GPS a base de vibraciones

Lucas Castronuovo

Resumen—

I. Introducción

I-A. Idea

Mi idea para el proyecto de investigación es la exploración de una aplicación móvil GPS que funcione a base de vibraciones. Esto le permitirá a los usuarios que no desean mirar su celular (por miedo a robos) saber hacia dónde ir o para aquellas personas con discapacidad auditiva o ceguera, poder dirigirse hacia su destino, de una forma dinámica, accesible y eficiente.

I-B. Investigación bibliografica

[1] Es un artículo que es muy similar a mi idea, pero con un enfoque especial para los invidentes y con el uso de un dispositivo externo. Contiene muchas referencias.

[2] Este artículo trata el mismo objetivo que el anterior, pero es más actual y también contiene muchas referencias.

[3] Documento que profundiza en el GPS de manera general. Cuenta con muchas referencias y fue muy citado en otras investigaciones.Es bastante viejo.

[4] Conferencia que profundiza en GPS. Bastante viejo

[5] Artículo que trata de la precisión del GPS presente en los teléfonos celulares. Artículo viejo pero bastante citado y con referencias.

[6] Documento que brinda una guía de cómo desarrollar una aplicación móvil para el uso de GPS.

[7] Libro para aprender sobre Android Studio.

[8] Artículo que habla sobre el desarrollo de la apreciación musical en niños sordos mediante vibraciones de una aplicación móvil. Aunque mi investigación no trate específicamente sobre eso, respalda el uso de vibraciones para la estimulacion en personas con discapacidad auditiva.

[9] Conferencia que destaca estrategias de accesibilidad en aplicaciones móviles para personas no videntes.

[10] Artículo que explica el uso de un dispositivo móvil para que una persona con discapacidad auditiva pueda reconocer, mediante vibraciones, distintos sonidos que ocurren en el ambiente, incluido el sonido de un teléfono celular.

[11] Artículo periodístico que profundiza sobre el robo de celulares en la vía pública. Destaca medidas de protección.

[12] Otro artículo periodístico que trata sobre la gran cantidad de hurtos del dispositivo móvil. Aconseja al lector los pasos a seguir tras sufrir el robo.

[13] Artículo que brinda información sobre la accesibilidad de personas con discapacidades.

Referencias

[1] R. Velázquez-Guerrero, E. Pissaloux, C. Del-Valle-Soto, M.-Á. Carrasco-Zambrano, A. Mendoza-Andrade, and J. Varona-Salazar, "Movilidad para invidentes utilizando el gps del teléfono inteligente y un dispositivo táctil vestible." *DYNA-Ingeniería e Industria*, vol. 96, no. 1, 2021.

[2] R. V. Guerrero, R. T. Gutiérrez, and C. D. V. Soto, "Zapato con gps para apoyar la autonomía de personas con discapacidad visual (gps shoe to support the autonomy of people with visual disabilities)," *Pistas Educativas*, vol. 45, no. 145, 2023.

[3] A. Pozo-Ruz, A. Ribeiro, M. García-Alegre, L. García, D. Guinea, and F. Sandoval, "Sistema de posicionamiento global (gps): descripción, análisis de errores, aplicaciones y futuro," *ETS ingenieros de Telecomunicaciones. Universidad de Malaga*, pp. 1–9, 2000.

[4] C. Rizos, "Trends in gps technology & applications," in *2nd International LBS Workshop*, 2003.

[5] P. A. Zandbergen and S. J. Barbeau, "Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones," *The Journal of Navigation*, vol. 64, no. 3, pp. 381–399, 2011.

[6] B. Andreea-Maria, "Development of an android location app based on gps signals," *BULETIN ŞTIINŢIFIC SUPLIMENT*, p. 139, 2021.

[7] GoalKicker.com, *Android:Notes for proffesionals*, 2018.

[8] A. E. F. Cortázar and J. R. Rojano-Cáceres, "Desarrollo de la apreciación musical en niños sordos mediante estimulación por vibraciones desde una aplicación móvil," *Revista Brasileira de Informática na Educação*, vol. 29, pp. 1007–1037, 2021.

[9] P. Santana Mansilla, G. E. Lescano, and R. Costaguta, "Accesibilidad de aplicaciones móviles para discapacitados visuales: problemas y estrategias de solución," in *II Simposio Argentino sobre Tecnología y Sociedad (STS)-JAIIO 44 (Rosario, 2015)*, 2015.

[10] M. Yağanoğlu and C. Köse, "Real-time detection of important sounds with a wearable vibration based device for hearing-impaired people," *Electronics*, vol. 7, no. 4, p. 50, 2018.

[11] M. Rufino, "Robo de celulares en amba: ya son 10.000 los ilícitos por día y crece el riesgo de estafas," *El Cronista*, 2024.

[12] M. Bianchi, "Robo de celulares. desaparece uno cada 30 segundos: qué hacen los delincuentes con los teléfonos y qué deben hacer las víctimas," *La Nación*, 2023.

[13] J. A. O. Zambrano, Y. T. C. Cujilan, M. d. C. T. Bernabe, and K. N. L. Castillo, "La usabilidad y accesibilidad: Estudio de guías para aplicaciones en dispositivos móviles," *Dominio de las Ciencias*, vol. 3, no. 3, pp. 1181–1209, 2017.

II. Robo de celulares en la vía pública

En Argentina, en 2022, se bloquean 1.039.727 dispositivos móviles por diversos motivos, como extravío, robo y hurto. En 2023, se reporta la desaparición de 3.000 dispositivos móviles por día, alcanzando un total de 731.292 hasta agosto. En la Ciudad Autónoma de Buenos Aires, durante ese mismo año, se registran aproximadamente 3.000 denuncias mensuales por robo de celulares, lo que equivale a 113 denuncias diarias[12]. En 2024, según BTR Consulting, el 27 % de los robos en en el país corresponde a hurtos de dispositivos móviles en vías públicas, con un promedio de 10.000 dispositivos arrebatados diariamente, es decir, 416 por hora[11].

El país registra niveles más altos de robos de dispositivos móviles en comparación con países vecinos. Según datos de la Policía Nacional de Perú, se reportan 4.800 robos de dispositivos móviles por día, lo que equivale a 200 por hora.

En Colombia, la cifra es menor, con 3.500 casos diarios, es decir, 146 por hora[11].

Los principales objetivos de los robos son adolescentes y jóvenes adultos, quienes constituyen los mayores consumidores de tecnología[11]. Se observa que los delincuentes prefieren realizar el hurto cuando las víctimas están utilizando el dispositivo, ya que en ese momento el teléfono suele estar desbloqueado[12].

El objetivo principal de este delito es la reventa de dispositivos móviles o sus partes en el mercado negro, a un precio entre un 30 % y 40 % inferior al valor promedio de venta. Este es el método mediante el cual los delincuentes obtienen ingresos. Para facilitar el proceso, se utiliza una herramienta conocida como ”Sigma Box”, que permite conectar el dispositivo robado, borrar su contenido y restaurarlo a su configuración de fábrica. Además, es común que se acceda al dinero de las aplicaciones de homebanking y billeteras virtuales de las víctimas, y se lleven a cabo estafas y extorsiones[12].

BTR Consulting señala que existe una alta exhibición de dispositivos móviles tanto en el transporte público como en la vía pública. La recomendación es evitar su uso en estos entornos y no guardarlos en los bolsillos traseros del pantalón o en los bolsillos de las mochilas[11].

III. Antecedentes de la investigación

III-A. Movilidad para invidentes utilizando el GPS del teléfono inteligente y un dispositivo táctil vestible

Se desarrolla un sistema diseñado para asistir a personas invidentes o con dificultades visuales en su movilidad en entornos urbanos. Este sistema utiliza el sensor GPS del dispositivo móvil para captar la ubicación del usuario en tiempo real y envía instrucciones sobre la dirección a seguir a través de patrones de vibración generados por una interfaz táctil integrada en el calzado. Para este proyecto, se realizan dos experimentos.

- En el primer experimento, participan de forma voluntaria 20 estudiantes jóvenes sin discapacidades de la Universidad Panamericana de México. Tras recibir instrucciones generales y familiarizarse con el dispositivo, se les da la opción de elegir la frecuencia de vibración de la interfaz; todos optan por 55 Hz, que es la frecuencia máxima disponible. Se evalúa únicamente la eficacia de percepción de las vibraciones, con un asistente que maneja y envía las instrucciones de vibración sin utilizar el dispositivo móvil ni el programa de navegación. La interfaz cuenta con cuatro puntos de contacto, que indican las instrucciones de avanzar (A), retroceder (R), girar a la izquierda (I), girar a la derecha (D) y detenerse (S) mediante patrones de vibración. Los resultados muestran una eficacia del 100 % para A, 97,78 % para R, 88,89 % para I, 90 % para D y 100 % para S.
- En el segundo experimento, se prueba el sistema en situaciones reales con la participación de dos hombres adultos invidentes. El participante A es ciego de nacimiento, mientras que el participante B desarrolla retinitis pigmentaria a una edad temprana. Ambos participantes no presentan discapacidades en el tacto de sus pies ni en sus funciones cognitivas. En este experimento, se lleva a cabo un entrenamiento previo para que los participantes transiten por entornos urbanos con poca concurrencia vehicular, enfrentándose a objetos estáticos y dinámicos, como personas en movimiento. Los participantes informan que el sistema es intuitivo y requiere de poca concentración.

Los resultados obtenidos demuestran que el sistema es operacional.[1]

III-B. Desarrollo de la apreciación musical en niños Sordos mediante estimulación por vibraciones desde una aplicación móvil

Se desarrolla una aplicación móvil para niños que permite, mediante vibraciones generadas por el dispositivo y el sentido del tacto del usuario con discapacidad auditiva, proporcionar estimulaciones que simulan la apreciación musical. Esto facilita la enseñanza de conceptos musicales sin depender del sentido del oído.

La aplicación cuenta con un video de bienvenida de una persona que, mediante lenguajes de señas, introduce al usuario sobre el fin de la aplicación, el contenido que cuenta la aplicación y como utilizar cada función:

- Función ”Piano”: Consiste en teclas de piano el cual cada nota cuenta con unn color asociado. Al presionar cada tecla, el dispositivo movil libera una correspondiente vibración y sonido.
- Función .ªprender las notas”: Consiste en que el usuario se familiariza con las notas musicales a través de niveles con desafíos. Se inicia con un video introductorio que explica, mediante lenguajes de señas, la temática de la función. Primero, se indican las notas a practicar en el nivel correspondiente, permitiendo que el usuario asocie cada nota mediante las vibraciones. Luego, otro video presenta un desafío en el cual el usuario debe identificar la nota correspondiente a la vibración liberada para ayudar a un animal a alcanzar la meta.
- Función ”Descubre el ritmo”: Mediante un video, indica al usuario que las pelotas se moverán de forma aleatoria. Debe seleccionar la pelota que se mueve según el patrón rítmico de las vibraciones que se liberan a intervalos regulares.

En sesiones presenciales, se introducen conceptos de música a 12 niños, de los cuales 5 participan en un experimento para probar la aplicación móvil. La evaluación demuestra mejoras en la percepción de las vibraciones a medida que los niños continúan utilizando la aplicación, despertando su interés por seguir aprendiendo conceptos musicales[8].

III-C. Accesibilidad de aplicaciones móviles para discapacitados visuales: problemas y estratégias de solución

La investigación destaca que los avances tecnológicos han dificultado la utilización de las aplicaciones móviles por parte de discapacitados visuales. En esta conferencia se mencionan las barreras de accesibilidad que enfrentan y posibles soluciones que se pueden implementar para mitigar las complejidades que ellos enfrentan:

- Solución a problemas de accesibilidad con síntesis de voz: Hay dos factores que dificultan la comprensión del sintetizador de voz: Los sonidos emitidos de otras aplicaciones y la privacidad. Para el primer caso se recomienda establecer una función que desactive estos sonidos de manera automática o que el usuario pueda configurar el audio. Para la privacidad se brinda la solución de poder modificar el sonido del sintetizador de voz para que los sonidos sean incomprensibls para otras personas.
- Solución a problemas de accesibilidad con el uso de mapas: Para la asistencia en la exploración de ambientes exteriores se destacan las aplicaciones TouchOver Map y PointNav. La primera permite permite que los usuarios no videntes obtengan información geoespacial mediante representaciones no visuales, utilizando retroalimentación auditiva y táctil. El usuario coloca su dedo sobre la pantalla y, cuando toca una calle, se menciona el nombre de la misma mientras el dispositivo vibra de forma continua. Si el dedo se aleja de la calle, el dispositivo deja de vibrar, pero el nombre de la calle sigue mencionándose, lo que permite que la persona vuelva a localizar la calle correspondiente. Este sistema guía al usuario a lo largo de una ruta y le proporciona información de localización precisa.

  La segunda aplicación, PointNav, permite al usuario explorar y navegar mediante retroalimentación por vibraciones y voz. La interfaz de la aplicación consta de 9 botones, los cuales son identificados a través de vibraciones mientras el usuario desplaza su dedo entre ellos. Al detenerse en un botón, una voz anuncia su nombre, y la selección se realiza levantando el dedo de la pantalla. Entre los botones disponibles se encuentra el botón .[Es]canear", que permite al usuario apuntar su dispositivo móvil en la dirección deseada. Cuando el dispositivo detecta un Punto de Interés (POI) dentro de un rango predefinido, se genera una vibración corta, seguida por la indicación del nombre y la distancia del POI. Otro botón destacado es "Guía", que dirige al usuario hacia el POI seleccionado utilizando ángulos específicos para las instrucciones recto", "girar a la derecha", "girar a la izquierdaτ "dar la vuelta". Las indicaciones se proporcionan tanto mediante señales auditivas como a través de diferentes patrones de vibración. Al llegar al destino, el dispositivo emite una secuencia final de vibraciones y un mensaje de voz que confirma la llegada.
- Solución a problemas de accesibilidad con el teclado: Para este caso se plantean dos estrategias: la realmentación vibro-tactil y el uso de marcadores táctiles. Se mencionan proyectos que trajeron la solución mediante distintas implementaciones: Un trabajo busca la organización de funciones y botones en la interfaz, además de como combinar mensajes de voz con feedback vibrotáctil. La aplicación consiste en 4 puntos de referencias, ubicados en las esquinas de la pantalla del dispositivo, que están siempre presentes en el teclado para que el usuario obtenga información cuando lo requiera. En la esquina superior izquierda es el botón .[Ex]it/Back"para volver al paso previo - En la esquina superior derecha se encuentra el botón "Position.[el] cual permite conocer el estado de la posición actual - En la esquina inferior izquierda se encuentra el botón Repeat"que permite leer un número de teléfono - En la esquina inferior derecha se encuentra el botón "Done"que permite confirmar una elección. Además, para ayudar al usuario a percibir los números en el teclado virtual, se utiliza la vocalización de número junto con un patrón específico para los números pares, impares y otro para el número 5. Otro trabajo se dedicó a mejorar la accesibilidad del usuario no vidente mediante combinaciones de mensajes de voz, audios y vibraciones insertadas en código fuente de la IU, de manera que se permita combinaciones personalizables, concepto conocido como CAC, asistiendo al no vidente en la exploración y reconocimiento del contenido sin altera la organización origial de la IU.
- Solución a problemas de accesibilidad con reconocimiento de voz: La solución se basa en la aplicación VoiceMail el cual permite a los usuarios no videntes enviar y recibir mails grabando su voz y en vez de convertir la voz en texto, el sistema envía directamente el mensaje de voz grabado a la dirección de e-mail correspondiente como un archivo adjunto. De esta manera se evitan los problemas que surjen por dificultados de la precisión del reconocimiento de voz, revisió y edición del texto reconocido e interferencias y ruidos ambientales.

[9]

III-D. Detección en tiempo real de sonidos importantes con un dispositivo portátil basado en vibraciones para personas con discapacidad auditiva

Se desarrolla un dispositivo para informar a personas con discapacidad auditiva sobre sonidos importantes del entorno, mediante vibraciones que permiten identificar el tipo de sonido. El sistema opera en tiempo real y utiliza el método de huellas dactilares de audio, logrando una precisión del 98 % para identificar el timbre de una puerta, 99 % para alarmas y timbres telefónicos, entre el 91 % y 97 % para otros sonidos como bocinazos, frenos, ladridos de perros y voces humanas. La huella dactilar de audio es un breve resumen perceptual de un archivo de audio.

El dispositivo fue probado 100 veces al día durante 100 días, tanto en cinco personas con discapacidad auditiva como en 50 personas oyentes, cuyos oídos estaban cubiertos con auriculares para simular pérdida auditiva. Los resultados mostraron

que las personas sordas valoraron la claridad del sistema en un 90 %, su utilidad en un 97 %, y un 100 % de ellas expresó su disposición a seguir usando el dispositivo. Este estudio tiene como objetivo mejorar la calidad de vida de las personas con discapacidad auditiva[10].

IV. Marco teórico

IV-A. GPS y A-GPS

IV-A1. GPS: : El Sistema de Posicionamiento Global (GPS) consiste en una constelación de 24 satélites que orbitan a 12.000 millas de la superficie terrestre y que son alimentados por energía solar. Estos satélites son operados por el Departamento de Defensa de los Estados Unidos en colaboración con otras entidades gubernamentales.

El funcionamiento del GPS se basa en la transmisión de señales de radio desde los satélites, las cuales son recibidas por el receptor GPS del dispositivo móvil. A través de estas señales, se calcula la posición del receptor, expresada en coordenadas de latitud y longitud.

IV-A2. A-GPS: : El Sistema de Posicionamiento Global Asistido (A-GPS) es una tecnología utilizada por los teléfonos móviles. La red del dispositivo brinda información útil para ayudar al receptor GPS a calcular la posición del usuario con mayor precisión y rapidez. Reduce el tiempo de adquisición de la señal al eliminar secciones del espacio de búsqueda de la señal.

La principal diferencia entre el Sistema de Posicionamiento Global (GPS) y el Sistema de GPS Asistido (A-GPS) radica en la precisión y velocidad de obtención de la información de posicionamiento. El GPS tradicional es más preciso, ya que presenta un menor margen de error en la determinación de la ubicación del usuario en comparación con el A-GPS. No obstante, el A-GPS permite obtener la información de posicionamiento de manera más rápida, especialmente en entornos complejos, como interiores o áreas urbanas densamente pobladas, donde el GPS tradicional puede requerir más tiempo para fijar la ubicación[5].

IV-B. Fuentes de error en los GPS

IV-B1. Perturbación ionosférica: : Las señales de radio del Sistema de Posicionamiento Global (GPS) experimentan alteraciones en su velocidad al atravesar la ionosfera, que está compuesta por partículas cargadas eléctricamente.

IV-B2. Fenómenos meteorológicos: : En la troposfera, el vapor de agua presente puede afectar las señales electromagnéticas, alterando su velocidad. Los errores que causan estas condiciones son similares en magnitud a los que se provocan por la ionosfera; sin embargo, en este caso, la corrección de dichos errores resulta más difícil.

IV-B3. Imprecisión en los relojes: : Los relojes atómicos de los satélites, a pesar de ser cuidadosamente ajustados y controlados, presentan ligeras desviaciones. De manera análoga, los relojes de los receptores también experimentan variaciones similares.

IV-B4. Interferencias eléctricas imprevistas: : Las interferencias eléctricas pueden causar correlaciones incorrectas en los códigos pseudoaleatorios o un redondeo inadecuado en el cálculo orbital. Aunque los errores significativos son fáciles de detectar, las pequeñas desviaciones pueden resultar en errores de hasta un metro, dificultando su identificación.

IV-B5. Error multisenda: : Las señales emitidas por los satélites pueden experimentar reflexiones antes de llegar al receptor. Para minimizar este tipo de error, los receptores modernos utilizan técnicas avanzadas de procesamiento de señal y antenas diseñadas específicamente. Sin embargo, la modelización de este error es compleja, ya que depende del entorno en el que se encuentre la antena GPS.

IV-B6. Interferencia "Disponibilidad Selectiva S/A": : Esta fuente de error constituye la más significativa y es introducida deliberadamente por las autoridades militares.

IV-B7. Topología receptor-satélites: : Es fundamental que los receptores consideren la geometría entre el receptor y los satélites visibles en el cálculo de distancias, dado que una disposición espacial específica puede afectar la precisión de las mediciones. Los receptores más sofisticados utilizan un factor multiplicativo para ajustar el error de medición de la distancia, conocido como "dilución de la precisión geométrica".

[3]

IV-C. La precisión del GPS en interacción con las fuentes

Al ocurrir estas fuentes de errores, se genera un cierto grado de incertidumbre. Es por eso que se determina una estimación de la posición, valor medio, dentro de un intervalo de tiempo con una determinada dispersion. El error se puede representar de diferentes maneras:

IV-C1. Error Cuadrático Medio: :

$$ECM = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

IV-C2. Distribución Normal: : Se utiliza cuando la mayoría de los errores se concentran alrededor del valor medio, y se pueden deducir probabilidades asociadas a estos errores en función de su dispersión.

IV-C3. Error Circular Probable: : Se define como el radio del círculo que contiene la estimación más probable de la posición.

[3]

IV-D. Accesibilidad para personas invidentes y sordas

IV-D1. Accesibilidad para personas invidentes:

- Lector de pantalla: Software que realiza la traducción de texto a sonido mediante un sintetizador de voz.
- Magnificador de pantalla: Tecnología que permite aumentar el tamaño de texto o imágenes.
- Tamaños de fuente ajustables: Función incorporada en algunos dispositivos móviles. Aumenta el tamaño de la fuente.
- Reconocimiento de voz: Herramienta que le permite al usuario, mediante un dispositivo de entrada, completar una función o control del dispositivo móvil a través de comandos de voz.
- Controles de contraste/brillo: Permite que el usuario pueda modificar el color de la pantalla y ajustar el brillo.
- Pantalla Braille: Hardware dedicado a la obtención y entrega de información electrónica desde una computadora al usuario, en formato Braille, a través de la lectura y su traducción[9].
- Entorno gráfico: Sustituye las letras y números por grafismos. Se cuenta con la posibilidad de ampliar los íconos.
- Asociador de contactos con imágenes y sonidos: Tecnología que permite relacionar un determinado sonido o imágen a cada persona para su identificación.
- Asociador de íconos con palabras: Permite que programas que traducen a voz el texto, puedan leer cada grafismo.
- VoiceOver: Tecnología que lee la pantalla y le brinda al usuario descripciones audibles de elementos y acciones en la aplicación[13].

IV-D2. Accesibilidad para personas sordas:

- Modulador del volumen del sonido: Tecnología que ecualiza sonidos.
- Alarmas visuales o vibradores: Herramienta que adapta funcionalidades como alarmas para la notificación de correos electrónicos, despertador, llamadas entrantes y citas de agenda.
- Videoconferencia: Permite la comunicación cara a cara utilizando gestos de señales, evitando el envío de textos.
- Subtítulos: Mediante texto, ayuda al usuario al seguimiento de una reproducción de video o película.
- Correo de voz: Mensajes de voz almacenados en el buzón de correo.
- Mono audio: El sonido de los canales izquierdo y derecho son combinados en una señal mono.

[13]

IV-E. Programación en Android y Android Studio

Android Studio constituye el entorno oficial para el desarrollo de aplicaciones en Android, fundamentado en la tecnología de JetBrains. Integra diversas herramientas que facilitan el proceso de desarrollo, tales como un sistema de compilación denominado Gradle y un emulador que permite la prueba de aplicaciones. Se adapta a múltiples plataformas de escritorio, incluyendo Mac, Windows y Linux[6].

V. Componentes

V-A. A-GPS del dispositivo móvil

La tecnología es utilizada para aplicaciones móviles que requieren información de ubicación en tiempo real, garantizando una disponibilidad y precisión elevadas de las correcciones de posición en diversas condiciones[5].

V-B. Android Studio para el desarrollo de la app

Es el entorno utilizado en [6] para el desarrollo de la aplicación móvil ”GpsCoordinates”.

VI. Implementación

VI-A. Declaración de variables

```
private static final int REQUEST_LOCATION_PERMISSION =
 1;

private static final int REQUEST_VOICE_RECOGNITION = 2;

private GoogleMap myMap;

private SearchView mapSearchView;

private Location currentLocation;

private FusedLocationProviderClient fusedLocationClient
 ;

private LocationCallback locationCallback;

private final String TAG = "MainActivity";

private final String API_KEY =
"AIzaSyBCYZFSfYNrjS0_rdIQlvBvMiZsv_KcDAk";

private Vibrator vibrator;

private ImageButton voiceSearchButton; // Boton para
 busqueda por voz
```

VI-B. Inicialización de vistas y servicios

```
mapSearchView = findViewById(R.id.mapSearch);

voiceSearchButton = findViewById(R.id.voiceSearchButton
 ); // Inicializa el boton de busqueda por voz

fusedLocationClient = LocationServices.
getFusedLocationProviderClient(this);

vibrator = (Vibrator) getSystemService(Context.
 VIBRATOR_SERVICE);
```

VI-C. Verificar permisos de ubicación al iniciar

```
if (ActivityCompat.checkSelfPermission(this, Manifest.
 permission.ACCESS_FINE_LOCATION) != PackageManager.
 PERMISSION_GRANTED)

{

    ActivityCompat.requestPermissions(this,
    new String[]{Manifest.permission.
    ACCESS_FINE_LOCATION},
    REQUEST_LOCATION_PERMISSION);

}

else
{

    getLastLocation();

}
```

VI-D. Verificar permisos de ubicación al iniciar

```
SupportMapFragment mapFragment = (SupportMapFragment)
  getSupportFragmentManager().findFragmentById(R.id.map);

if (mapFragment != null) {

     mapFragment.getMapAsync(this);

}
```

VI-E. Configuración del SearchView para búsquedas de texto

```
mapSearchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {

    @Override
    public boolean onQueryTextSubmit(String s) {
        searchLocation(s);
        return false;
    }

    @Override
    public boolean onQueryTextChange(String s) {
        return false;
    }
});
```

VI-F. Configuración del botón de búsqueda por voz

```
voiceSearchButton.setOnClickListener(v ->
  startVoiceRecognition());
```

VI-G. Configuración del callback para recibir actualizaciones de ubicación

```
 locationCallback = new LocationCallback()
 {

    @Override
    public void onLocationResult(LocationResult
 locationResult) {

        if (locationResult == null) {

            return;

         }

         for (Location location : locationResult.
 getLocations()) {

            currentLocation = location;

            updateMapWithCurrentLocation();

        }
    }
};
```

VI-H. Iniciar actualizaciones de ubicación periódicas

```
 startLocationUpdates();
```

VI-I. Método para iniciar el reconocimiento de voz

```
private void startVoiceRecognition() {

    Intent intent = new Intent(RecognizerIntent.
 ACTION_RECOGNIZE_SPEECH);

    intent.putExtra(RecognizerIntent.
 EXTRA_LANGUAGE_MODEL, RecognizerIntent.
 LANGUAGE_MODEL_FREE_FORM);

    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "
  Que  lugar estas buscando?");

    startActivityForResult(intent,
 REQUEST_VOICE_RECOGNITION);
}
```

VI-J. Método para manejar el resultado del reconocimiento de voz

```
@Override
protected void onActivityResult(int requestCode, int
 resultCode, Intent data) {

    super.onActivityResult(requestCode, resultCode,
 data);

    if (requestCode == REQUEST_VOICE_RECOGNITION &&
 resultCode == RESULT_OK) {

        ArrayList<String> results = data.
 getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS)
 ;

        if (results != null && !results.isEmpty()) {

            String spokenText = results.get(0);

            mapSearchView.setQuery(spokenText, true);
 // Establece el texto hablado en el SearchView

        }
    }
}
```

VI-K. Método para vibrar cuando el usuario debe girar a la derecha

```
private void vibrateRightTurn() {

    long[] pattern = {0, 300, 100, 300}; // Esperar 0
 ms, vibrar 300 ms, esperar 100 ms, vibrar 300 ms

    vibrator.vibrate(pattern, -1); // -1 para que no se
  repita

}
```

VI-L. Método para vibrar cuando el usuario debe girar a la izquierda

```
private void vibrateLeftTurn() {

    long[] pattern = {0, 1000, 100, 1000, 100, 1000,
 100, 1000}; // 4 veces de vibracion de 1 segundo

    vibrator.vibrate(pattern, -1);

}
```

VI-M. Método para vibrar si el usuario se aleja de la ruta

```
private void vibrateOffRoute() {

    long[] pattern = {0, 500}; // Vibrar de manera
 continua

    vibrator.vibrate(pattern, 0); // 0 para que se
 repita indefinidamente

}
```

VI-N. Método para indicar que el usuario ha llegado a su destino

```
private void vibrateArrived() {

    long[] pattern = {0, 500, 500, 500}; // Vibrar 500
 ms, esperar 500 ms, vibrar 500 ms, esperar 500 ms

    vibrator.vibrate(pattern, -1); // -1 para que no se
  repita

}
```

VI-Ñ. Método para detener la vibración

```
private void stopVibration() {

    vibrator.cancel();

}
```

VI-O. Configuración de actualizaciones de ubicación

```
private void startLocationUpdates() {

    LocationRequest locationRequest = LocationRequest.
 create();

    locationRequest.setInterval(10000); // Cada 10
 segundos

    locationRequest.setFastestInterval(5000); // Cada 5
  segundos

    locationRequest.setPriority(LocationRequest.
 PRIORITY_HIGH_ACCURACY);

    if (ActivityCompat.checkSelfPermission(this,
 Manifest.permission.ACCESS_FINE_LOCATION) ==
 PackageManager.PERMISSION_GRANTED) {

        fusedLocationClient.requestLocationUpdates(
 locationRequest, locationCallback, Looper.getMainLooper
 ());

    }
}
```

VI-P. Obtener última ubicación conocida

```
private void getLastLocation() {

    if (ActivityCompat.checkSelfPermission(this,
 Manifest.permission.ACCESS_FINE_LOCATION) ==
 PackageManager.PERMISSION_GRANTED) {

        Task<Location> task = fusedLocationClient.
 getLastLocation();

        task.addOnSuccessListener(this, location -> {
            if (location != null) {

                currentLocation = location;

                updateMapWithCurrentLocation();

            } else {

                Toast.makeText(MainActivity.this, "No
 se pudo obtener la ubicacion actual", Toast.
 LENGTH_SHORT).show();

            }
        });
    } else {

        ActivityCompat.requestPermissions(this, new
 String[]{Manifest.permission.ACCESS_FINE_LOCATION},
 REQUEST_LOCATION_PERMISSION);

    }
}
```

VI-Q. Actualizar el mapa con la ubicación actual

```
private void updateMapWithCurrentLocation() {

    if (myMap != null && currentLocation != null) {

        LatLng currentLatLng = new LatLng(
 currentLocation.getLatitude(), currentLocation.
 getLongitude());

        myMap.clear();

        myMap.addMarker(new MarkerOptions().position(
 currentLatLng).title("Ubicacion Actual"));

        myMap.moveCamera(CameraUpdateFactory.
 newLatLngZoom(currentLatLng, 15)); // Ajusta el zoom
 del mapa

    }
}
```

VI-R. Buscar ubicación introducida en el SearchView

```
private void searchLocation(String location) {

    Geocoder geocoder = new Geocoder(this);

    List<Address> addressList = null;

    if (location != null && !location.isEmpty()) {

        try {

            addressList = geocoder.getFromLocationName(
 location, 1);

        } catch (IOException e) {

            e.printStackTrace();
        }

        if (addressList != null && !addressList.isEmpty
 ()) {

            Address address = addressList.get(0);
```

```
            LatLng latLng = new LatLng(address.
 getLatitude(), address.getLongitude());

            myMap.addMarker(new MarkerOptions().
 position(latLng).title(location));

            myMap.moveCamera(CameraUpdateFactory.
 newLatLngZoom(latLng, 15));

            getRoute(currentLocation, latLng); //
 Obtener la ruta hacia la ubicacion
        }
    }
}
```

VI-S. *Obtener ruta de la ubicación actual al destino*

```
private void getRoute(Location origin, LatLng
 destination) {

    String url = "https://maps.googleapis.com/maps/api/
 directions/json?origin=" + origin.getLatitude() + "," +
  origin.getLongitude()
            + "&destination=" + destination.latitude +
 "," + destination.longitude + "&key=" + API_KEY;

    StringRequest request = new StringRequest(Request.
 Method.GET, url, response -> {
        try {

            JSONObject jsonResponse = new JSONObject(
 response);

            JSONArray routes = jsonResponse.
 getJSONArray("routes");

            if (routes.length() > 0) {

                JSONArray steps = routes.getJSONObject
 (0).getJSONArray("legs").getJSONObject(0).getJSONArray(
 "steps");

                List<LatLng> points = new ArrayList<>()
 ;

                for (int i = 0; i < steps.length(); i
 ++) {

                    String polyline = steps.
 getJSONObject(i).getJSONObject("polyline").getString("
 points");
                    points.addAll(decodePoly(polyline))
 ;

                }

                myMap.addPolyline(new PolylineOptions()
 .addAll(points).width(10).color(0xFF0000FF)); //
 Dibujar la ruta

            } else {

                Toast.makeText(this, "No se encontro
 ruta", Toast.LENGTH_SHORT).show();

            }
        } catch (JSONException e) {

            e.printStackTrace();

        }
    }, error -> Log.e(TAG, "Error en la solicitud de
 direccion: " + error.getMessage()));

    RequestQueue queue = Volley.newRequestQueue(this);

    queue.add(request);
}
```

VI-T. *Decodificar polilínea en una lista de coordenadas*

```
private List<LatLng> decodePoly(String encoded) {

    List<LatLng> poly = new ArrayList<>();

    int index = 0, len = encoded.length();

    int lat = 0, lng = 0;

    while (index < len) {

        int b, shift = 0, result = 0;

        do {

            b = encoded.charAt(index++) - 63;

            result |= (b & 0x1f) << shift;

            shift += 5;

        } while (b >= 0x20);

        lat += ((result & 1) != 0 ? ~(result >> 1) : (
 result >> 1));

        shift = 0;

        result = 0;

        do {

            b = encoded.charAt(index++) - 63;

            result |= (b & 0x1f) << shift;

            shift += 5;
        } while (b >= 0x20);

        lng += ((result & 1) != 0 ? ~(result >> 1) : (
 result >> 1));

        poly.add(new LatLng((double) lat / 1E5, (double
 ) lng / 1E5));

    }

   return poly;

}
```

VI-U. *Método que se ejecuta cuando el mapa está listo*

```
public void onMapReady(GoogleMap googleMap) {

    myMap = googleMap;

    if (ActivityCompat.checkSelfPermission(this,
 Manifest.permission.ACCESS_FINE_LOCATION) !=
 PackageManager.PERMISSION_GRANTED && ActivityCompat.
 checkSelfPermission(this, Manifest.permission.
 ACCESS_COARSE_LOCATION) != PackageManager.
 PERMISSION_GRANTED) {

        return;

    }

    myMap.setMyLocationEnabled(true);

    getLastLocation(); // Obtener la ultima ubicacion
}
```

VI-V. *Método que se ejecuta cuando el mapa está listo*

```
public void onMapReady(GoogleMap googleMap) {

    myMap = googleMap;

    if (ActivityCompat.checkSelfPermission(this,
 Manifest.permission.ACCESS_FINE_LOCATION) !=
 PackageManager.PERMISSION_GRANTED && ActivityCompat.
 checkSelfPermission(this, Manifest.permission.
 ACCESS_COARSE_LOCATION) != PackageManager.
 PERMISSION_GRANTED) {
```

```
            return;

        }

        myMap.setMyLocationEnabled(true);

        getLastLocation(); // Obtener la ultima ubicacion
    }
```

VI-W. Manejar la respuesta de los permisos solicitados

```
    @Override
    public void onRequestPermissionsResult(int requestCode,
      @NonNull String[] permissions, @NonNull int[]
     grantResults) {

        super.onRequestPermissionsResult(requestCode,
     permissions, grantResults);

        if (requestCode == REQUEST_LOCATION_PERMISSION) {

            if (grantResults.length > 0 && grantResults[0]
     == PackageManager.PERMISSION_GRANTED) {

                getLastLocation();

            } else {

                Toast.makeText(this, "Permiso de ubicacion
     no concedido", Toast.LENGTH_SHORT).show();

            }
        }
    }
```

[6] [7]

VII. Conclusión