

**UNIVERSIDAD DE TARAPACÁ**



**FACULTAD DE INGENIERÍA**

Departamento de Ingeniería en Computación e Informática



**Segundo Informe**

**Sensor referencial GPS para discapacitados visuales**

**Autor(es): Javier Mamani**

**Mauricio Mamani**

**Esteban Ovando L.**

**Asignatura: Proyecto II**

**Profesor(es): Diego Aracena P.**

**Arica 23 12 2020**

## 1. Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
06/10/2020	1.0	Definición de escenario	Mauricio Mamani Javier Mamani Esteban Ovando
28/10/2020	1.1	Actualización de objetivos específicos y objetivo estratégico, cambio en alcance	<i>Mauricio Mamani</i>
29/10/2020	1.2	Actualización del plan de proyecto.	<i>Javier Mamani</i> <i>Mauricio Mamani</i>
30/10/2020	1.3	Actualización del plan de proyecto	<i>Javier Mamani</i> <i>Mauricio Mamani</i> Esteban Ovando
3/11/2020	1.4	Actualización del plan de proyecto	<i>Javier Mamani</i> <i>Mauricio Mamani</i> Esteban Ovando
30/11/2020	2	Corrección del informe	<i>Javier Mamani</i> <i>Mauricio Mamani</i> Esteban Ovando
17/12/2020	2.1	Actualización de Carta gantt	<i>Javier Mamani</i>
22/12/2020	2.2	Actualización de informe de avance 2, análisis y diseño	<i>Javier Mamani</i> <i>Mauricio Mamani</i> Esteban Ovando
23/12/2020	2.3	Agregación de arquitectura del sistema	<i>Esteban Obando</i> <i>Javier Mamani</i>
3/01/2021	3	Agregación de la Implementación	<i>Mauricio Mamani</i> <i>Esteban Obando</i> <i>Javier Mamani</i>



## 2. Tabla de contenido

1. Historial de Cambios .....	2
3. Introducción.....	5
4. Escenario .....	7
5. Panorama General.....	8
4.1 Propósito .....	8
4.2 Alcance .....	8
4.3 Objetivo general .....	8
4.4 Objetivo específico. ....	8
4.5 Suposiciones .....	9
4.6 Restricciones .....	9
4.7 Entregables del Proyecto .....	9
6. Organización del Proyecto.....	10
5.1 Personal y entidades internas .....	10
5.2 Roles y responsabilidades .....	10
5.3. Mecanismos de Comunicación .....	11
7. Planificación de los procesos de gestión .....	12
6.1 Planificación inicial del proyecto .....	12
6.2 Lista de actividades .....	13
Asignación de tiempo.....	13
6.3 Planificación de la gestión de riesgos .....	14
8. Análisis .....	15
7.1 Documento de diseño de interfaz usuario .....	15
7.2 Descripción de la arquitectura .....	19

---

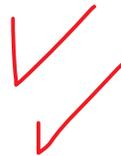
7.3	Requerimientos funcionales y no funcionales .....	20
7.4	Modelos de casos de uso .....	21
9.	Diseño .....	<del>26</del> 25
8.1	Diagrama de secuencia .....	26
8.2	Modelo de clases .....	28
10.	Implementación .....	29
9.1	Plan de integración .....	29
9.2	Modelo de implementación .....	30
9.3	Módulos Implementados .....	31
9.4	Reporte de revisión .....	40
11.	Problemas encontrados .....	44
12.	Soluciones propuestas .....	44
13.	Conclusión .....	45
14.	Referencias .....	46
15.	Anexos .....	47

### 3. Introducción

Entre una de las discapacidades que puede enfrentar una persona está la discapacidad visual, esta deja a la persona sin una de las más importantes cualidades para el ser humano, esto, ya que mediante la vista una persona es capaz de reconocer su entorno, apreciar las distintas maravillas del mundo, llegar a lugares observando señales, disfrutar de películas, libros o juegos de mesa, o simplemente ver cómo cambia el mundo que se conoce. Es por esto que es tan importante la vista para el ser humano, datos que ha entregado la organización mundial de salud dicen que 285 millones de personas tiene discapacidad visual y de las cuales 39 millones son ciegas en todo el mundo, estas cifras dejan claro que no son pocos los afectados. Es por ello que el equipo "JEM GPS" a propuesto desarrollar una solución que estará enfocada a guiar al usuario no vidente, mediante una aplicación móvil, la cual se encargará de guiar al discapacitado visual a través de instrucciones por audio, con la finalidad de que el individuo pueda llegar a una ubicación deseada.

Este proyecto tiene la finalidad de, lograr la entrega de un producto funcional que permita guiar al discapacitado visual, es por esto que, mediante este informe de avance, se establecerán objetivos específicos, los cuales son un pilar importante para lograr el objetivo general, estos objetivos específicos a su vez se podrán realizar por medio subtareas o trabajos. También se establecen restricciones las cuales dejan en claro cuáles serán los factores que no se pueden modificar para la realización del proyecto, luego de esto se mostrará los entregables del proyecto con una breve descripción de dicho entregable. También se establecen roles, estos roles están definidos según los requerimientos que se necesitan para la realización del proyecto, los roles tendrán a su ~~cargo~~ uncargo un integrante del equipo que es quien debe dar solución a los problemas que surjan. Se tendrá una planificación inicial del proyecto donde se tendrá, una planificación de estimaciones la cual muestra los costos totales de recursos software, hardware y horas persona que se tiene en el proyecto, luego se muestra cómo será repartido el recurso humano, esto mostrando cuantos integrantes estarán destinados a

una labor en específico. Luego se mostrará la lista de actividades, en esta sección se tiene una Carta Gantt, en la cual se da a conocer en detalle el avance de las actividades mostrando porcentaje realizado, fechas iniciales y fechas límites, además de, nombre del integrante que está encargado de la subtarea. Otra sección que se tiene es planificación y gestión de riesgos, en esta sección se tendrá una tabla la cual muestra los riesgos identificados, su probabilidad de riesgo (que tan probable es que ocurra), nivel de impacto, este estará categorizado con números del uno al tres, siendo uno el nivel más insignificante y tres el riesgo más importante, después se tendrá una acción remedial, la cual puede ser una forma de dar soporte o solución a dicho riesgo. Se definen las secciones de análisis y diseño con el fin de plasmar los procesos previos que son útiles para la realización de la aplicación, esto ya que, en una sección posterior de implementación, será una base fundamental para realizarla. Posteriormente se tienen las secciones de implementación, en esta sección se aborda el plan de integración que cuenta cómo se integra cada parte de la aplicación, luego viene el modelo de implementación el cual muestra cómo se compone la aplicación mediante botones y librerías que fueron utilizados, luego viene módulos implementados, mediante esta parte se da cuenta de los módulos que se utilizaron para la realización de la aplicación dando una descripción detallada de estos mismos, después viene reporte de revisión, donde se da a conocer las pruebas realizadas para la aplicación. Finalmente se tendrá una conclusión que dé a conocer lo aprendido por cada integrante en el desarrollo del informe, además de dar a conocer los conocimientos previos que ayudaron a realizar dicho informe.



## 4. Escenario



Ilustración 1 Escenario

Con el objetivo de poder ayudar al discapacitado visual se les proporciona ayuda mediante GPS y audio, el usuario podrá llegar a una ubicación mediante el software que estará en su dispositivo móvil, este tendrá almacenado ubicaciones de locales cercanos, le mandará audio de tal manera que lo guíe paso a paso para llegar a la ubicación deseada.

## 5. Panorama General

### 4.1 Propósito

El proyecto permitirá desarrollar una aplicación de ayuda a las personas invidentes, o con problemas de vista.

### 4.2 Alcance

Para lograr una aplicación funcional capaz de guiar al invidente, se hace uso del sensor de GPS junto con una salida de audio que permita indicar el camino que debe seguir el invidente, esto estará montado sobre un Smartphone con sistema operativo Android.

### 4.3 Objetivo general.

Realizar una aplicación móvil que pueda guiar al discapacitado visual, permitiendo que este llegue desde un punto de partida, a una ubicación deseada, mediante sensor GPS e indicaciones de ~~movimiento~~ movimiento a través de audio.

### 4.4 Objetivo específico.

- Desarrollo de la documentación mediante informes que muestren en detalle los avances realizados (informe, bitácora, escenario, etc).
- Realizar estudio de la problemática, definiendo el problema más relevante, del cual se establece una solución, esto debe mostrarse en un escenario experimental.
- Estudiar aspectos fundamentales con respecto al lenguaje de programación que se utilizara para el desarrollo de la aplicación.
- Estudiar y establecer módulos que permitan el funcionamiento apropiado de la aplicación móvil.
- Estudiar y establecer un entorno de programación apropiado para poder trabajar con el dispositivo móvil.
- Diseño de la interfaz adecuada para el usuario de la aplicación móvil.

- Realizar pruebas experimentales dejando en evidencia el funcionamiento de la aplicación móvil.
- Entrega del producto final, cumpliendo cada parámetro establecido.

**Comentado [u1]:** Mucho objetivos específicos.. normalmente 3 a 4 es lo apropiado

### 4.5 Suposiciones

Lo que se espera con este proyecto es facilitar la orientación a los invidentes mediante una aplicación móvil, que utilizará el GPS del celular, con el objetivo de que el invidente pueda ir a lugares solicitados.

### 4.6 Restricciones

- El equipo será formado por un máximo de 3 integrantes.
- El proyecto se culminará en un tiempo no mayor a un semestre académico.
- Se debe contemplar un tiempo de entrenamiento para que el usuario pueda utilizar la aplicación.
- El sistema de visión para invidente, no debe ser invasivo, este debe ser evaluado por los organismos de seguridad y salud.

### 4.7 Entregables del Proyecto

Entregables	Descripción
Informe del proyecto	Estos consisten en. <ul style="list-style-type: none"> <li>● Informe 1</li> <li>● Informe 2</li> <li>● Informe final</li> </ul>
Bitácoras semanales	Es donde se registrara lo que se realizó en la semana, además de contener actividades futuras y sus responsables.
Carta Gantt	Cronología de las actividades además de su nivel de desarrollo actual.
Producto	Aplicación para smartphone que ayuda a los invidentes.
Escenario	Es el diseño explicativo en donde se detalla la problemática puntual a la que se le debe

	dar solución, mostrando dicha solución propuesta.
--	---

Tabla 1 Entregables del proyecto

## 6. Organización del Proyecto

### 5.1 Personal y entidades internas

- Programador: Esteban Ovando
- Diseñador: Mauricio Mamani
- Jefe de proyecto: Mauricio Mamani
- Encargado de informe: Javier Mamani
- Encargado de Redmine: Javier Mamani

### 5.2 Roles y responsabilidades

- Programador: codifica las especificaciones detalladas en el diseño según el lenguaje de trabajo.
- Diseñador: Diseña el ambiente visual que ayuda a la utilización del software.
- Jefe de-Proyecto: Encargado de supervisar y organizar el trabajo de cada integrante del equipo de proyecto.
- Encargado de informe: Es el encargado de que se lleve a cabo el informe del proyecto.
- Encargado de redmine: Es el encargado de subir y revisar los archivos requeridos.

Obs: entonces tendremos un buen wiki del proyecto

### 5.3. Mecanismos de Comunicación

Plataforma online: Google Drive, Discord y zoom

- Drive: Es una plataforma, donde se almacenarán los informes, bitácoras, tratamientos de riesgos, etc. También se usa como herramienta para el desarrollo de informes y presentaciones, ya que, cuenta con un sistema de trabajo online en el cual se puede avanzar paralelamente en ello.
- Discord: Software gratuito de comunicación online, este cuenta con un sistema de salas, el proyecto cuenta con una sala de trabajo con un chat y llamada de voz grupal. Este funciona tanto en Windows como en Android.
- Zoom: Es un servicio de videoconferencia que se puede usar para reunirse virtualmente con otras personas, ya sea por video o solo audio o ambos, todo mientras realiza chats en vivo, y le permite grabar esas sesiones para verlas más tarde.



*Ilustración 2 Mecanismos de comunicación*

## 7. Planificación de los procesos de gestión

### 6.1 Planificación inicial del proyecto

- Planificación de estimaciones:

Nombre	Cantidad	Costo total
Teléfono con gps	1	\$70.000
Computadoras portátiles	3	\$1.500.000
Seguro de teléfono	1	\$10.000
horas de trabajo(persona)	3	\$8.000(por hora) 8 horas semanales (c/u) \$192.000(por semana) 16*192.000= \$3.072.000(total)
Producto	1	\$4.652.000

Tabla 2 Planificación de Estimaciones

- Planificación de Recursos Humanos

Programador: 3

Diseñador gráfico: 1

Jefe de proyecto: 1

Encargado de informes: 3

**Comentado [u2]:** Tiene su complejidad su proyecto .. barato se venden

Encargado de redmine: 1

## 6.2 Lista de actividades

### Actividades de trabajo

- 1) Documentación del proyecto (14 semanas).
- 2) Bitácoras semanales (14 semanas).
- 3) Estudio de la problemática propuesta (2 semanas).
- 4) Estudiar sensor GPS, entrada y salida de audio (1 semana).
- 5) Estudiar lenguaje de programación (1 semana).
- 6) Realizar módulo de reconocimiento de ubicación (2 semanas).
- 7) Realizar módulo de entrada de instrucciones por micrófono (2 semanas).
- 8) Realizar módulo de salida de audio (2 semanas).
- 9) Realizar diseño de interfaz (2 semanas).
- 10) Realizar prueba de funcionamiento (4 semanas).
- 11) Documentación del código (1 semana).
- 12) Corrección del informe (1 semana).

### Asignación de tiempo

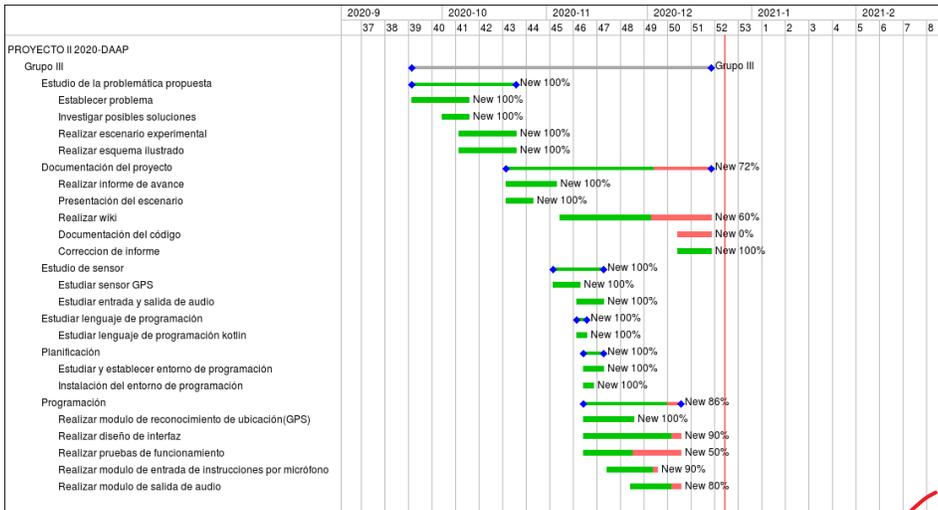


Ilustración 3 Carta Gantt

### 6.3 Planificación de la gestión de riesgos

Riesgos	Probabilidad de riesgo	Nivel de impacto	Acción remedial
Falta de respaldo del proyecto	5%	2	Volver a desarrollar lo perdido
Falta del servicio net	50%	2	Postular a la beca de conectividad
Tiempo insuficiente para finalizar el proyecto	10%	2	Planificar y establecer labores más prioritarias.
Conflicto con algún integrante del equipo	5%	3	Conversar con el integrante para llegar a un acuerdo
Compatibilidad de software con dispositivo móvil	25%	3	Comprar un dispositivo móvil que sea compatible

			con el software
Abandono de integrante	5%	1	Distribuir las actividades con los demás integrantes

Tabla 3 Planificación de la Gestión de riesgos

## 8. Análisis

### 7.1 Documento de diseño de interfaz usuario

Esta aplicación es para cualquier tipo de usuario, pero va más dedicado a las personas no videntes. Es por esto que la interfaz está diseñada de una manera sencilla para que el usuario lo utilice sin mucha ~~interacción, y~~ interacción, y sin necesidad de la vista.

Para la interfaz principal, tenemos 2 botones, para que el usuario no tenga tanta complejidad en interactuar con la interfaz. Como primer botón, de hecho va hacer el primer botón que el usuario debe presionar, es el botón ~~“listar”~~ “listar lugares”, donde al usuario se le dictara por medio de un audio, las listas de los lugares que están registrados. Ahora, tenemos al otro botón “Elegir lugar”

Comentado [u3]: Se tiene



*Ilustración 4 Interfaz de usuario*

Para poder orientar al usuario sobre el uso de la aplicación, se le facilitara un tutorial, donde se le indica el uso de la aplicación. Ahora, para acceder a esta tutorial, el usuario debe girar el móvil hacia la derecha en 90°

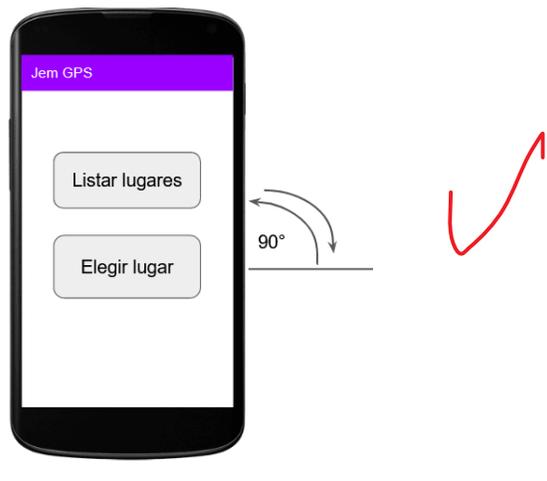
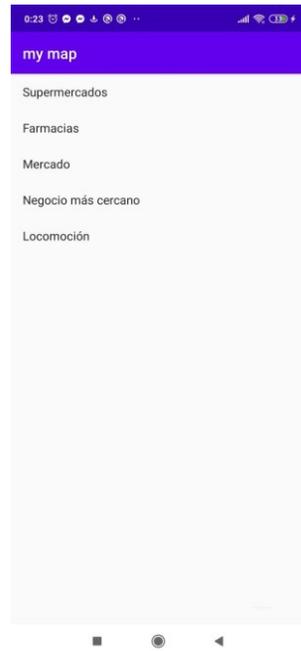


Ilustración 5 Interfaz de usuario evento

El usuario tiene disponible N destinos registrados en la aplicación, estos destinos están guardado en la aplicación, y solo son llamados por el usuario al solicitarlo, en la ilustración 6 se muestra los destinos que tiene disponible el usuario en la aplicación.



*Ilustración 6 Interfaz funcionalidad de ubicaciones*

En la ilustración 7, se muestra el destino que eligió el usuario que es "Farmacias", donde se muestra la ubicación del usuario, el destino asignado y la ruta para llegar a este destino.



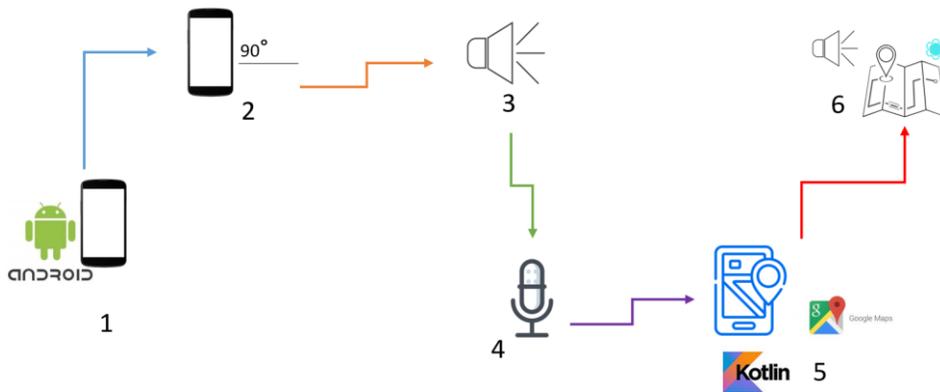


Ilustración 8 Descripción de la arquitectura

- 1) Interfaz definida para el usuario invidente, con la que podrá acceder a las funcionalidades de la aplicación. Si el usuario selecciona algún botón, se podrá escuchar mediante audio la opción que seleccionó.
- 2) Evento que proporciona tutorial en audio, sirve como guía rápida del uso de la interfaz de usuario.
- 3) Sensor de audio, del cual se reciben las instrucciones que dicta cada evento activado por el usuario.
- 4) Sensor de micrófono del dispositivo móvil con el cual se recibe la ubicación que el usuario quiere visitar.
- 5) Procesamiento de la orden de voz del usuario con la cual se obtienen datos coincidentes que previamente ya están almacenados en la aplicación.
- 6) Se le brinda una serie de instrucciones para poder llegar a la ubicación de destino solicitada previamente.



### 7.3 Requerimientos funcionales y no funcionales

Requerimientos funcionales	Especificación
Sensor gps	Se utilizará el sensor GPS para obtener la ubicación del usuario, esto servirá para poder comparar ubicaciones.
Salida de audio	Se utilizará para guiar, tanto en la ruta hacia al destino como en el uso de la interfaz de usuario.
Entrada de audio	Se utiliza para recibir instrucciones del usuario.
Sensor giroscopio	Se utiliza para provocar eventos con el movimiento del teléfono.

Tabla 47 Especificación de requerimientos

Lista de requerimientos no funcionales:

- Reconocimiento de obstáculos mediante visión computacional.
- Implementación de brújula.
- Obtener la ruta desde cualquier ubicación.

## 7.4 Modelos de casos de uso

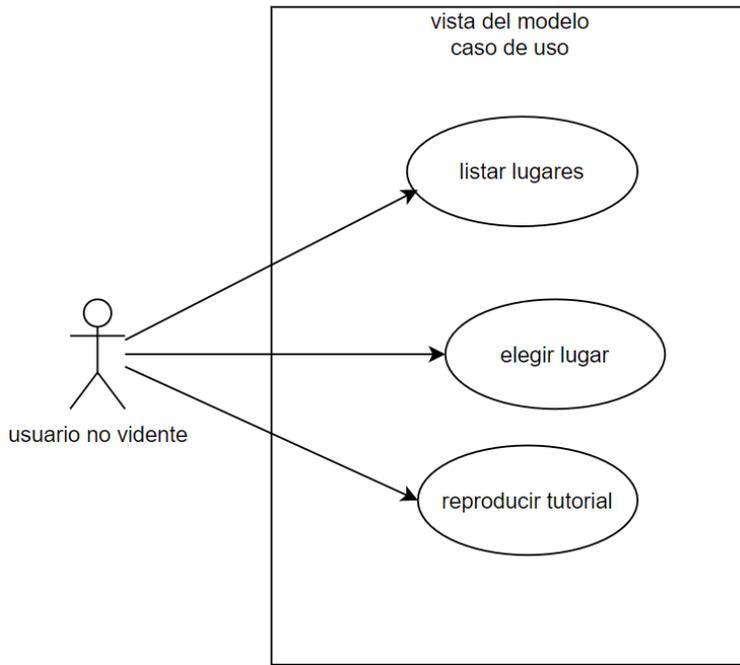


Ilustración 9 Modelo caso de uso

Obs: falta los casos de uso de todo lo referente a la aplicación (captura, seguimiento, orientar ..

Nombre	Elegir lugar
Descripción	Se elige el lugar que se quiere como destino. Luego, se recibe la información necesaria para poder llegar a dicho destino
Actores	Usuario no vidente
Precondiciones	Tiene que haber lugares guardados en la aplicación
Flujo normal	
<p style="text-align: center;">ACTOR</p> <p>1. Solicitar Lugar de destino</p> <p>2. Ingresar el lugar de su destino</p>	<p style="text-align: center;">SISTEMA</p> <p>3. Dicta las instrucciones de orientación</p>
Flujo alternativo:	
1.1. No se encontró el lugar de destino	1.2.. Dicta mensaje de error.
Glosario:	
Postcondiciones: Se guió al usuario al lugar de destino.	

Tabla 54 Caso de uso elegir lugar

Con formato: Fuente: 16 pto

Nombre	Listar lugares	
Descripción	Se listan los lugar guardados	
Actores	Usuario no vidente	
Precondiciones	Deben de haber lugares almacenados con anterioridad en la aplicación.	
Flujo normal		
	ACTOR	SISTEMA
	1.- Solicitar <b>lista de lugares</b> .	2.- Dictar <b>lista de lugares</b> .
Flujo alternativo		
Glosario: <b>lista de lugares</b> : hace referencia a los lugares a los que pueden acceder.		
Postcondiciones: Se conocen los lugares a los que se puede dirigir el usuario.		

Tabla 65 Caso de uso listar lugares

Con formato: Fuente: 16 pto

Nombre	Reproducir tutorial	
Descripción	Se reproduce un tutorial simple	
Actores	Usuario no vidente	
Precondiciones	Debe de existir el diálogo del tutorial.	
Flujo normal		
	ACTOR	SISTEMA
	1.- Solicitar tutorial.	2.- Dicta tutorial de la interfaz.
Flujo alternativo		
Glosario:		
Postcondiciones: Se comprende el funcionamiento de la interfaz.		

Tabla 76 Caso de uso Reproducir tutorial

## 9. Diseño

### 8.1 Diagrama de secuencia

#### Diagrama de secuencia elegir lugar

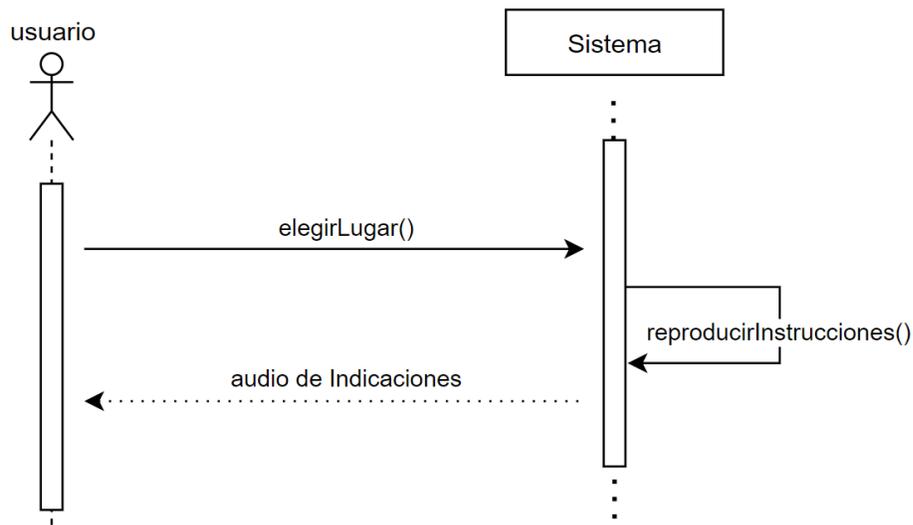


Ilustración 10 Diagrama de secuencia elegir lugar

### Diagrama de secuencia listar lugar

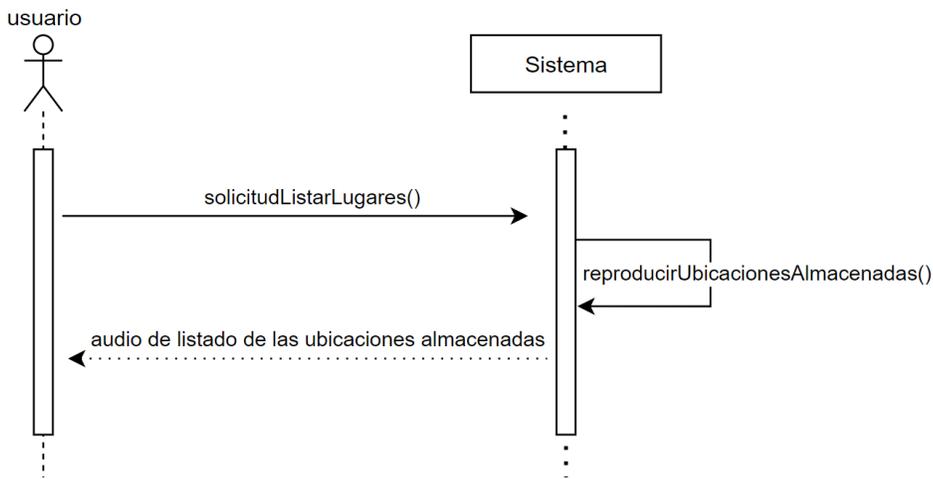


Ilustración 11 Diagrama de secuencia listar lugar

### Diagrama de secuencia reproducir tutorial

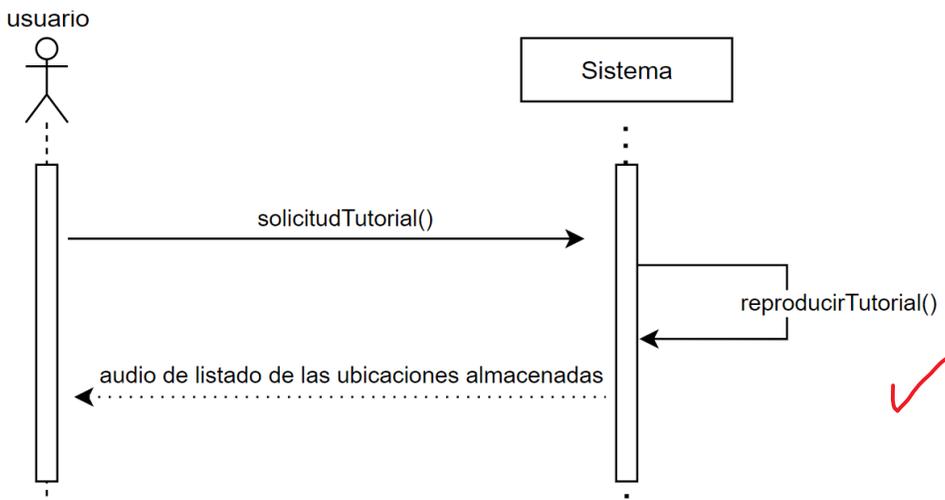


Ilustración 12 Diagrama de secuencia reproducir tutorial

## 8.2 Modelo de clases

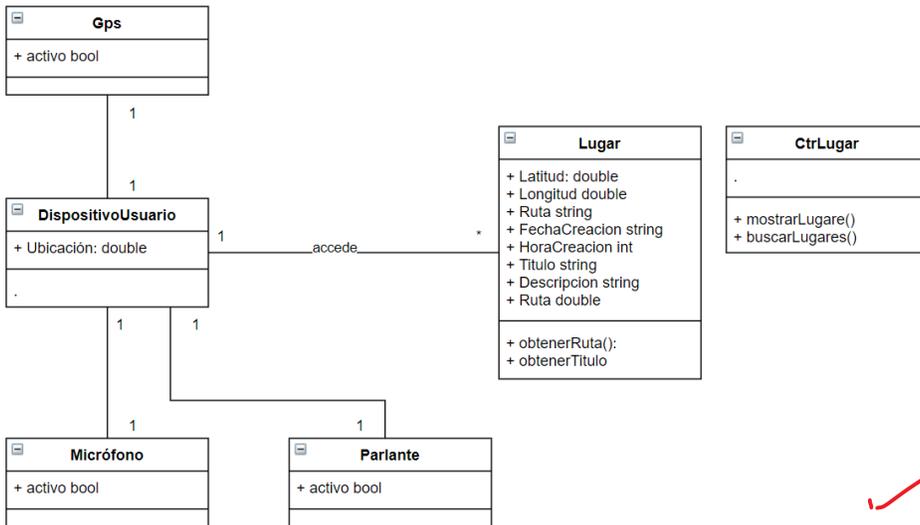


Ilustración 13 Modelo de clases

## 10. Implementación

### 9.1 Plan de integración

A grandes rasgos la aplicación consta de dos apartados, el de diseño gráfico y el del mapa con sus respectivas funciones, estas dos partes son necesarias para que el usuario, que en este caso es discapacitado visual pueda interactuar con la aplicación y para que la aplicación cumpla con los requerimientos específicos.

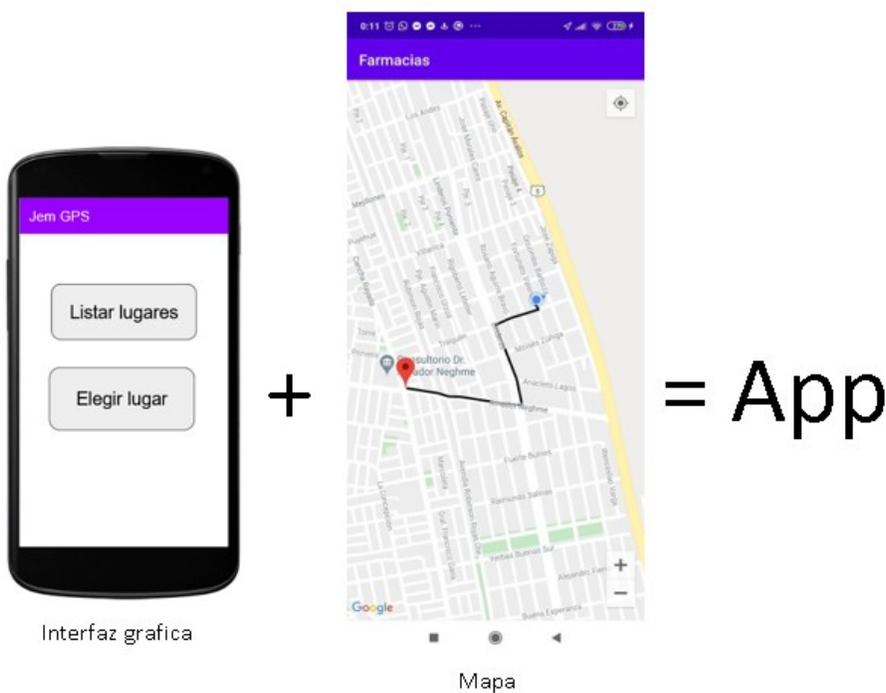


Ilustración 14 Plan de integración

La manera en que se integran es que el usuario mediante la interfaz gráfica ingresa una entrada de audio del lugar solicitado, la aplicación transforma esta entrada de audio a texto y con eso busca el lugar en una lista de lugares ya establecida en la aplicación, si



este lugar es encontrado, la aplicación procede a entregar una ruta en el mapa desde su ubicación al lugar solicitado.

## 9.2 Modelo de implementación

Para el desarrollo del proyecto se hizo uso de 2 API, son “text to speech” y “google map”.

**API “text to speech”:** Convierte de texto a voz y de voz a texto, por lo que es una herramienta que va a tener de mucha utilidad, ya que el usuario al no poder hacer uso de su vista, no puede interactuar con cualquier interfaz, es por esto que a través de la voz, podrá hacer uso de la aplicación. Como esta aplicación se trata de guiar al usuario hacia su destino, a través de la voz producida por la aplicación, el usuario podrá orientarse, y así llegar a su destino.

**API “google map”:** Con esta herramienta podemos acceder al mapa de google, y en esta agregar funciones, por ejemplo, la localización del móvil, agregar marcadores de algún lugar, etc. Esta herramienta va a ser esencial para guiar al usuario, ya que a través del mapa se puede saber qué calles o que dirección se debe dirigir el usuario.

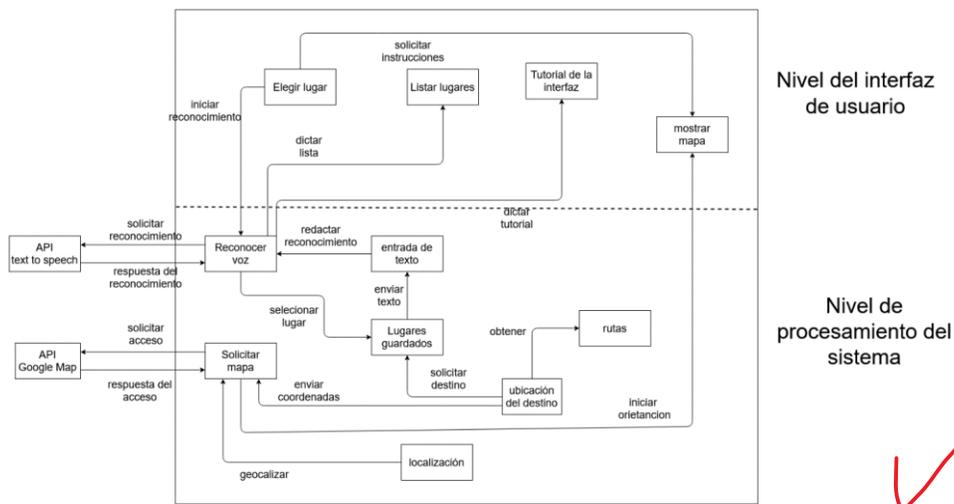
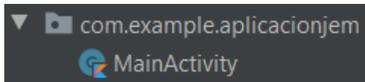


Ilustración 15 Modelo de implementación

Comentado [u4]: Excelente.. no se refleja en los casos de uso??

### 9.3 Módulos Implementados

- Módulo de interfaz gráfica



*Ilustración 16 Módulo de interfaz gráfica*

Toda la implementación de eventos que componen a la interfaz de usuario se definen mediante el archivo MainActivity.kt, dentro de dicho archivo se mantiene los módulos necesarios para poder tener una óptima comunicación usuario y aplicación.

Los eventos que tiene la interfaz de usuario tratan de.

```
import android.speech.tts.TextToSpeech
```

*Ilustración 17 Librería TextToSpeech*

Pasar de texto a voz, esto es posible mediante la librería que muestra la ilustración [16]. Mediante esta librería se puede hacer uso del parlante del dispositivo móvil permitiendo que el usuario reciba tanto el audio de rótulo de los botones como las instrucciones necesarias para poder llegar a una ubicación.

```
import android.speech.RecognizerIntent
```

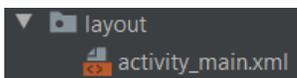
*Ilustración 18 Librería RecognizerIntent*

Pasar de voz a texto, esto es posible mediante la librería que muestra la ilustración [17]. Mediante esta librería se puede acceder al micrófono del dispositivo móvil permitiendo que el usuario mediante la pulsación de un botón pueda hablar y se guarde en forma de texto lo que dijo el usuario.

```
import android.hardware.*
```

*Ilustración 19 Librería hardware*

Activar audio con el movimiento del teléfono, esto es posible mediante la librería que muestra la ilustración [18], Mediante esta librería se puede acceder al giroscopio del dispositivo móvil permitiendo que este reciba un instructivo de la interfaz de usuario mediante un movimiento simple. Para poder hacer uso de este evento no es necesario seleccionar un botón de la interfaz solo se mueve el dispositivo móvil en un ángulo de 90° hacia la derecha y lo volvemos a su posición original, a continuación, se escuchará dicho instructivo.



Mediante el archivo activity\_main.xml se define el diseño que tendrá la aplicación. En la ilustración [19], se puede visualizar como es el diseño de la interfaz.



*Ilustración 20 Diseño interfaz de usuario*

- Módulo de entrada y salida de audio

El módulo de entrada y salida de audio, que ya se nombró en el módulo de interfaz de usuario, es posible mediante las librerías que se ilustran en la imagen x y x'. Luego de

que se hace uso de las librerías se define, tanto para el paso de texto a voz como para el paso de voz a texto, lo siguiente.

### Entrada de audio.

Esto es posible mediante las funciones `speak()` y `onActivityResult()` ilustradas en la imagen [20] y [21], mediante la función `speak` se implementa la activación de permisos para poder hacer uso del capturador de voz, creando una instancia "Intent" llamada `btnAction` y accediendo a las propiedades que este le brinda, también, la función `speak` contiene un `try catch` que nos permiten recibir el audio del usuario y capturarlo para su uso posterior. La función `onActivityResult()` nos permite hacer la conversión del audio del usuario a texto, y luego ser almacenado en el en "val result" para posteriormente ser utilizado.

```
private fun speak(){
    val btnAction = Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
    btnAction.putExtra(
        RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM
    )
    btnAction.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault())
    btnAction.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "hola habla alguna cosa")

    try {
        startActivityForResult(btnAction, REQUEST_CODE_SPEECH_INPUT)
    } catch (e: Exception){
        Toast.makeText(context: this, e.message, Toast.LENGTH_SHORT).show()
    }
}
```

Ilustración 21 Función `speak`

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    when(requestCode){
        REQUEST_CODE_SPEECH_INPUT -> {
            if (resultCode == Activity.RESULT_OK && null != data) {
                val result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS)
            }
        }
    }
}
```

Ilustración 22 Función onActivityResult

### Salida de audio.

Para lograr la salida de texto audio, se necesita de las funciones onInit() y pressTextToSpeech() ilustradas en la imagen [22] y [23], mediante la función onInit se establecen las condiciones iniciales del paso de texto a voz, definiendo el tipo de lenguaje que se utilizara para reproducir el audio del texto, además, se establecen las condiciones de posibles fallos a la hora de hacer la conversión. La función pressTextToSpeech, recibe un texto, que luego se reproducirá por el parlante del dispositivo móvil.

```
override fun onInit(status: Int) {
    if(status == TextToSpeech.SUCCESS) {
        val result = tts!!.setLanguage(Locale(language: "spa", country: "MX"))//ESP
        if(result == TextToSpeech.LANG_MISSING_DATA || result == TextToSpeech.LANG_NOT_SUPPORTED){
            Log.e(tag: "TTS", msg: "el lenguaje especificado no es soportado")
        }
        else{
            btnElegir.isEnabled = true
            btnListar.isEnabled = true
        }
    } else {
        Log.e(tag: "TTS", msg: "inicialización fallida")
    }
}
```

Ilustración 23 Función onInit

```
private fun processTextToSpeech(text: String)
{
    tts!!.speak(text, TextToSpeech.QUEUE_FLUSH, params: null, utteranceId: "")
}
```

Ilustración 24 Función processTextToSpeech

- Módulo de mapa

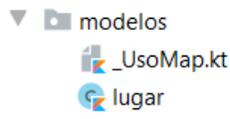


Ilustración 25 Modulo de mapa

En la carpeta “modelos” estan los archivos “\_UsoMap.kt” y “lugar.kt”.

```
import java.io.Serializable

data class UsoMap(val title: String, val places: List<lugar>): Serializable
```

Ilustración 26 Clase UsoMap

En el archivo “\_UsoMap” se crea la clase UsoMap, donde éste guarda los lugares que el usuario llama como destino. Tenemos como atributo a “title” que seria el titulo de lugar, “description” una breve descripción del lugar y la llamada a la lista de las coordenadas que se quiere llamar que están en el archivo “lugar”.

```
import java.io.Serializable

data class lugar(val title: String, val description: String, val latitude: Double, val longitude: Double) : Serializable
```

Ilustración 27 Clase lugar

En el archivo “lugar.kt” se tiene la clase lugar donde se guarda el titulo de un lugar en especifico, description de ese lugar en especifico, la latitude y longitude de la ubicación de dicho lugar. También al ser una clase serializable, sirve solamente para especificar

que todo el estado de un objeto instanciado podrá ser escrito o enviado en la red como una trama de bytes, al ser “lugar” serializable “UsoMap” también debe ser serializable.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    UsoMaps = generateSampleData().toMutableList()
    rvMaps.LayoutManager = LinearLayoutManager( context: this)

    mapAdapter = MapsAdapter( context: this, UsoMaps, object: MapsAdapter.OnClickListener{
        override fun onitemClick(position: Int) {
            Log.i(TAG, msg: "onItemClick $position")
            val intent = Intent( packageContext: this@MainActivity, DisplayMapActivity::class.java)
            intent.putExtra(EXTRA_USER_MAP, UsoMaps[position])
            startActivity(intent)
        }
    })
}
```

Ilustración 28 Función onCreate

Ahora en el archivo “MainActivity.kt” tenemos la función “onCreate” donde primero muestra la interfaz “activity\_main” en esa interfaz se muestra la lista de los lugares que están disponible para el usuario como destino, y este puede dirigirse con un click en el lugar que se quiere, pero para este proyecto el usuario no tendrá interacción de esta interfaz(figura: x).

Existe el archivo “displayMapActivity” donde estan ubicados las funciones relacionado con el mapa, y para poder acceder a este archivo, se asigno la funcion “onitemClick(position: Int)”.



Ilustración 29 Asignación lugares de destino

En la figura [29], se puede observar cómo se asigna los lugares que son los destinos posibles para el usuario, estos lugares son llamados por el usuario, donde a este se le muestra la ruta correspondiente para llegar al lugar elegido.

```

private fun generateSampleData(): List<Usomap> {
    return listOf(
        Usomap(
            title: "Santa isabel",
            listOf(
                lugar( title: "Santa Isabel (renato roca)", description: "Many long nights in this basement", latitude: -18.464886, longitude: -70.291856)
            )
        ),
        Usomap( title: "Farmacia vida sana",
            listOf(
                lugar( title: "Farmacia amador neeghme", description: "farmacia cercana", latitude: -18.438240, longitude: -70.290330)
                //Lugar("Banchi", "Family visit + wedding!", 23.34, 85.31),
                //Lugar("Singapore", "Inspired by \"Crazy Rich Asians\"", 1.35, 103.82)
            )
        ),
        Usomap( title: "Mercado"
    )
}

```

Ilustración 30 Función generateSampleData

En la figura [30], se observa que en el archivo "DisplayMapActivity.kt" en la función "onCreate(savedInstanceState:Bundle?)" se inicia su instancia en la interfaz "activity\_display\_map" donde está ubicado el mapa.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_display_map)

    usoMap = intent.getSerializableExtra(EXTRA_USER_MAP) as UsoMap

    supportActionBar?.title = usoMap.title
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    val mapFragment : SupportMapFragment = supportFragmentManager
        .findFragmentById(R.id.map) as SupportMapFragment
    mapFragment.getMapAsync(this)

    fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
}
```

Ilustración 31 Función onCreate

Para agregar una ruta hacia el marcador, se debe agregar las coordenadas que den hacia el marcador, y estas coordenadas se van a unir a través de una línea en el mapa. Ahora, en la figura x podemos observar un "if" donde se confirma si el lugar elegido por el usuario es la "Farmacia vida sana", si es así, va a entrar a la función del "if", y se mostrará la ruta hacia ese destino, los "LatLng" son las coordenadas que se van uniando con respecto a su siguiente coordenada, y de esa manera va formando la ruta.

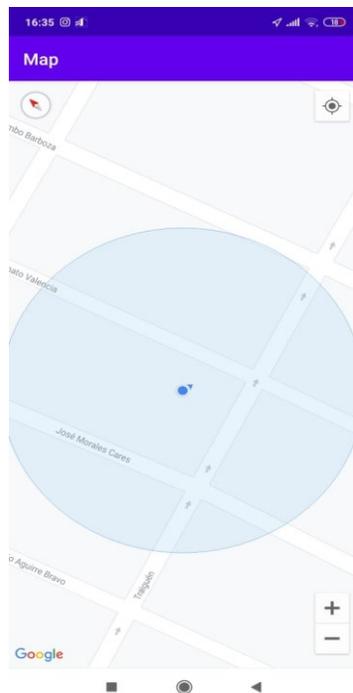
L

```
if(usoMap.title=="Farmacia vida sana") {  
    val polyline2 : Polyline! = googleMap.addPolyline(  
        PolylineOptions()  
            .clickable(true)  
            .add(  
                LatLng(-18.435473, -70.286098),  
                LatLng(-18.435716, -70.285958),  
                LatLng(-18.435858, -70.286399),  
                LatLng(-18.435995, -70.286884),  
                LatLng(-18.436206, -70.287369),  
                LatLng(-18.437073, -70.287041),  
                LatLng(-18.437947, -70.286675),  
                LatLng(-18.438787, -70.286496),  
                LatLng(-18.438715, -70.287206),  
                LatLng(-18.438622, -70.287684),  
                LatLng(-18.438513, -70.288532),  
                LatLng(-18.438502, -70.288918),  
                LatLng(-18.438398, -70.289259),  
                LatLng(-18.438238, -70.290304)  
            )  
        )  
    }  
}
```

Ilustración 32 Cordenadas agregadas

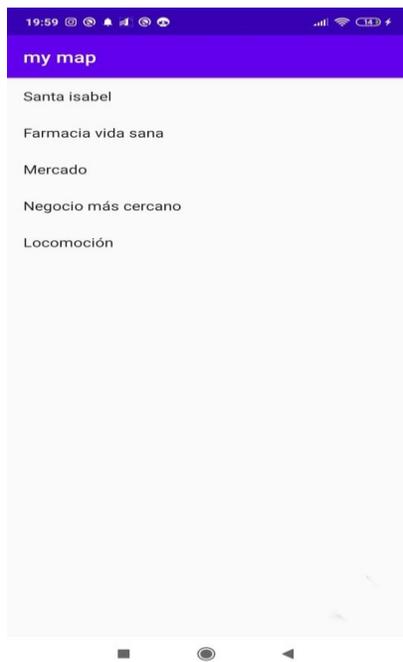
## 9.4 Reporte de revisión

**Primera prueba:** Como primer paso del proyecto, se implementó el mapa de Google a través de su API, después se agregó la función del GPS para el usuario, es decir en el mapa se manifiesta la ubicación del móvil del usuario. Luego esto se llevo a prueba usando la aplicación que nos entregó Android Studio y el teléfono móvil, esta prueba resultó satisfactoria.



*Ilustración 33 Primera prueba*

**Segunda prueba:** Para poder solicitar la ruta y el marcador del lugar que el usuario quiere solicitar como destino, se hizo la interfaz que muestra la figura [33], en donde el usuario elige el lugar.



*Ilustración 34 Segunda prueba*

**Tercera prueba:** En esta prueba, se implementó la ruta, para la ruta se agregó las coordenadas correspondiente para hacer el camino hacia el marcador del destino.

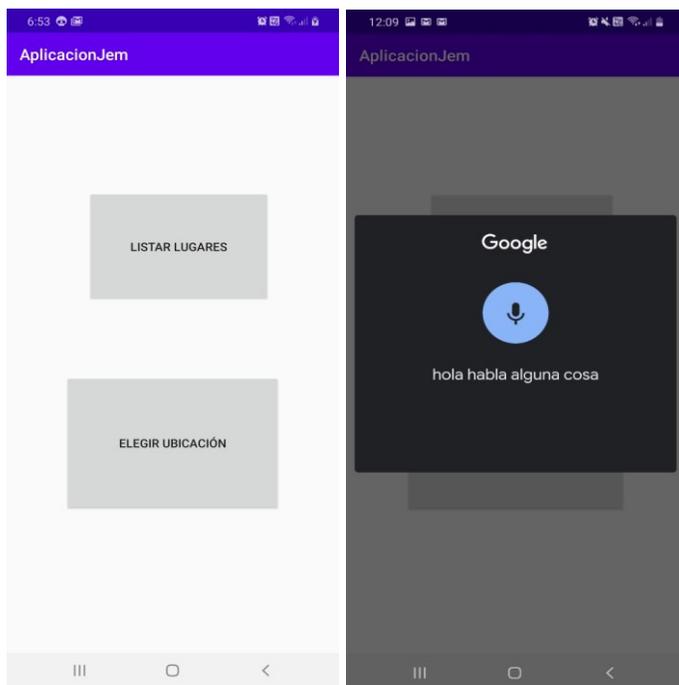


*Ilustración 35 Tercera prueba*

**Cuarta prueba:** Se probó el funcionamiento de los eventos de audio de botones, quedando en evidencia el funcionamiento correcto de este mismo.

**Quinta prueba:** Se prueba la captura de audio del usuario tras seleccionar el botón “elegir lugar”, al seleccionar el botón elegir lugar se puede ver como guarda en texto el audio del usuario logrando cumplir con dicho funcionamiento.





*Ilustración 36 Quinta prueba*

**Sexta prueba:** Se prueba el sensor de giroscopio del dispositivo móvil, observando que, logra reproducir audio con el movimiento del dispositivo móvil.

## 11. Problemas encontrados

Durante el desarrollo de la aplicación se usó la técnica dividir para conquistar, cuando se quiso probar las partes que se habían dividido, surgieron problemas, ya que la aplicación no ejecutaba y se originaron múltiples errores.

## 12. Soluciones propuestas

Eventualmente el grupo decidió hacer las partes divididas desde cero con la ayuda del encargado de la parte deseada de la aplicación.

## 13. Conclusión

Mediante el desarrollo del informe que se trabajó durante el semestre, se puede destacar cuatro partes importantes las cuales fueron de gran importancia para el equipo. Como primera parte se destaca, la búsqueda de información necesaria para poder ofrecer una aplicación para el usuario invidente, además de, confrontar ideas adversas de cada integrante con el fin de lograr dicha aplicación.

Como segunda parte se destaca la división del trabajo, ya que, aplicando esta estrategia se logra equilibrar un trabajo complejo en sub trabajos, esto fue clave para realizar la interfaz y la geolocalización en tiempo real del usuario, puesto que, se logró trabajar en paralelo ambos trabajos.

Como tercera parte se destaca la descripción de la arquitectura, esta parte, permite mostrar la funcionalidad de la aplicación y cómo logra cada una de estas funcionalidades, muestra el paso a paso que tendrá que realizar la aplicación en conjunto con los datos proporcionados por el usuario para poder cumplir con la principal funcionalidad de la aplicación (guiar al invidente).

Finalmente se concluye que el trabajo realizado durante el semestre, a pesar de las complicaciones y de no alcanzar todo lo planificado nos fue de gran utilidad ya que da evidencia de las falencias y fortalezas que tiene cada integrante del equipo, además de reforzar cada conocimiento de cursos anteriores, puesto que, se mantiene un constante seguimiento del desarrollo de cada grupo de trabajo, ya sea con correcciones, retroalimentación y consejos proporcionados por el profesor a cargo.



## 14. Referencias

- [1] J. C. G. Diego Aracena Pizarro, Escritor, *Sistema Asistencial para invidentes mediante visión computacional*. [Performance]. Departamento de ingeniería en computación e informática, 2020.
- [2] J.-P. Lang, «Redmine,» area de ingenieria en computación e informatica, 2006. [En línea]. Available: <http://pomerape.uta.cl/redmine>. [Último acceso: 29 octubre 2020].
- [3] J. G. M. B. M. J. N. María José Ortín, de *El Modelo del Negocio como base del Modelo de Requisitos*, mursia, pp. 10-13.

## 15. Anexos

Ilustración 1 Escenario .....	7
Ilustración 2 Mecanismos de comunicación .....	11
Ilustración 3 Carta Gantt.....	14
Ilustración 4 Interfaz de usuario .....	16
Ilustración 5 Interfaz de usuario evento .....	17
Ilustración 6 Interfaz funcionalidad de ubicaciones .....	18
Ilustración 7 Interfaz funcionalidad de recorridos .....	19
Ilustración 8 Descripción de la arquitectura.....	20
Ilustración 9 Modelo caso de uso .....	22
Ilustración 10 Diagrama de secuencia elegir lugar.....	26
Ilustración 11 Diagrama de secuencia listar lugar.....	27
Ilustración 12 Diagrama de secuencia reproducir tutorial .....	27
Ilustración 13 Modelo de clases .....	28
Ilustración 14 Plan de integración.....	29
Ilustración 15 Modelo de implementación .....	30
Ilustración 16 Módulo de interfaz gráfica.....	31
Ilustración 17 Librería TextToSpeech .....	31
Ilustración 18 Librería RecognizerIntent.....	31
Ilustración 19 Librería hardware .....	31
Ilustración 20 Diseño interfaz de usuario .....	32
Ilustración 21 Función speak.....	33
Ilustración 22 Función onActivityResult.....	34
Ilustración 23 Función onInit .....	34
Ilustración 24 Función processTextToSpeech .....	35
Ilustración 25 Modulo de mapa .....	35
Ilustración 26 Clase UsoMap .....	35
Ilustración 27 Clase lugar .....	35
Ilustración 28 Función onCreate.....	36

---

Ilustración 29 Asignación lugares de destino.....	37
Ilustración 30 Función generateSampleData.....	37
Ilustración 31 Función onCreate.....	38
Ilustración 32 Cordenadas agregadas .....	39
Ilustración 33 Primera prueba .....	40
Ilustración 34 Segunda prueba .....	41
Ilustración 35 Tercera prueba .....	42
Ilustración 36 Quinta prueba.....	43
Tabla 1 Entregables del proyecto .....	10
Tabla 2 Planificación de Estimaciones.....	12
Tabla 3 Planificación de la Gestión de riesgos .....	15
Tabla 7 Especificación de requerimientos.....	21
Tabla 4 Caso de uso elegir lugar.....	23
Tabla 5 Caso de uso listar lugares .....	24
Tabla 6 Caso de uso Reproducir tutorial .....	25