

# UCUENCA

## Universidad de Cuenca

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

### **Desarrollo de un sistema para la transmisión de la ubicación de un teléfono inteligente hacia los servicios de emergencia por medio de tonos DTMF**


Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

**Autor:**

Galo Roberto Aguirre Cobo

**Director:**

Andrés Marcelo Vazquez Rodas

ORCID:  0000-0002-6114-1179

**Cuenca, Ecuador**

2023-09-18

## Resumen

Obtener la ubicación de un usuario quien realiza una llamada a los servicios de emergencia es importante para brindar la urgente atención que requiere. Cuando la llamada se origina desde un teléfono móvil, discernir esta información es complicado. El usuario puede desconocer su ubicación, o no poder comunicarla, debido a su condición o a sus circunstancias. Los teléfonos móviles en la actualidad pueden obtener la ubicación del servicio de ubicación del S.O. Existe también el sistema DTMF que asigna dos señales sinusoidales a cada dígito del teclado numérico, este se utiliza para comunicarse con sistemas telefónicos automáticos. En el presente trabajo de titulación, se desarrolla un sistema para transmitir las coordenadas geográficas de un teléfono móvil con el uso de DTMF. El sistema consiste de una aplicación para el S.O. Android, que obtiene las coordenadas geográficas del teléfono, las codifica y las reproduce por medio de tonos DTMF en el audio de la llamada. Los tonos son captados por un dispositivo electrónico que los decodifica, para así recuperar las coordenadas geográficas del usuario. El dispositivo despliega las coordenadas en una pantalla LCD y transmite una URL hacia un ordenador; al hacer clic sobre este enlace, la ubicación del usuario se visualiza en un mapa del sitio web de Google Maps. Se realizaron tres series de pruebas bajo varios niveles de cobertura de telefonía móvil, en tres ubicaciones diferentes. Los resultados demuestran que este sistema funciona con un alto grado de fidelidad.

*Palabras clave:* coordenadas geográficas, servicios de ubicación, Arduino, Android, módulo MT8870



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

**Repositorio Institucional:** <https://dspace.ucuenca.edu.ec/>

## Abstract

Obtaining the location of a user who calls emergency services is important in order to give them the urgent care they need. The call originating from a mobile phone makes discerning this information difficult. The user might not know his location or can't communicate it, because of their condition or due to the situation they're in. Modern cellphones can obtain their position from the O.S.'s location services. Meanwhile, the DTMF tone system assigns two sinusoidal audio signals to each digit of a phone keypad; this is used to communicate with various automatic phone systems: to navigate through menus, as well as to enter various information. In the present undergraduate thesis project, a system is developed to transmit the geographic coordinates of a mobile phone. The system consists of an Android application that obtains the location coordinates from the mobile phone's O.S., encodes these with the use of DTMF tones, which are then played in the audio of the ongoing call. These tones are received by an electronic device that then decodes the tones to recover the original geographic coordinates of the user. The device displays the coordinates on a LCD screen, as well as transmits a URL with the coordinates to a computer, when the URL is clicked on, the location of the user is displayed in Google Maps. A series of trials are carried out under various mobile phone coverage levels, at three different locations. The results demonstrate that this system performs with a high degree of fidelity.

*Keywords:* geographic coordinates, location services, Arduino, Android, MT8870 module



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

**Institutional Repository:** <https://dspace.ucuenca.edu.ec/>

**Índice de contenido**

<b>Resumen .....</b>	<b>2</b>
<b>Abstract .....</b>	<b>3</b>
<b>Índice de contenido .....</b>	<b>4</b>
<b>Índice de figuras .....</b>	<b>7</b>
<b>Índice de tablas.....</b>	<b>10</b>
<b>Agradecimientos .....</b>	<b>11</b>
<b>Dedicatoria.....</b>	<b>12</b>
<b>Abreviaciones y acrónimos.....</b>	<b>13</b>
<b>1 1. Introducción.....</b>	<b>14</b>
1.1 Definición del problema.....	14
1.2 Justificación .....	14
1.3 Alcance de la propuesta.....	15
1.4 Objetivos Generales y Específicos .....	15
1.4.1 Objetivo general.....	15
1.4.2 Objetivos específicos .....	16
1.5 Estructura del documento y contribuciones .....	16
<b>2 2. Fundamentos teóricos .....</b>	<b>18</b>
2.1 Sistema de Posicionamiento Global (GPS).....	18
2.1.1 Componente Espacial .....	20
2.1.2 Componente de Control .....	20
2.1.3 Componente de usuario.....	21
2.2 API de Ubicación de Android .....	22
2.3 Tono Dual de Múltiples Frecuencias (DTMF) .....	22
2.4 Arduino.....	23
2.5 Intervalos de confianza de una distribución binomial .....	24



<b>3</b>	<b>3. Trabajos relacionados .....</b>	<b>26</b>
<b>4</b>	<b>4. Metodología de Desarrollo .....</b>	<b>28</b>
4.1	Diseño e implementación del sistema .....	28
4.1.1	Esquema general del sistema .....	28
4.1.2	Diseño e implementación del dispositivo de decodificación.....	36
4.1.3	Diseño e implementación de la aplicación transmisora .....	37
4.1.3.1	Marcación de los tonos DTMF .....	40
4.1.3.2	Recepción de la ubicación usando la API .....	40
4.1.3.3	La obtención y el marcado de las coordenadas geográficas de la API de ubicación del S.O. Android .....	51
4.1.4	Integración y funcionamiento del sistema .....	54
4.1.5	Materiales y características de los componentes .....	56
4.1.6	Funcionalidades adicionales .....	58
4.1.6.1	Visualización de la ubicación en Google Maps .....	59
4.1.6.2	Obtención del dato de precisión de la ubicación .....	63
4.1.6.3	Obtención del dato de altitud de la ubicación .....	65
4.1.6.4	Obtención del dato del nivel de batería del móvil .....	66
4.1.6.5	Implementación de un dígito de paridad .....	67
4.2	Evaluación experimental .....	68
4.2.1	Esquema para la evaluación de la fidelidad en la transmisión de las coordenadas .....	69
4.2.2	Ejecución de la evaluación de la fidelidad en la transmisión de las coordenadas .....	70
4.2.2.1	Instalación del sitio de recepción en Cuenca .....	70
4.2.2.2	Serie de pruebas con niveles altos de cobertura.....	72
4.2.2.3	Serie de pruebas con niveles medios de cobertura.....	74
4.2.2.4	Instalación del sitio de recepción en Ambato .....	75
4.2.2.5	Serie de pruebas con niveles bajos de cobertura.....	75
<b>5</b>	<b>5. Resultados .....</b>	<b>79</b>
5.1	Resultados - Pruebas en el cantón Cuenca, provincia del Azuay .....	79
5.2	Resultados - Pruebas en la parroquia rural Checa, cantón Cuenca .....	79

5.3	Resultados - Pruebas en el km. 2 de la Vía a Aguaján, cantón Ambato, provincia de Tungurahua .....	80
5.4	Resumen de los resultados .....	80
<b>6</b>	<b>6. Conclusiones y recomendaciones .....</b>	<b>82</b>
6.1	Conclusiones .....	82
6.2	Recomendaciones .....	83
6.3	Trabajos futuros .....	84
	<b>Referencias .....</b>	<b>86</b>

## Índice de figuras

2.1 Problema de la trilateración de la geometría. Fuente: adaptada de [1].	18
2.2 En un espacio 3-dimensional, se necesita la distancia hacia 4 puntos (los centros de las esferas), para ubicar a un punto con precisión. Fuente: adaptada de [2].	19
2.3 El receptor de GPS calcula su distancia hacia uno de los satélites de la constelación GPS. Fuente: adaptada de [3].	19
2.4 Constelación de satélites GPS en 6 planos orbitales. Fuente: adaptada de [4].	20
2.5 Mapa de las estaciones de monitoreo y control y de las antenas del segmento de control del GPS. Fuente: adaptada de [5].	21
2.6 Teclado numérico. Fuente: adaptada de [6].	22
2.7 Señal sinusoidal de baja frecuencia ( $697\text{ Hz}$ ). Fuente: autor, 2023.	23
2.8 Señal sinusoidal de alta frecuencia ( $1209\text{ Hz}$ ). Fuente: autor, 2023.	23
2.9 Señal DTMF, consiste de la suma de dos señales sinusoidales ( $697\text{ Hz} + 1209\text{ Hz}$ ), que representa al dígito 1 del teclado numérico. Fuente: autor, 2023.	24
2.10 La tarjeta de un Arduino UNO, el modelo más popular y masificado de la plataforma Arduino. Fuente: adaptada de [7].	24
4.1 Esquema de simulación.	29
4.2 Gráfica de la precisión de la ubicación por GPS.	30
4.3 Gráfica de la precisión de la ubicación por WiFi.	30
4.4 Gráfica de la precisión de ubicación suplementada por sensores.	31
4.5 Ordenador de las consolas del SIS ECU 911.	31

4.6	Caja de Voz de las consolas del SIS ECU 911. ....	32
4.7	Teléfono de las consolas del SIS ECU911. ....	32
4.8	Puerto RJ9 de los teléfonos GN Netcom 7100. ....	33
4.9	Módulo decodificador MT8870. ....	33
4.10	Esquema de integración para el SIS ECU 911. ....	34
4.11	Arreglo de teclas original del estándar UTI-T Q.23. ....	35
4.12	Diagrama de conexión del dispositivo receptor. ....	37
4.13	Aplicación de llamadas, Simple Dialer. ....	38
4.14	Archivos de interfaces gráficos de Simple Dialer. ....	38
4.15	Listado de los módulos de Simple Dialer. ....	39
4.16	Interfaz gráfica del teclado de Simple Dialer. ....	40
4.17	Captura de pantalla de los mensajes log en el módulo de llamadas. ....	41
4.18	Flujograma del funcionamiento del sistema. ....	55
4.19	Configuración de PuTTY-url para la comunicación serial con el Arduino. ..	60
4.20	Confirmación del puerto del Arduino en Windows. ....	61
4.21	Configuración de PuTTY-url, para que guarde una bitácora de datos. ....	61
4.22	Se guarda las configuraciones en PuTTY-url. ....	62
4.23	Despliegue de texto en PuTTY-url, con la dirección URL. ....	62
4.24	Página web con la ubicación en Google Maps. ....	63
4.25	Despliegue de la ubicación incluyendo el dato de precisión. ....	65
4.26	Despliegue de la ubicación con el mensaje de 'Error de Paridad' ....	68
4.27	Mapa de la ubicación del edificio de la Cámara de la Construcción. ....	70
4.28	Comprobación del nivel de cobertura en la oficina del edificio de la Cá- mara de la Construcción de Cuenca. ....	71
4.29	Arreglo de los equipos para la recepción de datos. ....	71
4.30	Se configura al primer celular para que conteste automáticamente las llamadas entrantes. ....	72
4.31	Mapa del sitio en donde se llevo a cabo la primer serie de pruebas. ....	72
4.32	Comprobación del nivel de cobertura en el Parque Calderón, Cuenca. ....	73
4.33	Mapa que demuestra la proximidad con la antena de la radio base. ....	73
4.34	Captura del interfaz de llamadas con el despliegue por transmitirse. ....	74
4.35	Mapa del sitio de la segunda serie de pruebas. ....	75

4.36 Comprobación de las condiciones de cobertura en la Parroquia Checa. ...	76
4.37 Mapa del sitio en Ambato donde se ubicó al receptor para la tercera serie de pruebas. ....	76
4.38 Comprobación de los niveles de cobertura en el domicilio en Ambato, para la tercera serie de pruebas. ....	77
4.39 Mapa del sitio para la tercera serie de pruebas. ....	78
4.40 Comprobación de los bajos niveles de cobertura en el sitio para la tercera serie de pruebas. ....	78
5.1 Resultados numéricos de la serie de pruebas realizadas en el centro histórico de la ciudad de Cuenca, provincia del Azuay. Fuente: autor, 2023. ....	79
5.2 Resultados numéricos de la serie de pruebas realizadas en la parroquia Checa, perteneciente al cantón Cuenca, provincia del Azuay. Fuente: autor, 2023. ....	80
5.3 Resultados numéricos de la serie de pruebas realizadas en el km. 2 de la Vía a Aguaján, perteneciente al cantón Ambato, provincia de Tungurahua. Fuente: autor, 2023. ....	81

**Índice de tablas**

4.1	Esquema de codificación. ....	35
4.2	Tabla de los 4 bits que codifican cada tono DTMF (Módulo MT8870).....	36
4.3	Lista de materiales.....	57
4.4	Lista de software.....	58
4.5	Tabla con números correspondientes para el cálculo del dígito de paridad. 68	
4.6	Tabla con las tres franjas de niveles de cobertura para las series de pruebas. ....	69
4.7	Tabla de los sitios y fechas de las 3 series de pruebas del sistema.....	77
5.1	Tabla de resultados porcentuales. ....	80
6.1	Lista de materiales y costos.....	83

## Agradecimientos

Con mucha sinceridad y una gran humildad, deseo agradecer a las siguientes personas e instituciones:

A toda mi familia, que ha estado en mi corazón siempre. En particular a mi querida madre, Irma, que en paz descanse, que me crió y me dio todo lo que pudo. A mi querida tía Beatriz, que en paz descanse, que fue como una segunda madre para mí. A mi prima Caty, por su incansable espíritu y paciencia. A mi tío Luis, por su apoyo incondicional y la escucha que me ofreció. A mi tío Hugo, que me apoyo en innumerables ocasiones y me dio su ejemplo como profesional. Y al resto de familiares, tíos y primos: nada de esto sería posible sin ustedes.

A mi compañera, Geoconda, que ha estado de mi lado y me ha apoyado en todo durante los últimos años: espero trazar el futuro a su lado.

A mi amigo, Guido, quien ha estado en las buenas y en las malas.

A la Universidad de Cuenca, una institución de alto nivel académico y altos estándares. A los profesores y profesoras, que brindaron su conocimiento técnico y profesionalismo durante este largo trayecto. A todos los compañeros y compañeras de la carrera, de quienes me serví de su cordial apoyo y trate de ser recíproco las veces que pude.

Un particular agradecimiento al Ing. Andrés Vazquez, PhD., por haberme guiado en el desarrollo del presente trabajo de titulación.

A la bella ciudad de Cuenca, que me acogió, como si fuera mi ciudad natal, con toda su cultura y folclor. A la Parroquia Checa, por permitir a un desconocido realizar la segunda serie de pruebas ahí, a pesar de la situación de inseguridad actual. Y a la ciudad de Ambato, ciudad natal de mi padre y tía Betty; un sitio que quiero mucho.

Mantengo una deuda pendiente con todos ustedes, que no podre saldar. Honraré sus sacrificios, sirviendo a la comunidad como profesional, con mis mejores habilidades.

**El Autor**

## **Dedicatoria**

Para mi padre Galo, para mi tío John Felipe, para mi tía Judy, para mi tía Beatriz, quien me fue como una segunda madre, y en especial para mi madre Irma, quien siempre extraño: Que en paz descansen, mi corazón está con ustedes.

**Galo Roberto Aguirre Cobo**



## Abreviaciones y acrónimos

**AGPS** GPS Asistido. [21](#), [56](#)

**API** Interfaz de Programación de Aplicaciones. [16](#), [22](#), [29](#), [33](#), [39](#), [63](#)

**DTMF** Tonos Duales de Múltiples Frecuencias. [2](#), [3](#), [7](#), [10](#), [15](#), [16](#), [22](#), [24](#), [26](#), [28](#), [33](#), [35](#), [36](#), [39](#), [59](#), [67](#), [69](#), [82](#), [84](#)

**GNSS** Sistema global de navegación por satélite. [18](#), [56](#)

**GPS** Sistema de Posicionamiento Global. [7](#), [14](#), [15](#), [18–22](#), [26](#), [28](#), [30](#), [44](#), [56](#), [63](#)

**I2C** Circuito inter-integrado. [37](#)

**LCD** Pantalla de cristal líquido. [2](#), [3](#), [28](#), [37](#), [56](#)

**O.S.** Operating System. [3](#)

**PBX** Central Privada Automática. [15](#)

**PEMEA** Aplicación Pan-Europea Móvil para Emergencias. [26](#)

**PSTN** Red Telefónica Pública Conmutada. [22](#)

**S.O.** Sistema Operativo. [2](#), [16](#), [22](#), [29](#), [35](#), [37–39](#), [64](#), [65](#), [82](#), [84](#)

**SIS ECU 911** Servicio Integrado de Seguridad ECU 911. [7](#), [8](#), [32–34](#), [82](#), [83](#)

**SSH** Intérprete de órdenes seguro. [59](#), [60](#)

**TOA** Tiempo de llegada. [18](#)

**URL** Localizador de recursos uniforme. [2](#), [3](#), [8](#), [56](#), [59](#), [60](#), [62](#), [63](#)

**UTC** Tiempo Universal Coordinado. [21](#)

**UTI-T** Sector de Normalización de las Telecomunicaciones de la UIT. [8](#), [35](#)

## 1. Introducción

### 1.1. Definición del problema

En una situación de emergencia, sea esta de naturaleza policial, médica o una que requiera del cuerpo de bomberos, el tiempo que se demoren los agentes en responder y llegar al sitio de los hechos es crucial: unos minutos de retraso pueden significar la diferencia entre la vida o la muerte. En una llamada a los servicios de emergencia, se debe seguir un protocolo en el cual se comunica a la operadora de estos servicios, entre otros datos, la dirección de los acontecimientos. Usualmente, dicho protocolo puede tardar entre 3 a 5 minutos en total.

Para llamadas realizadas desde un teléfono convencional, se ubica el origen de la llamada por medio de la dirección con la cual esté registrado el abonado [2]. Para el caso de una llamada originada desde un teléfono móvil, es necesario que la operadora de los servicios de emergencia solicite de la persona su ubicación, la intersección de calles más próximas a donde se ubica y una referencia, para así mejor dirigir a los agentes. Sin embargo, en algunos casos, quien inicia la llamada es un menor de edad, o un adulto mayor, o alguien con una discapacidad, o una persona herida o en shock, y puede que no sea posible que este individuo conteste las preguntas con precisión, e incluso la misma naturaleza de la emergencia puede impedir que la persona conteste; por ejemplo, en caso de un atragantamiento, un paro cardíaco, un asalto, entre otros.

Algunos servicios de emergencia cuentan con un servicio de ubicación, sin necesitar de la cooperación del usuario, que utiliza la trilateración para aproximar la ubicación del usuario [2], sin embargo, con una precisión limitada de entre 50 y 200 metros a la redonda.

### 1.2. Justificación

Se sabe que la penetración de la telefonía móvil avanzada en el mercado ecuatoriano es de un 94.7% [3]. Además, los teléfonos inteligentes generalmente cuentan con un módulo GPS, que en promedio reportan la ubicación con una precisión de 5 metros bajo condiciones de cielos despejados [4]. Adicionalmente, estos dispositivos pueden

transmitir información en medio de una llamada telefónica por medio de los Tonos Duales de Múltiples Frecuencias (DTMF) [5], lo cual se utiliza para interactuar con sistemas de atención automatizada (p. ej.: PBX[8], buzones de mensajería, etc.).

Entonces, un teléfono inteligente podría codificar las coordenadas de la ubicación geográfica obtenida del módulo GPS por medio de tonos DTMF y reproducirlos en el audio de una llamada dirigida hacia los servicios de emergencia [6]. Mientras que, en el lado de la operadora de los servicios de emergencia, un equipo receptor podría decodificar los tonos de regreso a coordenadas geográficas. Esta información podría entonces ser visualizada para el beneficio de la operadora, con el fin de que esta dirija a los agentes de emergencia al sitio de los hechos.

### **1.3. Alcance de la propuesta**

Con el fin de transmitir la ubicación de quien acuda a llamar a los servicios de emergencia de forma más rápida y precisa, en este trabajo se pretende desarrollar una aplicación para teléfonos móviles inteligentes que extraiga las coordenadas geográficas del módulo GPS del teléfono, codifique estas coordenadas con tonos DTMF y que reproduzca y transmita los tonos por medio de la llamada; todo esto de manera transparente para el usuario. Esta aplicación se podría desarrollar tanto para el sistema operativo Android como para el S.O. iOS; por conveniencia se desarrollará para Android. Para el lado de la operadora de los servicios de emergencia, se piensa diseñar e implementar un dispositivo que, desde que se conteste la llamada, estaría continuamente a la escucha para captar los tonos ingresados. Al traducirse los tonos de nuevo a coordenadas geográficas, se sabrá la ubicación del teléfono del usuario que llama.

### **1.4. Objetivos Generales y Específicos**

#### **1.4.1. Objetivo general**

- Diseñar, implementar y poner a prueba un sistema que permita transmitir las coordenadas geográficas de un teléfono inteligente hacia una operadora de los ser-

vicios de emergencia por medio de tonos DTMF, con el fin de brindar un servicio de rescate más expedito.

#### 1.4.2. Objetivos específicos

- Desarrollar una aplicación para el S.O. de teléfonos inteligentes Android, que obtenga las coordenadas geográficas de la API de ubicación del S.O. Android, las codifique con tonos DTMF y las emita por medio del audio de la llamada.
- Desarrollar un dispositivo para captar y decodificar los tonos DTMF que podría ser instalado en la consola de una operadora de los servicios de emergencia, con el fin de recuperar las coordenadas geográficas de la ubicación del usuario.
- Generar un protocolo para la codificación de las coordenadas de longitud, latitud, y de un valor de precisión de la ubicación.
- Implementar pruebas simuladas del sistema desarrollado, llevando a cabo llamadas entre el teléfono móvil con el aplicativo instalado, hacia un segundo teléfono conectado al dispositivo electrónico, bajo diferentes condiciones de cobertura de telefonía móvil.

#### 1.5. Estructura del documento y contribuciones

Para registrar de forma sistemática el desarrollo del trabajo de titulación, el presente documento se divide en seis secciones, sin incluirse la presente introducción. En la Sección 2 se presentan los conceptos teóricos que fueron necesarios para el desarrollo de este trabajo de titulación. La Sección 3 resume algunos de los trabajos previos más relevantes al presente trabajo, del campo de las telecomunicaciones. En la Sección 4, se presentan los detalles del proceso que se siguió para el desarrollo del sistema para la transmisión de coordenadas de un teléfono móvil.

Una vez desarrollado un prototipo funcional del sistema, se llevó a cabo tres series de pruebas bajo diferentes condiciones de cobertura; estas series de pruebas son descritas también en la Sección 4. Mientras que en la Sección 5, se presentan los resultados de las evaluaciones de la fidelidad del sistema en transmitir las coordenadas.

Finalmente, tanto la aplicación para Android que codifica y transmite las coordenadas por medio del audio de la llamada, como el programa decodificador que se ejecuta en la tarjeta Arduino del dispositivo receptor, fueron almacenadas en la plataforma de desarrollo colaborativo, GitHub [9].

A pesar de que el presente trabajo no innova en generar nuevas tecnologías ni técnicas, la aplicación de las tecnologías presentes hacia esta problemática resulta en un posible valioso aporte, ya que si este fuese integrado en la actualidad, podría salvar la vida de un ciudadano o ciudadana a futuro.

## 2. Fundamentos teóricos

### 2.1. Sistema de Posicionamiento Global (GPS)

El Sistema de Posicionamiento Global (GPS, por sus siglas en inglés), es un sistema global de navegación por satélite (GNSS), desarrollado por el ejército de los EE.UU. En general, los GNSSs se basan en un problema de la geometría, llamado trilateración (Fig. 2.1), en donde, utilizando las distancias conocidas hacia los tres vértices de un triángulo, se debe ubicar un punto en su interior con precisión.

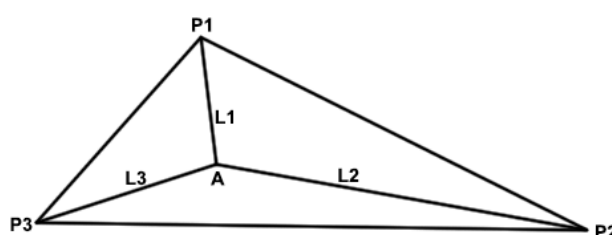


Figura 2.1: Problema de la trilateración de la geometría.  
Fuente: adaptada de [1].

Sin embargo, tres puntos son suficientes para ubicar a un punto en un plano bidimensional. Para el caso del espacio tridimensional, es necesario cuatro puntos de referencia (Fig. 2.2). En el caso del sistema GPS, se calcula la distancia hacia cuatro satélites en órbita, utilizando el tiempo de llegada de señales electromagnéticas provenientes de estos satélites y la velocidad de las señales, para despejar la distancia que recorrieron las mismas (Ecu. 2.1). En el caso de la radiación electromagnética, se sabe que la velocidad es el de la luz (de aproximadamente  $300,000 \text{ km/s}$ ). A la distancia calculada, es necesario también llevar a cabo varias correcciones relativistas debido a la velocidad de los satélites y su ubicación en el campo gravitacional de la tierra.

$$Dist_{sat.-rec.} = c \cdot TOA \quad (2.1)$$

En donde:

$Dist_{sat.-rec.}$  es la distancia entre uno de los satélites de la constelación GPS y el receptor,

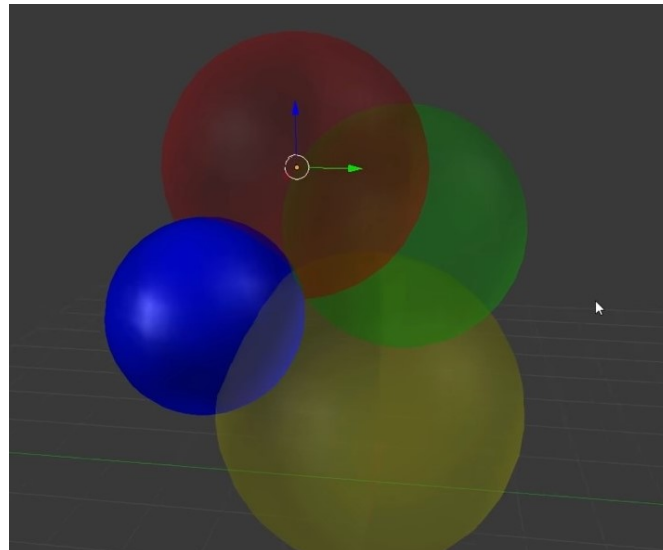


Figura 2.2: En un espacio 3-dimensional, se necesita la distancia hacia 4 puntos (los centros de las esferas), para ubicar a un punto con precisión.

Fuente: adaptada de [2].

$c$  es la velocidad de la luz ( $\approx 300,000km/s$ ), y

$TOA$  es el Tiempo de llegada de la señal electromagnética, proveniente del satélite, al receptor.

El receptor calcula su ubicación con respecto a las ubicaciones de los satélites, utilizando su distancia a cada uno de estos (Fig. 2.3). El sistema GPS consiste de tres componentes: el componente espacial, el componente de control, y el componente de usuario. Los tres componentes se describen a continuación:

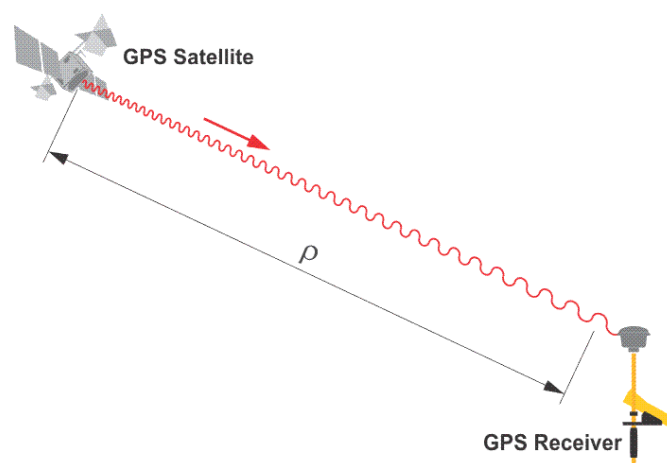


Figura 2.3: El receptor de GPS calcula su distancia hacia uno de los satélites de la constelación GPS.

Fuente: adaptada de [3].

### 2.1.1. Componente Espacial

El componente espacial consiste de un mínimo de 24 satélites en órbita (31 en la actualidad), a unos 20,200 km de altura sobre la superficie terrestre. Se colocan a 4 satélites (mínimo) en 6 planos orbitales equidistantes entre sí (Fig. 2.4). De este modo, se podrán observar a un mínimo de 4 de estos satélites desde cualquier lugar del mundo, debido a que se necesita este número de satélites para el cálculo de la posición del receptor.

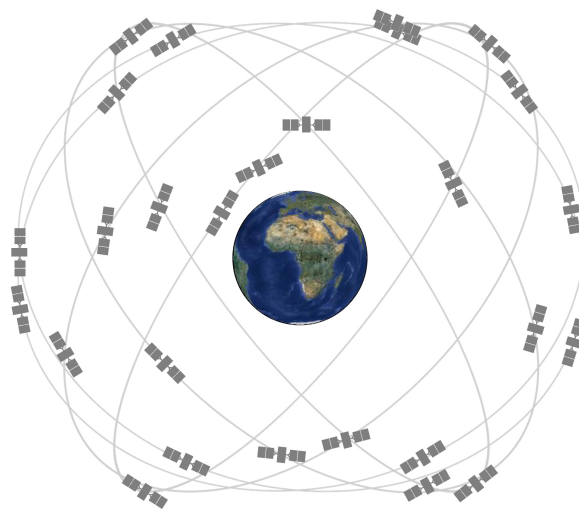


Figura 2.4: Constelación de satélites GPS en 6 planos orbitales.  
Fuente: adaptada de [4].

### 2.1.2. Componente de Control

El componente de control, consiste de varias estaciones terrestres de monitoreo (una de ellas ubicada en el Ecuador [Fig. 2.5]) y de varias antenas a nivel mundial. Las estaciones monitorean las órbitas de cada satélite, para saber la ubicación precisa de cada uno, como también para controlar y corregir cualquier desviación en sus trayectos. La información de la ubicación de cada satélite es transmitida desde los satélites hacia los receptores, para que estos últimos puedan calcular su ubicación, utilizando a la ubicación relativa de los satélites como marco de referencia. Entre los datos, también se incluyen la información de la ubicación precisa del resto de la constelación de satélites.

El tiempo es un componente crucial para el cálculo de la ubicación de los satélites.



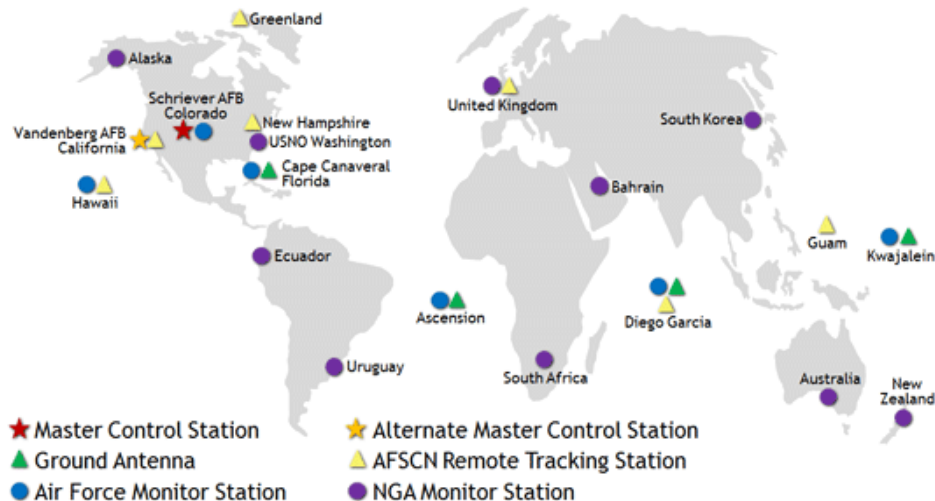


Figura 2.5: Mapa de las estaciones de monitoreo y control y de las antenas del segmento de control del GPS.

Fuente: adaptada de [5].

Se utiliza la señal de una red mundial de relojes atómicos, para calcular un dato de corrección para el tiempo que marcan los relojes de los satélites (adicionalmente, hay una diferencia de varios segundos entre el Tiempo Universal Coordinado (UTC) y el tiempo GPS (el estándar UTC agrega segundos adicionales [segundos intercalar] para sincronizarse con el tiempo solar medio). Este dato de corrección se transmite hacia los receptores. Con esta información, los receptores calculan su ubicación sobre la faz de la tierra, con una precisión de menos de 1 metro [10]. La red de antenas se encarga de subir los datos de tiempo y de ubicación a los satélites, para que estos últimos remitan esta información en los tiempos precisos a los receptores.

### 2.1.3. Componente de usuario

El componente de usuario consiste de los equipos receptores que se ocupan para calcular la ubicación sobre la superficie terráquea. Esto incluye en la actualidad la mayoría de los teléfonos móviles, que incluyen módulos receptores de las señales satelitales de la constelación GPS. En el caso de ubicarse dentro de un túnel o de haber otras obstrucciones que interrumpan a estas señales, los celulares ocupan la tecnología AGPS (GPS Asistido), la cual ocupa otras fuentes de información, como las torres de celular y los puntos de acceso de WiFi, para calcular la ubicación de los usuarios. Adicionalmente, el AGPS descarga parte de la información de la constelación GPS

utilizando las redes de datos a las que estos estén conectados, permitiendo un más rápido y preciso cálculo de la ubicación.

## 2.2. API de Ubicación de Android

La API de ubicación para el desarrollo y funcionamiento de las aplicaciones de Android consiste en un modelo cliente-servidor. Las aplicaciones implementan instancias de los clientes de ubicación, los mismos que solicitan al servicio de ubicación del S.O. Android.

El servicio de ubicación puede utilizar como fuente los módulos GPS presentes en los equipos, como también las señales provenientes de las torres de las radiobases, o las ubicaciones de los puntos de acceso de las redes WiFi dentro del entorno inmediato del teléfono móvil.

## 2.3. Tono Dual de Múltiples Frecuencias (DTMF)

La tecnología de Tono Dual de Múltiples Frecuencias (DTMF), fue adoptada para reemplazar al marcado por pulsos en la telefonía fija (PSTN). El DTMF consiste básicamente en asignar a cada número de un teclado numérico (Fig. 2.6), un par de frecuencias: una baja y una alta. Estas frecuencias son asignadas según un arreglo matricial, en donde cada frecuencia baja corresponde a una fila del arreglo, mientras que cada frecuencia alta corresponden a una columna del mismo (Fig. 4.11).



Figura 2.6: Teclado numérico.  
Fuente: adaptada de [6].

Se utiliza este sistema para marcar los dígitos del número del abonado con quien se quiera comunicar. El momento de presionar una tecla correspondiente a un dígito del número de la contraparte, el teléfono emite el par de señales sinusoidales con las frecuencias correspondientes a ese dígito (Figs. 2.7, 2.8 y 2.9). La central telefónica reconoce las dos frecuencias de cada número y tras recibir el último dígito, establece el circuito de comunicación con el teléfono destinatario. El equipo del destinatario timbra y, si el usuario contesta, se establece el flujo bidireccional de las señales de audio entre los dos usuarios.

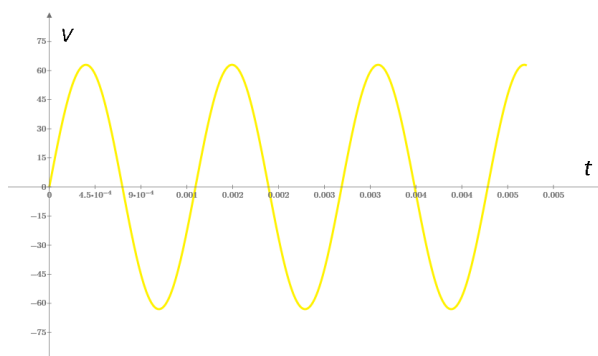


Figura 2.7: Señal sinusoidal de baja frecuencia ( $697\text{ Hz}$ ).  
Fuente: autor, 2023.

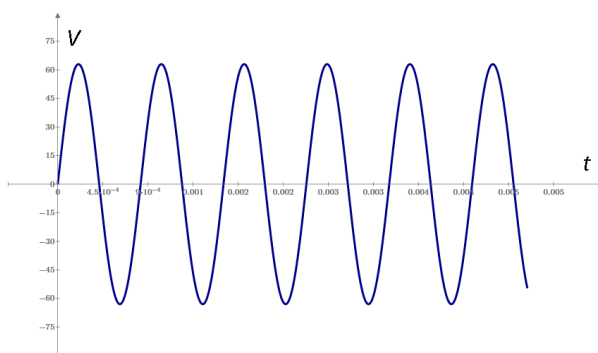


Figura 2.8: Señal sinusoidal de alta frecuencia ( $1209\text{ Hz}$ ).  
Fuente: autor, 2023.

## 2.4. Arduino

El Arduino es una plataforma ideada para el desarrollo acelerado de prototipos, que consiste de tarjetas electrónicas diseñadas con base en un microcontrolador, y de una versión simplificada del lenguaje de programación C++. Estos dispositivos cuentan con pines analógicos y digitales para recibir señales de entrada, como también

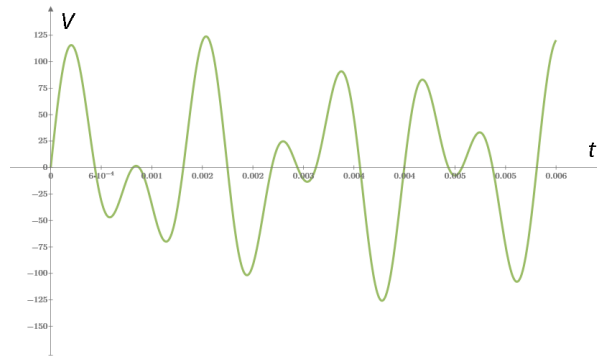


Figura 2.9: Señal DTMF, consiste de la suma de dos señales sinusoidales ( $697 \text{ Hz} + 1209 \text{ Hz}$ ), que representa al dígito 1 del teclado numérico.

Fuente: autor, 2023.

emitir señales de salida (Fig. 2.10). El programa que se ejecuta en el Arduino media la interacción entre las señales de entrada que ingresan, y las que Arduino emite. Con esta funcionalidad, se pueden diseñar sistemas, por ejemplo, para controlar las revoluciones de un motor eléctrico o dirigir a un robot según las señales que capten sus sensores.



Figura 2.10: La tarjeta de un Arduino UNO, el modelo más popular y masificado de la plataforma Arduino.

Fuente: adaptada de [7].

## 2.5. Intervalos de confianza de una distribución binomial

Una distribución binomial es aquella donde una población estadística[11] se puede caracterizar por encajar en una de dos categorías. Las categorías pueden denominarse como Verdadero y Falso, o Acierto y Desacierto. Las estadísticas sobre estas distribuciones se caracterizan por el número de muestras de aciertos,  $x$ , y el número

ro de muestras totales,  $n$ . La denominada proporción es el cociente entre estos dos números,  $p = x/n$ .

Para calcular los intervalos de confianza de estas estadísticas, se utiliza la fórmula ajustada de Wald [12] (Ecuación 2.2).

$$\hat{p}_{adj} \pm z_{(1-\frac{\alpha}{2})} \sqrt{\frac{\hat{p}_{adj}(1 - \hat{p}_{adj})}{n_{adj}}} \quad (2.2)$$

En esta fórmula, aparecen ajustes al número de muestras,  $n_{adj}$  (Ecuación 2.3) y la proporción,  $p_{adj}$  (Ecuación 2.4). En estas ecuaciones también figura el valor crítico para una distribución normal,  $z$ , que para un valor de confianza de un 95 % es igual a 1.96.

$$n_{adj} = n + z^2 \quad (2.3)$$

En donde:

$n_{adj}$  es el número de muestras ajustado,

$n$  es el número de muestras original, y

$z$  es el valor crítico para una distribución normal.

$$\hat{p}_{adj} = \frac{x + z^2}{n_{adj}} \quad (2.4)$$

Y, en donde:

$\hat{p}_{adj}$  es la proporción ajustada.

$x$  es el valor promedio de aciertos,

$z$  es el valor crítico para una distribución normal, y

$n_{adj}$  es el número de muestras ajustado.

### 3. Trabajos relacionados

Entre los trabajos relacionados con la presente propuesta, en [13], se presenta un sistema con un fin similar, pero ya con un nivel de sofisticación mucho más alto y con más fines particulares: un sistema que utiliza una red de servidores especializados que actúan como intermediarios entre los teléfonos de quienes acuden a llamar a los sistemas de emergencia y los mismos sistemas de emergencia, basado en el protocolo de la aplicación pan-europea móvil para emergencias (PEMEA). Adicionalmente, a remitir la ubicación del usuario a los servicios de emergencia, también permite transmitir información vital del usuario, como también iniciar chats y videollamadas con las operadoras y los agentes de emergencia. Frente a esta solución de alta gama, la presente propuesta compite en términos de simplicidad, costo, escalabilidad y un mayor grado de universalidad (dado que al acceder a la señal analógica de la llamada, se podría implementar en prácticamente cualquier parte del mundo). Pero claro, con un enfoque mucho más limitado: únicamente con el fin de compartir la ubicación del usuario.

En [14], se demuestra la posibilidad de insertar tonos DTMF de forma dinámica en los momentos de silencio de la transmisión de una emisora para comunicar señales de control hacia radios, lo que permite automáticamente cambiar la estación al que estén sintonizados hacia un canal de emergencias. Este canal emitiría información pertinente durante situaciones de emergencia, tales como terremotos, deslaves, inundaciones, entre otros. De forma similar, en la solución que se propone, se insertarían los tonos DTMF en el audio de una llamada, pero únicamente para transmitir datos codificados con estos tonos, y no precisamente para controlar unos dispositivos.

En [15], se comprobó la precisión del uso de la localización por medio de los módulos GPS y WiFi de los teléfonos inteligentes para ubicar a los usuarios en situaciones de emergencia, con 232 pruebas en 11 países. También se demostró un sistema internacional de apoyo para viajeros en circunstancias de emergencia. Este sistema constaba de una aplicación que, al presionar un botón, informaba a un centro de emergencias en Alemania sobre la emergencia, el cual remitía la información a los servicios de emergencia en el país extranjero, por medio de una agencia local. Es una propuesta que puede ofrecer mucho valor a un viajero. Sin redundar mucho en

el tema, con respecto a este trabajo, se retoma la universalidad del sistema que se propone: este no dependería del idioma que se hable en algún país en particular, y así, mientras que la interfaz del aplicativo instalado en el teléfono inteligente este traducido al idioma del usuario, la segunda etapa del sistema se encargaría de comunicar a la operadora las coordenadas del mismo usuario.

En [16], se presenta el esquema que se propone como un método para transmitir la ubicación hacia los servicios de emergencia. Lo presenta como una alternativa secundaria a la propuesta principal, la que se basa en usar la capacidad de síntesis de voz del teléfono inteligente para enunciar las coordenadas en medio de la llamada hacia la operadora. Ambas propuestas las presentan como conceptos, sin llevar a cabo la implementación, ni tampoco lleva a cabo pruebas prácticas de lo que propone. La ventaja que se tiene en enunciar las coordenadas, es que no requiere de dispositivos electrónicos en las instalaciones de los servicios de emergencia, ni de alteración alguna de sus sistemas existentes, solo requiere de la comprensión del idioma por la operadora. Las fallas, sin embargo, podrían darse en caso de que el usuario viaje a otro país o también, como requeriría que la operadora transcriba las coordenadas, esta puede fallar debido al error humano.

En [17], se detalla la creación y estructura del Servicio Integrado de Seguridad ECU 911 hasta el año 2018. Resalta que el caso del SIS ECU 911 puede servir como una referencia para la implementación en otros países en vías de desarrollo. Aparte, enumera los requisitos para llevar a este servicio de emergencias a la siguiente generación (Next Generation 911), entre el que se encuentra justamente poder ubicar a los usuarios con mayor precisión. Se espera que la implementación del sistema propuesto en el presente documento sirva como un aporte para este fin.

## 4. Metodología de Desarrollo

### 4.1. Diseño e implementación del sistema

#### 4.1.1. Esquema general del sistema

El esquema general de este sistema es el que se presenta en la Fig. 4.1. En esta se observa un teléfono móvil inteligente en el cual se ha instalado la aplicación que se desarrolló. Por medio de este se realiza la llamada hacia un segundo teléfono celular cuyo rol sería simular la consola de la operadora de los servicios de emergencia. En el transcurso de la llamada, se transmite las coordenadas geográficas codificadas como tonos DTMF. Al puerto de auriculares del segundo celular se conecta un adaptador divisor para bifurcar la señal analógica hacia dos puertos. Al primer puerto se conectaría el dispositivo electrónico que se diseñó para decodificar y recuperar las coordenadas, y al otro puerto se conectan auriculares para escuchar la llamada.

El dispositivo electrónico consistiría de:

- Un módulo decodificador de tonos DTMF (MT8870).
- Un Arduino Uno, en el cual se ejecutaría un programa encargado en decodificar las señales provenientes del módulo DTMF y recuperar las coordenadas geográficas.
- Una pantalla LCD en la cual se pueda visualizar las coordenadas geográficas.
- Opcionalmente, se puede conectar el Arduino Uno a un ordenador portátil al cual se transmitiría las coordenadas por medio de una comunicación serial por el puerto USB.

Se sabe que la precisión de la ubicación obtenida desde el módulo GPS de un teléfono inteligente es generalmente muy buena al aire libre [18] [19]; mientras que, para los interiores, por ejemplo, de una casa o un edificio, en donde no se logra captar la señal de la constelación de satélites de GPS, la precisión baja casi a cero (Fig. 4.2). Para estos casos, se puede ubicar a un usuario por medio de las señales de los puntos de acceso de WiFi presentes en los alrededores (Fig. 4.3) [19] [20]. Mientras que, para las transiciones entre estos dos medios, se puede extrapolar la ubicación



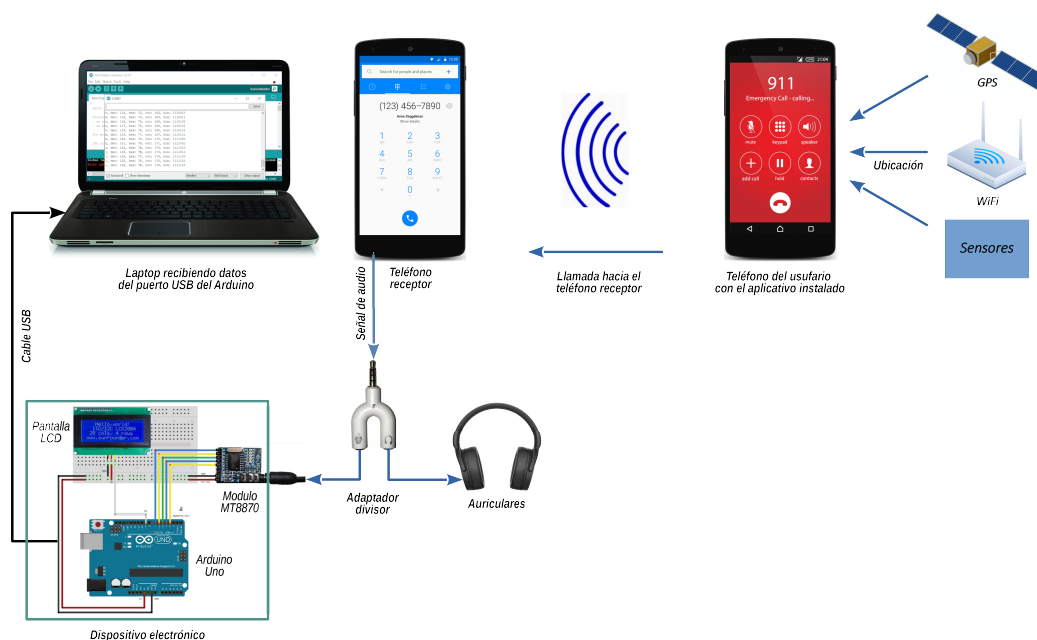


Figura 4.1: Esquema para simular el sistema.

Fuente: autor, 2023.

(**dead reckoning** o navegación por estima) por medio de los varios sensores del teléfono, tales como el acelerómetro, el giroscopio y el magnetómetro (Fig. 4.4) [19] [21]. Actualmente, este proceso se lleva a cabo mediante los algoritmos del Proveedor de Ubicación Fusionada (**Fused Location Provider**) del S.O. Android. Este servicio está disponible para el desarrollador a través de una API simplificada que permite disponer de una ubicación precisa y de forma continua, ubicua e instantánea. La ventaja adicional de este esquema es que se gestiona a estos recursos de forma eficiente en términos de consumo de energía, haciendo uso de cada uno únicamente cuando sea necesario [21]. En el presente trabajo se utilizará, por lo tanto, esta combinación de fuentes de información geográfica para una mejor ubicación del usuario en el lugar que se encuentre.

Con el fin de investigar la factibilidad de la integración del sistema que se propone a los sistemas de los servicios de emergencia, se llevó a cabo una visita a la Coordinación Zonal del Servicio Integrado de Seguridad ECU911 en la ciudad de Cuenca. Se observó que las consolas que ocupan las operadoras encargadas de contestar las llamadas de emergencia constan de:

- Un ordenador (Fig. 4.5).

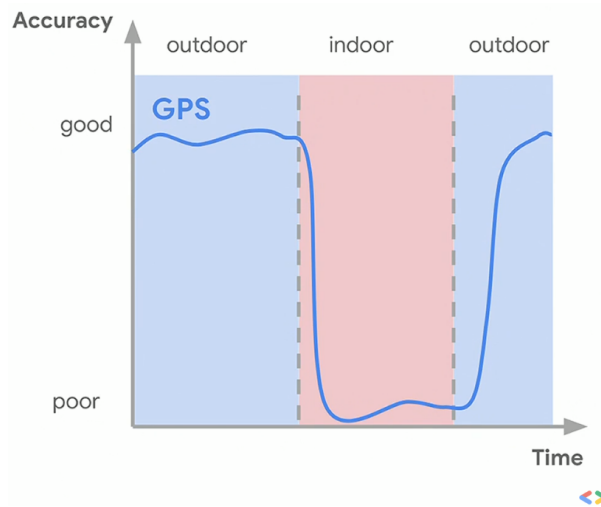


Figura 4.2: Gráfica de la precisión de la ubicación por GPS en las transiciones del aire libre al interior de algún edificio y viceversa.  
Fuente: adaptada de [19].

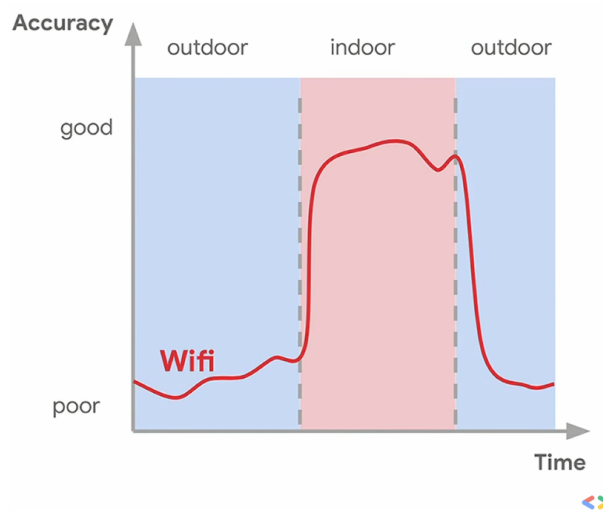


Figura 4.3: Gráfica de la precisión de la ubicación por WiFi en las transiciones del aire libre al interior de algún edificio y viceversa.  
Fuente: adaptada de [19].

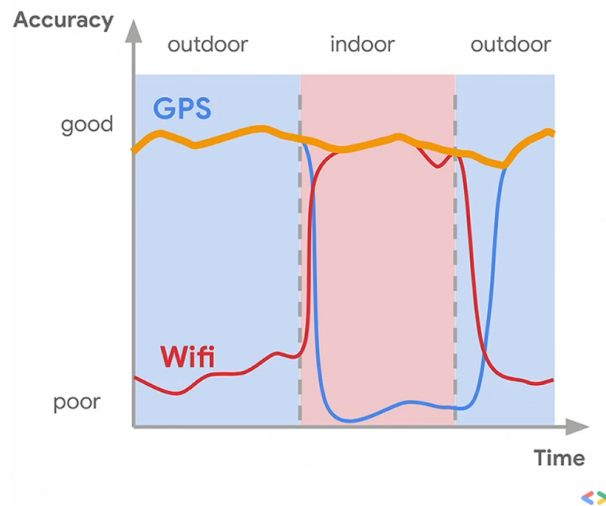


Figura 4.4: La precisión de la ubicación en las transiciones del aire libre al interior de algún edificio y viceversa se puede suplementar con el uso de los sensores del teléfono (acelerómetro, giroscopio, magnetómetro, etc.).

Fuente: adaptada de [19].

- Una caja de voz conectada al ordenador (Fig. 4.6)
- Un teléfono analógico (Fig. 4.7).

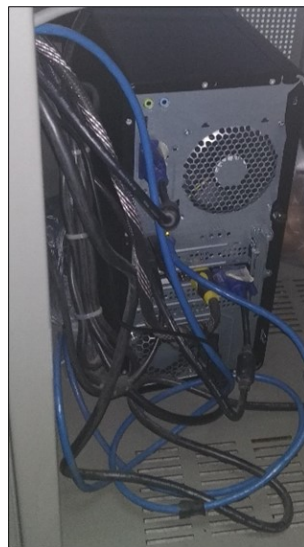


Figura 4.5: Ordenador que forma parte de cada consola que manejan las operadoras del SIS ECU911.

Fuente: autor, 2023.

El modelo particular de teléfono se identificó como el GN Netcom 7100, al cual se conectan auriculares con los cuales la operadora responde a las llamadas. Estos auriculares se conectan a un puerto RJ-9 (Fig. 4.8). Para probar la integración del sistema propuesto, sería necesario interponer entre el teléfono y los auriculares un adaptador



Figura 4.6: Caja de Voz que forma parte de cada consola que manejan las operadoras del SIS ECU911.

Fuente: autor, 2023.

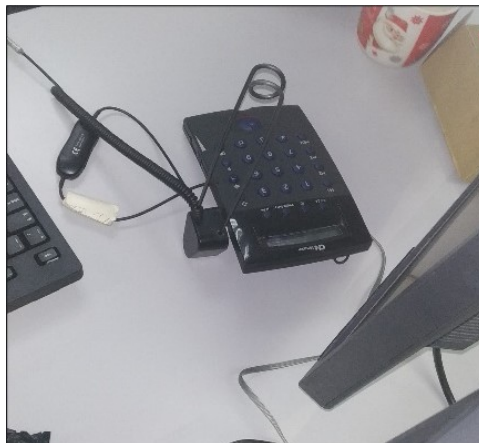


Figura 4.7: Teléfono que forma parte de cada consola que manejan las operadoras del SIS ECU 911.

Fuente: autor, 2023.

divisor para conectores RJ-9. Ya que el módulo MT8870 únicamente cuenta con un puerto de audio de 3.5 mm (Fig. 4.9), al segundo puerto de este divisor se conectaría un adaptador de conexiones RJ-9 a conexiones de audio de 3.5 mm, para posteriormente conectar el dispositivo electrónico, el mismo que se prevé crear (Fig. 4.10). Los adaptadores mencionados se pueden conseguir en el mercado en la actualidad. Si bien se podría llevar a cabo el procesamiento del audio de forma digital directamente en el ordenador de la consola, con este esquema se gana universalidad, ya que se puede despreocupar del sistema operativo, del software propietario que este posiblemente ocupe, del codec particular del audio, etc.; sin hablar de futuras actualizaciones ni modificaciones a cualquiera de estos.



Figura 4.8: Puerto RJ9 del teléfono, modelo GN Netcom 7100, al que se le conectan auriculares con los cuales contestan las llamadas las operadoras del SIS ECU 911. Fuente: autor, 2023.

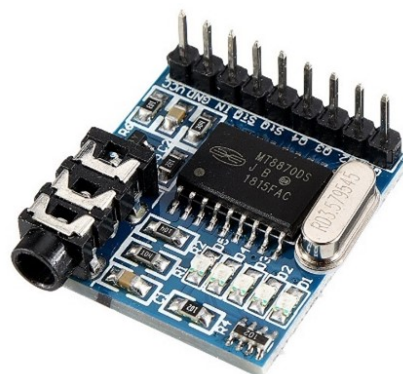


Figura 4.9: Módulo MT8870: decodifica los tonos DTMF de la señal de audio entrante al puerto de 3.5 mm. Establece un código de 4 bits en sus pines de salida que comunica cuál tono DTMF fue recibido. Fuente: adaptada de [22].

Al respecto del esquema de codificación, inicialmente se pensó transcribir la cadena de bits que representarían las coordenadas geográficas a hexadecimal, dado que originalmente la especificación que describe la codificación DTMF [23] permitía el uso de 16 símbolos (Fig. 4.11). Sin embargo, el API disponible en el ámbito de desarrollo de

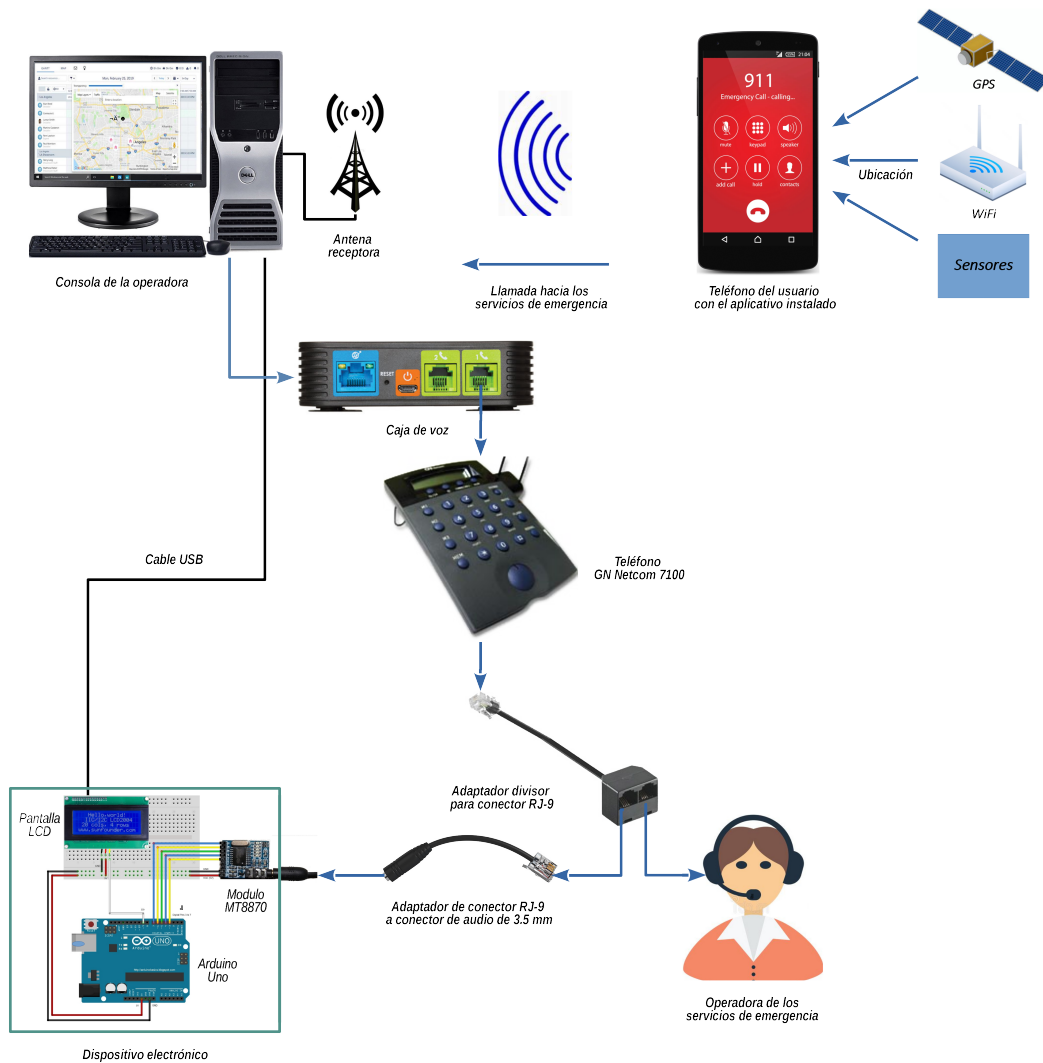


Figura 4.10: Esquema de como se integraría el sistema propuesto a los sistemas del SIS ECU 911.

Fuente: autor, 2023.

aplicaciones para el S.O. Android permite únicamente el marcado de los 12 símbolos convencionales (los dígitos del 0 al 9, el asterisco y el símbolo numeral). Por lo que se opta por el esquema de codificación que se observa en la Tabla 4.1:

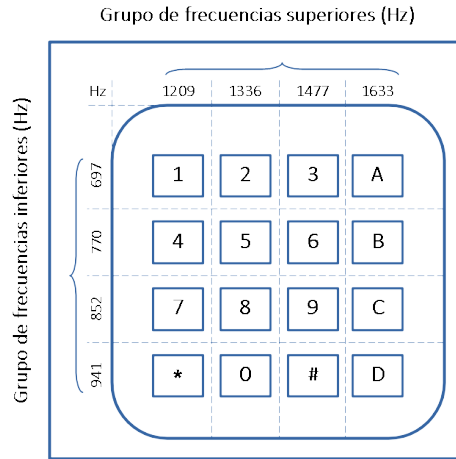


Figura 4.11: Arreglo de teclas originalmente propuesto en el estándar UTI-T Q.23 para el sistema de tonos DTMF, con 16 teclas en total.  
Fuente: autor, 2023.

Tabla 4.1: Esquema de codificación.

Símbolo(s)	Código(s)
Numerales de 0 a 9	Numerales de 0 a 9
Símbolo negativo (-)	# o ##
Punto (.)	**
Coma (,)	*#
Fin de la secuencia	#*

Con esta base entonces, a continuación se muestran dos ejemplos del uso de esta codificación:

$$\begin{array}{ccc}
 -2.90860,78.99191 & \rightarrow & ##2**90860*#78**99191#* \\
 -2.90860,-78.99191 & \rightarrow & ##2**90860*##78**99191#*
 \end{array} \tag{4.1}$$

**Símbolos**
**Codificación**

Nótese que, en el segundo ejemplo, el signo negativo correspondiente a la latitud (el primero número decimal) se codifica como dos símbolos numerales. Mientras que para representar el signo negativo correspondiente a la longitud (el segundo número decimal) que viene inmediatamente después de la coma se representa con un único símbolo numeral. Esto para evitar codificar dos símbolos negativos en vez de uno solo

(i.e.: la decodificación compara dos símbolos a la vez).

Cabe recalcar que para que esta propuesta funcione, no requerirá de servicio de datos móviles, únicamente contar con suficiente cobertura para enviar llamadas. Con este esquema se posibilitará una respuesta más ágil y con una ubicación más precisa de los servicios de emergencia, lo cual a su vez podrá mitigar los efectos sobre los que sufran alguna calamidad.

#### 4.1.2. Diseño e implementación del dispositivo de decodificación

Para el diseño del dispositivo decodificador, se valió de diseños existentes para este fin [24], los cuales se sirven del módulo MT8870 [25], para decodificar los tonos DTMF de la señal analógica de audio de la llamada y de ahí remitir un código binario correspondiente de 4 bits al Arduino, según lo que se muestra en la Tabla 4.2.

Tabla 4.2: Tabla de los 4 bits que codifican cada tono DTMF (Módulo MT8870).

Dígito	Q0	Q1	Q2	Q3
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	0	1	0	1
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0
Cualquiera	Z	Z	Z	Z

Basándose en un programa de ejemplo para Arduino [24], que permite decodificar los tonos DTMF utilizando los 4 bits provenientes módulo MT8870, se adaptó a este programa para recuperar las coordenadas geográficas codificadas con tonos DTMF, según el esquema de codificación de la Tabla 4.1.



Una vez obtenidas las coordenadas geográficas, estas son desplegadas a una pantalla LCD de 20x4 caracteres, utilizando el protocolo I2C [26]. Adicionalmente, se puede transmitir las coordenadas por la interfaz serial del Arduino, conectando este al puerto USB de un ordenador.

El diagrama de conexión de los diferentes componentes es el que se observa en la Figura 4.12

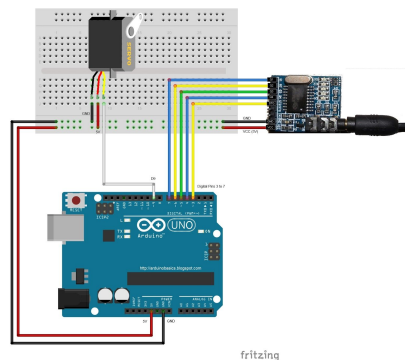


Figura 4.12: Diagrama de conexión del dispositivo receptor.  
Fuente: adaptada de [27].

Esto resume el funcionamiento del dispositivo receptor/decodificador. El programa completo de decodificación fue almacenado en la plataforma GitHub [9].

### 4.1.3. Diseño e implementación de la aplicación transmisora

Para el diseño de la aplicación encargada de obtener las coordenadas geográficas del usuario que acude a llamar a los servicios de emergencia, se decidió modificar a una aplicación para llamadas telefónicas de código abierto, llamado **Simple Dialer** (Fig. 4.13), cuyo código fuente está disponible en la plataforma de GitHub [28]. Simple Dialer está programado para el S.O. Android, utilizando Kotlin, el lenguaje de programación de tipado estático, desarrollado por la empresa checa JetBrains [29].

Al examinar el código fuente de la aplicación, se observa que el mismo está organizado en varios archivos de interfaz gráfica (Fig. 4.14), y con sus respectivos programas (Fig. 4.15), estos últimos que controlan el funcionamiento lógico de la aplicación.

Al observar en particular el programa correspondiente a la interfaz de llamadas (la interfaz que se muestra durante una llamada, [Fig. 4.16]) de la aplicación, se nota que

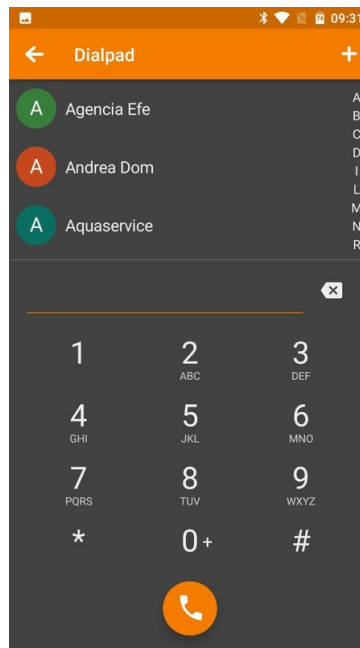


Figura 4.13: Aplicación de llamadas de código abierto, Simple Dialer, para el S.O. Android.

Fuente: adaptada de [30].

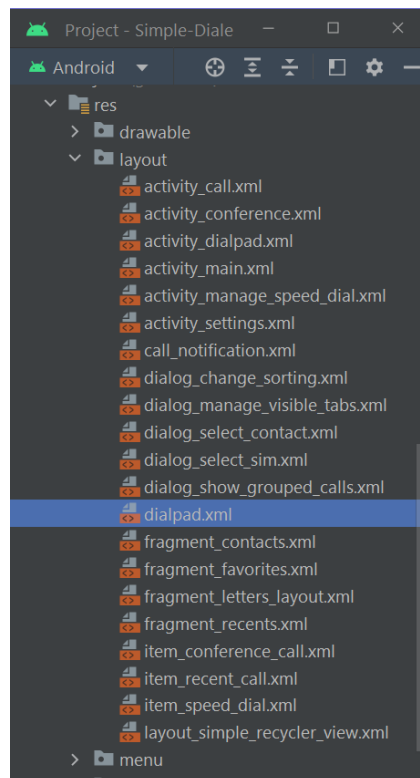


Figura 4.14: Listado de archivos de interfaces gráficas de la aplicación, Simple Dialer. Fuente: autor, 2023.

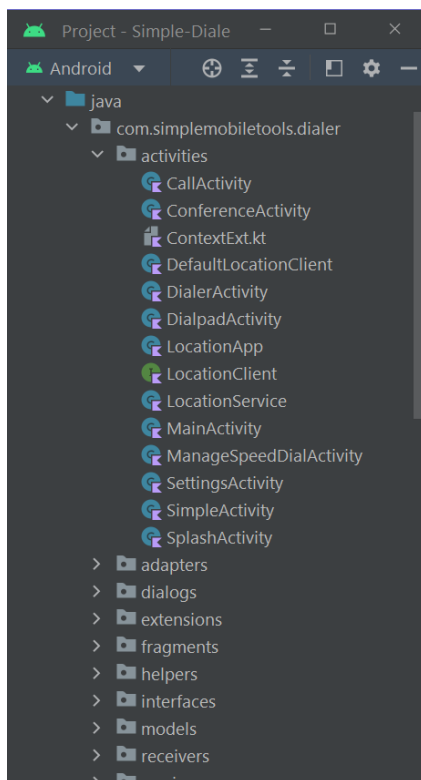


Figura 4.15: Listado de módulos que determinan el funcionamiento programático de la aplicación, Simple Dialer.

Fuente: autor, 2023.

el marcar una tecla en medio de una llamada inicia un llamado a una función. Esta función se encarga de reproducir el tono DTMF correspondiente a la tecla marcada en medio del audio de la llamada (Extracto de Código 4.1). Es esta función a la que se acudirá para emitir los tonos DTMF para codificar las coordenadas geográficas obtenidas de la API de ubicación del S.O. Android.

```
private fun dialpadPressed(char: Char) {  
    ...  
    CallManager.keypad(this, char) // Se ejecuta al marcar  
    una tecla.  
    ...  
}
```

Extracto de Código 4.1: Función llamada al marcar una tecla.

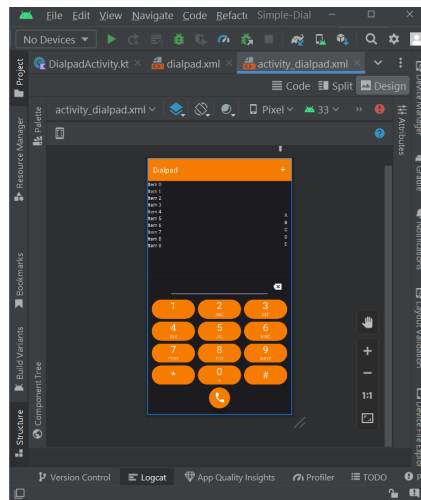


Figura 4.16: La interfaz gráfica del teclado para el marcado de llamadas de la aplicación, Simple Dialer.  
Fuente: autor, 2023.

#### 4.1.3.1. Marcación de los tonos DTMF

Para determinar en qué ubicación del programa se insertará el código que marcará las coordenadas geográficas, se insertó mensajes de registro log [31] en cada función del módulo que rige la lógica de las llamadas en la aplicación; observando cuáles de estas se ejecutan al momento que la contraparte contesta la llamada (Fig. 4.17). Se eligió como candidata la función **updateCallAudioState()**, encargada de dirigir por qué salida de audio se transmite el audio de la llamada (altoparlante, auricular, bluetooth, etc.).

#### 4.1.3.2. Recepción de la ubicación usando la API

Basándose en un videotutorial que indica como obtener el dato de la ubicación en el desarrollo de un programa de Android [32], para este fin, primero es necesario especificar como dependencia al servicio de ubicación en el archivo **build.gradle** (El archivo correspondiente a la aplicación.) (Extracto de Código 4.2).

```
dependencies {
    ...
    \\ Servicios de ubicacion
```

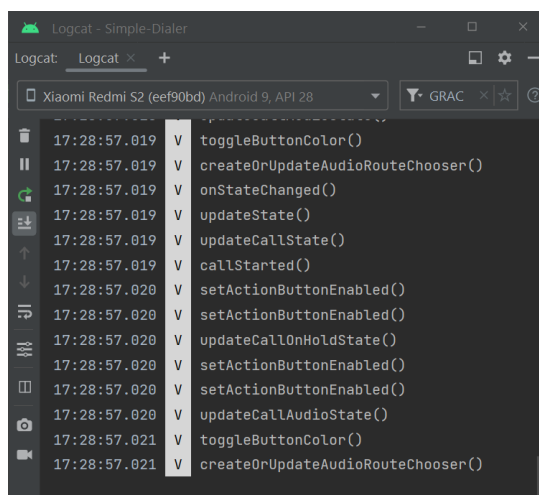


Figura 4.17: Captura de pantalla de los mensajes de log, indicando cuáles funciones del módulo de llamadas se ejecutan en la aplicación, Simple Dialer, una vez que la contraparte conteste la llamada.

Fuente: autor, 2023.

```

        implementation 'com.google.android.gms:play-services-
            location:20.0.0'
        ...
    }

```

Extracto de Código 4.2: Código para incluir los servicios de ubicación en la aplicación.

Posteriormente, se deberá especificar los permisos que el usuario deberá acceder para que se pueda obtener la ubicación del móvil. Esto se lleva a cabo en el archivo de manifiesto del proyecto, **AndroidManifest.xml** (Extracto de Código 4.3). Se deben incluir permisos tanto para la ubicación de baja precisión, como también la de alta precisión. Adicionalmente, se requiere solicitar el permiso de un servicio de primer plano, que ejecutará la obtención de las coordenadas cuando la aplicación no esté en primer plano en el móvil. Finalmente, se requiere del permiso para mostrar notificaciones al usuario mientras se esté accediendo a la ubicación, ya que por precaución por la privacidad del usuario, el entorno de Android exige que el usuario sea notificado.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/
    android"
        xmlns:tools="http://schemas.android.com/tools"
        package="com.simplemobiletools.dialer"

```

```
    android:installLocation="auto">
    ...
    <!-- PERMISOS REQUERIDOS -->
    <uses-permission android:name="android.permission.INTERNET
        " />
    <uses-permission android:name="android.permission.
        ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.
        ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.
        FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.
        POST_NOTIFICATIONS" />
    ...
</manifest>
```

Extracto de Código 4.3: Inclusión de los permisos en el archivo manifiesto de la aplicación

A continuación se implementó una interfaz [33], llamada **LocationClient**, para obtener la ubicación del usuario, mismo que se observa en el Extracto de Código 4.4. En esta interfaz se implementa una función que brindará actualizaciones de la ubicación; esta función recibe como parámetro el intervalo en milisegundos para estas actualizaciones. El resultante de esta función será un flujo de datos [34] en el cual se ingresarán las actualizaciones de la ubicación del móvil.

```
package com.simplemobiletools.dialer.activities

import android.location.Location
import kotlinx.coroutines.flow.Flow

interface LocationClient {
    fun getLocationUpdates(interval: Long): Flow<Location>
```

```
class LocationException(message: String): Exception()
}
```

Extracto de Código 4.4: Interface para los clientes del servicio de ubicación.

El Extracto de código 4.5 muestra la implementación de la interfaz `LocationClient`. Los parámetros que se le pasa a la función son el Contexto actual del programa [35] y un cliente del Proveedor de Servicio de Ubicación Fusionada [21] [36]. La interfaz primero asegura que el usuario haya accedido a los permisos requeridos (Extracto de Código 4.6), por medio de una función que extiende la funcionalidad del objeto contexto (Extracto de Código 4.7). En el caso de que el usuario no haya accedido a alguno de los permisos, este módulo devolverá una excepción señalando esto.

```
package com.simplemobiletools.dialer.activities

import ...

class DefaultLocationClient(
    private val context: Context,
    private val client: FusedLocationProviderClient
): LocationClient {
```

Extracto de Código 4.5: Ingresan argumentos al cliente del servicio de ubicación.

```
@SuppressWarnings("MissingPermission")
override fun getLocationUpdates(interval: Long): Flow<Location
> {

    return callbackFlow {
        if(!context.hasLocationPermission()) {
            throw LocationClient.LocationException("Falta un
                permiso de ubicacion.")
        }
    }
}
```

Extracto de Código 4.6: El cliente ejecuta una función del objeto Contexto.

```
fun Context.hasLocationPermission(): Boolean {  
  
    return ContextCompat.checkSelfPermission(  
        this,  
        Manifest.permission.ACCESS_COARSE_LOCATION  
    ) == PackageManager.PERMISSION_GRANTED &&  
        ContextCompat.checkSelfPermission(  
            this,  
            Manifest.permission.ACCESS_FINE_LOCATION  
        ) == PackageManager.PERMISSION_GRANTED  
}
```

Extracto de Código 4.7: Función del objeto Contexto para verificar los permisos de la aplicación.

De ahí, en la implementación de la interfaz se verifica si está posibilitado obtener la ubicación, ya sea por el módulo GPS del celular o por la red inalámbrica (WiFi o Red de Datos) al que esté conectado el equipo (Extracto de Código 4.8), caso contrario, de nuevo se emitirá una excepción.

```
val locationManager = context.getSystemService(Context.  
    LOCATION_SERVICE) as LocationManager  
  
val isGpsEnabled = locationManager.isProviderEnabled(  
    LocationManager.GPS_PROVIDER)  
  
val isNetworkEnabled = locationManager.isProviderEnabled(  
    LocationManager.NETWORK_PROVIDER)  
  
if(!isGpsEnabled && !isNetworkEnabled) {  
    throw LocationClient.LocationException("Los proveedores de  
        ubicacion estan deshabilitados.")  
}
```



```
}
```

Extracto de Código 4.8: Verificación de la habilitación para obtener la ubicación por GPS o por Internet.

A continuación, se genera la solicitud de la ubicación, especificando como parámetros; la frecuencia con la cual se requiere las actualizaciones y la prioridad de la precisión de la ubicación (Extracto de Código 4.9). También se implementa una retrollamada que se ejecutará cada vez que al cliente del servicio de ubicación le llegue una actualización de la ubicación (Extracto de Código 4.10). En la variable de esta función llamada resultados estará incluida la última ubicación obtenida del servicio. El programa entonces inserta dicha ubicación en el flujo de datos, para acceder a esta ubicación desde otros módulos del programa (específicamente en el módulo **CallActivity.kt**, que maneja la lógica de las llamadas).

```
val request = LocationRequest.create()
    .setInterval(interval)
    .setFastestInterval(interval)
    .setPriority( Priority.PRIORITY_HIGH_ACCURACY )
```

Extracto de Código 4.9: Se genera el pedido de la ubicación.

```
val locationCallback = object : LocationCallback() {
    override fun onLocationResult(result: LocationResult) {
        super.onLocationResult(result)
        result.locations.lastOrNull()?.let { location ->
            launch { send(location) }
        }
    }
}
```

Extracto de Código 4.10: Retrollamada que se ejecuta al llegar una nueva ubicación.

Al cliente del proveedor del servicio de ubicación fusionada, se le proporciona la petición por la ubicación y la retrollamada como parámetros, para que el proveedor del servicio a su vez devuelva la ubicación (Extracto de Código 4.11). Finalmente, se es-

pecifica que, cuando el módulo que accede a este servicio se cierre, también se deje de recibir actualizaciones de la ubicación (Extracto de Código 4.12).

```
client.requestLocationUpdates(  
    request,  
    locationCallback,  
    Looper.getMainLooper()  
)
```

Extracto de Código 4.11: Se pasan parámetros al cliente de ubicación.

```
awaitClose {  
    client.removeLocationUpdates(locationCallback)  
}
```

Extracto de Código 4.12: Función de cierre del cliente de ubicación.

Como siguiente paso, se deberá implementar un servicio de primer plano, que se encargará de proveer la ubicación, en el caso que la aplicación de llamadas no este en primer plano del móvil (p. ej.: cuando el usuario cambie enfoque a otra aplicación). Este servicio, llamado **LocationService** (Extracto de Código 4.13), hereda de la clase **Service** (Clase Servicio). En el método requerido para vincular el servicio a otro módulo, llamado **onBind**, se devuelve un objeto nulo, ya que no se pretende acceder a esta funcionalidad (Extracto de Código 4.14). Para la función **onCreate**, que se ejecuta el momento que este servicio se cree, se le pasa el objeto Contexto de la aplicación y el cliente del servicio de ubicación de Android (Extracto de Código 4.15). La función **onStartCommand()**, se encarga de llamar a las funciones de inicio o detención de este servicio (Extracto de Código 4.16).

```
class LocationService: Service() {  
  
    private val serviceScope = CoroutineScope(SupervisorJob()  
        + Dispatchers.IO)  
    private lateinit var locationClient: LocationClient  
    ...  
}
```

```
}
```

Extracto de Código 4.13: Servicio de ubicación de segundo plano.

```
override fun onBind(p0: Intent?): IBinder? {  
    return null  
}
```

Extracto de Código 4.14: Función de vinculación.

```
override fun onStartCommand(intent: Intent?, flags: Int,  
    startId: Int): Int {  
    when(intent?.action) {  
        ACTION_START -> start()  
        ACTION_STOP -> stop()  
    }  
    return super.onStartCommand(intent, flags, startId)  
}
```

Extracto de Código 4.15: Función que se ejecuta al iniciarse el servicio de segundo plano.

```
override fun onCreate() {  
    super.onCreate()  
    locationClient = DefaultLocationClient(  
        applicationContext,  
        LocationServices.getFusedLocationProviderClient(  
            applicationContext)  
        )  
}
```

Extracto de Código 4.16: Función que se ejecuta en la creación del servicio de segundo plano.

La función de inicio crea la notificación que será desplegada en el área de notificaciones, dejando saber al usuario que su ubicación se está rastreando por la aplicación. Adicionalmente, utiliza el cliente del servicio de ubicación, para recibir actualizaciones

de la ubicación y modificar la notificación con las nuevas ubicaciones (Extracto de Código 4.17). La función de detención a cambio detiene el despliegue de la notificación y también detiene al servicio (Extracto de Código 4.18).

```
private fun start() {
    val notification = NotificationCompat.Builder(this, "
        location")
        .setContentTitle("Raestreando la ubicacion...")
        .setContentText("Ubicacion: nula")
        .setSmallIcon( R.drawable.ic_launcher_background )
        .setOngoing(true)

    val notificationManager = getSystemService(Context.
        NOTIFICATION_SERVICE) as NotificationManager

    locationClient
        .getLocationUpdates(10000L)
        .catch { e -> e.printStackTrace() }
        .onEach { location ->
            val lat = location.latitude.toString()
            val long = location.longitude.toString()
            val accur = location.accuracy .toString()
            val updatedNotification = notification.
                setContentText(
                    "Ubicacion: ($lat, $long, $accur)"
                )
            notificationManager.notify(1, updatedNotification.
                build())
        }
        .launchIn(serviceScope)

    startForeground(1, notification.build())
}
```

```
}
```

Extracto de Código 4.17: Función de inicio del servicio de segundo plano.

```
private fun stop() {  
    stopForeground(true)  
    stopSelf()  
}
```

Extracto de Código 4.18: Función de detención del servicio de segundo plano.

Es necesario registrar el servicio de primer plano en el archivo de manifiesto de la aplicación (Extracto de Código 4.19). Para que el servicio de primer plano pueda desplegar notificaciones, es necesario crear una clase de tipo aplicación, para que esta a su vez se encargue de crear un canal de notificaciones para poder desplegar estas notificaciones (Extracto de Código 4.20).

```
<service android:name=".activities.LocationService" android:  
    foregroundServiceType="location"/>
```

Extracto de Código 4.19: Se registra el servicio en el archivo manifiesto de la aplicación.

```
override fun onCreate() {  
    super.onCreate()  
    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        val channel = NotificationChannel(  
            "location",  
            "Location",  
            NotificationManager.IMPORTANCE_LOW  
        )  
        val notificationManager = getSystemService(Context.  
            NOTIFICATION_SERVICE) as NotificationManager  
        notificationManager.createNotificationChannel(channel)  
    }  
}
```

```
}
```

Extracto de Código 4.20: Clase de tipo Aplicación que crea el canal de notificaciones para el servicio de segundo plano.

Finalmente, antes de acceder a la ubicación dentro de otros módulos de la aplicación, en el módulo principal de la aplicación (**MainActivity.kt**) es necesario solicitar al usuario que acceda a los permisos de ubicación (Extracto de Código 4.21).

```
class MainActivity : SimpleActivity() {  
    ...  
    override fun onCreate(savedInstanceState: Bundle?)  
    {  
        ...  
        if (isDefaultDialer()) {  
            checkContactPermissions()  
  
            ActivityCompat.requestPermissions(  
                this,  
                arrayOf(  
                    Manifest.permission.ACCESS_COARSE_LOCATION ,  
                    Manifest.permission.ACCESS_FINE_LOCATION ,  
                ),  
                0  
            )  
            ...  
        }  
    }  
}
```

Extracto de Código 4.21: Se solicita acceder a los permisos al usuario.

#### 4.1.3.3. La obtención y el marcado de las coordenadas geográficas de la API de ubicación del S.O. Android

Tras la programación preparativa anterior, para obtener la ubicación del usuario al realizar una llamada, en el método **onCreate()** del programa **CallActivity.kt**, que se ejecuta al iniciar una llamada, es necesario iniciar el servicio de ubicación (Extracto de Código 4.22). Adicionalmente es necesario iniciar el cliente que será servido con la ubicación (Extracto de Código 4.23). Tras esto, el cliente solicita recibir actualizaciones de la ubicación cada cierto intervalo de tiempo. Para los fines de probar el funcionamiento de la aplicación, se eligió recibir actualizaciones cada cinco minutos (300,000 milisegundos). En cada llegada de una nueva actualización de la ubicación, el cliente almacena los valores de latitud y longitud en una variable de texto, la misma a la que se accederá en otros métodos de este módulo (Extracto de Código 4.24).

Cabe recalcar que, para los fines que se destina este sistema, se limitó a transmitir únicamente los primeros cinco decimales de la latitud y longitud (truncando, no redondeando), ya que esto daría una precisión máxima de 11.1 metros a la redonda [37]. Se pensó que este nivel de precisión sería suficiente para ubicar a un usuario en una situación de emergencia, aunque esta configuración se podrá modificar en futuras iteraciones del sistema, según se considere apropiado.

```
override fun onCreate(savedInstanceState: Bundle?)
{
    // Iniciar el servicio de ubicacion
    Intent(applicationContext, LocationService::class.java).
        apply {
            action = LocationService.ACTION_START
            startService(this)
        }
}
```

Extracto de Código 4.22: Código en que inicia el servicio de ubicación.

```
// INICIAR EL CLIENTE DE UBICACION
locationClient = DefaultLocationClient(
    applicationContext,
```

```
        LocationServices.getFusedLocationProviderClient(  
            applicationContext)  
    )
```

Extracto de Código 4.23: Código que inicia el cliente del servicio de ubicación.

```
locationClient  
    .getLocationUpdates(300000L)  
    .catch { e -> e.printStackTrace() }  
    .onEach { location ->  
        lat = "%09.5f".format(location.latitude)  
        long = "%010.5f".format(location.longitude)  
  
        Todito = " " + lat + "," + long  
  
    }  
    .launchIn(serviceScope)
```

Extracto de Código 4.24: Código que crea la solicitud de ubicación.

Entonces, una vez que se inicia una llamada, en el método **onCreate()**, se inician el servicio y el cliente del servicio. Al recibir una actualización de la ubicación, el cliente almacena esta, en una variable de texto (la variable **Todito**).

Al momento que la llamada sea contestada, en el método **updateCallAudioState()** (Extracto de Código. 4.25), se itera por la variable de texto que almacena la ubicación, marcando sucesivamente el símbolo de código correspondiente de cada carácter, según el esquema de codificación (Tabla 4.1). Tras iterar por todo el texto de la variable de ubicación, se transmite el símbolo de fin de la secuencia, señalando al receptor que se han enviado todos los datos.

```
private fun updateCallAudioState(route: AudioRoute?)  
{  
    ...  
    if (route != null)  
    {
```



```
...
// Inicio de la secuencia
CallManager.keypad( this, '#' )
CallManager.keypad( this, '*' )
for ( i in 1..Todito.length - 1 )//+ 1 )
{
    if ( Todito[ i ] == ' ' )
    {
        //      NO SE HACE NADA EN CASO DE LOS
        ESPACIOS
    }
    else if ( Todito[ i ] == '-' )
    {
        if ( Todito[ i - 1 ] == ',' )
        {
            CallManager.keypad( this, '#' )
        }
        else
        {
            CallManager.keypad( this, '#' )
            CallManager.keypad( this, '#' )
        }
    }
    else if ( Todito[ i ] == ',' )
    {
        CallManager.keypad( this, '*' )
        CallManager.keypad( this, '#' )
    }
    else if ( Todito[ i ] == '.' )
    {
        CallManager.keypad( this, '*' )
        CallManager.keypad( this, '*' )
    }
}
```

```
    }  
    else  
    {  
        CallManager.keypad( this, Todito[ i ] )  
    }  
}  
// Fin de la secuencia, #* :  
CallManager.keypad( this, '#' )  
CallManager.keypad( this, '*' )  
}  
...  
}
```

Extracto de Código 4.25: Código para codificar y marcar las coordenadas y marca los tonos DTMF.

Esto resume el funcionamiento de la aplicación, diseñada para transmitir las coordenadas geográficas hacia el dispositivo decodificador. La aplicación completa fue almacenada en la plataforma de GitHub [9].

#### 4.1.4. Integración y funcionamiento del sistema

El flujograma del funcionamiento completo de sistema se visualiza en la Figura 4.18. Este consiste primero, en la aplicación que obtiene las coordenadas geográficas, las codifica y las reproduce en el audio de la llamada, y segundo, del receptor, que está a la escucha continua por estos tonos, los decodifica para finalmente recuperar las coordenadas geográficas.

Para comprobar el funcionamiento conjunto del sistema, se llevó a cabo unas pruebas básicas [38]. En el transcurso de estas pruebas se encontró con la situación que, si se corta la llamada durante el proceso de estar recibiendo las coordenadas, antes que se logre enviar el código de fin de secuencia, en algunos casos quedaban coordenadas parciales en la memoria del receptor. Estos restos de la transmisión anterior, interferían con la transmisión actual. Para solucionar esto, se resolvió transmitir la secuencia

de fin de secuencia tanto al inicio como al fin de la transmisión de coordenadas. Esto borraría la memoria y permitiría que las nuevas coordenadas ingresen exitosamente.

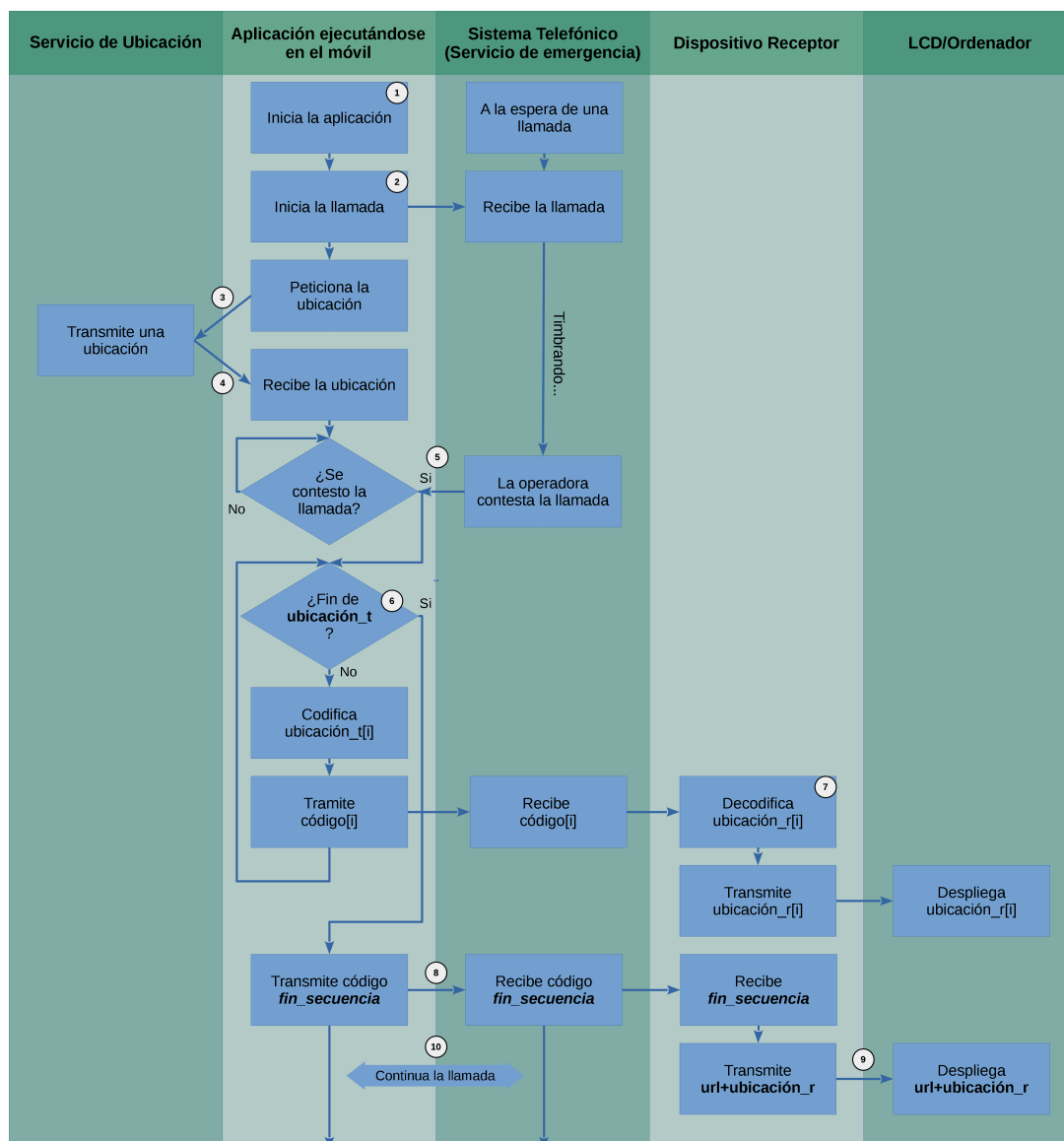


Figura 4.18: Flujograma del funcionamiento del sistema.

Fuente: autor, 2023.

Tal como se describen en la Figura 4.18, las etapas del funcionamiento del sistema son las siguientes:

1. Se inicia la aplicación.
2. Se inicia la llamada. Al servicio de emergencias se le notifica la llamada por medio del timbrado.
3. Se solicita al servicio de ubicación del S.O. la ubicación del móvil.

4. El servicio de ubicación eventualmente entrega a la aplicación la ubicación actual del móvil, la misma que se almacena en una variable.
5. Se espera que la operadora conteste la llamada iniciada por el usuario.
6. Una vez que se contesta la llamada, se envía el código de inicio de secuencia. De ahí, se itera sobre la variable de la ubicación, marcando el/los código(s) correspondiente a cada carácter según el esquema de codificación.
7. El equipo receptor/decodificador recibe los códigos transmitidos y decodifica cada símbolo correspondiente a estos, recuperando así las coordenadas geográficas. Así mismo, remite estas coordenadas una por una a la pantalla LCD y al ordenador por la conexión serial.
8. Tras iterar por todos los caracteres de la variable de la ubicación, se transmite el código de fin de la secuencia, señalando al receptor que no se transmitirá más símbolos.
9. En este momento, el equipo receptor/decodificador puede transmitir la URL completa, compuesta por unir la primera parte de la URL con las coordenadas que hayan sido recibidas (este funcionamiento se describe en detalle a la Subsección 4.1.6.1).
10. Finalizada la transmisión de las coordenadas, la conversación entre el usuario y la operadora puede continuar normalmente.

#### **4.1.5. Materiales y características de los componentes**

En la Tabla 4.3 se detallan la lista de materiales utilizados en la implementación y en las subsiguientes pruebas del sistema.

Con respecto de especificaciones técnicas del teléfono móvil, Redmi S2, del fabricante Xiaomi [39], es importante notar que, además de utilizar el Sistema de Posicionamiento Global (GPS), también funciona con las tecnologías de AGPS, y con los GNSS de GLONASS (el análogo ruso del sistema GPS) y BeiDou (el análogo chino del sistema GPS) [40].

Tabla 4.3: Lista de materiales.

<b>No.</b>	<b>Componente</b>	<b>Unidad</b>	<b>Cantidad</b>
1	Arduino Uno	u	1
2	Modulo MT8870	u	1
3	Cable USB A a B, macho-macho (Incluido con el Arduino)	u	1
4	Cables Dupont Jumper, 20 cm, macho-hembra	u	11
5	Cable de audio de 3.5 mm, 100 cm, macho-macho	u	1
6	Pantalla LCD 2004, azul, 20x4 caracteres	u	1
7	Teléfono celular Redmi S2, 2018, Android 9 (Pie)	u	1
8	Teléfono celular Samsung J1 Prime, 2016, Android 6 (Marshmallow)	u	1
9	Laptop HP Pavilion dv7, Intel Core i5-2410M, Windows 10 Enterprise 64-bits	u	1

En la Tabla 4.4, se detallan las aplicaciones utilizados para el desarrollo y funcionamiento del Sistema.

Tabla 4.4: Lista de software.

<b>Aplicación</b>	<b>Versión</b>
Arduino IDE	2.0.4
Android Studio	2022.1.1
Simple Dialer	5.17.3
PuTTY-url	0.78

#### 4.1.6. Funcionalidades adicionales

Se investigó y se implementó otras funcionalidades suplementarias para el sistema que podrían ser útiles para un mejor funcionamiento del mismo. Estas incluyen:

- El formateo y acople de la comunicación serial de los datos a un programa, que permita visualizar la ubicación del móvil en un sitio web de información cartográfica (p.ej.: el sitio de Google Maps).
- Obtener la altitud como dato adicional de la ubicación del móvil.
- Obtener el nivel de batería del móvil, lo cual puede ser útil en los casos en que los agentes que respondan a una llamada de emergencia estén lejos o con un difícil acceso a los sitios de los hechos y fuera necesario limitar la comunicación para no desgastar por completo las reservas de energía del móvil.
- Obtener el dato de precisión de las coordenadas, para informar mejor a los agentes en el caso de que se necesite llevar acabo una búsqueda acotada en el sitio de los hechos para encontrar a los afectados por la emergencia.
- Diseño de un dígito de paridad: A pesar que en las pruebas que se llevaron acabo, se tuvo excelente fidelidad en el envío de los datos, dada la delicada naturaleza de los mismos, se consideró necesario implementar un dígito de paridad. Esto para alertar ante una errónea transmisión de los datos.

Se describen cada una de estas funcionalidades a continuación.

#### 4.1.6.1. Visualización de la ubicación en Google Maps

Con el fin de facilitar la presentación de la ubicación geográfica para la operadora de los servicios de emergencia; para que esta pueda comunicar mejor la información hacia los agentes que responden ante una emergencia, se propuso desplegar la ubicación en un navegador que acceda a un sitio web de información cartográfica, como es el de Google Maps. Para permitir esto, fue necesario dos pasos:

1. Formatear las coordenadas para que formen parte de una dirección URL [41], y que al acceder a este enlace, se despliegue automáticamente la ubicación en un sitio web de mapas, como el de Google Maps.
2. Configurar para poder acceder a los datos seriales que provienen del Arduino en un cliente de SSH, de tal forma que al dar clic sobre el enlace, este abra el navegador y acceda al sitio web de la URL.

Para lo primero, fue necesario almacenar el inicio de la URL en una variable en el programa del Arduino (Extracto de Código 4.26). De ahí que, fue necesario simplemente adjuntar las coordenadas geográficas que se recuperen, de los tonos DTMF del audio de la llamada al final de la URL. En el mismo sentido, fue necesario codificar la coma que separa la latitud y la longitud en la URL con el símbolo '%2C', debido al código URL [42]. Posteriormente, se junta la primera parte de la URL con las coordenadas por medio de la impresión consecutiva de estas al puerto serial (Extracto de Código 4.27).

```
char url[] = "https://www.google.com/maps/search/?api=1&query  
=";
```

Extracto de Código 4.26: Variable con la primera parte de la URL.

```
Serial.print( url );  
  
for (byte i = 0; i < counter-12; i = i + 1) {  
  Serial.print(secuencia[i]);  
}
```

Extracto de Código 4.27: La consecutiva impresión de las variables al puerto serial.

Para lo segundo, se conoce que es posible acceder al flujo de datos del puerto serial proveniente de un Arduino, utilizando clientes de SSH, tal como PuTTY [43]. Adicionalmente, una modificación no oficial de PuTTY, llamado PuTTY-ur1 [44], permite hacer clic sobre un enlace que se presente sobre el despliegue de texto del emulador de terminal de PuTTY-ur1.

Valiéndose de este aplicativo, es simplemente un tema de configurar el cuadro de diálogo de PuTTY-ur1 para conexiones seriales y configurar el número de puerto al que este conectado el Arduino, como se observa en la Figura 4.19. El número del puerto al que este conectado el Arduino se puede averiguar en el **Administrador de Dispositivos** de Windows, en la sección de puertos (Fig. 4.20). Adicionalmente es posible (y recomendable) guardar una bitácora o **log** de los datos recibidos, para mayor seguridad. Para ello se configura que se guarden todos los datos, y que se agreguen nuevos datos al final del archivo, en lugar de sobrescribir sobre el archivo (Fig. 4.21). Finalmente, también sería recomendable guardar las configuraciones para futuras sesiones de recepción de datos del Arduino (Fig. 4.22).

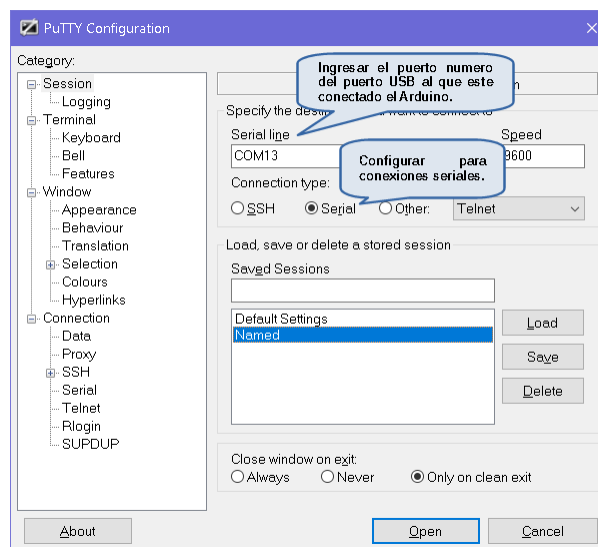


Figura 4.19: Configuración de PuTTY-ur1 para la comunicación serial y con el número de puerto del Arduino.

Fuente: autor, 2023.

Una vez que se haga clic en **Open**, se abre una sesión de comunicación serial con el Arduino (Fig. 4.23). Al recibir una nueva ubicación por medio de este canal, se puede hacer clic sobre la dirección URL, ante lo cual se abrirá una ventana del navegador por defecto en el ordenador; con ello se accederá al sitio web con la ubicación señalada



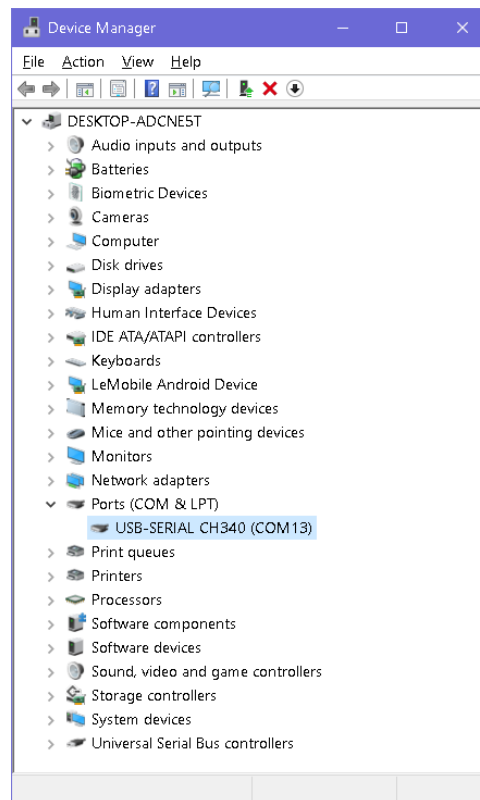


Figura 4.20: Confirmación del puerto del Arduino en el Administrador de Dispositivos de Windows.

Fuente: autor, 2023.

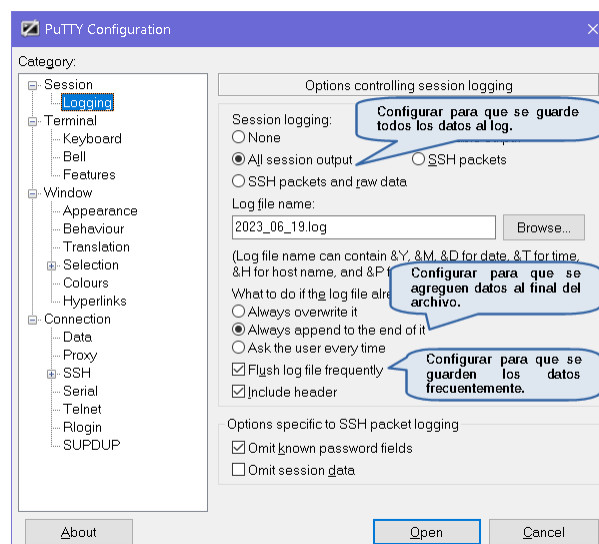


Figura 4.21: Configuración de PuTTY-url, para que este guarde una bitácora de los archivos que llegue a este del Arduino.

Fuente: autor, 2023.

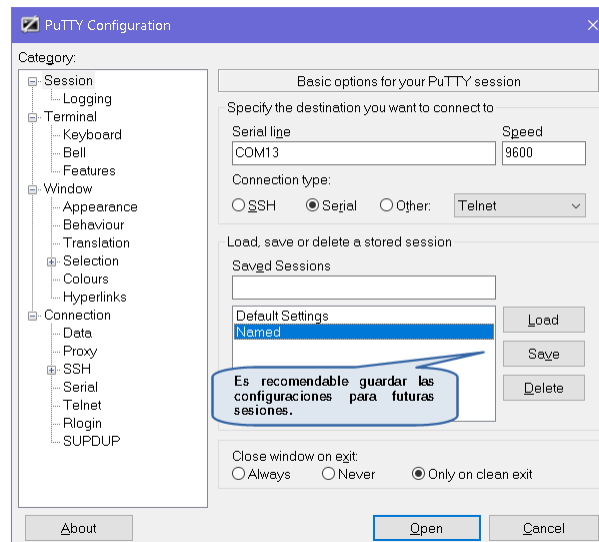


Figura 4.22: Se guarda las configuraciones en PuTTY-url, para futuras sesiones de la recepción de datos.

Fuente: autor, 2023.

en un mapa de la localidad (Fig. 4.24).

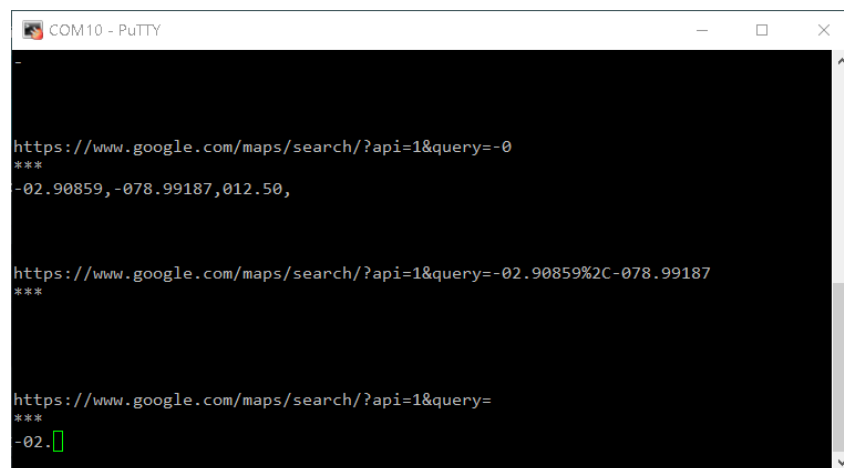


Figura 4.23: Despliegue de texto en la consola de PuTTY-url, que incluye la dirección URL completa, que se puede hacer clic para abrir la página web de con la ubicación en un navegador.

Fuente: autor, 2023.

Esta funcionalidad se considera que podrá brindar un mayor valor agregado al sistema, ya que la operadora guiará remotamente a los agentes que respondan a una emergencia con la información del contexto inmediato del sitio de los hechos. Esto podría incluir dando referencias de locales o direcciones para una mejor movilización a la ubicación del usuario.

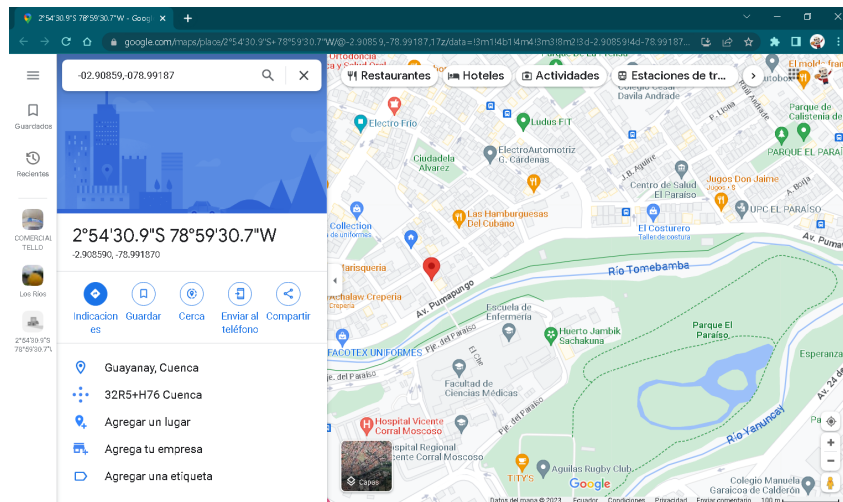


Figura 4.24: Pagina web de la direccion URL, con la ubicación desplegada en Google Maps.

Fuente: adaptada de [45].

#### 4.1.6.2. Obtención del dato de precisión de la ubicación

El dato de ubicación que brinda el servicio de ubicación de Android siempre tiene una precisión limitada. Dicha precisión puede variar según la fuente específica que brinda la ubicación (GPS vs. WiFi/Sensores o una combinación de estas), por las condiciones climatológicas, por la topografía de la ubicación o también por la calidad de los módulos del modelo de teléfono particular.

En un caso de emergencia, es importante conocer la limitación de la precisión, caso contrario los agentes podrían fiarse de la ubicación que reporta el sistema y fallar en responder al sitio correcto de los acontecimientos. Para los casos en que la precisión que reporta el sistema sea baja, puede ser necesario que los agentes emprendan una búsqueda limitada en el sitio aproximado que brindó el sistema. Otra posibilidad sería que la operadora pida al usuario que proporcione una referencia para precisar mejor la ubicación del usuario.

El API del servicio de ubicación de Android ofrece una estimación de la precisión de la ubicación en metros a la redonda. Esta estimación tiene un nivel de confianza del 68° percentil [46]. Una forma de interpretar esto, es que la ubicación verdadera se encontraría dentro de un círculo del radio que reporta el servicio, un 68 % por ciento del tiempo.

La estimación de precisión de la ubicación que reporta el servicio de ubicación del S.O. Android es automáticamente incluido en el objeto Ubicación que devuelve este servicio. Por lo tanto, el incluir este dato entre los datos que transmitirá el sistema hacia los servicios de emergencia es simple (Extracto de Código 4.28). Dada la importancia del dato de precisión para que el sistema informe a los servicios de emergencia de una forma apropiada, se incluye el mismo en el funcionamiento básico del sistema (Fig. 4.25). En las siguientes secciones se presentan otros datos que también podrían complementar el funcionamiento del sistema.

```
class CallActivity : SimpleActivity() {
    ...
    override fun onCreate(savedInstanceState: Bundle?)
    {
        ...
        locationClient
            .getLocationUpdates(300000L)
            .catch { e -> e.printStackTrace() }
            .onEach { location ->
                ...
                accur = ("%05.1f").format(location.accuracy )
                    // SE OBTIENE EL DATO DE LA PRECICION DEL
                    OBJETO DE UBICIACION.

                Todito = " " + lat + "," + long+", "+accur
                    // SE UNE ESTE DATO A LA VARIABLE DE
                    UBICACION.
            }
        }
    }
}
```

Extracto de Código 4.28: Código para la obtención del dato de precisión de la ubicación.

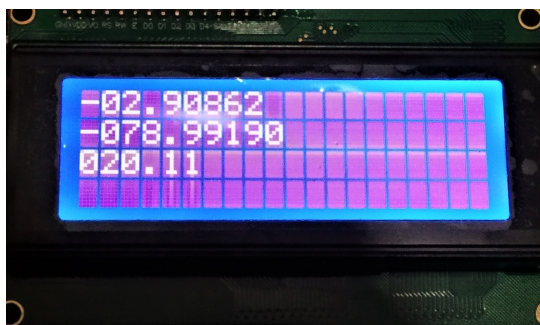


Figura 4.25: Despliegue de la ubicación en la pantalla LCD, incluyendo el dato de precisión.

Fuente: autor, 2023.

#### 4.1.6.3. Obtención del dato de altitud de la ubicación

En algunos casos, el dato de altitud sería importante incluirse entre los que se transmite hacia la operadora de los servicios de emergencia; en particular para ubicaciones dentro de urbes densas con edificios con 10 o más pisos o incluso en ambientes con altos gradientes verticales, como puede ser un cerro o montaña. Esto permitirá ubicar mejor a los afectados de una emergencia, dirigiendo a los agentes que respondan con más precisión y en un menor tiempo.

El dato de altitud se incluye automáticamente en el objeto Ubicación, que entrega el servicio de ubicación del S.O. Android. Por lo tanto, se lo puede incluir entre los datos que transmita el sistema hacia la operadora (Extracto de Código 4.29). Aunque no se incluye entre el funcionamiento básico del sistema, es importante presentar este como una posible mejora al sistema.

```
override fun onCreate(savedInstanceState: Bundle?)
{
    ...
    locationClient
        .getLocationUpdates(300000L)
        .catch { e -> e.printStackTrace() }
        .onEach { location ->
            lat = "%09.5f".format(location.latitude)
            long = "%010.5f".format(location.longitude)
```

```
alt = "%.5f".format(location.altitude ).
    toString() // Se obtiene el dato de altitud
    de la ubicacion.

    Todito = " " + lat + "," + long+", "+altitud
    // Se adjunta la altitud entre los datos
    de la uticacion.
}
    .launchIn(serviceScope)
}
```

Extracto de Código 4.29: Código para la obtención del dato de la altitud de la ubicación.

#### 4.1.6.4. Obtención del dato del nivel de batería del móvil

En la vida cotidiana, circunstancias como que se apague el teléfono móvil por que se haya agotado la batería puede ser un inconveniente. Pero, en una situación de emergencia, que ocurra lo mismo, puede significar la diferencia entre recibir la ayuda que urgentemente se necesita o no. Por ello, es importante que el sistema propuesto tenga la posibilidad de reportar este dato a la operadora de emergencia; para que esta pueda decidir continuar con la llamada o interrumpirla en el tiempo preciso para retomar la comunicación en otro momento.

El Extracto de Código 4.30 muestra la implementación para obtener el porcentaje de batería en el entorno de Android [47]. Una vez obtenido el porcentaje de batería, se puede incluir este, entre los datos que se transmite a la operadora. Una vez más, no se incluyó esto en el funcionamiento básico del sistema, pero fue importante presentar esta posibilidad de funcionalidad. En una futura iteración de la aplicación, se podría pensar en que se transmita este dato, en caso que el nivel de batería este por de debajo de algún umbral crítico.

```
class CallActivity : SimpleActivity() {
    ....
}
```

```
override fun onCreate(savedInstanceState: Bundle?)
{
    ...
    val batteryStatus: Intent? = IntentFilter(Intent.
        ACTION_BATTERY_CHANGED).let { ifilter ->
        applicationContext.registerReceiver(null, ifilter)
    }

    val batteryPct: Float? = batteryStatus?.let { intent
        ->

        val level: Int = intent.getIntExtra(BatteryManager
            .EXTRA_LEVEL, -1)
        val scale: Int = intent.getIntExtra(BatteryManager
            .EXTRA_SCALE, -1)
        level * 100 / scale.toFloat()
    }
}
...
}
```

Extracto de Código 4.30: Código para la obtención del dato del nivel de batería del móvil.

#### 4.1.6.5. Implementación de un dígito de paridad

Como ya se ha expuesto, debido a la delicada naturaleza de enviar correctamente la ubicación a los sistemas de emergencia, a pesar de la alta fidelidad con que rindió el sistema en las pruebas que se ejecutaron, se decidió programar un dígito de paridad para agregar mayor seguridad a la transmisión.

Para aquello, a cada código DTMF utilizado en la codificación, se le asignó un número del 1 al 12, como se observa en la Tabla 4.5. Cada vez que se transmita un código DTMF, se sumará a una variable acumuladora el número correspondiente. Después

de transmitir el último código, se divide el valor almacenado en la variable acumuladora para 2. Si la división tiene residuo (por entenderse que la suma es impar), se adjunta un 1 al final de la secuencia, antes de transmitir el símbolo de fin de la secuencia; caso contrario se adjunta un 0 (por entenderse que la suma es par).

Tabla 4.5: Tabla con números correspondientes para el cálculo del dígito de paridad.

<b>Código</b>	<b>Número</b>
Numerales del 1 al 9	Numerales del 1 al 9
0	10
*	11
#	12

En el lado del receptor, se hace la misma operación y se compara este resultado con el último dígito de la secuencia: si los dos dígitos generados no coinciden, se detectaría que hubo una distorsión en la transmisión de los datos. Esta funcionalidad, al igual que el dato de precisión de la ubicación transmitida, se incluye en el funcionamiento básico del sistema (Fig. 4.26).



Figura 4.26: Despliegue de la ubicación en la pantalla LCD, con el mensaje de 'Error de Paridad'.

Fuente: autor, 2023.

## 4.2. Evaluación experimental

Para investigar la efectividad del sistema propuesto, se planteó un esquema experimental, que permite probar la fidelidad del sistema para transmitir correctamente las coordenadas geográficas, obtenidas del servicio de ubicación del teléfono inteligente. Este esquema se describe a continuación.



#### 4.2.1. Esquema para la evaluación de la fidelidad en la transmisión de las coordenadas

El esquema de evaluación consiste en probar el sistema bajo diferentes condiciones de cobertura de telefonía móvil. Se decidió probar bajo tres franjas de calidad de cobertura: buena, media y mala. Estas franjas se especifican en la Tabla 4.6.

Tabla 4.6: Tabla con las tres franjas de niveles de cobertura para las series de pruebas.

Nivel de cobertura.	Rango de potencia.
Alta	$\geq -85 \text{ dBm}$
Media	$\leq -85 \text{ dBm y } \geq -105 \text{ dBm}$
Baja	$\leq -105 \text{ dBm}$

El procedimiento es el siguiente:

1. Encender el dispositivo receptor/decodificador y conectarlo a un teléfono móvil, así como también a un ordenador en un sitio con buenas condiciones de cobertura ( $\geq -85 \text{ dBm}$ ). Para los fines de estas pruebas, sería necesario configurar el teléfono móvil, para que conteste automáticamente las llamadas entrantes.
2. Ubicarse con el segundo celular, en el cual se haya instalado previamente la aplicación de llamadas desarrollada, en un sitio que cumpla con una de las tres franjas de cobertura mencionadas. Comprobar los niveles de cobertura utilizando la aplicación, **network signal info** [48].
3. Realizar 100 llamadas consecutivas del segundo teléfono al primero, transmitiendo a este último las coordenadas geográficas de donde está situado. El dispositivo electrónico conectado al celular captará y decodificará los tonos DTMF, recuperando así las coordenadas geográficas. Estas coordenadas se transmitirán por la conexión serial al ordenador.
4. Se almacenan tanto las coordenadas transmitidas, como las que fueron recibidas en el ordenador. Así se podrá comparar la coincidencia entre estas, comprobando de este modo la fidelidad del sistema de transmisión.

Este procedimiento se llevó a cabo para los 3 sitios, en donde se encontró las condiciones de cobertura de las 3 franjas seleccionadas.

## 4.2.2. Ejecución de la evaluación de la fidelidad en la transmisión de las coordenadas

A continuación, se detalla cómo se llevaron a cabo las pruebas sobre el sistema.

### 4.2.2.1. Instalación del sitio de recepción en Cuenca

Siguiendo el procedimiento detallado anteriormente, primero se eligió un sitio con buenas condiciones de cobertura. Este sitio fue una oficina en el segundo piso, dentro del Edificio de la Cámara de Construcción, situado en la Ave. Fray Vicente Solano y Nicanor Aguilar, en la ciudad de Cuenca, en la provincia del Azuay del Ecuador (Fig. 4.27). Por medio de la aplicación, **network signal info** [48], se comprobó que hubiese un buen nivel de cobertura en este sitio (Fig. 4.28).



Figura 4.27: Mapa de la ubicación del edificio de la Cámara de la Construcción de Cuenca.

Fuente: adaptada de [45].

En esta oficina, se encendió y se conectó el dispositivo receptor. A su vez, se conectó el módulo MT8870 de este dispositivo al puerto de auriculares del primer celular. Por otro lado, se conectó el Arduino del dispositivo, a un ordenador para transmitir y almacenar las coordenadas geográficas que capte y decodifique el dispositivo (Fig. 4.29). Para ello, fue necesario configurar el primer celular para que este conteste automáticamente las llamadas entrantes (Fig. 4.30).

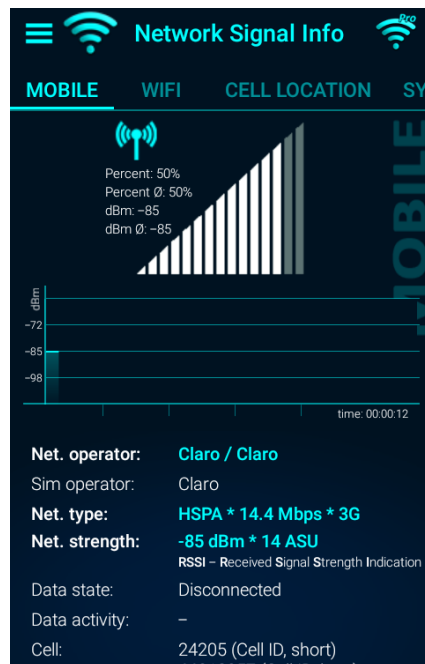


Figura 4.28: Captura de la comprobación de un buen nivel de cobertura en la oficina del edificio de la Cámara de la Construcción de Cuenca.

Fuente: adaptada de [48].

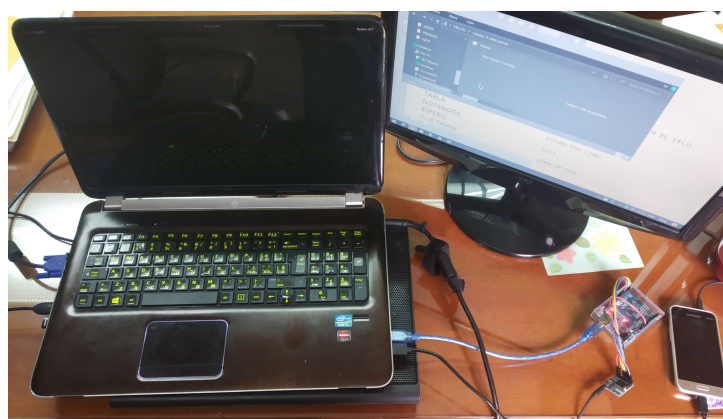


Figura 4.29: Arreglo de los equipos para la recepción de datos para las primeras dos series de pruebas (centro histórico de Cuenca y la parroquia Checa).

Fuente: autor, 2023.

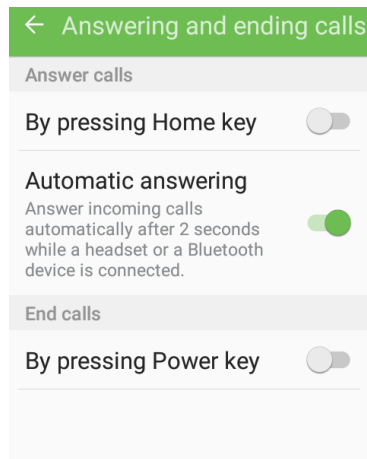


Figura 4.30: Se configura al primer celular (Samsung J1 Prime), para que este conteste automáticamente las llamadas entrantes.

Fuente: autor, 2023.

#### 4.2.2.2. Serie de pruebas con niveles altos de cobertura

Para probar dentro de la primera franja de cobertura, con el segundo celular, en el que previamente se instaló la aplicación de llamadas desarrollado, se ubicó en un sitio con buena cobertura de telefonía móvil ( $\geq -85$  dBm). Este sitio fue en el Parque Calderón en el centro histórico en la ciudad de Cuenca (Fig. 4.31). Nuevamente, con la aplicación, network signal info, se comprobó las condiciones de cobertura en este lugar. En este caso, hubieron excelentes condiciones de cobertura, con un nivel constante de  $-51$  dBm (Fig. 4.32), situado a aproximadamente 10 metros del la antena de la estación base (Fig. 4.33).



Figura 4.31: Mapa del sitio en el Parque Calderón, en el centro histórico de Cuenca, donde se llevó a cabo la primer serie de pruebas.

Fuente: adaptada de [45].

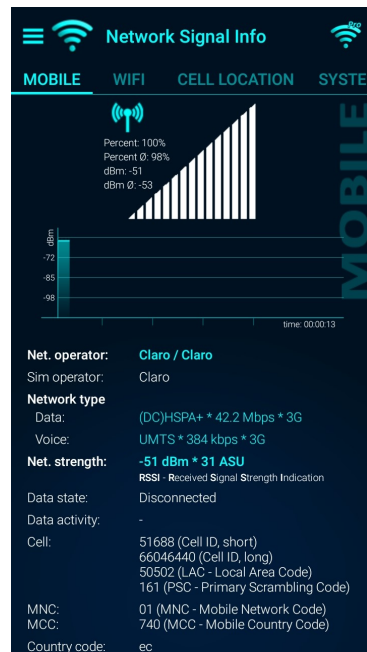


Figura 4.32: Captura de la comprobación del buen nivel de cobertura en el Parque Calderón del centro histórico de Cuenca.

Fuente: adaptada de [48].

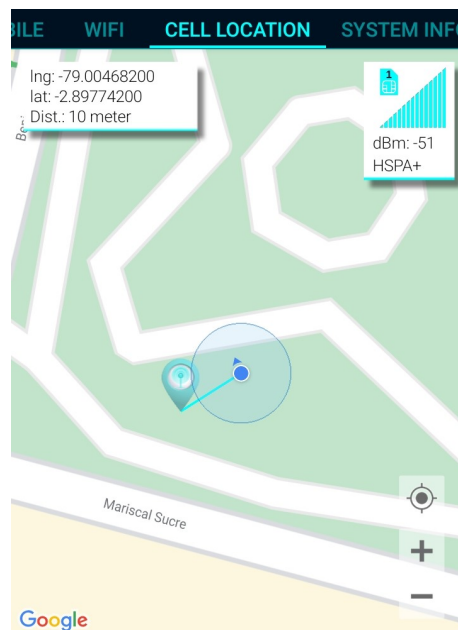


Figura 4.33: Captura de un mapa que demuestra la proximidad entre la antena de la radio base y el sitio donde se llevó a cabo la primera serie de pruebas.

Fuente: adaptada de [48].

A continuación, se realizaron cien llamadas consecutivas desde el segundo celular al primero, utilizando la aplicación desarrollada. Al inicio y fin de la llamada, se realizaron capturas de pantalla en donde se visualizaron las coordenadas que serían transmitidas (Fig. 4.34). La duración de cada llamada fue planificada para un mínimo de 35 segundos aproximadamente. Al colgar la llamada, se esperó unos segundos y se repetía el procedimiento.

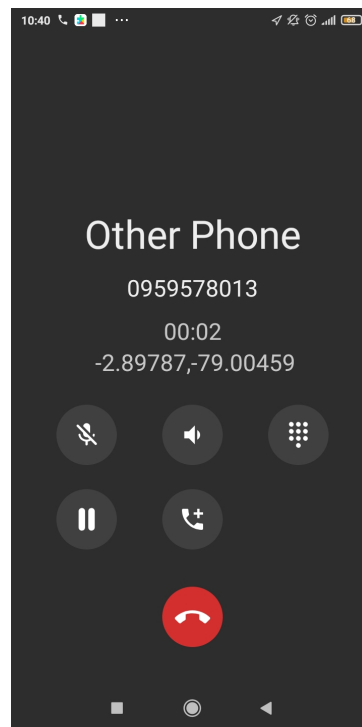


Figura 4.34: Captura del interfaz de llamadas de la aplicación, donde se despliega las coordenadas que serán transmitidas hacia el dispositivo receptor.  
Fuente: autor, 2023.

#### 4.2.2.3. Serie de pruebas con niveles medios de cobertura

Para probar la fidelidad del sistema bajo condiciones de cobertura media ( $\leq -85$  y  $\geq -105$  dBm), se movilizó a Checa, parroquia rural del Cantón Cuenca, provincia de Azuay (Fig. 4.35). Se acudió a esta parroquia, debido a que no se encontró un sitio en el área urbana de Cuenca, que mantenga consistentemente los niveles de cobertura requeridos. Nuevamente, con la aplicación, network signal info, se comprobó durante 4 minutos que el sitio elegido mantenía niveles de cobertura entre los límites de esta franja (Fig. 4.36). De ahí se procedió de forma similar, a realizar un mínimo de 100 llamadas consecutivas al primer celular, que previamente se encontraba conectado al

dispositivo receptor y este a su vez al ordenador. Así mismo, se realizó capturas de las coordenadas transmitidas para ser comparadas con las recibidas por el dispositivo.

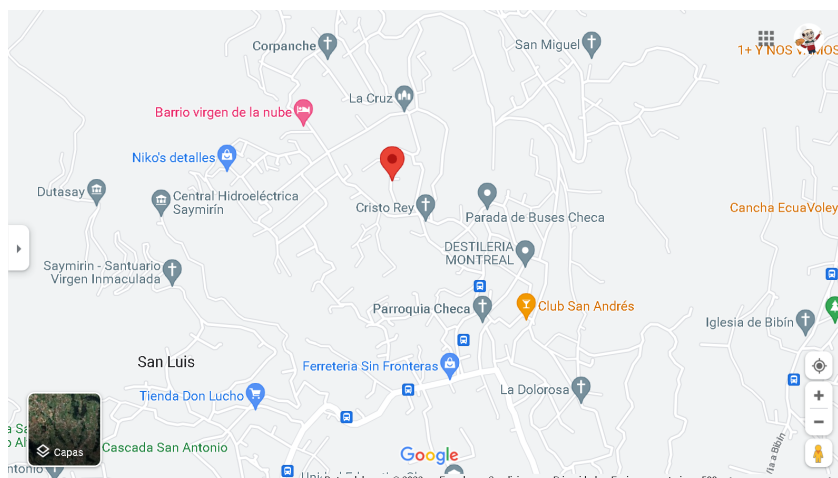


Figura 4.35: Mapa del sitio en la Parroquia Checa, donde se llevo a cabo la segunda serie de pruebas del sistema.  
Fuente: adaptada de [45].

#### 4.2.2.4. Instalación del sitio de recepción en Ambato

Finalmente, se llevaron a cabo las pruebas del funcionamiento del sistema propuesto, con bajos niveles de cobertura ( $\leq 105$  dBm), en la ciudad de Ambato, en la provincia de Tungurahua. Se situó el receptor en una vivienda ubicada en la calle Río Talantag entre la Ave. Cervantes y Yanayacu (Fig. 4.37). Siguiendo el mismo procedimiento, se encendió el dispositivo y se conectó este al primer celular y al ordenador; comprobándose primero que haya buenos niveles de cobertura para la recepción de las llamadas (Fig. 4.38).

#### 4.2.2.5. Serie de pruebas con niveles bajos de cobertura

El sitio elegido, con los niveles de cobertura necesarios, se ubicó en el kilómetro 2 de la Vía a Aguaján, en la ciudad de Ambato, provincia de Tungurahua (Fig. 4.39). Nuevamente, se comprobó durante 4 minutos que en este sitio se mantenían niveles de cobertura por debajo del umbral requerido (Fig. 4.40). Posteriormente, se realizaron las cien llamadas consecutivas hacia el primer celular, tomando capturas de lo transmitido. Cabe señalar que durante esta serie de pruebas, por unos pocos minutos no

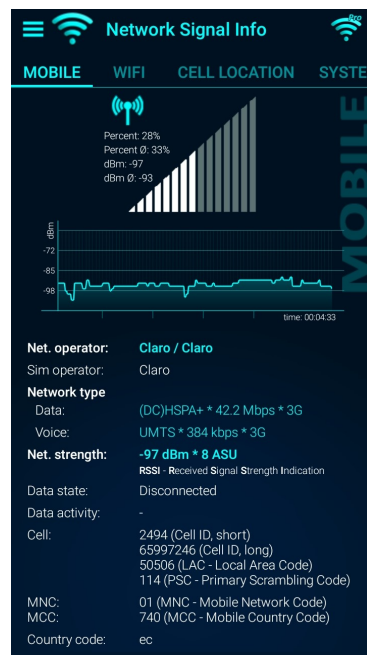


Figura 4.36: Captura de la comprobación de las condiciones medias de cobertura en la Parroquia Checa.

Fuente: adaptada de [48].

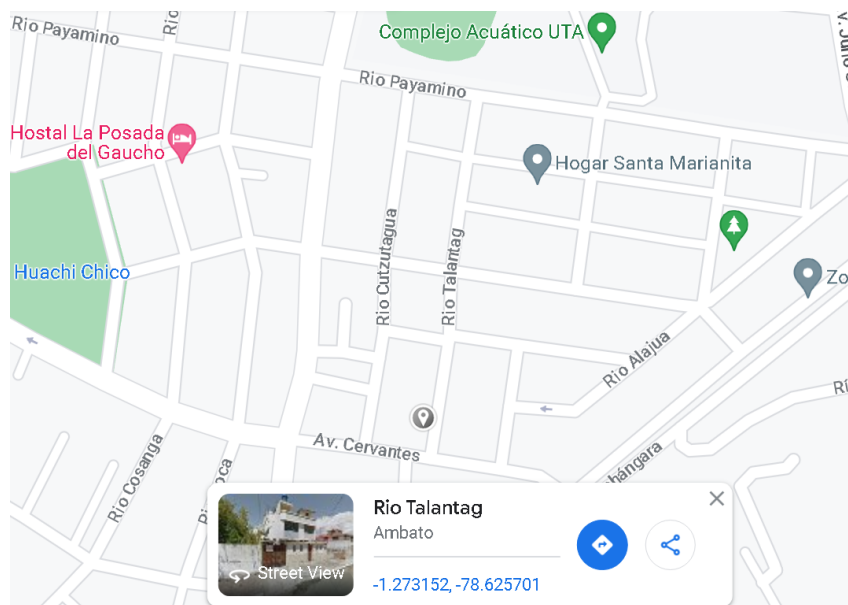


Figura 4.37: Mapa del domicilio en Ambato, donde se situó al receptor para la tercera serie de pruebas del sistema.

Fuente: adaptada de [45].





Figura 4.38: Captura de la comprobación de los buenos niveles de cobertura en el domicilio en Ambato, para la recepción de llamadas, para la tercera serie de pruebas del sistema.

Fuente: adaptada de [48].

se conectaban las llamadas desde el celular número 2, debido justamente a los bajos niveles de cobertura.

Las fechas en que se llevaron a cabo las diferentes series de pruebas se resumen en la Tabla 4.7. Todas las pruebas se llevaron a cabo en días laborables, durante horarios laborables, para probar bajo condiciones de tráfico de red nominales.

Tabla 4.7: Tabla de los sitios y fechas de las 3 series de pruebas del sistema.

Sitio del experimento.	Fecha en el que se llevó a cabo.
Parque Calderón, cantón Cuenca	21 de abril de 2023
Parroquia Rural Checa, cantón Cuenca	28 de abril de 2023
Km. 3 de la Vía a Aguaján, ciudad de Ambato	2 de mayo de 2023

Los resultados de las anteriores series de pruebas ser presentan en el siguiente capítulo.



Figura 4.39: Mapa del sitio en el km. 2 de la Vía a Aguaján, de donde se llevó a cabo las llamadas para la tercera serie de pruebas del sistema.  
Fuente: adaptada de [45].

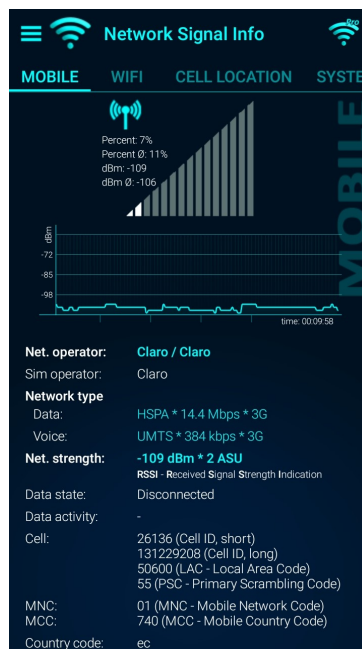


Figura 4.40: Captura de la comprobación de los bajos niveles de cobertura en el sitio en el km. 3 de la Vía a Aguaján, de donde se llevó a cabo las llamadas para la tercera serie de pruebas del sistema.  
Fuente: adaptada de [48].

## 5. Resultados

En las siguientes subsecciones se presentan gráficas de los resultados numéricos de las tres series de pruebas realizadas que corresponden a cada una de las tres franjas de nivel de cobertura (Tabla 4.6).

### 5.1. Resultados - Pruebas en el cantón Cuenca, provincia del Azuay

En la Figura 5.1, se presentan los resultados de la serie de pruebas llevadas a cabo en el centro histórico del cantón Cuenca, Provincia del Azuay. Como se puede observar, se tuvo más del 99 % de transmisiones acertadas, entre las 201 transmisiones dadas.

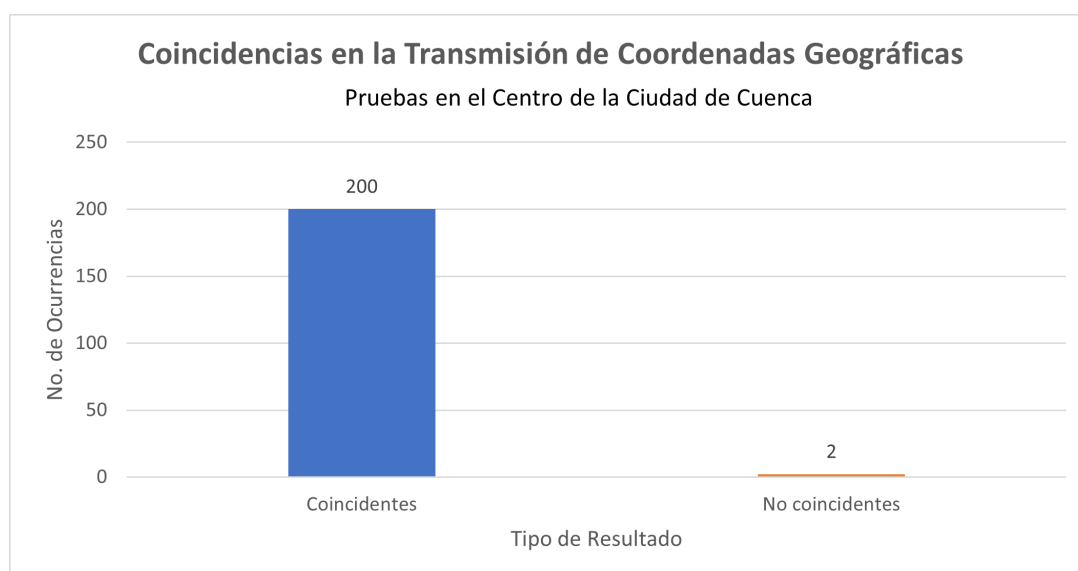


Figura 5.1: Resultados numéricos de la serie de pruebas realizadas en el centro histórico de la ciudad de Cuenca, provincia del Azuay.

Fuente: autor, 2023.

### 5.2. Resultados - Pruebas en la parroquia rural Checa, cantón Cuenca

En la Figura 5.2, se presentan los resultados de la serie de pruebas llevadas a cabo en la parroquia rural Checa, perteneciente al cantón Cuenca, Provincia del Azuay. Como se aprecia en la figura, de las 210 transmisiones dadas, más del 99 % de estas fueron exitosas.

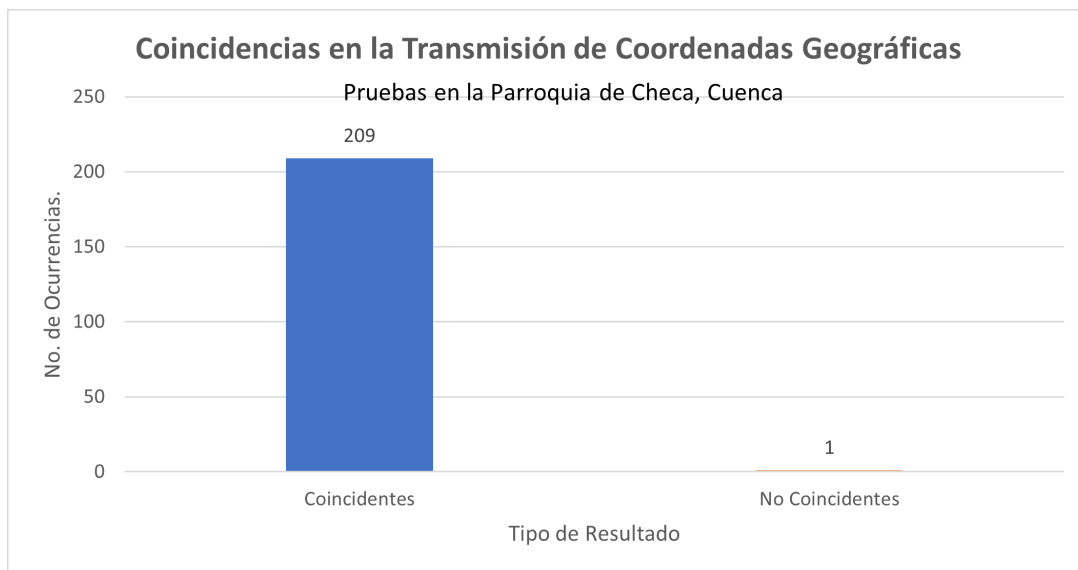


Figura 5.2: Resultados numéricos de la serie de pruebas realizadas en la parroquia Checa, perteneciente al cantón Cuenca, provincia del Azuay.  
Fuente: autor, 2023.

### 5.3. Resultados - Pruebas en el km. 2 de la Vía a Aguaján, cantón Ambato, provincia de Tungurahua

En la Figura 5.3, se presentan los resultados de la serie de pruebas llevadas a cabo en el km. 2 de la Vía a Aguaján, del cantón Ambato, provincia de Tungurahua. Como se puede apreciar en esta figura, a pesar de las bajas condiciones de cobertura, se logró tener un 98 % de transmisiones exitosas de las 253 transmisiones que se realizaron.

### 5.4. Resumen de los resultados

Los porcentajes de fidelidad en la transmisión de coordenadas y los intervalos de confianza del 95 %, basados en la fórmula ajustada de Wald, descrita en la Sección 2.5, que se obtuvieron en las 3 series de experimentos, se resumen en la Tabla 5.1.

Tabla 5.1: Tabla de resultados porcentuales.

Serie de experimentos	Porc. de fidelidad	Inter. de confianza del 95 %
Centro histórico de Cuenca	99.01 %	±1.867 %
Parroquia rural Checa, Cuenca	99.52 %	±1.556 %
Km. 2, Vía a Aguaján, Ambato	98.4 %	±1.835 %

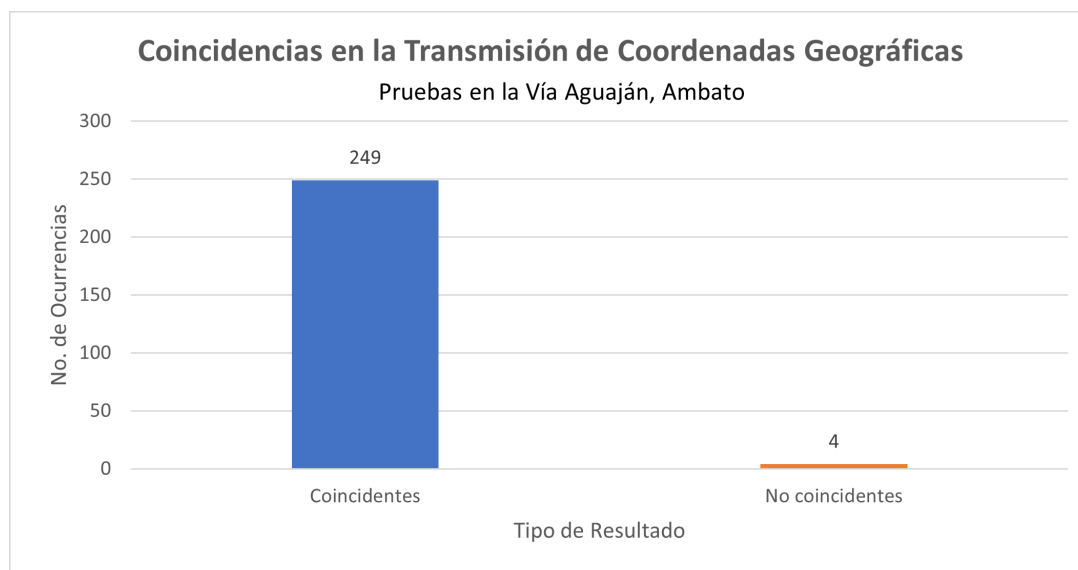


Figura 5.3: Resultados numéricos de la serie de pruebas realizadas en el km. 2 de la Vía a Aguaján, perteneciente al cantón Ambato, provincia de Tungurahua.

Fuente: autor, 2023.

Como se puede observar, el sistema obtuvo altos grados de fidelidad en la transmisión de las coordenadas entre el teléfono celular y el equipo receptor. Ninguno de los promedios de aciertos estuvieron por debajo del 95 %, incluso tomando en cuenta los límites inferiores de los intervalos de confianza.

Estos resultados sugieren que se podría confiar en la transmisión de la ubicación de un usuario con este sistema, incluso bajo circunstancias críticas, cuando los usuarios que acuden a los servicios de emergencia esperan un servicio urgente y oportuno. La subsecuente implementación del dígito de paridad, descrito en la Subsubsección 4.1.6.5, se espera brindará aún mayor seguridad y confianza en la transmisión de coordenadas durante los momentos críticos de una emergencia.

## 6. Conclusiones y recomendaciones

### 6.1. Conclusiones

Con respecto de los objetivos que se planteó al inicio de este proyecto, se puede considerar lo siguiente:

- Efectivamente, se desarrolló la aplicación para el S.O. Android. Se comprobó su funcionamiento exitoso en un modelo de celular; este funcionamiento fue estable, sin percances en su ejecución y además funcionó precisamente como se esperaba.
- Se desarrolló el dispositivo receptor, con pocos implementos, con un diseño simple y relativamente económico. Para su plena integración en los equipos del SIS ECU 911, fuera necesario ciertos adaptadores, para que se acople a los teléfonos que ahí se ocupan.
- El protocolo que se desarrolló fue directo y poco complicado: aparte de los dígitos numéricos que se representaron directamente con los tonos DTMF correspondientes, únicamente fue necesario representar los caracteres que forman parte de las coordenadas geográficas con los dos símbolos restantes del teclado numérico (el # y el \*). Utilizando un par de estos símbolos fue suficiente para representar los caracteres restantes.
- En términos de la integración de estos elementos: la aplicación, el código y el dispositivo electrónico, su rendimiento fue mejor de lo que se esperaba. Se tuvo que corregir el protocolo de transmisión de las coordenadas, al incluir el símbolo de **fin de secuencia** al inicio de la transmisión, para que esta acción borre los datos de la transmisión anterior.
- El rendimiento que presentó el sistema en las series de pruebas que se desarrollaron, de igual manera, fue exitoso. La fidelidad del sistema fue cerca del 100 % en algunos casos, e incluso tuvo alta fidelidad para niveles bajos de cobertura.
- Además de los objetivos anteriores, se diseñó e implementó varias mejoras para suplementar el funcionamiento del sistema. Las que se consideró más pertinentes, se incluyeron entre las capacidades básicas del sistema.

Tomando en consideración lo anterior, se podría decir que se cumplió plenamente con todos de los objetivos específicos planteados. Y en su conjunto, estos aportaron para cumplir con el objetivo general: el diseño e implementación de un sistema para la transmisión de las coordenadas geográficas desde el teléfono móvil de un usuario hacia las operadoras de los servicios de emergencia, con la finalidad de brindar una respuesta pronta y oportuna ante cualquier emergencia.

## 6.2. Recomendaciones

Para considerar que esta tecnología pueda ser aplicada en uno u otro ámbito, aparte de su rendimiento técnico, se deberá considerar los costos que implica la misma. Para ello, presentamos en la Tabla 6.1 los costos de los materiales que se utilizaron en la implementación del dispositivo receptor. Los gastos de la aplicación son nulos, ya que el código en el que se basó su desarrollo es abierto y sin costo.

Tabla 6.1: Lista de materiales y costos.

No.	Componente	Precio/Uni. (USD)	Cantidad	Subtotal (USD)
1	Arduino Uno	12.49	1	12.49
2	Módulo MT8870	3.59	1	3.59
3	Cables Jumper 20 cm,	0.4	11	0.44
4	Cable de audio, 3.5 mm,	2.00	1	2.00
5	Pantalla LCD, 20x4,	10.79	1	10.79
6	Adaptador divisor RJ-9	9.00	1	9.00
7	Adaptador, RJ-9 a 2.5 mm	35.00	1	35.00
8	Adaptador, 2.5 mm a 3.5 mm	1.50	1	1.50
<b>Total</b>				<b>74.81</b>

Como ejemplo, se puede considerar la implementación de esta tecnología en las instalaciones de la Coordinación Zonal 6 del SIS ECU 911; posiblemente como un proyecto piloto. Hasta el año 2017, esta Coordinación Zonal contaba con 19 consolas para el área que recibe las llamadas de los usuarios [49]. Extrapolando el valor total de la Tabla 6.1, resultaría en un costo total de 1421.39 USD para equipar a todas las consolas de esta coordinación. Se considera esto un precio razonable, dado la importancia de preservar la vida humana.

Para lograr una implementación e integración exitosa de este sistema, sería necesario

primero llevar a cabo las siguientes actividades:

- Llevar a cabo pruebas más extensas en conjunto con la Coordinación Zonal 6.
- Instalar y probar el funcionamiento de la aplicación en varios modelos de teléfonos Android.
- Expandir el desarrollo de la aplicación para múltiples S.O.s de teléfonos móviles (iOS y Android), utilizando plataformas de desarrollo, tal como Flutter y React.
- Publicar las aplicaciones en las respectivas tiendas de aplicaciones (Play Store y App Store), para facilitar el acceso.
- Continuar el desarrollo a medio y largo plazo, respondiendo a las inquietudes y percances que reporten los usuarios de la aplicación.

### 6.3. Trabajos futuros

Para expandir sobre lo desarrollado en este proyecto de tesis, entre los posibles trabajos futuros que se podrían planificar, estarían los siguientes:

- Crear una versión de esta aplicación para un modo de pruebas, que automatice lo que en este trabajo se hizo manualmente.
- Llevar a cabo estudios para verificar la precisión del sistema en diferentes zonas del país y dentro de varias geografías (p. ej.: en el Parque Nacional Cajas o en las regiones costa y Amazonía). Con estos datos, se podría prever las diferencias de como el sistema respondería bajo estas circunstancias y tomar esto en cuenta para mejor responder a las emergencias.
- Generar una aplicación de botón de pánico basado en la transmisión de la ubicación, utilizando los tonos DTMF. Esto serviría para casos particulares de emergencia, como sería los casos de violencia doméstica o en caso de un rapto o secuestro.
- Investigar los casos de uso que meriten incluir datos adicionales en la transmisión, tal como la altitud, posiblemente para ciertas actividades de turismo.



- Investigar e implementar un algoritmo que permita comunicar el piso aproximado en que se encuentre un usuario dentro de una edificación, basándose en la altitud reportada con la ubicación, comparando esta con la altitud de la intersección de calles en donde se encuentre el edificio. Esto permitiría a ambulancias y bomberos responder como más agilidad a las emergencias que se lleven a cabo en urbes densas, con estructuras de 10 o más pisos.

Se considera que estos trabajos posibilitarán nuevas aplicaciones para esta tecnología y enriquecerán el contenido técnico que forma parte de la misma.

## Referencias

- [1] J. V. Sickle, "Trilateration | geog 862: Gps and gnss for geospatial professionals," 2022. [En línea]. Disponible: <https://www.e-education.psu.edu/geog862/node/1731>
- [2] "(63) adventures in science: How gps works - youtube," 2017. [En línea]. Disponible: [https://www.youtube.com/watch?v=6m0xGwkYYy0&ab\\_channel=SparkFunElectronics](https://www.youtube.com/watch?v=6m0xGwkYYy0&ab_channel=SparkFunElectronics)
- [3] J. V. Sickle, "One-way ranging | geog 862: Gps and gnss for geospatial professionals," 2022. [En línea]. Disponible: <https://www.e-education.psu.edu/geog862/node/1733>
- [4] "Gps.gov: Space segment," 2022. [En línea]. Disponible: <https://www.gps.gov/systems/gps/space/>
- [5] "Gps.gov: Control segment," 2021. [En línea]. Disponible: <https://www.gps.gov/systems/gps/control/>
- [6] "The keypad of an old fashioned wireless phone stock photo - alamy," 2013. [En línea]. Disponible: <https://www.alamy.com/the-keypad-of-an-old-fashioned-wireless-phone-image61977569.html>
- [7] "uğur'un bloğu: Robokar," 2012. [En línea]. Disponible: [http://uguraykar.blogspot.com/p/blog-page\\_9.html](http://uguraykar.blogspot.com/p/blog-page_9.html)
- [8] "Qué es un central telefónica pbx •• beneficios i 3cx." [En línea]. Disponible: <https://www.3cx.es/voip-sip/central-telefonica-pbx/>
- [9] G. Aguirre, "Github - galoaguirre-ucuenca/eculocalizador911: Eculocalizador911 desarrollo de un sistema para la transmisión de la ubicación de un teléfono inteligente hacia los servicios de emergencia por medio de tonos dtmf. trabajo de titulación previo a la obtención del título de ingeniero en electrónica y telecomunicaciones." 2023. [En línea]. Disponible: <https://github.com/galoAguirre-UCuenca/EcuLocalizador911>

- [10] J. V. Sickle, “Introduction | geog 862: Gps and gnss for geospatial professionals,” 2022. [En línea]. Disponible: <https://www.e-education.psu.edu/geog862/node/1884>
- [11] “Población estadística - qué es, definición y concepto | 2023 | economipedia,” 2019. [En línea]. Disponible: <https://economipedia.com/definiciones/poblacion-estadistica.html>
- [12] J. Sauro, *Quantifying the user experience : practical statistics for user research*. Morgan Kaufmann, 2016.
- [13] U. Sedlar, J. Winterbottom, B. Tavcar, J. Sterle, J. Cijan, M. Volk, y M. Casoni, “Next generation emergency services based on the pan-european mobile emergency application (pemea) protocol: Leveraging mobile positioning and context information,” *Wireless Communications and Mobile Computing*, vol. 2019, 2019.
- [14] L. Wu, H. Feng, Z. Zhang, y Y. Shi, “Research on the implementation method of am medium-wave emergency broadcasting based on dtmf signaling insertion,” *Communications in Computer and Information Science*, vol. 163 CCIS, pp. 448–453, 2011. [En línea]. Disponible: [https://link.springer.com/chapter/10.1007/978-3-642-25002-6\\_63](https://link.springer.com/chapter/10.1007/978-3-642-25002-6_63)
- [15] M. Weinlich, P. Kurz, M. B. Blau, F. Walcher, y S. Piatek, “Significant acceleration of emergency response using smartphone geolocation data and a worldwide emergency call support system,” *PLoS ONE*, vol. 13, 6 2018.
- [16] G. Al-Nuaimi, M. Lloyd, H. Kachali, S. U. Rehman, y P. Gardner-Stephen, “Fixing mobile emergency call geo-location once and for all.” *IEEE*, 6 2021, pp. 62–67. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/9612463/>
- [17] D. Corral-De-Witt, E. V. Carrera, J. A. Matamoros-Vargas, S. Munoz-Romero, J. L. Rojo-Álvarez, y K. Tepe, “From e-911 to ng-911: Overview and challenges in ecuador,” *IEEE Access*, vol. 6, pp. 42 578–42 591, 6 2018.
- [18] “Gps.gov: Gps accuracy,” 2022. [En línea]. Disponible: <https://www.gps.gov/systems/gps/performance/accuracy/>

- [19] “Seamless and smooth location everywhere with the new fusedlocationprovider (google i/o’19) - youtube,” 2019. [En línea]. Disponible: [https://www.youtube.com/watch?v=MEjFW\\_tLrFQ](https://www.youtube.com/watch?v=MEjFW_tLrFQ)
- [20] “Wi-fi location tracking & positioning systems | inpixon,” 2020. [En línea]. Disponible: <https://www.inpixon.com/technology/standards/wifi>
- [21] I. Kumarage, “Real-time location tracking using google’s fused location provider api | by imal kumarage | the startup | medium,” 2020. [En línea]. Disponible: <https://medium.com/swlh/real-time-location-tracking-using-googles-fused-location-provider-api-2fc8023f2c1c>
- [22] “Mt8870 dtmf voice decoding module speech audio decoder voice board,” 2019. [En línea]. Disponible: <https://zemfiraaksenova.blogspot.com/2019/08/mt8870-dtmf-voice-decoding-module.html>
- [23] “Itu-t recommendation q.23: Technical features of push-button telephone sets,” 1988. [En línea]. Disponible: <https://www.itu.int/rec/T-REC-Q.23/en>
- [24] “Mt8870 dtmf decoder interfacing with arduino uno | arduino,” 2023. [En línea]. Disponible: <https://www.electronicwings.com/arduino/mt8870-dtmf-decoder-interfacing-with-arduino-uno>
- [25] “Sensors modules mt8870 dtmf decoder | sensors modules,” 2023. [En línea]. Disponible: <https://www.electronicwings.com/sensors-modules/mt8870-dtmf-decoder>
- [26] A. Rahman, “Arduino lcd i2c - tutorial with arduino uno - youtube,” 2020. [En línea]. Disponible: <https://www.youtube.com/watch?v=EAeuxjtkumM>
- [27] “Oip.tpxoyitqa9mx2qznnwomvghagz (474x409),” S.f. [En línea]. Disponible: <https://www.pinterest.fr/pin/419538521532565190/>
- [28] “Github - simplemobiletools/simple-dialer: A handy phone call manager with phonebook, number blocking and multi-sim support,” 2023. [En línea]. Disponible: <https://github.com/SimpleMobileTools/Simple-Dialer>

- [29] E. Canorea, “¿qué es kotlin y para qué sirve? - plain concepts,” 2022. [En línea]. Disponible: <https://www.plainconcepts.com/es/kotlin-android/>
- [30] “bed7bb1bbbc4c0bfa47e06b15badef7b53cfd7e9ce8c2941e6bf8c6390e64552:800 (450×800),” S.f. [En línea]. Disponible: <https://img.utdstc.com/screen/bed/7bb/bed7bb1bbbc4c0bfa47e06b15badef7b53cfd7e9ce8c2941e6bf8c6390e64552:800>
- [31] “Log | android developers,” 2023. [En línea]. Disponible: <https://developer.android.com/reference/kotlin/android/util/Log>
- [32] P. Lackner, “How to track your users location in the background in android - android studio tutorial - youtube,” 2022. [En línea]. Disponible: [https://www.youtube.com/watch?v=Jj14sw4Yxk0&ab\\_channel=PhilippLackner](https://www.youtube.com/watch?v=Jj14sw4Yxk0&ab_channel=PhilippLackner)
- [33] “Interfaces en programación orientada a objetos,” S.f. [En línea]. Disponible: <https://informaticapc.com/poo/interfaces.php>
- [34] “Flujos de kotlin en android | android developers,” 2023. [En línea]. Disponible: <https://developer.android.com/kotlin/flow?hl=es-419>
- [35] “Context | android developers,” 2023. [En línea]. Disponible: <https://developer.android.com/reference/kotlin/android/content/Context>
- [36] “Fused location provider api | google for developers,” S.f. [En línea]. Disponible: <https://developers.google.com/location-context/fused-location-provider?hl=es-419>
- [37] A. Fox, “Precision matters: The critical importance of decimal places | blis,” 2022. [En línea]. Disponible: <https://blis.com/precision-matters-critical-importance-decimal-places-five-lowest-go/>
- [38] G. Aguirre, “2023 03 17 23 25 35 - youtube,” 2023. [En línea]. Disponible: <https://www.youtube.com/watch?v=L6WxDutwHKU>
- [39] “Mi chile,” S.f. [En línea]. Disponible: <https://www.mi.com/cl/redmi-s2/specs>
- [40] Y. Fernández, “Gps, glonass, beidou y galileo: qué son y cuáles son las diferencias,” 2023. [En línea]. Disponible: <https://www.xataka.com/basics/gps-glonass-beidou-galileo-que-cuales-diferencias>

- [41] “Get started | maps urls | google developers,” 2023. [En línea]. Disponible: <https://developers.google.com/maps/documentation/urls/get-started>
- [42] “Código uri - ecyt-ar,” 2011. [En línea]. Disponible: [https://cyt-ar.com.ar/cyt-ar/index.php/C%C3%B3digo\\_URI](https://cyt-ar.com.ar/cyt-ar/index.php/C%C3%B3digo_URI)
- [43] M. A. D. Gómez, “Arduino a muete: Utilizando putty,” 2015. [En línea]. Disponible: <http://arduinoamuete.blogspot.com/2015/12/utilizando-putty.html>
- [44] D. Tengu, “Clickable urls in putty | spaceholder,” 2011. [En línea]. Disponible: <https://www.daitengu.com/2011/09/11/clickable-urls-in-putty/>
- [45] “Google maps,” 2023. [En línea]. Disponible: <https://www.google.com/maps/>
- [46] “Location | android developers,” 2023. [En línea]. Disponible: [https://developer.android.com/reference/kotlin/android/location/Location#getAccuracy\(\)](https://developer.android.com/reference/kotlin/android/location/Location#getAccuracy())
- [47] W. Memom, “How to get battery state/level (percentage) in android ? androprogrammer,” S.f. [En línea]. Disponible: <https://www.androprogrammer.com/2013/12/how-to-get-battery-statelevel.html>
- [48] “Network signal info - apps on google play,” 2023. [En línea]. Disponible: <https://play.google.com/store/apps/details?id=de.android.telnet>
- [49] (2017) Rendicion de cuentas 2017 coordinacion zonal 6 servicio integrado de seguridad ecu 911 contenidos. [En línea]. Disponible: <https://www.ecu911.gob.ec/wp-content/uploads/2018/03/Presentaci%C3%B3n-Rendici%C3%B3n-de-Cuentas-Zona-6.pdf>