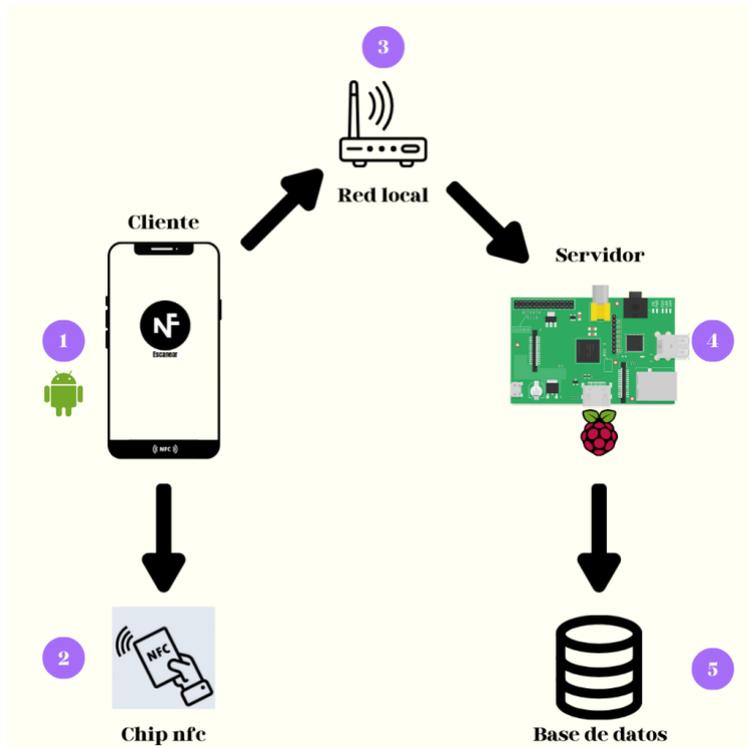


Ilustración 4 Descripción de Arquitectura.

1. Software (cliente) diseñado para el usuario con discapacidad visual que permitirá acceder al sensor NFC integrado en el dispositivo, con el que se podrá escanear la tarjeta nfc.
2. Tarjeta NFC NTAG215 asignada a un objeto, contendrá el identificador del objeto almacenado a la base de datos.
3. Red local WIFI que permitirá a la aplicación realizar solicitudes HTTP al servidor (Raspberry-Pi3) para recibir o enviar información.
4. Servidor alojado en una Raspberry-Pi3. Maneja solicitudes HTTP para recibir o emitir datos, y controla la base de datos.
5. Base de datos dentro de la Raspberry-Pi3 que almacena las instancias de los objetos.

4.1.3. Diseño de Interface de Usuario



Al momento de iniciar la aplicación móvil (pantalla “Principal”), aparece un botón en medio de la pantalla que dice “Escanear” (1), el cual se debe presionar para escanear un objeto ya registrado (*Ilustración 5*).

Ilustración 5 Vista Principal.



En la pantalla “Registrar objeto”, primeramente, se tiene una pantalla con las etiquetas “Título” (1), la cual en una caja de entrada de texto se ingresa el título del objeto. En segundo lugar, está la etiqueta “Descripción” (2), donde se ingresa la descripción del objeto a agregar. Por último, se tiene una etiqueta denominada “Tipo” (3), donde en un combo box se presentan diferentes opciones para clasificar al objeto. Cada información por agregar es comunicada por el usuario mediante su voz y para ser guardada, se encuentra el botón “Escanear” (4) (*Ilustración 6*).

Ilustración 6 Vista Registrar objeto.



En la pantalla “Lista de objetos”, por medio de la reproducción de audio, cada objeto registrado se listará por su título. Por cada objeto, se ofrecerán dos opciones: Eliminar (1) o Ver detalle (2), las cuales deben ser comunicadas al dispositivo móvil mediante la voz del usuario.

En caso de ser escogida la opción “Eliminar [título del objeto]”, el objeto será removido de la lista. De lo contrario, si es escogida la opción “Ver detalle” (véase *Ilustración 7*).

En caso contrario, si es escogida la opción “Ver detalle [título del objeto]”, mediante la reproducción de audio, se indicará la descripción y el tipo de objeto.



Ilustración 7 Vista Ver detalle.

4.1.4. Caso de Uso General

En el caso de este trabajo, el enfoque se encuentra dirigido en crear diagramas secuenciales para visualizar las interacciones entre los actores y el sistema (que comprende la aplicación móvil, la Raspberry Pi y la base de datos) en diferentes escenarios en base a los casos de usos definidos previamente. Estos diagramas proporcionarán una perspectiva detallada de las acciones y respuestas del sistema, lo que facilitará la identificación de puntos clave de interacción y flujos de información.

Asimismo, se emplearán los diagramas de clases para representar la estructura estática del sistema. Estos diagramas funcionarán como una herramienta para facilitar la visualización de las clases, sus atributos y las relaciones entre ellas.

A continuación, se muestra el caso de uso general donde se puede apreciar la interacción del usuario con discapacidad visual con el sistema.

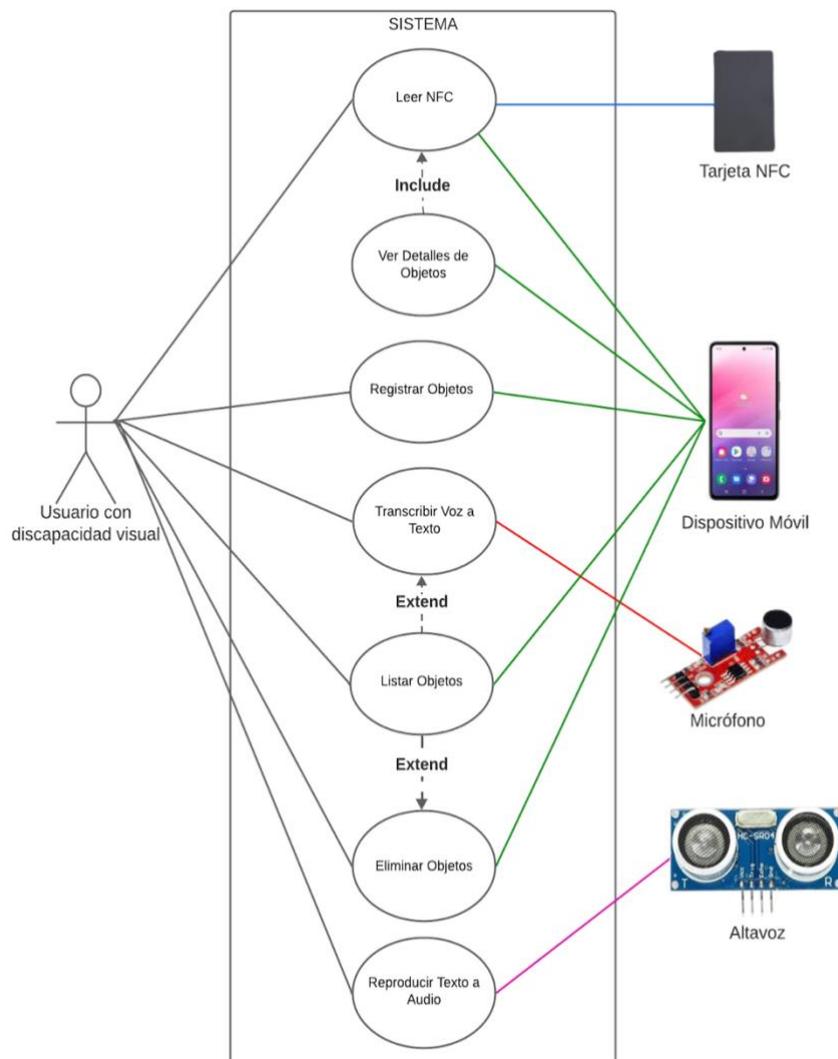


Ilustración 8 Caso de Uso General.

4.1.5. Casos de Uso de Sistema

4.1.5.1. Leer NFC

Autor / Fecha: Andrew Campos / 20 - 12 - 2023	
Descripción: El usuario comunica la tarjeta NFC (iniciador; pasivo) con el dispositivo móvil (receptor; activo) para saber qué ID de objeto contiene.	
Actor: Usuario con discapacidad visual, Tarjeta NFC, Dispositivo Móvil	
Precondición: El dispositivo móvil debe tener la opción activada "Pagos NFC y Sin contacto" en sus ajustes.	
Flujo Principal: Usuario 1. Acerca la tarjeta NFC al dispositivo móvil para establecer la comunicación.	Flujo Principal: Sistema 2. El sistema obtiene la información (ID) proveniente de la tarjeta NFC.
Postcondiciones: No posee.	

Tabla 9 CUS Leer NFC.

Junto a su respectivo diagrama de secuencia:

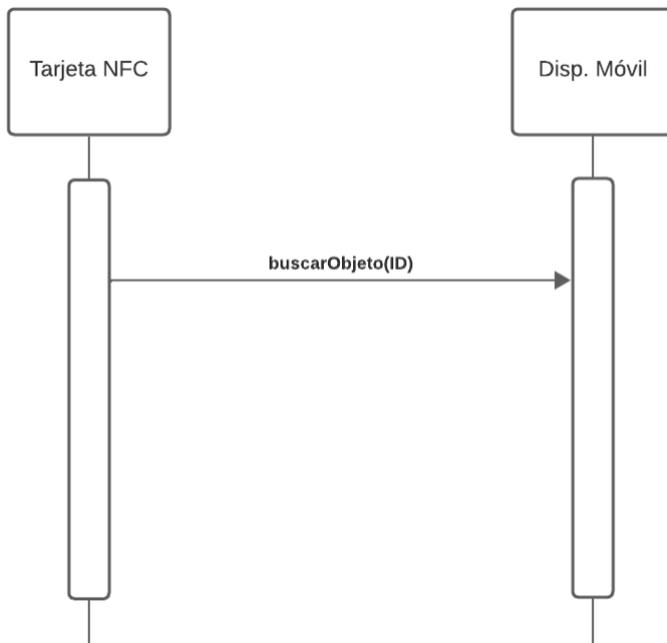


Ilustración 9 Diagrama de Secuencia - Leer NFC.

4.1.5.2. Ver detalles de objetos

Autor / Fecha: Andrew Campos / 22 - 12 - 2023	
Descripción: Posteriormente a leer una Tarjeta NFC a través del Dispositivo Móvil, la información que recibe el sistema (ID) la procesa con la Raspberry Pi para obtener la información de ese objeto.	
Actor: Dispositivo móvil	
Precondición: El Dispositivo Móvil debe estar conectado a la misma red wifi que la Raspberry Pi.	
Flujo Principal: Dispositivo Móvil, Altavoz 1. << Incluye el CUS "Leer NFC" >> 2. Envía el ID del objeto.	Flujo Principal: Sistema 3. Muestra de ese ID en específico: <ul style="list-style-type: none"> • Título • Descripción • Categoría
Postcondiciones: No posee.	

Tabla 10 CUS Ver detalle de objetos.

Junto a su respectivo diagrama de secuencia:

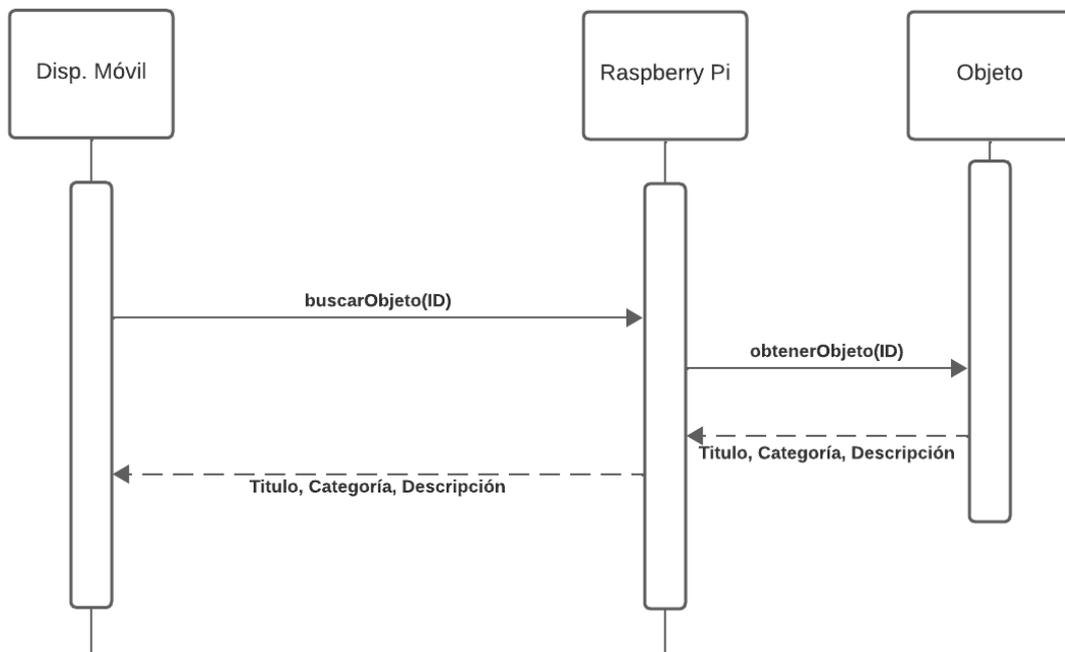


Ilustración 10 Diagrama de Secuencia - Ver detalle de objetos.

4.1.5.3. Registrar objetos

Autor / Fecha: Andrew Campos / 20 - 12 - 2023	
Descripción: El Usuario con discapacidad visual crea un nuevo objeto en la Raspberry Pi a través del Dispositivo Móvil.	
Actor: Usuario con discapacidad visual, Dispositivo móvil	
Precondición: El Dispositivo Móvil debe estar conectado inalámbricamente a la Raspberry Pi.	
Flujo Principal: Rellenar elemento NFC 1. Ingresar a la vista "Registrar objetos" 3. Ingresar los datos solicitados.	Flujo Principal: Sistema 2. Solicita los siguientes datos del objeto: <ul style="list-style-type: none"> • Título • Categoría • Descripción 4. Indica que el objeto se ha creado exitosamente, se genera y muestra el ID generado para el objeto. 5. Agrega el ID al chip NFC.
Flujo alternativo: 3.1. << Extiende CUS Transcribir voz a texto >>	
Postcondiciones: Se registra un nuevo objeto en la Raspberry Pi y retorna su ID al Dispositivo Móvil.	

Tabla 11 CUS Registrar objetos.

Junto a su respectivo diagrama de secuencia:

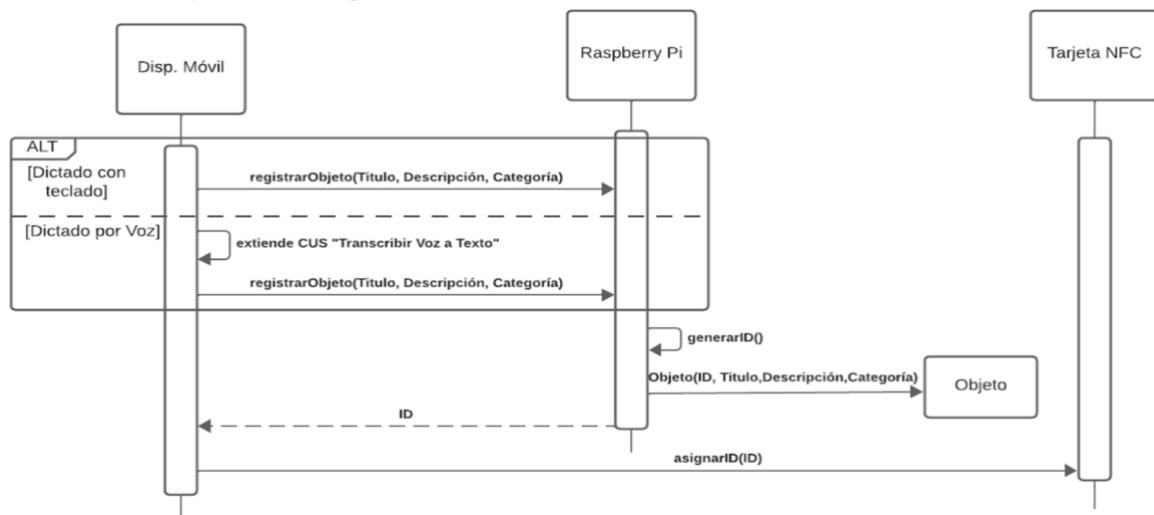


Ilustración 11 Diagrama de Secuencia - Registrar objetos.

4.1.5.4. Listar objetos

Autor / Fecha: Andrew Campos / 20 - 12 - 2023	
Descripción: Se listan en el Dispositivo Móvil todos los objetos de la base de datos que está en la Raspberry Pi.	
Actor: Usuario con discapacidad visual, Dispositivo Móvil, Raspberry Pi.	
Precondición: El Dispositivo Móvil debe estar conectado a la misma red wifi que la Raspberry Pi.	
Flujo Principal: Usuario 1. Ingresar a la vista "Listar objetos"	Flujo Principal: Sistema 2. Indica por cada objeto en la Raspberry Pi: <ul style="list-style-type: none"> • Título • Opción "Ver detalles" • Opción "Eliminar"
Flujo Alternativo: "Ver detalles" 2.1. Selecciona la opción "Ver detalles" de un objeto en específico.	Flujo Alternativo: 2.1.1 Muestra la siguiente información de la instancia: <ul style="list-style-type: none"> • Categoría • Descripción
Flujo Alternativo: "Eliminar" 2.2. Selecciona la opción "Eliminar" de un objeto en específico.	2.2.1. Obtiene el ID del objeto el cual se desea eliminar. 2.2.1. << Extiende el CUS "Eliminar objeto" >>
Postcondición: No posee.	

Tabla 12 CUS Listar objetos.

Junto a su respectivo diagrama de secuencia:

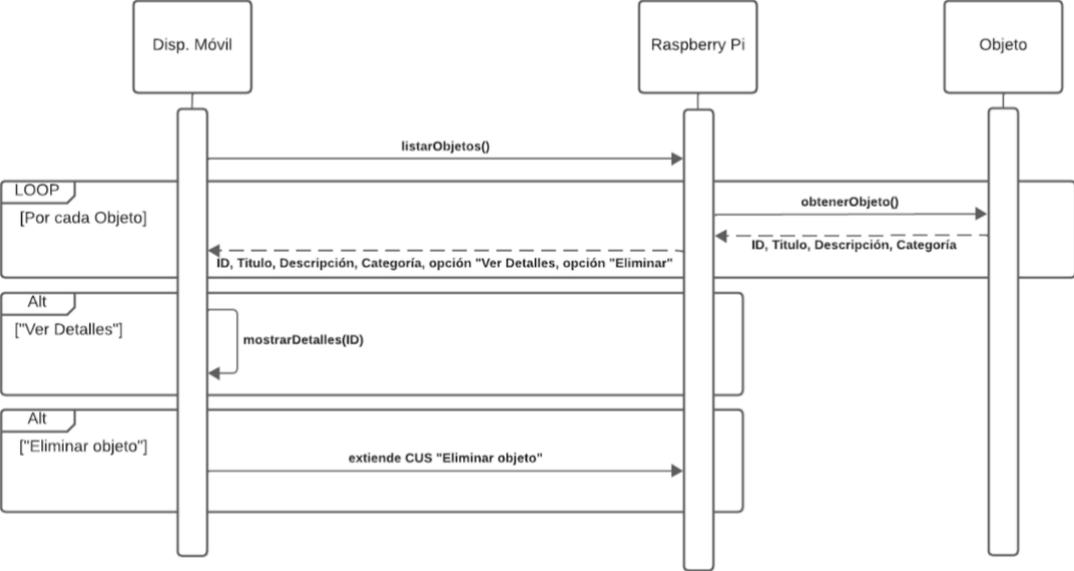


Ilustración 12 Diagrama de Secuencia - Listar objetos.

4.1.5.5. Eliminar objeto

Autor / Fecha: Andrew Campos / 22 - 12 - 2023	
Descripción: El Usuario con discapacidad visual a través del Dispositivo Móvil elimina un objeto existente en la Raspberry Pi	
Actor: Usuario con discapacidad visual, Dispositivo móvil	
Precondición: El Dispositivo Móvil debe estar conectado a la misma red wifi que la Raspberry Pi.	
Flujo Principal: Usuario con discapacidad visual. 2. Retorna a la vista "Listar objeto".	Flujo Principal: 1. Busca el objeto que tenga su ID idéntico al ingresado y lo elimina de la base de datos.
Postcondiciones: Se elimina el objeto de la base de datos presente en la Raspberry Pi.	

Tabla 13 CUS Eliminar objeto.

Junto a su respectivo diagrama de secuencia:

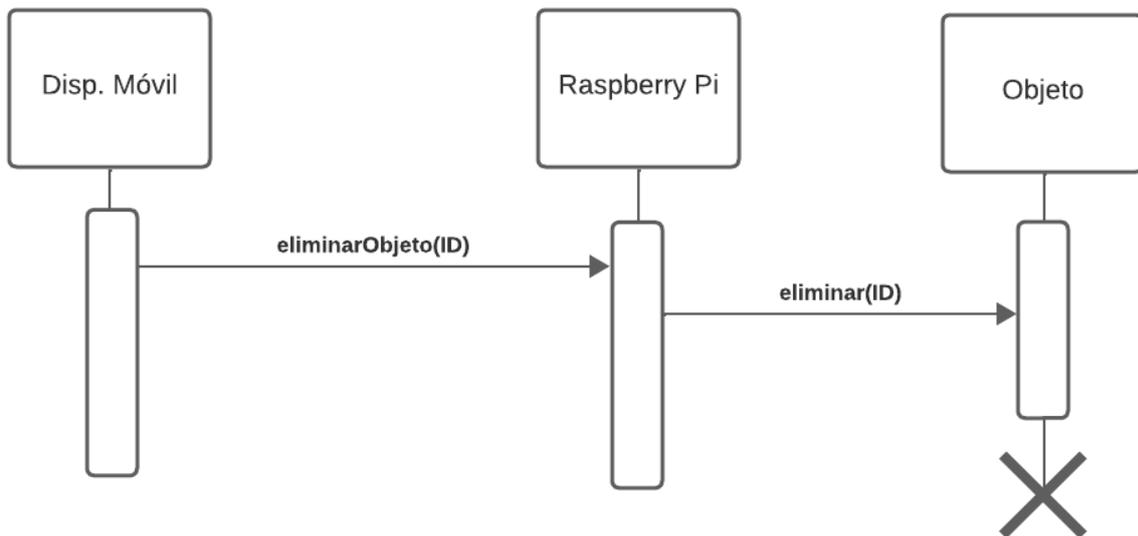


Ilustración 13 Diagrama de Secuencia - Eliminar objeto.

4.1.5.6. Transcribir voz a texto

Autor / Fecha: Andrew Campos / 23 - 12 - 2023	
Descripción: La función “dictado por voz” de teclado integrada por Google traducirá la información comunicada por voz a texto.	
Actor: Usuario con discapacidad visual, Micrófono	
Precondición: El micrófono debe estar habilitado, el usuario debe ser capaz de pronunciar correctamente las palabras a traducir y la opción “dictado por voz” debe estar activada.	
Flujo Principal: Usuario 1. En el teclado presiona la opción de “dictado por voz”.	Flujo Principal: Sistema 2. La grabación obtenida es traducida a texto.
Postcondiciones: No posee.	

Tabla 14 CUS Transcribir voz a texto.

Junto a su respectivo diagrama de secuencia:

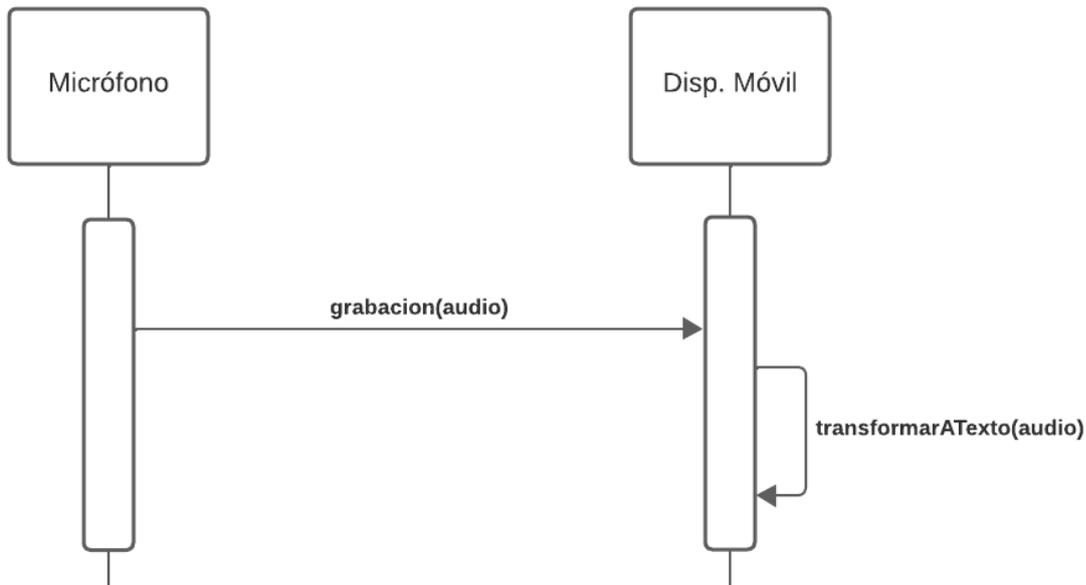


Ilustración 14 Diagrama de Secuencia - Transcribir voz a texto.

4.1.5.7. Reproducir texto a audio

Autor / Fecha: Andrew Campos / 24 - 10 - 2023	
Descripción: El sistema reproduce en audio cada elemento que se encuentra en la interfaz con el uso de Google Talk-Back.	
Actor: Usuario con discapacidad visual, Altavoz	
Precondición: La opción "Google Talk-Back" debe estar activada.	
Flujo Principal: Usuario 1. Mientras presiona la vista, desliza el dedo para ir cambiando en elementos en pantalla.	Flujo Principal: Sistema 2. El sistema reproduce a través del altavoz cada elemento presente en la interfaz.
Postcondiciones: No posee.	

Tabla 15 CUS Reproducir texto a voz.

Junto a su respectivo diagrama de secuencia:

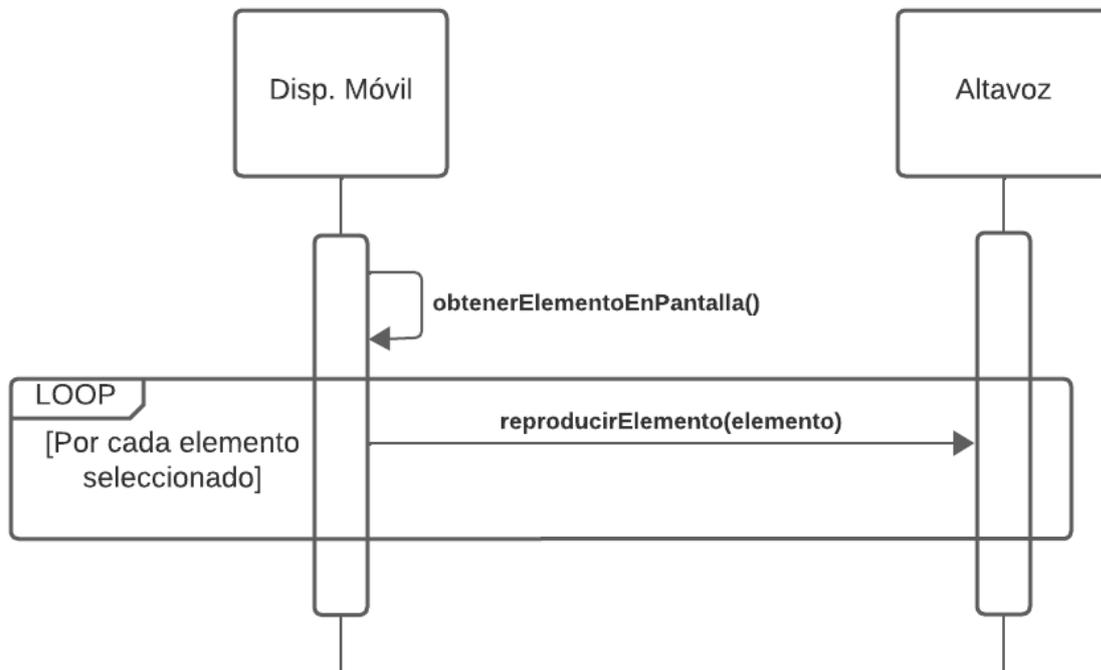


Ilustración 15 Diagrama de Secuencia - Reproducir texto a voz.

4.1.6. Diagrama de clases

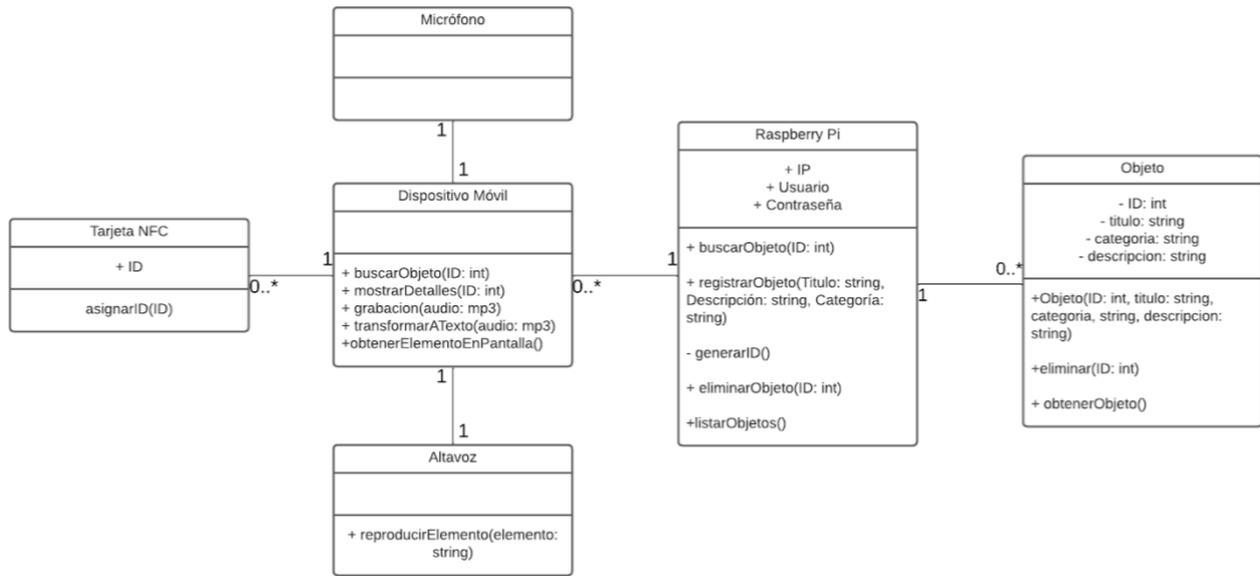


Ilustración 16 Diagrama de Clases.

4.2. HERRAMIENTAS

Herramientas y sensores:

1. *Visual Studio Code (VSCode)*: editor de código ligero y potente que es compatible con el desarrollo de proyectos web y móviles.
2. *Git*: sistema de control de versiones para el seguimiento de cambios en el código.
3. *SSH (Secure Shell)*: para acceder a la Raspberry Pi de forma remota desde una máquina de desarrollo.
4. *SCP/SFTP*: para transferir archivos entre una máquina de desarrollo y la Raspberry-Pi.
5. *Sensor activo*⁵: Dispositivo móvil con NFC. Puede generar señales NFC para comunicarse con otros dispositivos NFC, como tarjetas, etiquetas o incluso otros teléfonos móviles.
6. *Sensor pasivo*⁶: Chip NFC NTAG215. No generan activamente señales, pero responden cuando son alimentadas por la energía de la señal NFC emitida por el dispositivo activo, como un teléfono móvil.
7. *Sensor micrófono incorporado en el dispositivo móvil*.
8. *Sensor altavoz incorporado en el dispositivo móvil*.

Herramientas a utilizar para el desarrollo del servidor (Raspberry Pi - Flask):

1. *Python*: lenguaje de programación principal para el desarrollo del servidor.
2. *Flask*: framework web ligero para Python que facilita la creación de aplicaciones web.
3. *Flask-SQLAlchemy*: extensión de Flask que proporciona integración con SQLAlchemy, una biblioteca de mapeo objeto-relacional (ORM) para Python.
4. *SQLite (o MySQL/PostgreSQL)*: base de datos que puedes utilizar para almacenar la información en el servidor. SQLite es una opción liviana y fácil de usar para proyectos más pequeños.
5. *Flask-CORS*: Extensión de Flask para manejar los encabezados CORS (Cross-Origin Resource Sharing) y permitir solicitudes desde dominios diferentes al del servidor.

⁵ Mundo Sensor (s.f.). *Sensores activos y pasivos* - Accedido el 27 de noviembre de 2023.
<https://mundosensor.blogspot.com/2014/11/sensores-activos-y-pasivos.html>

⁶ Krypton Solid (s.f.). *¿Qué es sensor activo? - Definición de Krypton Solid* - Accedido el 27 de noviembre de 2023.
<https://mundosensor.blogspot.com/2014/11/sensores-activos-y-pasivos.html>

Herramientas para el desarrollo de la Aplicación Móvil (Ionic):

1. *Node.js* y *npm*: Node.js es necesario para instalar y ejecutar herramientas relacionadas con el desarrollo web, y npm es el gestor de paquetes de Node.js.
2. *Ionic Framework*: framework para el desarrollo de aplicaciones móviles con tecnologías web (Angular, HTML, CSS).
3. *Angular*: framework de desarrollo web utilizado por Ionic para construir la parte frontend de la aplicación móvil.
4. *Cordova*: plataforma para el desarrollo de aplicaciones móviles híbridas que permite acceder a las API nativas del dispositivo desde JavaScript.
5. *HttpClient (Angular)*: módulo de Angular para realizar solicitudes HTTP desde la aplicación móvil.

4.3. TÉCNICA UTILIZADA

La técnica a utilizar en el proyecto para realizar pruebas y ver los defectos que tiene la aplicación móvil es la de extremo a extremo. Según el sitio web parasoft.com, metodología de prueba que emplean los probadores para probar el flujo de trabajo de una aplicación de principio a fin. El objetivo de las pruebas de extremo a extremo es validar la integración del sistema y asegurarse de que los datos estén seguros. [4]

5. IMPLEMENTACIÓN

5.1. PLAN DE INTEGRACIÓN

En el siguiente apartado, se establece el plan de integración para el desarrollo de la aplicación móvil asistencial basado en IoT, cuyo objetivo es la identificación de chips NFC para personas con discapacidad visual.

Para llevar a cabo la integración de la aplicación móvil, se ha dividido en cuatro partes:

I. Comunicación entre el usuario y la aplicación:

Interfaz de Usuario con Ionic:

Se ha creado una interfaz de usuario utilizando el framework Ionic para proporcionar una experiencia cómoda y accesible. Esta interfaz se complementa con opciones de accesibilidad provenientes del sistema Android, tales como Google Talk-Back y dictado por voz.

Librerías Utilizadas:

- *Angular Core*: Proporciona las funcionalidades básicas de Angular.
- *Angular Router*: Módulo de enrutamiento para la navegación entre diferentes vistas.
- *Angular HTTP Client*: Módulo para realizar solicitudes y recibir respuestas HTTP.

II. Prueba en la Comunicación:

Se realizan 2 tipos de pruebas para supervisar la efectividad de la comunicación entre la aplicación y el usuario. La primera prueba verifica la conexión de la base de datos, y la segunda revisa el correcto funcionamiento de botones y cajas de texto en las vistas de la aplicación.

III. Identificación de NFC:

Para la lectura y escritura de NFC, se ha utilizado la librería *@awesome-cordova-plugins/nfc/ngx*, parte de los plugins de Cordova⁷, que interactúa con la funcionalidad NFC en aplicaciones móviles.

IV. Pruebas en la Identificación de NFC:

Se realizan dos pruebas: la primera verifica la funcionalidad independiente de la identificación de tarjetas NFC utilizando la aplicación externa NFC Tools, y la segunda garantiza la comunicación efectiva entre el usuario y la identificación de tarjetas NFC mediante la codificación que escribe y lee objetos mediante mensajes después de crear el APK.

⁷ Apache Cordova (s.f.). *Plugin Search – Apache Cordova* - Accedido el 24 de diciembre de 2023. <https://mundosensor.blogspot.com/2014/11/sensores-activos-y-pasivos.html>

5.2. MODELO DE IMPLEMENTACIÓN

Para realizar operaciones relacionadas con la comunicación entre la aplicación móvil y el servidor remoto mediante protocolo HTTP, se hace uso de la API Service. Este servicio es propio de Angular interactuando con el servidor API, lo cual logra y asegura una comunicación eficiente y manejo adecuado de las respuestas.

Entre sus funciones se encuentra la realización de operaciones CRUD en objetos gestionados por el servidor. Dichas operaciones se incluye la obtención de todos los objetos, la obtención de un objeto en específico mediante su ID, la creación de nuevos objetos y la eliminación de ellos. Como segunda función, tiene la capacidad de notificar eventos relacionados con la actualización de los objetos en la aplicación.

5.3. MÓDULOS IMPLEMENTADOS

5.3.1. Conexión Base de Datos con Aplicación Móvil

Para este módulo, se emplea el servicio API Service para realizar solicitudes HTTP al servidor creado con Flask. Para las operaciones CRUD, API Service utiliza las rutas definidas en Flask. Por ejemplo, el método **getObjetos()** en el servicio API Service obtiene todos los objetos accediendo a la ruta **/objetos** en el archivo creado con Flask.

```
class Objeto(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nombre = db.Column(db.String(50), nullable=False)
    categoria = db.Column(db.String(80), nullable=False)
    descripcion = db.Column(db.String(200))

# Rutas para operaciones CRUD en objetos
@app.route('/', methods=['GET'])
@app.route('/objetos', methods=['GET'])
def obtener_objetos():
    objetos = Objeto.query.all()
    return render_template('objetos.html', objetos=objetos)
```

Ilustración 18 Base de datos creada.

```
export class ApiService {
    private apiUrl = 'http://192.168.1.125:5000'; // 'http://192.168.71.226'; // Ajuste

    constructor(private http: HttpClient, private eventoService: EventoService) {}

    getIp(): string {
        return this.apiUrl;
    }

    getObjetos(): Observable<any> {
        return this.http.get(`${this.apiUrl}/objetos`);
    }
}
```

Ilustración 17 API Service.

5.3.2. Identificar tarjetas NFC (Lectura)

Las operaciones de lectura de la tarjeta NFC comienzan en el método **startListenToNFC()**, el cual se llama cuando se pulsa el botón de escaneo en la interfaz de usuario. Luego, inicia la escucha de eventos NFC mediante el uso de **nfc.addNdefListener()**. Al momento de que se detecte un evento NFC, se realiza la lógica donde se obtiene el ID de la tarjeta NFC.

```
startListenToNFC() {
  this.mostrarInfo = true;
  this.nfcListener = this.nfc.addNdefListener(() => {
    this.escaneando = true;
    this.mensaje = 'Se ha comenzado a escuchar, nfc listener iniciado.';
    console.log('NFC listener started');
  }, (err: any) => {
    this.error = 'Error al inicializar nfc listener + error: ' + err;
    console.error('Error starting NFC listener', err);
  }).subscribe((event) => {
    this.zone.run(() => {
      this.mensaje = 'NFC detectado';
      console.log('NFC tag detected', event);
      let payload = event.tag.ndefMessage[0].payload;
      let tagContent = this.nfc.bytesToString(payload).substring(3);
      console.log('Tag content:', tagContent);
      this.nfcData = tagContent;
      //===== La logica para obtener el obj. =====
      // Convertir el ID a número
      const objectId = parseInt(tagContent, 10);

      // Verificar si la conversión fue exitosa
      if (!isNaN(objectId)) {
        this.apiService.getObjetoPorId(objectId).subscribe((objeto) => {
          console.log('Objeto obtenido por ID:', JSON.stringify(objeto));
          //Abrir modal
          this.abrirModal();
          this.objeto = objeto; // Almacena el objeto obtenido de la API
        }, (error) => {
          console.error('Error al obtener objeto por ID', error);
        });
      } else {
        console.error('EL contenido de la etiqueta NFC no es un ID válido');
      }
    });
  });
}
```

Ilustración 19 Lectura NFC.

5.3.3. Identificar tarjetas NFC (Escritura)

Las operaciones de escritura de la tarjeta NFC solo se realizan después de que se haya detectado un chip NFC y la lógica de la aplicación haya sido ejecutada. Al pulsar el botón de escritura que aparece en la interfaz de usuario, inicia la escucha de eventos NFC y cuando detecta una etiqueta NFC, escribe el ID en el chip NFC utilizando el método **nfc.write()**.

```
92
93 escribirNFC(id: string) {
94   this.abrirModal();
95   this.nfcListener = this.nfc.addNdefListener(
96     () => {
97     this.mensaje = 'Se ha comenzado a escuchar';
98     this.ifMensaje = true;
99     console.log('NFC listener started');
100    },
101    (err: any) => {
102    this.error = 'Error al iniciar nfc listener, error: ' + err;
103    console.error('Error starting NFC listener', err);
104    }
105  ).subscribe((event) => {
106    this.zone.run(() => {
107      this.mensaje = 'NFC detectado';
108      console.log('NFC tag detected', event);
109      // Se escribe el id en la tarjeta NFC.
110      let message = this.ndef.textRecord(id);
111      this.nfc.write([message]).then((writeResp) => {
112        this.error = 'id escrito en la nfc: ' + id;
113        console.log('Error: ', this.error);
114        console.log('ID escrito en la etiqueta NFC');
115        console.log("writeResp: ", writeResp);
116      })
117      .catch((err:any)=> {
118        this.error = 'Error al escribir id en la NFC';
119        console.log("written error : ", err);
120      });
121    });
122  });
```

Ilustración 20 Escritura NFC.

5.4. REPORTE DE REVISIÓN

5.4.1. Prueba Número Uno

5.4.1.1. Descripción

Se verificó la capacidad de la aplicación "NFTouch" para conectarse a la base de datos alojada en el servidor Raspberry Pi. Se realizaron múltiples intentos de conexión para asegurar la estabilidad y la fiabilidad de la comunicación entre la aplicación y la base de datos.

5.4.1.2. Resultados obtenidos

La aplicación "NFTouch" logró establecer una conexión consistente y segura con la base de datos en todas las instancias de prueba. Se registraron los tiempos de respuesta y la integridad de los datos recuperados o enviados para asegurar la eficacia del proceso.

5.4.1.3. Conclusiones

La exitosa conexión entre la aplicación y la base de datos confirma la eficiencia del diseño de la red y la correcta implementación de los protocolos de comunicación. Esto indica que "NFTouch" está bien equipada para manejar la información de los usuarios y operar de manera efectiva en un entorno real.

5.4.2. Prueba Número Dos

5.4.2.1. Descripción

Se centró en validar la funcionalidad de registro de la aplicación "NFTouch". Consistió en crear un nuevo objeto en la aplicación y confirmar que se añadía adecuadamente a la base de datos alojada en el servidor Raspberry Pi.

5.4.2.2. Resultados obtenidos

Cada intento de registro de un nuevo objeto desde la aplicación resultó en una entrada correspondiente y correcta en la base de datos. Se comprobó la consistencia de los datos entre la entrada realizada desde la aplicación y los registros almacenados en la base de datos.

5.4.2.3. Conclusiones

Los resultados demuestran que la funcionalidad de registro de "NFTouch" es operativa y fiable. La aplicación es capaz de crear registros precisos en la base de datos, lo que es esencial para el funcionamiento efectivo y la confiabilidad del sistema.

5.4.3. Prueba Número Tres

5.4.3.1. Descripción

Se examinó la capacidad de "NFTouch" para leer y sobrescribir información en un chip NFC, así como la integración y funcionamiento del escáner NFC del dispositivo móvil. Se efectuaron pruebas de lectura y escritura de datos en los chips NFC para verificar esta funcionalidad esencial.

5.4.3.2. Resultados obtenidos

La aplicación "NFTouch" mostró una lectura precisa de la información existente en los chips NFC y sobrescribió con éxito los datos cuando se requirió. Además, el escáner NFC del teléfono móvil funcionó sin interrupciones, detectando y procesando los chips NFC de manera eficiente en cada intento.

5.4.3.3. Conclusiones

Los resultados confirman que "NFTouch" realiza con éxito la lectura y escritura en los chips NFC y que la aplicación interactúa eficazmente con el escáner NFC del teléfono. Esto garantiza que la aplicación puede gestionar de manera efectiva la información en los chips NFC durante su uso cotidiano.

6. PROBLEMAS ENCONTRADOS Y SOLUCIONES

PROPUESTAS

1. Para establecer la conexión remota entre la aplicación móvil y la base de datos, se requería que ambos estuvieran conectados a la misma dirección IP. Sin embargo, durante las pruebas y al cambiar de red Wi-Fi, la dirección IP cambiaba constantemente, lo que dificultaba la conexión con la base de datos. Como solución, se optó por utilizar la misma red Wi-Fi y se introdujo una variable llamada **APIUrl** en el código, donde se puede modificar según sea necesario.
2. En la planificación del proyecto, se determinó que el dispositivo móvil utilizado para las pruebas sería un Galaxy A51, ya que contaba con un sensor NFC. Sin embargo, al integrar el APK en el dispositivo y tratar de realizar pruebas de lectura y escritura de un chip NFC, surgieron problemas. Estos inconvenientes se debieron a que la biblioteca utilizada para crear el APK no era compatible con Android 12 en adelante. Una de las soluciones propuestas fue integrar la APK en dispositivos iOS. Sin embargo, la política de distribución de aplicaciones en iOS es más rigurosa que en Android. Como consecuencia, se optó por cambiar el dispositivo móvil a uno que tuviera NFC y ejecutara una versión de Android 11.
3. Al comienzo, se utilizaba la librería PhoneGap-NFC, el cual era una librería para Cordova, pero al utilizar Capacitor, no era compatible. Como solución propuesta, se averiguó una librería que integre funciones NFC compatible con Capacitor, siendo esta [@awesome-cordova-plugins/nfc/ngx](https://github.com/awesome-cordova-plugins/nfc-ngx)

7. CONCLUSIONES

En conclusión, respondiendo a la pregunta inicial, el IoT puede utilizarse para mejorar la calidad de vida de las personas no videntes mediante el desarrollo de una aplicación móvil. Se logró planificar un proyecto destinado a crear una aplicación móvil asistencial enfocada en personas con discapacidad visual, capaz de reconocer y detectar agentes externos en su entorno hogareño. Además, se analizaron los diversos desafíos que enfrentan las personas con discapacidad visual con el propósito de abordar un problema específico en sus vidas.

Para esto, por un lado, se colaboró con una persona no vidente, lo que permitió identificar un problema y encontrar una solución. Se desarrolló el diseño de una aplicación móvil utilizando herramientas como IoT, chips NFC y Raspberry Pi. Por otro lado, se establecieron las suposiciones y restricciones de este proyecto, lo que proporcionó una mayor claridad sobre los desafíos que se tuvieron que enfrentar durante la fase final del proyecto.

En el momento actual, la aplicación móvil ya diseñada e implementada contiene funcionalidades de lectura y escritura de objetos en el chip NFC. Los objetos se registran mediante voz, y los elementos en la vista pueden describirse a través de Google Talk-Back, facilitando así la experiencia para usuarios con discapacidad visual al guiarse en el dispositivo móvil.

En relación con la creación de aplicaciones móviles, se concluye que es menos compleja en dispositivos Android. Aunque la revisión de la App Store puede proporcionar un entorno más seguro y controlado, también puede complicar el proceso de instalación. Por otro lado, la apertura de Android permite una mayor flexibilidad, ya que los usuarios pueden instalar aplicaciones desde fuentes distintas a la Google Play Store. Sin embargo, esto puede implicar riesgos de seguridad si los usuarios instalan aplicaciones desde fuentes no verificadas.

La necesidad de crear nuevas y más innovadoras formas de mejorar la calidad de vida humana es una tarea ardua. La importancia de las nuevas tecnologías, en concordancia con la creatividad humana, es clave para ayudar a grupos humanos específicos y evitar que sus necesidades sean olvidadas. Este proyecto representa una iniciativa más en la búsqueda de una mejor calidad de vida para todos, aunque se enfoque solo en una comunidad. Sin embargo, es relevante destacar que existen innumerables otros problemas para esta y muchas otras comunidades que pueden resolverse con la ayuda de la tecnología y el ingenio humano.

8. TRABAJO FUTURO

Actualmente, la aplicación móvil cumple con las funcionalidades establecidas por el cliente, es decir, es efectiva y cómoda para las personas con discapacidad visual. Además, se implementa una Raspberry Pi aplicando IoT. Sin embargo, hay características que se pueden mejorar, y para el futuro, sería gratificante innovarlas para que la calidad de la aplicación funcione de manera exponencial.

El enfoque principal por el cual NFTouch se destaca es la aplicación de tecnologías como las tarjetas NFC. Por lo tanto, una excelente idea sería adoptar el formato de stickers, ya que es más práctico. De forma paralela, también se espera actualizar la interfaz gráfica, implementando una vista de GPS para que el usuario pueda tener una idea más precisa sobre dónde se encuentra cada objeto

Por otro lado, una proyección a futuro es implementar la aplicación en dispositivos iOS con la finalidad de tener un público más amplio.

REFERENCIAS

- [1] A. Beltramen, «Prototipado rápido de proyectos IoT sin programación,» SEDICI - Repositorio de la Universidad Nacional de La Plata, Septiembre 2018. [En línea]. Available: <http://sedici.unlp.edu.ar/handle/10915/71753>.
- [2] G. R. Pontiggia, «Las soluciones tecnológicas que mejoran la vida de las personas ciegas,» El Dínamo, 28 Abril 2023. [En línea]. Available: <https://www.eldinamo.cl/buen-dato/2023/04/28/las-soluciones-tecnologicas-que-mejoran-la-vida-de-las-personas-ciegas/>.
- [3] SAP, «¿Qué es el internet de las cosas (IoT)?,» SAP, [En línea]. Available: <https://www.sap.com/latinamerica/products/artificial-intelligence/what-is-iot.html>.
- [4] “Las pruebas de extremo a extremo mejoran la calidad del software”. Parasoft. [En línea]. Available: <https://es.parasoft.com/solutions/end-to-end-testing/#:~:text=La%20prueba%20de%20extremo%20a,que%20los%20datos%20estén%20seguros>
- [5] “Salario para Jefe De Proyecto en Chile - Salario Medio”. Talent.com. [En línea]. Disponible: <https://cl.talent.com/salary?job=Jefe+de+proyecto>
- [6] “Salario para Analista Programador en Chile - Salario Medio”. Talent.com. [En línea]. Disponible: <https://cl.talent.com/salary?job=Analista+programador>
- [7] “Salario para Desarrollador Frontend en Chile - Salario Medio”. Talent.com. [En línea]. Disponible: <https://cl.talent.com/salary?job=Desarrollador+frontend>
- [8] “Salario para Desarrollador Backend en Chile - Salario Medio”. Talent.com. [En línea]. Disponible: <https://cl.talent.com/salary?job=Desarrollador+backend>
- [9] “Sensores activos y pasivos”. Todo acerca de sensores. [En línea]. Disponible: <https://mundosensor.blogspot.com/2014/11/sensores-activos-y-pasivos.html>
- [10] “Angular”. Angular. [En línea]. Disponible: <https://angular.io/api/core>
- [11] “Ionic Docs - Ionic Documentation”. Ionic: Enterprise App Platform. [En línea]. Disponible: <https://ionic.io/docs>
- [12] “Plugin Search - Apache Cordova”. Apache Cordova. [En línea]. Disponible: <https://cordova.apache.org/plugins/>