

Sistema de seguridad para bicicletas

Jorge Edison Jiménez Leguizamón

Universitaria Agustiniana
Facultad de Ingeniería
Programa de Ingeniería en Telecomunicaciones
Bogotá, D.C.
2020

Sistema de seguridad para bicicletas

Jorge Edison Jiménez Leguizamón

Director

Edgar Fabian Rodríguez

Trabajo de grado para optar al título de Ingeniero en Telecomunicaciones

Universitaria Agustiniana

Facultad de Ingeniería

Programa de Ingeniería en Telecomunicaciones

Bogotá, D.C.

2020

Resumen

En la ciudad de Bogotá, Colombia existen un alto índice de hurto a bicicletas, el cual se ha incrementado aún más el presente año, tanto así, que el diario británico The Guardian señala a Bogotá, “capital de la muerte” para los ciclistas por tal razón, en el presente trabajo se desarrolla la implementación de un sistema de geolocalización para disminuir el hurto de bicicletas en la ciudad de Bogotá, mediante la creación de una aplicación móvil para dispositivos Android, la cual le permita al bici-usuario encontrar o monitorear la bicicleta en caso de un robo y el bici-usuario pueda recuperar su bicicleta. Así mismo, en el presente documento se investiga y se indican cuales lugares con mayor índice de hurto de bicicleta, con la intención que el bici-usuario si por algún motivo deba transitar por estas zonas se encuentre alerta ante un posible hurto. Por otra se detalla la manera en la cual se debe configurar el Gps Localizador Tracker Tk102, el cual se utiliza en el sistema de geolocalización del presente proyecto de grado, así como también, la programación y el diseño de la aplicación móvil.

Palabras claves: GPS, GPRS, Android, PHP, hosting, Java.

Abstract

In the city of Bogota, Colombia, there is a high rate of bicycle theft, which has increased even more this year, so much so that the British newspaper The Guardian points to Bogota, "capital of death" for cyclists for that reason, This work develops the implementation of a geolocation system to reduce bicycle theft in the city of Bogota, by creating a mobile application for Android devices, which allows the bike-user to find or monitor the bike in case of a theft and the bike-user can recover his bike. Likewise, this document investigates and indicates the places with the highest rate of bicycle theft, with the intention that the bicycle-user if for some reason must transit through these areas is alert to a possible theft. On the other hand, it details the way in which the Gps Localizer Tracker Tk102 should be configured, which is used in the geolocation system of the present degree project, as well as the programming and design of the mobile application.

Keywords: GPS, GPRS, Android, PHP, hosting, Java.

Tabla de contenido

Introducción	10
Problemática.....	11
Justificación.....	12
Objetivos	13
Objetivo general	13
Objetivos específicos.....	13
Marco de referencia.....	14
Estado del arte	14
Marco teórico.....	16
Marco legal.....	17
Metodología	18
Cronograma.....	19
Actividades.....	19
Presupuesto.....	21
Desarrollo	22
Adquisición del GPS	22
Topología de red del sistema de geolocalización	28
Investigación de lugares inseguras de la ciudad de Bogotá.....	29
Creación de base de datos.....	30
Diseño de la aplicación móvil	31
Lógica de la aplicación móvil.....	33
Diagramas de flujo de información	38
Programación de activities.....	45
Botones menú principal	49
Funcionamiento del sistema de geolocalización	56
Tiempos de retardo del sistema de geolocalización	64

Comportamiento de la batería	67
Conclusiones	68
Recomendaciones.....	69
Referencias	70

Lista de figuras

Figura 1.	Configuración GPS 1	26
Figura 2.	Configuración GPS 2	26
Figura 3.	Configuración GPS 3	27
Figura 4.	Configuración GPS 4	27
Figura 5.	GPS Localizador Tracker Tk102.....	28
Figura 6.	Diagrama de flujo del sistema de geolocalización	28
Figura 7.	Hurto de bicicletas en Bogotá	29
Figura 8.	Tabla de registro	30
Figura 9.	Tabla de ubicación	31
Figura 10.	Diagrama de flujo activity main y registro.....	33
Figura 11.	Diagrama de flujo activity sincronizar GPS.....	34
Figura 12.	Diagrama activity sitios inseguros	34
Figura 13.	Diagrama de flujo activity ubicación	35
Figura 14.	Diagrama de flujo activity emergencia	36
Figura 15.	Diagrama activity configuración.....	37
Figura 16.	Diagrama de flujo de datos Activity Main.....	38
Figura 17.	Diagrama de flujo de datos Activity Registro.....	39
Figura 18.	Diagrama de flujo de datos Activity MenUser.....	40
Figura 19.	Diagrama de flujo de datos Activity Registro.....	41
Figura 20.	Diagrama de flujo de datos Activity Maps.....	42
Figura 21.	Diagrama de flujo de datos Activity Bicicleta robada	43
Figura 22.	Diagrama de flujo de datos Activity Ubicación	44
Figura 23.	Diagrama de flujo de datos Activity configuración	45
Figura 24.	Funcionamiento librería Volley	46
Figura 25.	ProgressDialog	47
Figura 26.	Comprobación campos vacíos.....	48
Figura 27.	Activity Google Maps.java	51
Figura 28.	Activity Google Maps	52
Figura 29.	Repuesta GPS.....	52
Figura 30.	Logcat.....	53

Figura 31.	Activity Emergencia.....	53
Figura 32.	Confirmación de monitoreo	54
Figura 33.	Monitoreo 1	54
Figura 34.	Monitoreo 2	54
Figura 35.	Monitoreo 3	55
Figura 36.	Activity Configuración.....	55
Figura 37.	Permisos activos.....	56
Figura 38.	Vista del GPS en la bicicleta	57
Figura 39.	Registro de nuevo usuario	57
Figura 40.	Visualización del registro de usuario	58
Figura 41.	Inicio de sesión.....	58
Figura 42.	Menú principal.	59
Figura 43.	Opción ubicación.....	59
Figura 44.	Ubicación de la bicicleta	60
Figura 45.	Verificación de envío de ubicación.....	60
Figura 46.	Verificación de ubicación.....	61
Figura 47.	Registro de contacto	62
Figura 48.	Confirmación.....	62
Figura 49.	Activity configuración.....	63
Figura 50.	Toast de confirmación del mensaje enviado	63
Figura 51.	Respuesta del GPS	64

Lista de tablas

Tabla 1. Actividades	19
Tabla 2. Presupuesto	21
Tabla 3. Comparación de GPS	24
Tabla 4. Sitios inseguros.....	50
Tabla 5. Tiempo de encendido GPS Tracker.....	64
Tabla 6. Sincronización GPS.....	65
Tabla 7. Tiempo de retardo de ubicación	66
Tabla 8. Tiempo de carga de Google Maps.....	66

Introducción

El presente documento evidenciará la implementación de un sistema de geolocalización para bicicletas para reducir el hurto de estos vehículos de transporte que en la actualidad se han puesto de moda, pero así mismo se han incrementado los robos a estos vehículos, se detalla el procedimiento para realizar dicho sistema de geolocalización el cual cuenta con el desarrollo de una aplicación móvil para teléfonos Android y la compra del Gps Localizador Tracker Tk102 el cual enviará la ubicación de la bicicleta y se podrá visualizar en la aplicación. Por tanto, en este documento se detallan los pasos para la elaboración de la aplicación y la comunicación entre el GPS y teléfono Android. En el transcurso del documento igualmente, se indica las configuraciones que requiere el GPS con el teléfono, la creación de una base de datos para almacenar los datos de registro de usuarios y la ubicación que proporciona el GPS y el diseño y programación de la aplicación móvil mediante Android Studio.

Problemática

En la ciudad de Bogotá en los últimos tres años el hurto de bicicletas ha aumentado el 400% (Gómez, 2019), por tanto, mi idea es disminuir el hurto de estos vehículos cada vez más apetecidos por la delincuencia en una ciudad donde se realizan casi 800.000 viajes diarios (Penagos, 2018). Por medio de las nuevas tecnologías como es la IoT (Internet de las cosas), la cual nos permite conectar cosas cotidianas con el Internet, se desea conectar la bicicleta a Internet para disminuir las tasas de robos y asesinatos que se presentan en la ciudad de Bogotá.

Se desea incentivar un medio de transporte amigable con el medio ambiente puesto que, no produce gases tóxicos y 16 bicicletas ocupan lo mismo que un vehículo, al momento de parquearse en un sitio público (fundación Aquae, 2019), pero los hurtos y asesinatos en este medio de transporte, causan un miedo en las personas que habitualmente no usan este medio de transporte y no se dan la oportunidad de probarlo.

Justificación

Mediante de la realización del este proyecto de grado, se desea que los bici-usuarios tenga una manera más eficiente de encontrar su bicicleta, en caso un posible hurto o también que tengan la oportunidad de conocer las zonas más inseguras de la ciudad de Bogotá en tiempo real, por medio de una aplicación para móviles Android el cual recurso tecnológico, que la mayoría de las bici-usuarios utilizan a diario, dicha aplicación les notificara cuando se localicen en de dichas zonas, así mismo una página web en la cual puedan consultar los movimientos que realicen en la bicicleta o ver la ubicación de su bicicleta cuando pueda ser robada, con el fin de prevenir el hurto en la ciudad de Bogotá, que es la problemática del este proyecto. Por otra parte, la realización de este proyecto permitirá la adquisición de nuevos conocimientos como lo son programación en Android, páginas web, dado que en el comienzo del proyecto eran desconocidos estos lenguajes de programación que a su vez contribuyen a tener un mejor perfil profesional, como ingeniero en telecomunicaciones, como lo es también el uso de la tecnología IoT (Internet de las cosas) que busca conectar las cosas que utilizan a diario las personas con Internet, en este caso la bicicleta.

Objetivos

Objetivo general

Implementar un sistema de geolocalización para la disminución de hurtos de bicicletas en la ciudad de Bogotá.

Objetivos específicos

- Realizar un análisis técnico de los diferentes dispositivos de GPS en el mercado.
- Diseñar una aplicación móvil para la transmisión y procesamiento de los datos de ubicación obtenidos del sistema de geolocalización.
- Verificar mediante pruebas del funcionamiento del sistema de geolocalización.

Marco de referencia

Estado del arte

Para el proyecto a realizar debo buscar información relacionada con el proyecto por ende al buscar en artículos, blogs o repositorios universitarios mi búsqueda se va baso en encontrar proyectos que tengan que ver sobre bicicletas, seguridad de estas por medio de tecnologías y ubicación por medio GPS o GMPS, además de proyectos de posición web de estas tecnologías para encontrar o visualizar en un tiempo real la ubicación de una bicicleta.

En el primer artículo, el autor diseña un “sistema de seguridad y monitoreo satelital para una bicicleta por medio redes GSM/GPRS” (Murillo, 2016). En el artículo el autor utiliza; un sensor de efecto Hall 49E, un módulo GPS NEO 6M ,un módulo GPRS/GSM M95 QUECTEL, un módulo y control PT2262, para hacer dicho proyecto, además anexa el programa que utiliza en la placa de Arduino uno, asimismo el autor da recomendaciones que a mi parecer me ayudaran a realizar mi proyecto una de estas recomendaciones es: “para asegurar un óptimo funcionamiento, se incluyó una antena GPRS para obtener más cobertura GSM y tener completa confiabilidad a la hora del envío de mensajes de texto al teléfono celular” (Murillo, 2016). Otra parte importante del artículo es en la cual indica ventajas y desventajas del dispositivo que diseñó, lo cual me servirá tener en cuenta al momento de implementar mi proyecto.

El segundo artículo, el proyecto está más enfocado hacia la web, es decir todo el proyecto que está enfocado en la arquitectura de un modelo de servicio web para que un usuario de bicicleta pueda ver en cualquier dispositivo, como por ejemplo un celular o una computadora, las coordenadas de su bicicleta usan tecnología 3G, GPS, entre otras para la elaboración del proyecto. Además, hablan acerca de una arquitectura de WSSBP, la cual usan para realizar su proyecto, algo interesante, es que, por medio de dicha arquitectura, “los dispositivos móviles con módulos GPS seguirán actualizando automáticamente la ubicación del usuario al servicio a través de la red 3G” (Yun-Yao Chen & Yueh-Yun Wang, 2010). Lo que interesó del artículo es como programan y diseñan la conexión entre el dispositivo y la web como anteriormente cite.

Por último, el tercer artículo habla acerca del internet de las cosas (IoT), y lo utilizan en las bicicletas, para evitar accidentes en las vías en las cuales transitan los bici-usuarios por medio de una tarjeta Arduino, una aplicación celular y el GPS. En el proyecto utilizan un módulo GPS NEO 6M, una tarjeta OpenLong y dos acelerómetros conectados a un Arduino NodeMcu. El sistema “puede transmitir datos, incluyendo coordenadas de ubicación y aceleraciones, a un espacio en la

nube directamente o a través de una aplicación móvil. Cualquier persona que se haya registrado puede acceder a la nube para descargar datos”. (P. Qiu, y otros, 2018). Al mismo tiempo, ellos diseñan una aplicación móvil desde la cual los usuarios pueden acceder a los datos como se mencionó anteriormente. De este artículo me intereso la manera en la cual ellos guardan los datos e intentan prevenir los accidentes que puedan llegar a tener los bici-usuarios; en mi caso me interesa avisarle al usuario que se encuentra en una zona peligrosa y así evitar un posible robo, todo esto por medio de la recolección de datos.

Por otra parte, (Patel, Rauniyar, Singh, Dwivedi, & Tripathi, 2018), presentan un sistema de rastreo de niños con base en Arduino usando GPS y GSM, con el fin de reducir la posibilidad de extraviar niños, en mi caso sería la bicicleta, hay que resaltar el uso de la placa ATmega328P, el módulo GSM (SIM900A), la programación la realizan mediante Arduino, que luego se implementaran a la placa Arduino. Además, cabe resaltar la precisión de Google Maps, que es usado en el proyecto para conocer la ubicación de los niños, la precisión del módulo GPS es de 72.2% mientras que al usar Google Maps la precisión del sistema es de 95%, por tanto, se utilizara esta herramienta menciona por estos autores, para desarrollar este proyecto, para mejorar la búsqueda de bicicleta cuando pueda llegar a darse un robo, y pueda encontrarse lo más pronto posible.

Así mismo, (Godfrey & Song, 2016), proponen un sistema de bicicletas antirrobo con el fin de proteger a las bici-usuarios del robo y ayudar a rastrear la ubicación de la bicicleta mediante un teléfono inteligente, utilizando un Arduino MEGA 2560 y el receptor GPS LS20031, este receptor trae una antena inteligente que rastreara hasta 66 satélites a la vez. Además, los autores diseñan una aplicación para móviles Android llamada “Safe Bike”, para notificar al usuario cuando la bicicleta de una persona cambie de posición o se detecte impactos en la bicicleta, es decir cuando la bicicleta se encuentre parqueada y pueda llegar hacer hurtada, y por medio de Google Maps siguen los movimientos de la bicicleta.

Además, (Torres, 2017) utiliza App Inventor, aplicación de Google Labs para la creación de aplicaciones Android, permite que las personas con poca experiencia en programación realizar sus propias aplicaciones, se diseñan desde la página de App Inventor y por medio de un código QR o una clave proporcionada por la página, pueda interactuar con la aplicación que se está diseñando y también, si se desea descargar el APK de la aplicación e instalarla en los móviles Android, es interesante acceder a este tipo aplicaciones para gente inexperta y logren aprender programación a

través de bloques, los cuales pueden ser botones, mapas, cuadros de texto, entre otras herramientas que se desean agregar al aplicación.

Marco teórico

Para la elaboración del proyecto debemos tener un claro varios términos de tecnologías y componentes del prototipo que se va a diseñar.

GPS.

El sistema de posición global (GPS), nos permite conocer la posición de un objeto en cualquier parte del mundo, funciona en bajo cualquier condición climática, en cualquier parte del mundo, las 24 horas del día, es decir nos da la hora exacta de dicho objeto además de sus coordenadas; lo cual en este proyecto será útil para conocer la ubicación exacta de bicicleta, además se usará el módulo NEO-6M el cual se configura por medio de Arduino. (GPS.GOV, 2019; IAC, 2018; Samsung, 2018)

Redes GSM.

El sistema global de comunicaciones móviles (GSM), es un protocolo de red el cual nos permite transmitir voz y datos digitales de bajo tamaño, como lo son mensajes de texto o mensajes de multimedia, a través de un dispositivo móvil. (Villagómez, 2017)

XAMPP.

XAMPP es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. De hecho, su nombre viene de ahí, X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP) y P (Perl). XAMPP es independiente de plataforma y tiene licencia GNU GPL. (Zepeda, 2015)

Hosting.

Hosting se refiere al servicio que permite que un sitio web permanezca en línea los 365 días del año, sin embargo, el Hosting no solamente sirve para hospedar un sitio web, sino que sirve para hospedar cualquier tipo de información, incluido aplicaciones, sistemas de correos, archivos, bases de datos etc. En pocas palabras si se trata de confiar la información en un proveedor externo, cualquiera que ella sea, se requiere este servicio. (Hosting Red, 2020)

Aplicación móvil.

Una aplicación móvil (o una app) es una aplicación informática diseñada para ser ejecutada en smartphones (teléfonos inteligentes), tabletas, u otros dispositivos móviles. El desarrollo de las aplicaciones tiene en cuenta las limitaciones de los dispositivos, como la batería o el software; y

comienzan probándose utilizando un emulador, para después ponerse al mercado en versión de prueba. (Appsmovilescavucm, 2020)

Android Studio.

Android Studio es un entorno de desarrollo, un software, que cuenta con herramientas y servicios para que los desarrolladores puedan crear nuevas aplicaciones para Android. Muchos de los sistemas operativos actuales cuentan con este tipo de entornos de desarrollo, algo que ocurre también en el sistema operativo de Google. (Ferreño, 2020)

Marco legal

Se deben tener en cuenta las siguientes leyes para el desarrollo del proyecto para percatarse de no incumplir ninguna norma establecida en la constitución política de Colombia:

Ley 1341 de 2009.

Artículo 3. Sociedad de la información y del conocimiento. El Estado reconoce que el acceso y uso de las Tecnologías de la Información y las Comunicaciones, el despliegue y uso eficiente de la infraestructura, el desarrollo de contenidos y aplicaciones, la protección a los usuarios, la formación de talento humano en estas tecnologías y su carácter transversal, son pilares para la consolidación de las sociedades de la información y del conocimiento. (Ministerio de Tecnologías de la Información y las Comunicaciones, 2020)

Artículo 4. Protección de los derechos de los usuarios. El Estado velará por la adecuada protección de los derechos de los usuarios de las Tecnologías de la Información y de las Comunicaciones, así como por el cumplimiento de los derechos y deberes derivados del Habeas Data, asociados a la prestación del servicio. Para tal efecto, los proveedores y/u operadores directos deberán prestar sus servicios a precios de mercado y utilidad razonable, en los niveles de calidad establecidos en los títulos habilitantes o, en su defecto, dentro de los rangos que certifiquen las entidades competentes e idóneas en la materia y con información clara, transparente, necesaria, veraz y anterior, simultánea y de todas maneras oportuna para que los usuarios tomen sus decisiones. (Ministerio de Tecnologías de la Información y las Comunicaciones, 2020)

Resolución No. 3501 de 2011.

Por la cual se determinan las condiciones de acceso a las redes de telecomunicaciones por parte de proveedores de contenidos y aplicaciones a través de mensajes cortos de texto (SMS) y mensajes multimedia (MMS) sobre redes de telecomunicaciones de servicios móviles, y se dictan otras disposiciones. (Comisión de Regulación de Comunicaciones, 2020)

Metodología

Para el proyecto se realizará una investigación con enfoque cuantitativo, la cual es secuencial. Cada etapa precede a la siguiente y no se puede “brincar” o eludir pasos. El orden es riguroso, aunque desde luego, se puede redefinir alguna fase (Sampieri, 2014), es decir, se comienza este proyecto con la búsqueda de información de una tecnología que ayude satisfacer la problemática planteada, lo que deriva en seguir un conjunto de pasos como lo son analizar las zonas inseguras de la ciudad, que lenguajes de programación son los adecuados para la elaboración de la base de datos, diseñar el código que permita la comunicación entre el GPS, el servidor y la aplicación móvil con sistema operativo Android, posteriormente una serie de pruebas, con la integración de los diferentes componentes del proyecto, como la base de datos, el móvil, el servidor local y el GPS. Esto que quiere decir que además de tener un enfoque cuantitativo también esta investigación tiene un enfoque científico, que busca solucionar una problemática y en la búsqueda de esta solución, se va adquiriendo conocimiento requerido para la elaboración del proyecto, como es la programación de una aplicación para móviles Android.

Finalmente se realizará una muestra, tomando un grupo de personas que sean parte de la población de bici-usuarios de la ciudad, se les dará la oportunidad de utilizar sistema de seguridad para bicicletas y en base a la observación, análisis de los datos que arroje el muestreo; tomar la ventajas y desventajas que tenga el sistema, corregir los errores que puedan llegar a existir y elaborar un informe con los resultados, lo que permitirá determinar, si se ha logrado responder a la pregunta problema y sacar las conclusiones de toda la elaboración del proyecto.

Cronograma

Tabla 1.
Actividades

Actividad	Mayo	Junio	Julio	Ago	Sept	Oct	Nov
Investigación de zonas inseguras de la ciudad de Bogotá	X						
Investigación de los diferentes lenguajes de programación de bases de datos y elección		X					
Investigación sobre base de datos y servidor		X					
Adquisición del GPS			X				
Diseño de la base de datos				X			
Diseño del código de comunicación del GPS y el móvil							
Creación de la base de datos y servidor				X	X		
Diseño de la aplicación móvil						X	X
Pruebas al sistema de geolocalización							

Nota. Autoría propia, 2020.

Actividades

Investigación de zonas inseguras de la ciudad de Bogotá: por medio de internet investigar en páginas de periódicos, revistas, bases de datos, la policía; cuales son las zonas en las que ocurren la mayoría hurto de bicicletas en la ciudad de Bogotá, lo cual permita diseñar un mapa de la ciudad con las zonas peligrosas, seguras y esto ayude a prevenir el hurto de bicicletas mediante alertas generadas en aplicación cuando una persona circule por la ciudad y esté utilizando la aplicación móvil.

Adquisición del GPS: ir al centro de la ciudad donde venden componentes electrónicos y cotizar el GPS, encontrar el mejor precio, así mismo investigar por internet donde vendan componentes electrónicos y cotizar; luego de esto, escoger el mejor precio de todas las opciones.

Investigación de los diferentes lenguajes de programación de bases de datos y elección: realizar una investigación, la cual me permita interactuar con los diferentes softwares de programación,

descargar, interactuar con la interfaz y la forma de conexión con un servidor web, después de esto escoger el más apropiado.

Diseño de la base de datos: Diseñar un modelo entidad relación para modelar la base de datos, saber que datos se requieren almacenar, por ejemplo, usuario, contraseña, etc. Después de ello empezar a escribir el código de la base de datos, ir realizando pruebas para mirar el funcionamiento de esta y conectarla al servidor local.

Diseño del código que comunique GPS con la aplicación: diseñar un diagrama de flujo para saber cómo se comunicaría del GPS, luego empezar a investigar cómo se realiza la comunicación, entre el GPS, el servidor local y la aplicación, para más adelante implementar el correspondiente código en el diseño de la aplicación móvil.

Investigación del funcionamiento de un servidor local: al no saber cómo se utiliza un servidor local, se realizará una búsqueda en libros, Internet, preguntándole a profesores como se une esta parte, que es la base de datos que está en el servidor local para verificar el registro del usuario y los movimientos que este realice, cuando este usando la aplicación.

Diseño de la aplicación móvil: investigar cómo se crea una aplicación móvil para Android, puesto que no se tiene el conocimiento para realizarlo, por medio de profesores, compañeros de universidad que tengan el conocimiento y pedirles una colaboración en el proyecto, es decir, me enseñen a programar en el lenguaje correspondiente a Android Studio. Así mismo buscar en Internet, libros, tutoriales para diseñar, crear aplicaciones móviles e ir adquiriendo el conocimiento necesario para realizar dicha aplicación y no tener que depender de alguien más en esta parte del proyecto. También se deben hacer pruebas para verificar que la aplicación funciones correctamente y se vincule el GPS al móvil y registre los datos de este dispositivo.

Presupuesto

Tabla 2.
Presupuesto

Presupuesto del proyecto				
Elemento	Cantidad	Valor Unitario	Total	Descripción
GPS	1	\$ 100.000	\$ 100.000	Componente electrónico con el cual se enviará la ubicación de la bicicleta.
Sim Card	1	\$ 5.000	\$ 5.000	Tarjeta Sim Card.
Plan de datos	1	\$ 20.000	\$ 20.000	2GB de Internet para poder enviar la ubicación del GPS.
Servicio de Internet	1	\$ 74.000	\$ 74.000	Conexión mediante la cual estará funcionando la base de datos.
Días de programación	80	\$ 30.000	\$ 2.400.000	Tiempo destinado a la programación de la aplicación, servidor local, base de datos.
Total			\$ 2.599.000	Valor total del proyecto

Nota. Autoría propia, 2020.

Desarrollo

Adquisición del GPS

Antes de adquirir el dispositivo de GPS Tracker se debe conocer, que es y cómo funciona. Para ello se puede definir a un GPS Tracker como un dispositivo que permite localizar cualquier objeto, por ejemplo, un vehículo, una maleta o ser vivo situado en la tierra, mediante la tecnología GPS (Sistema de Posicionamiento Global). (El Localizador GPS, 2020)

Este tipo de GPS funcionan mediante un microchip (Sim-Card), insertado en interior del vehículo, para este proyecto el vehículo será una bicicleta a la cual se le dará seguimiento. Este GPS permite rastrear la permanente ubicación del vehículo, mediante el apoyo de satélites de órbita baja. Que posteriormente envía señales mediante la red del proveedor de la Sim-Card del usuario, que indican la ubicación y otras informaciones, como pueden ser velocidad del vehículo, hora y fecha de la consulta de ubicación, porcentaje restante de la batería. (AutosOnline, 2020)

Además, el GPS debe estar en línea directa con un mínimo de 3 satélites diferentes para triangular una ubicación. Si se encuentra en un lugar abierto, pero cerca de un edificio que bloquea la mitad del cielo, es posible que no esté a la vista de 3 satélites. Los satélites conocen su ubicación en el espacio y conocen la distancia entre ellos. Luego miden la distancia entre ellos y su GPS Tracker, por lo que los satélites le darán la ubicación del rastreador con gran precisión. Los GPS Tracker más sofisticados también cruzan las ubicaciones de verificación entre GPS, WIFI, LBS y Bluetooth para mostrar la ubicación más precisa posible. (Tailit, 2020)

Como se menciona anteriormente una manera de verificar la ubicación es mediante el uso de LBS (estaciones base locales). Esto se utiliza cuando no hay señales de satélites. Como ejemplo, esto se usaría cuando el GPS Tracker este ubicado en sótanos o en el medio de edificios de concreto sin señal. En estos casos, con frecuencia se obtienen señales LBS. LBS te dará una ubicación aproximada incluso cuando reciba señal de solo 1 o 2 antenas. No será muy preciso, pero te dará una indicación de donde se encuentre el vehículo que se requiere ubicar. Si hay más antenas LBS en el área, la ubicación es más precisa (Tailit, 2020). Un sistema LBS utiliza tecnología de sistemas de información geográfica, alguna de las tecnologías de posicionamiento bien sea del lado cliente (Por ejemplo los sistemas GPS anteriormente mencionados) o del lado del servidor (Por ejemplo un servicio de posicionamiento suministrado por el operador de la red; generalmente algún método de triangulación espacial y/o ubicación por celdas) y tecnología de comunicación de redes para

transmitir información hacia una aplicación LBS que pueda procesar y responder la solicitud. (Ciampagna, 2013)

Otros tipos de rastreadores que se encuentran en el mercado son los Tracker Bluetooth, que son un pequeño localizador que funciona gracias a una aplicación para descargar en el propio smartphone. Una vez que smartphone y Tracker estén conectados entre ellos por Bluetooth, la aplicación permite la visualización en el smartphone de la posición del objeto al que el Tracker ha sido unido, poco por ejemplo unas llaves. Cuando el dispositivo de rastreo se encuentre a más de 10 a 25 metros de distancia del teléfono (depende del fabricante), este sonará dando una advertencia al usuario que se ha perdido la ubicación del objeto rastreado, y mediante el mapa de la aplicación, podrá rastrear la última ubicación del dispositivo. (Maikii, 2020)

Por otra, se debe resaltar los tipos de seguimientos que pueden utilizarse con un GPS, es decir los modos de uso, existe el modo pasivo que es donde el rastreador está inactivo hasta que se le solicite una ubicación, con esto se ahorra la batería. El otro modo será activo en el cual el GPS actualizará su ubicación (Tailit, 2020). Por último, la opción más importante en rastreadores GPS es el seguimiento en vivo, que actualiza la ubicación en tiempo real, esto se utiliza mayormente en flotas de buses, autos de valores o depende el uso que requiera. Esta es una característica realmente importante si se necesita encontrar a una mascota, un anciano, un vehículo o la persona que acaba de robar bicicleta, que es lo que requiere en este proyecto.

Para el desarrollo de proyecto se comienza con la búsqueda del dispositivo GPS que dará la ubicación de la bicicleta, para ello se investigó en distintos sitios webs de venta de dispositivos de rastreo para bicicletas, la principal característica que se buscó, es la precisión del GPS y su precio, la mayoría de dispositivos encontrados tienen la misma precisión de del GPS (5 metros). Respecto a su tamaño, los diferentes dispositivos encontrados, algunos son más grandes que otros, es decir, no existe un tamaño estándar para este tipo de dispositivos. En cuanto a la red que utilizan, la mayoría de ellos usan GSM y GPRS ya que, vienen con ranura de tarjeta SIM lo que permite comunicarse vía satélite y enviar mediante un mensaje de texto con las coordenadas de ubicación e información adicional de donde está el GPS, que estará ubicado en la bicicleta.

La mayoría de estos sitios web son de otros países y los precios están en dólares estadounidenses, dada la situación actual, se buscó el GPS en la plataforma online de MercadoLibre y se encontró la referencia Gps Localizador Tracker Tk102 por un valor de \$100.000 pesos

colombianos que en dólares estadounidenses serían aproximadamente US\$26.11, no se realizó la compra de manera online, sino se realizó la compra de manera presencial en el local del comprador, a continuación se observa una tabla comparativa de los distintos modelos de GPS investigados en Internet.

Tabla 3.
Comparación de GPS

Nombre del modelo	Características
GPS305	Red: GSM/GPRS Banda: 850/900/1800/1900 MHz Sensibilidad del GPS: -159dBm Precisión del GPS: 5 metros Ranura tarjeta SIM: Si Dimensiones: 23,5x110mm Batería: 3.7 V 500 mAh Costo: \$39 dólares
EV-09	Red: GPS/WIFI/BLE Banda: 850/900/1800/1900 MHz Sensibilidad del GPS: -165 dBm Precisión del GPS: 5 metros Ranura tarjeta SIM: Si Dimensiones: 43x14,5mm Batería: 3.7 V 500 mAh Costo: \$38 dólares
TK906	Red: GSM/GPRS Banda: 850/900/1800/1900 MHz Sensibilidad del GPS: -159dBm Precisión del GPS: 5 metros Ranura tarjeta SIM: Si Dimensiones: 40x85x28mm Batería: 3.7 V 1800 mAh Costo: \$32 dólares
Gps Localizador Tracker Tk102	Red: GSM/GPRS Banda: 850/900/1800/1900 MHz Sensibilidad del GPS: -159dBm Precisión del GPS: 5 metros Ranura tarjeta SIM: Si Dimensiones: 64x46x17mm

	Batería: 3.7 V 1800 mAh Costo: \$26 dólares
Primetracking	Red: No especifica Banda: No especifica Sensibilidad del GPS: 15.24 metros Precisión del GPS: 5 metros Ranura tarjeta SIM: Si Dimensiones: 68x38x23mm Batería: 3.7 V 500 mAh Costo: \$44 dólares

Nota. Autoría propia, 2020.

Como se mencionó anteriormente se adquirió el Gps Localizador Tracker Tk102 el cual cuenta con las siguientes características:

Seguimiento satelital vía SMS o GPRS.

Micrófono espía

Entrada para memoria micro SD para almacenar recorridos.

Alarma batería baja.

Alarma exceso de velocidad.

Alarma por golpes.

Alarma de movimiento.

Antes de describir algunos de los comandos que el GPS Tracker permite realizar, se comentará brevemente hacer de que son estos comandos y como función. La manera en la cual se el usuario de un GPS Tracker puede interactuar con dicho dispositivo es a través de comandos AT, estos comandos son las instrucciones codificadas que conforman un lenguaje de comunicación entre el usuario y un terminal (en este el GPS Tracker), los comando AT se denominan así por la abreviatura de attention (atención). Todos los GPS Tracker poseen una lista de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas, leer, escribir y enviar mensajes SMS. (3Cu Electrónica, 2020)

A continuación, se mostrarán algunos comandos que tienen el Gps Localizador Tracker Tk102 y que fueron de utilidad para desarrollar el sistema de geolocalización.

El primer paso con el GPS será la inicialización, para esto se envía un mensaje de texto a la sim card que se instala en el GPS y el comando es “begin+password” y el GPS responde “begin ok!”, como se puede observar en la *figura 1*.

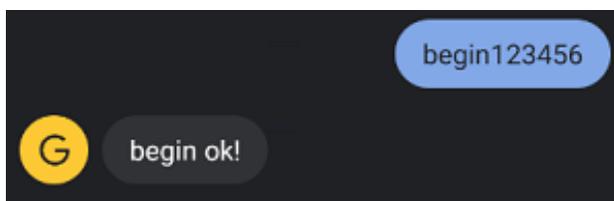


Figura 1. Configuración GPS 1. Autoría propia (2020).

La contraseña que viene de por defecto es “123456”, para cambiar dicha contraseña se debe usar el siguiente comando “password+old +password+space+new password”. Por otra parte, se debe configurar el número de teléfono autorizado, es decir el numero al cual queremos que llegue dicha ubicación para esto se usa el comando “admin+password+space+ cell pone number” y el GPS responderá “admin OK”, como se puede visualizar en la *figura 2*. Además, el GPS puede enviar la ubicación mediante una llamada al sim card que se encontré en este, a lo que el GPS responderá con un mensaje de texto con la siguiente información: latitud, longitud, velocidad, fecha y hora, porcentaje de batería y link de Google Maps.

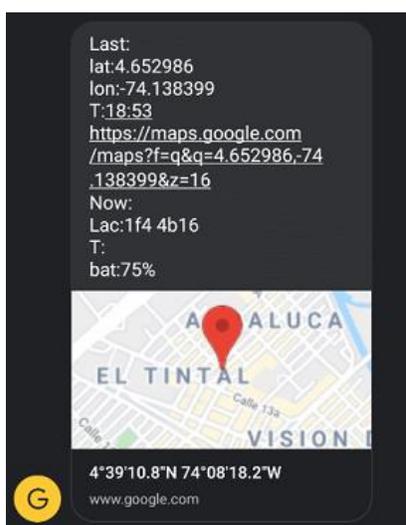


Figura 2. Configuración GPS 2. Autoría propia (2020).

Se debe configurar la hora del GSP puesto que el GPS está configurado con la hora UTC (Tiempo Universal Coordinado) que es el nombre más conocido de la zona horaria UTC+0, para cambiar a la zona horaria de Bogotá (UTC-5) se debe enviar un mensaje al GPS con el siguiente

comando “**time+space+zone+password+space+local time zone**”, el GPS enviara un mensaje de confirmación de cambio de zona horario el cual será “**time ok!**”. En la *figura 3* se observar el cambio de zona horaria hecho al GPS.

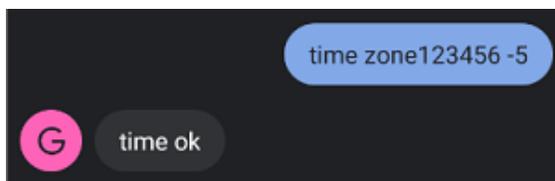


Figura 3. Configuración GPS 3. Autoría propia (2020).

Para sabe el estado actual de dispositivo GPS se cuenta con el comando “**check**”, se le debe enviar un mensaje al dispositivo GPS con el siguiente código “**check+password**”, en la *figura 4* se observa el estado actual en el que se encuentra el GPS, como se observa el dispositivo GPS no cuenta con señal GPRS ni GPS.

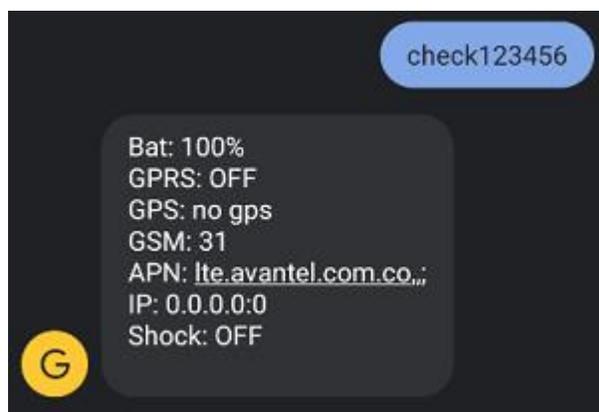


Figura 4. Configuración GPS 4. Autoría propia (2020).

Para evitar que el usuario debe aprenderse todos estos comandos, se desarrollará una aplicación móvil para dispositivos Android para que el usuario solamente tenga que registrar el número de la sim card que instalo en el GPS y la aplicación por medio de botones enviara los anteriores mensajes, y el usuario logrará visualizar la ubicación de su bicicleta por medio de la interfaz de la aplicación.



Figura 5. GPS Localizador Tracker Tk102. Autoría propia (2020).

Topología de red del sistema de geolocalización

Para desarrollo del proyecto debe especificarse la topología de red utilizada para el sistema de geolocalización, es decir, como estará diseñada la red para el intercambio de los datos, este caso la ubicación que transmite el GPS Tracker. En la **figura 6** se observa el mapa lógico de la red para el intercambio de datos que conforman el sistema de geolocalización.

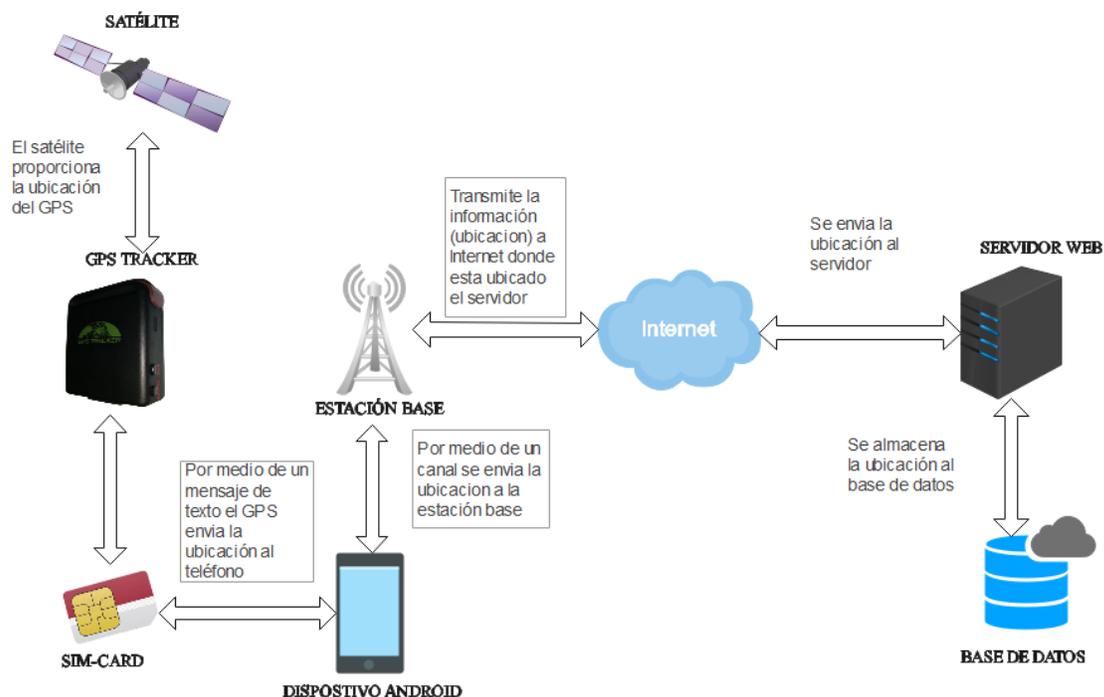


Figura 6. Diagrama de flujo del sistema de geolocalización. Autoría propia (2020).

Investigación de lugares inseguros de la ciudad de Bogotá

Para la elaboración de la aplicación y las alerta que la aplicación tendrá, lo primero que se debe realizar es una investigación acerca de los lugares que en la ciudad de Bogotá son más críticos o en los cuales se presenta un mayor porcentaje de hurto de bicicletas, se buscó en portales de periódicos de la ciudad de Bogotá. Cabe resaltar la investigación del periodista Javier Gonzales el cual nos entrega cifras interesantes como, por ejemplo, el promedio de bicicletas robadas diariamente entre enero y julio era de 11, en 2018 ese número ascendió a 19. Dicho en otras palabras, cada 75 minutos es hurtado uno de estos vehículos en la capital (Penagos, El Espectador, 2018). Y la información más importante que nos ofrece:

Al revisar por zonas, la más azotada por hurto de bicicletas es Suba, donde en los primeros siete meses de 2018 han sido hurtadas 647, es decir que allí ocurren el 16 % de los hechos. Le siguen Engativá (540), Kennedy (483), Usaquén (441), Teusaquillo (272) y Chapinero (268). En estas cinco zonas se concentran el 66 % de los robos. Por el contrario, las zonas más seguras para ciclistas son Candelaria (con nueve robos), Usme (19), San Cristóbal (32), Tunjuelito (66) y Antonio Nariño (67). En la **figura 7** se observa los sitios más inseguros de la ciudad de Bogotá.

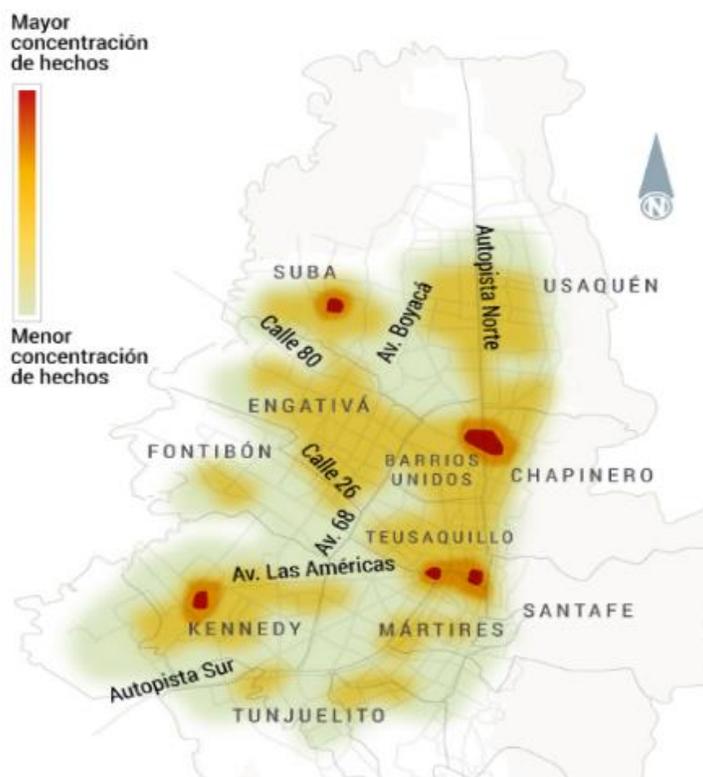


Figura 7. Hurto de bicicletas en Bogotá. (Penagos, El Espectador, 2018).

Creación de base de datos

La función de la base de datos será almacenar los datos usuarios que se registren en la aplicación, así como la ubicación que reporten desde la aplicación. Los datos que se requerirá el usuario serán los siguientes: nombre, email, contraseña, cédula, marco de bicicleta y número de sim card del GPS.

Los últimos dos datos que se solicita registrar al usuario serán fundamentales dado que, primero el marco es el identificativo que tiene cada bicicleta, es decir un dato único, como lo es para las personas el número de cedula, por otra parte el numero de la sim card que se utilice en el GPS se requiere dato, puesto que este dato además de almacenarse en la base de datos será con el cual la aplicación quedara configurada para enviar los mensajes o los reportes de ubicación de cada usuario. Para la elaboración de la base de datos se utilizó dos programas, XAMPP y Visual Studio Code, el primero permite la gestión de base de datos MySQL, además de usarse como un servidor local dado que XAMPP viene con Apache que es un servidor web, pero en este se utiliza como un servidor local y el según es un editor de código, en cual se realizó el código de la comunicación entre la aplicación y la base de dato. Para la aplicación fue necesario realizar una base datos con dos tablas, la primera tabla está diseñada para el registro de usuarios y la segunda para guardar la ubicación que envié el GPS. Se utiliza el lenguaje PHP para enviar la información a la base de datos, puesto que, PHP permite gestionar peticiones y dar respuesta, en este caso una petición puede ser el registro de un usuario y la respuesta puede ser exitosa, es decir se registró correctamente el usuario o errónea, es decir existió un error al momento de realizar el registro, que puede ser por falta de conexión de Internet del usuario o que el usuario registro un correo ya existente en la base de datos. En las **figuras 7 y 8** se puede observar las dos tablas anteriormente nombradas, así como algunos registros de prueba que se han realizado.

+ Opciones									
	id	nombre	email	contrasena	cedula	marcobici	simbici		
<input type="checkbox"/>	1	uff	hooj	jjj	868	xhx	868		
<input type="checkbox"/>	2	jorge	jorge	123456	123	n123	123		
<input type="checkbox"/>	3	edisson	jorge@mail.com	qwerty	147852	n567	369852		
<input type="checkbox"/>	4	Jorge Jimenez	jorge1@mail.com	123456	369852	N543	2550		
<input type="checkbox"/>	5	Felipe	felipe@mail.com	q2erty	147852	jgkj	562		
<input type="checkbox"/>	7	karen	karen@mail.com	qwertyu	124846	n73737	342181		
<input type="checkbox"/>	8	Andrés	andres@mail.com	1234567	316461	k73737	36161517		
<input type="checkbox"/>	9	Ana	ana@mail.com	asdfigh	12121212	j7373	3192929		

Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

Figura 8. Tabla de registro. Autoría propia (2020).

+ Opciones			id	prueba1	
<input type="checkbox"/>	Editar	Copiar	Borrar	8	https://maps.google.com/maps?f=q&q=4.653064,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	13	https://maps.google.com/maps?f=q&q=4.653075,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	24	https://maps.google.com/maps?f=q&q=4.653005,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	25	https://maps.google.com/maps?f=q&q=4.653005,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	26	https://maps.google.com/maps?f=q&q=4.652732,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	27	https://maps.google.com/maps?f=q&q=4.652732,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	28	https://maps.google.com/maps?f=q&q=4.652931,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	29	https://maps.google.com/maps?f=q&q=4.652999,-74.13...
<input type="checkbox"/>	Editar	Copiar	Borrar	30	https://maps.google.com/maps?f=q&q=4.652999,-74.13...

Figura 9. Tabla de ubicación. Autoría propia (2020).

Además, como se viene mencionando anteriormente se necesita un archivo PHP para la comunicación entre la base de datos y la aplicación por tal razón se necesitan tres archivos PHP, uno para el registro, otro para el login, y otro para registrar la ubicación de la bicicleta. El archivo de login se utiliza para consultar email y contraseña de la tabla registro para verificar primero que el usuario este registrado y que los datos suministrados al momento del login sean correctos.

Diseño de la aplicación móvil

Después de tener la base de datos se procede a realizar la aplicación, se debe tener en cuenta cuántas ventanas (activities) tendrá la aplicación, en otras palabras, cuantas funciones realizará la aplicación, que serían las siguientes: registro, login (inicio de sesión), menú principal, sitios inseguros, ubicación, bicicleta robada y configuración.

En total serán siete activities, los dos primeros ya sean mencionado anteriormente su funcionalidad, el siguiente menú principal tendrá los accesos a las diferentes funciones de la aplicación, como se mencionan anteriormente, sitios inseguros (permite al usuario visualizar los lugares de la ciudad de Bogotá con mayor robo), ubicación (permite al usuario enviar un mensaje de texto al GPS el cual responderá con su ubicación, la cual se puede visualizar mediante la aplicación Google Maps), bicicleta robada (permite al usuario que el GPS le envíe su ubicación cada determinado tiempo, se configuró que el GPS responda al usuario cada tres minutos su ubicación, además esta activity cuenta con los números de los caís más cercanos de los lugares con mayor robo de bicicleta en la ciudad de Bogotá) y la última activity permite al usuario configurar cuatro números de teléfono los cuales pueden ubicar al GPS, en caso que el usuario sea hurtado de su bicicleta y celular y desee ubicar su bicicleta, por medio de los otros cuatro números previamente configurados. Otro caso puede un padre que desee averiguar la ubicación de su hijo o

hija que este montando bicicleta, si el número del padre esta entre esos cuatro podrá llamar al GPS y este le responderá con su ubicación.

Al tener claro la cantidad de actividades que tendrá la aplicación referente al número de acciones o funcionalidad que se realizaran se procede a crear un nuevo proyecto en Android Studio y empezar a programar y diseñar la parte grafica de cada activity, dado que al crear una nueva activity Android Studio crea dos archivos uno JAVA y otro XML, en el primero se crea la parte lógica del activity y el segundo la parte grafica que tendrá dicho activity.

Lógica de la aplicación móvil

Mediante el uso de diagramas de flujo se explicará la lógica que conlleva realizar esta aplicación móvil. El primer diagrama de flujo **figura 10** se analiza el comportamiento de la primera activity iniciar sesión, en la cual el usuario tendrá la opción de ir directamente al menú principal digitando su correo y contraseña o en caso contrario que el usuario no encuentre registrado, tendrá la alternativa mediante un botón, de ir a otra activity en la cual podrá registrarse.

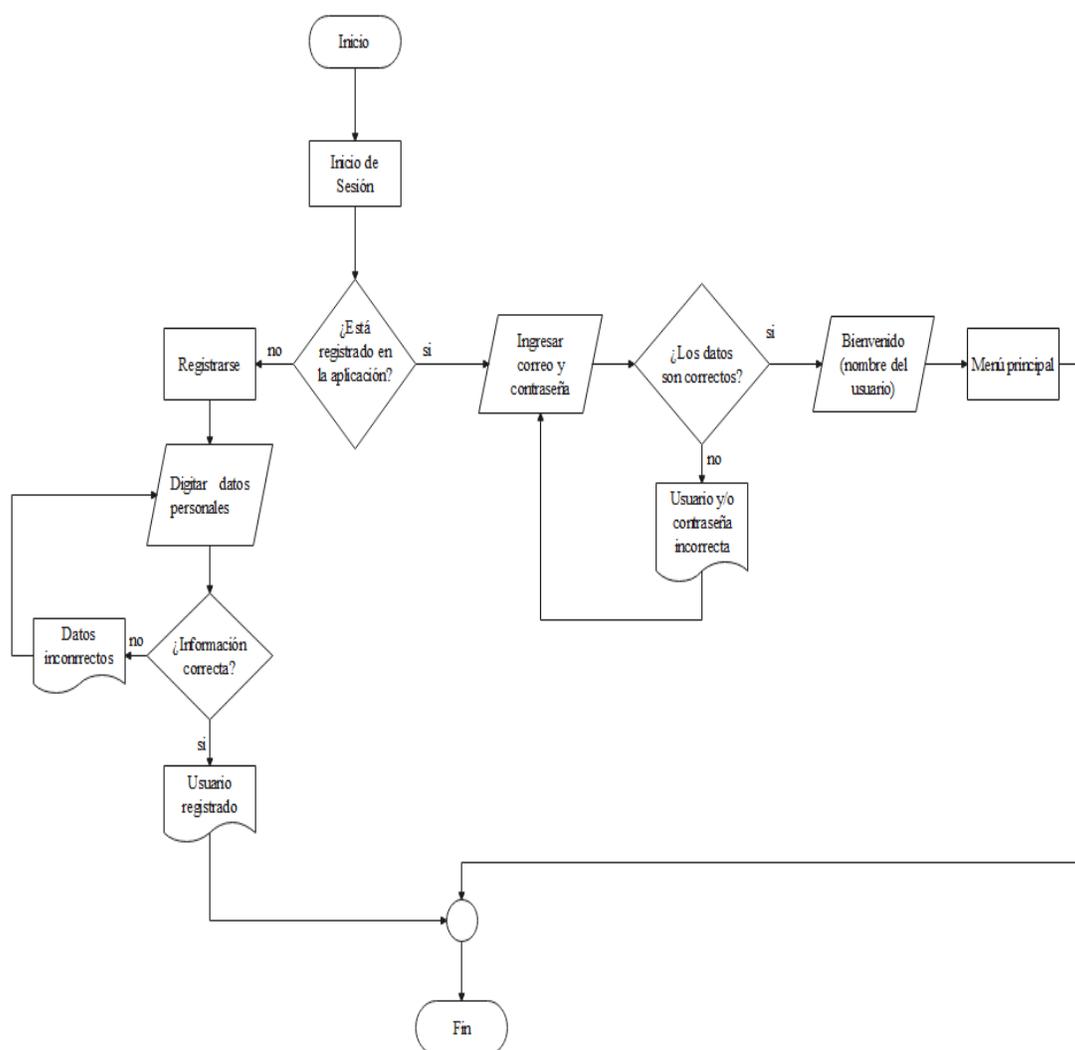


Figura 10. Diagrama de flujo activity main y registro. Autoría propia (2020).

El primer botón con el cual usuario podrá interactuar en el menú principal será el botón sincronizar GPS, mediante la **figura 11** se representa el proceso del botón anteriormente mencionado, su lógica

es sencilla y el usuario deberá utilizarlo solo una vez para sincronizar el GPS con el dispositivo móvil, cabe resaltar que si el usuario desinstala la aplicación deberá volver a sincronizar el GPS con el móvil.

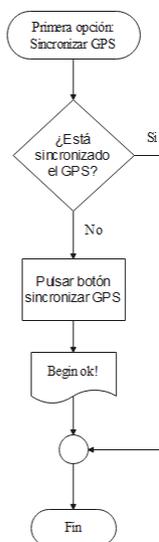


Figura 11. Diagrama de flujo activity sincronizar GPS. Autoría propia (2020).

El segundo botón lo podrá utilizar el usuario, para realizar una consulta visual acerca de los lugares más inseguros para circular por la ciudad de Bogotá en bicicleta, el proceso o actividad que el usuario puede desarrollar se evidencia en la *figura 12*.

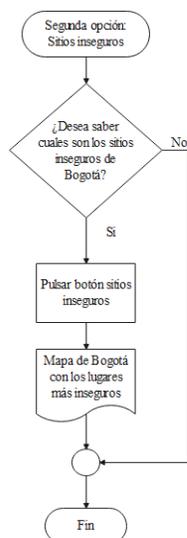


Figura 12. Diagrama activity sitios inseguros. Autoría propia (2020).

La tercera opción que tiene el usuario es la más importante o significativa de la aplicación, la cual le permitirá vigilar o rastrear su bicicleta, se decidió llamar al botón de esta opción ubicación, para que usuario la recuerde fácilmente, la lógica detrás de este botón se puede evidenciar en la **figura 13**, en dicha figura se observa todo el proceso desde que el usuario pulsa el botón ubicación hasta donde el usuario podrá observa la ubicación exacta de su bicicleta.



Figura 13. Diagrama de flujo activity ubicación. Autoría propia (2020).

En la *figura 14* se evidencia los procesos o actividades detrás de la opción número cuatro del menú principal de la aplicación, en la cual el usuario puede escoger en caso de un robo, que opciones le puede ofrecer la aplicación o en el caso de querer consultar algún número de un CAI cercano a los lugares más peligrosos de ciudad de Bogotá.

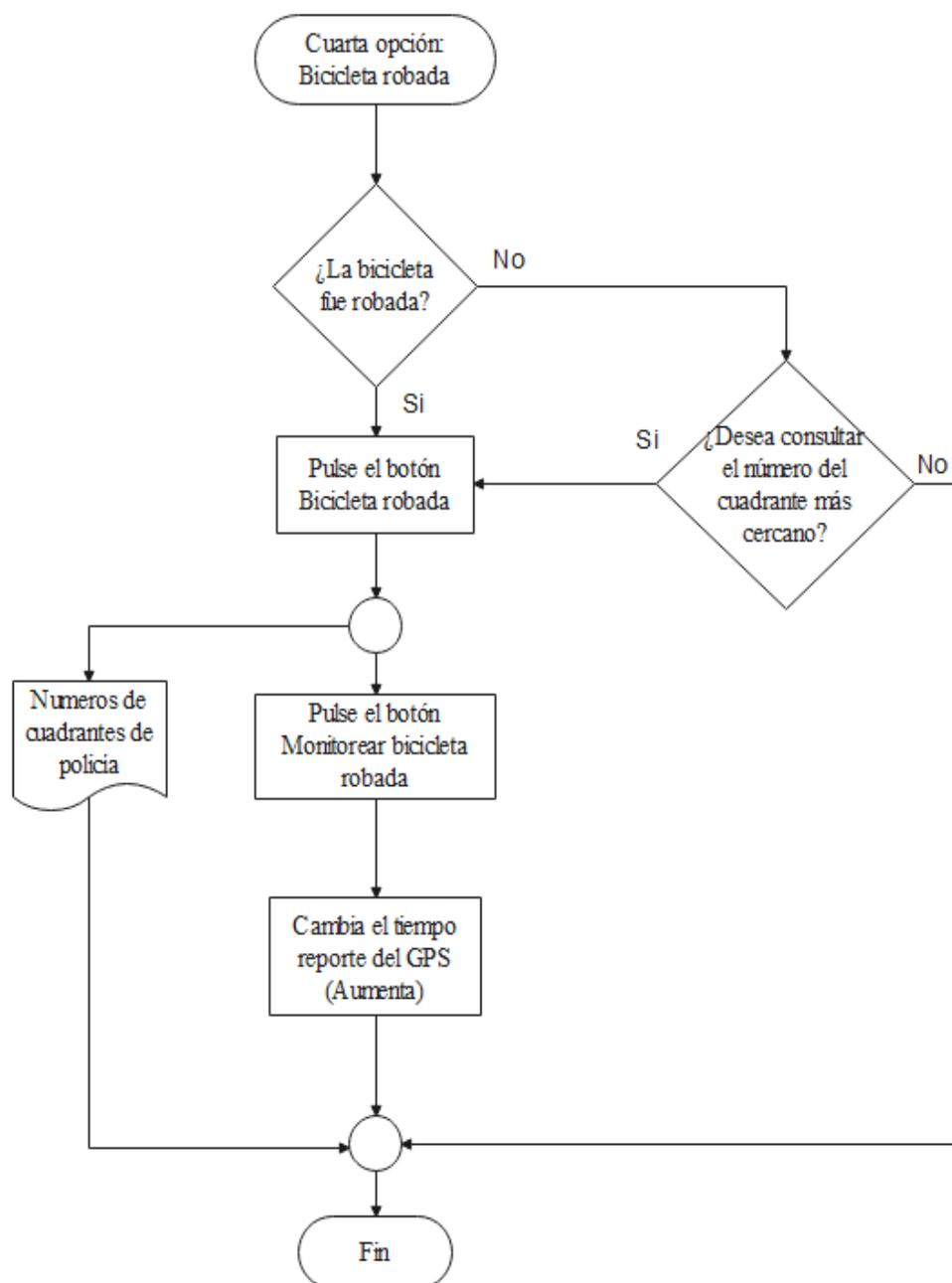


Figura 14. Diagrama de flujo activity emergencia. Autoría propia (2020)

En la **figura 15** se evidencia la última opción o quinta del menú principal, en la cual el usuario dispone de dos botones, el primero para realizar el registro de un número de emergencia, el cual puede verificar la ubicación del GPS, y el segundo le permite al usuario conocer cómo se encuentra el GPS respecto a su batería y señales (GPS, GMS, GPRS), además también de su ubicación.

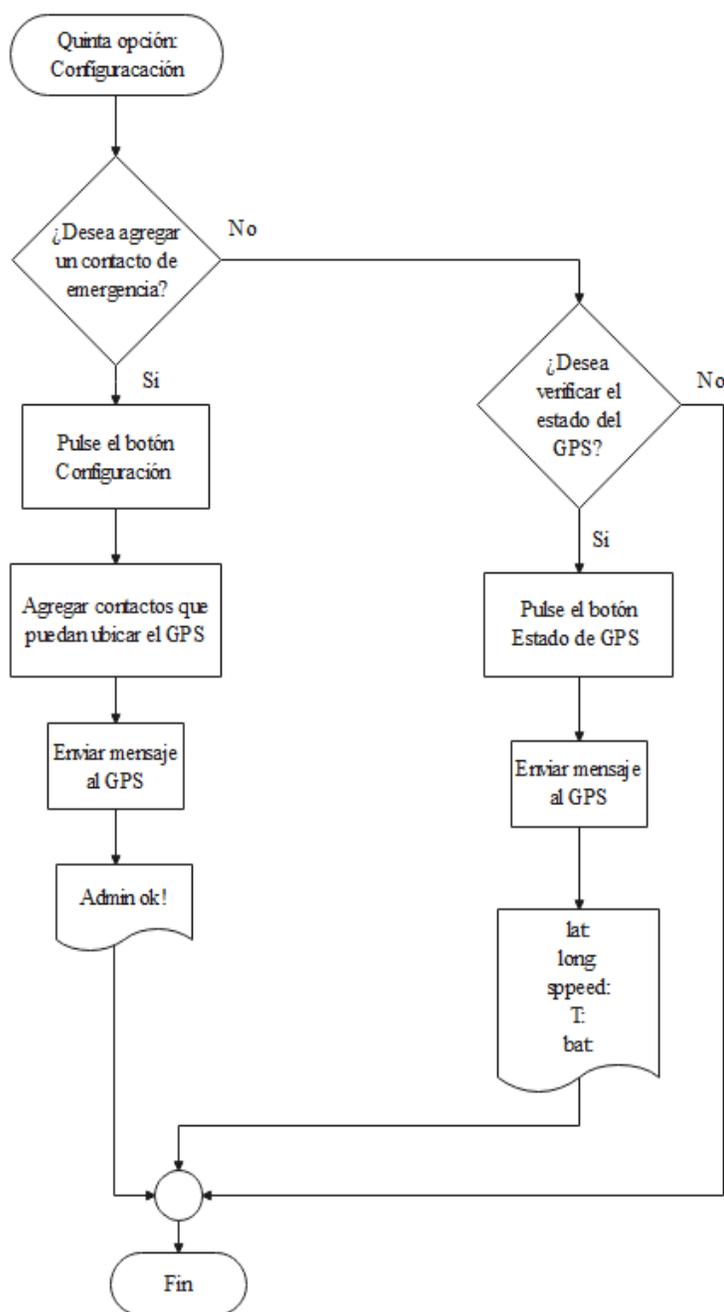


Figura 15. Diagrama activity configuración. Autoría propia (2020).

Diagramas de flujo de información

Para verificar o analizar el comportamiento de la aplicación respecto al manejo que se le da a la información, es decir cómo se captura, lee, y envía, se explicará mediante diagramas de bloques en los cuales se muestra flujo de dicha información. Para comenzar analizar el flujo de información en la aplicación móvil en la **figura 16**, se muestra como el activity main, se verifican los datos suministrados por el usuario y se realiza una consulta en la base de datos para verificar dicha información ingresada por el usuario, mediante el método POST se envía la información dicha información, cabe señalar que los datos enviados por este método tienen formato JSON. En caso que los datos ingresados sean correctos al realizar la consulta en la base de datos, no solamente se ingresa a la siguiente activity, si no también se realiza una captura de algunos datos (número de Sim-Card, nombre y cedula), que serán de utilidad en algunos activities posteriores.

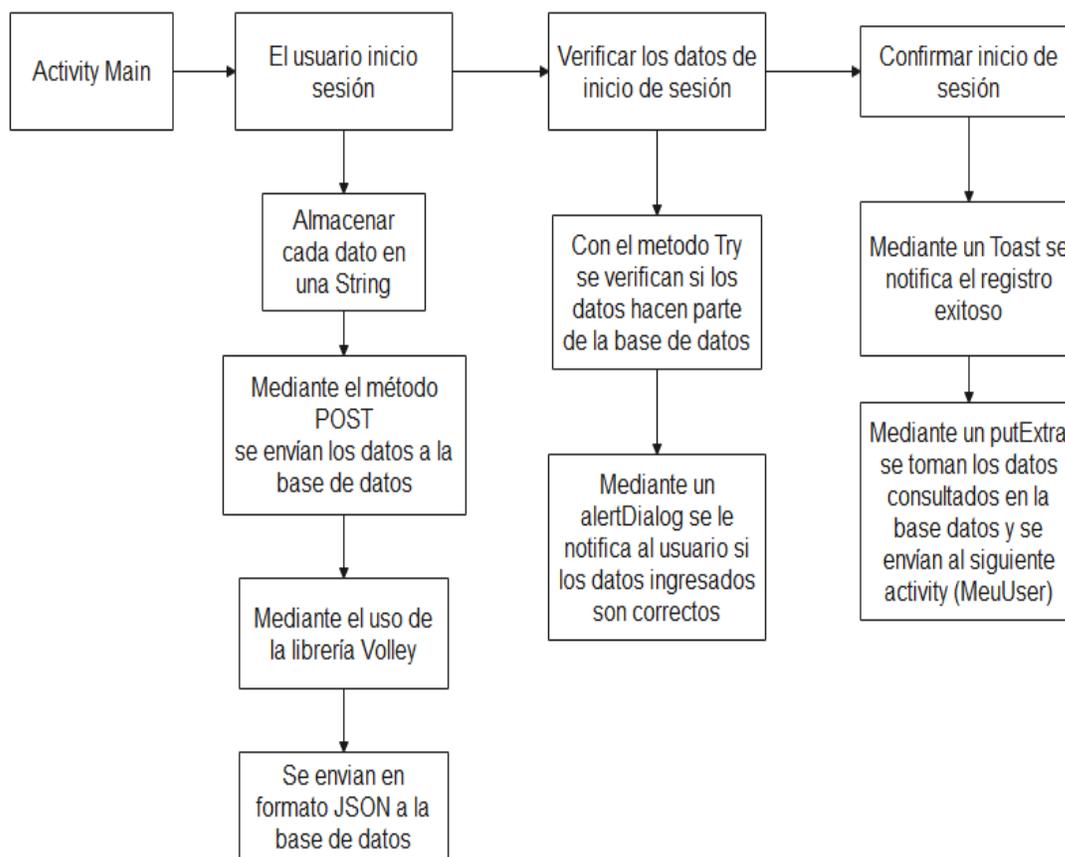


Figura 16. Diagrama de flujo de datos Activity Main. Autoría propia (2020).

Para el activity registro se puede observar cual es el flujo de datos en la **figura 17**, se comienza con almacenar o guardar los datos suministrados por el usuario (nombre, cedula, numero de Sim-Card del GPS, numero del marco de la bicicleta, email y contraseña). Estos datos se guardan en un String cada uno, después de esto se procede a su envío a la base de datos mediante el método POST y cabe señalar, que se envían en formato JSON.

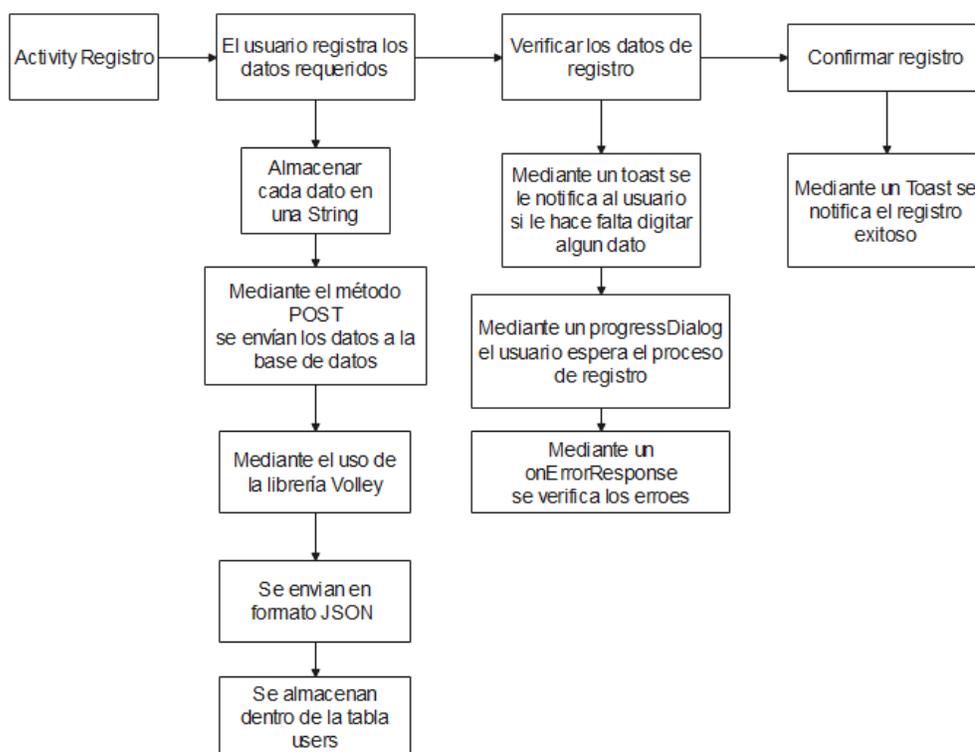


Figura 17. Diagrama de flujo de datos Activity Registro. Autoría propia (2020).

En el activity MenUser el flujo de datos que tiene se observa en la **figura 18**, en la cual se analiza el flujo de datos que se envía desde la activity anterior (Main) hasta la presente activity, se traen los datos del nombre para mediante un setText darle la bienvenida al usuario, referente al envío de datos de una activity a otra, se logra utilizando el método getIntent, lo otro a tener en cuenta es que se debe almacenar los datos en un String.

Para que la activity tenga acceso a estos datos, como se mencionó anteriormente se traen desde la anterior activity (main activity), mediante la consulta que se hace al iniciar de sesión, hay mismo se procede a traer estos datos y enviarlos a la siguiente activity (MenUser).

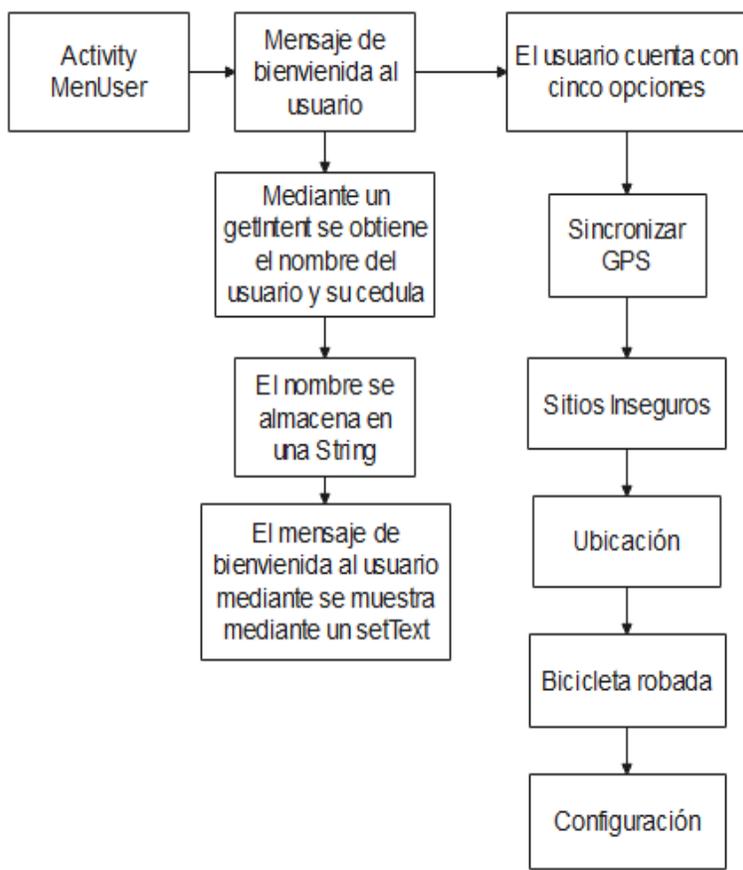


Figura 18. Diagrama de flujo de datos Activity MenUser. Autoría propia (2020).

En la **figura 19**, se observa el flujo de datos que tiene la primera opción en el menú principal la cual es sincronización GPS, para lograr enviar el mensaje de texto al GPS, se le debe indicar al GPS Tracker a cuál número de móvil debe enviar la información de su ubicación, es decir se debe sincronizar la aplicación con el GPS, en esta operación de sincronización, el flujo de información surge desde que se envía el mensaje de texto desde la aplicación, utilizando la clase SmsManager la cual se encarga de enviar el mensaje al GPS, la clase necesita el número del móvil, es decir, el número de la Sim-Card que se encuentra en el GPS y el mensaje que es el comando AT, que se consulta en el manual del GPS. Estos datos se almacenan dentro de un String cada uno y a su vez la clase utilizada se encuentra desde un método Try. Cabe resaltar que el número del GPS, se obtiene previamente al iniciar sesión, se utiliza el método getStringExtra, para obtenerlo, del activity MenUser, también se utiliza la clase sendTextMessage para enviar el mensaje.

Luego de enviar el mensaje de texto al GPS, se le confirma al usuario si el mensaje se envió con éxito o no, esto mediante un Toast, el cual se encuentra el método Try y Catch, este método Catch, solo se utiliza cuando el mensaje no se pudo enviar, la razón por la cual no se logró enviar el mensaje sería por falta de saldo en el móvil.

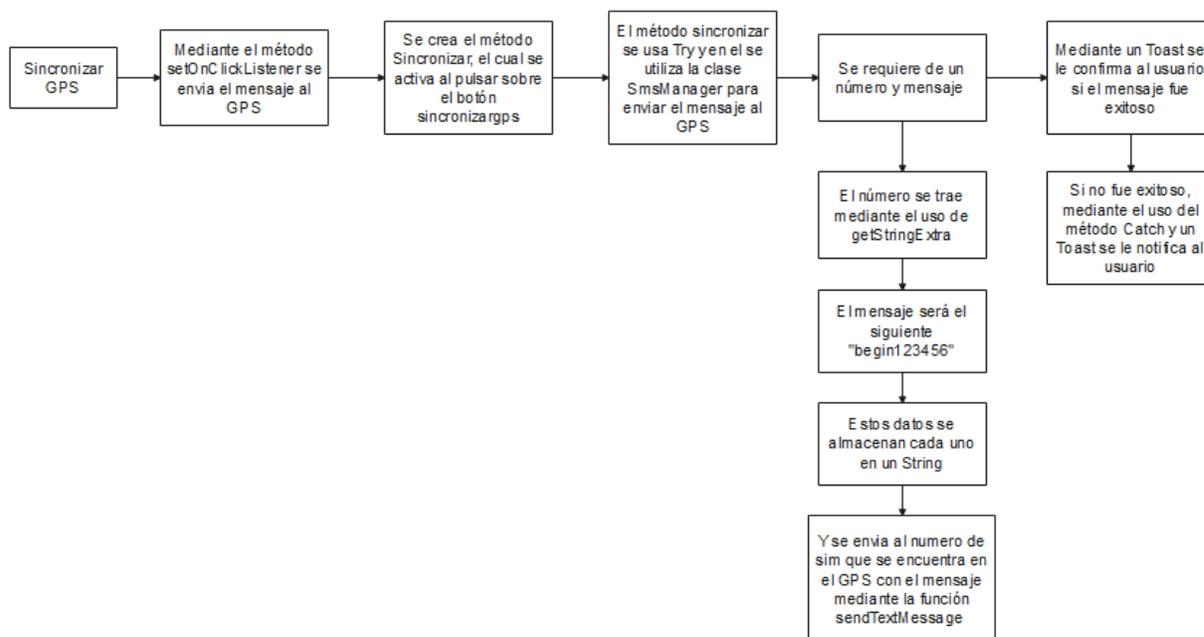


Figura 19. Diagrama de flujo de datos Activity Registro. Autoría propia (2020).

El siguiente flujo de información dentro de aplicación, sucede dentro del activity Maps, el cual le informa al usuario de ciertos lugares en los cuales es peligroso movilizarse en bicicleta, por la alta posibilidad de sufrir un robo.

Para esto se procedió a diseñar un mapa de la ciudad de Bogotá con los puntos claves de inseguridad. Para realizar dicho mapa se utiliza una herramienta de Google, la cual posee mapas de todo el planeta, es decir, Google Maps, mediante el Activity Maps requiere una credencial, la cual se obtiene desde Google Play Console, luego de obtenerla se debe proporcionar la credencial en la activity Maps, lo siguiente y como se observa en la **figura 20**, es indicar mediante coordenadas los lugares inseguros de la ciudad de Bogotá que se han investigado previamente.

Se utiliza el método `addMarker` para añadir dichas coordenadas, en este caso el flujo de datos parte, desde la elección de las coordenadas de los marcadores hasta su posterior visualización, es decir, al escoger los lugares inseguros, y luego como esos datos de coordenadas de lugares inseguros, se visualizan para en la aplicación y que el usuario los pueda conocer y los tenga en cuenta.

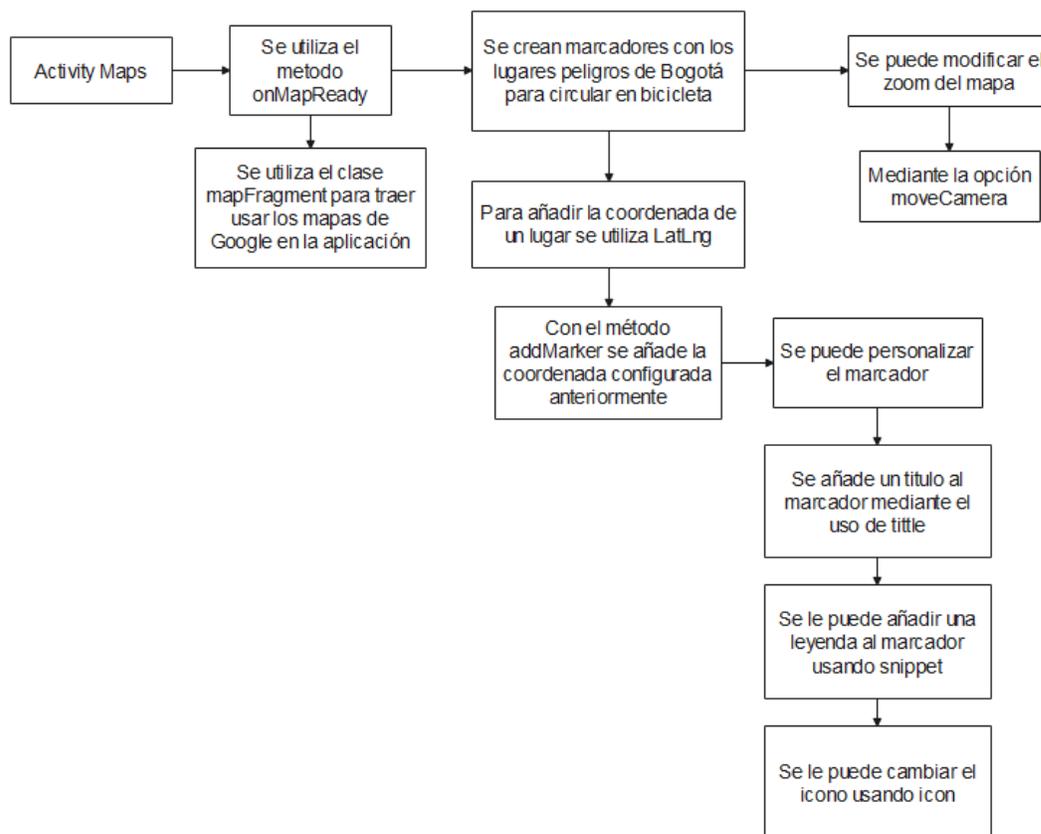


Figura 20. Diagrama de flujo de datos Activity Maps. Autoría propia (2020).

En el activity bicicleta robada, el flujo de datos se puede visualizar en la **figura 21**, en la cual se menciona la manera en la que el usuario puede realizar un monitoreo de su bicicleta en caso que sea robada, para esto se debe enviar un mensaje al GPS, el cual se envía mediante la clase `sendTextMessage`, en este mensaje se le indica al GPS que debe enviar su ubicación cada tres minutos.

Así mismo, el otro flujo de datos que surge es internamente en la aplicación, y es introducir los números de los Caís más cercanos a los lugares inseguros de la ciudad de Bogotá, para que el

usuario cuando ingrese en la opción bicicleta robada, logre identificar el número del CAI más cercano y le brinden ayuda a recuperar la bicicleta hurtada.

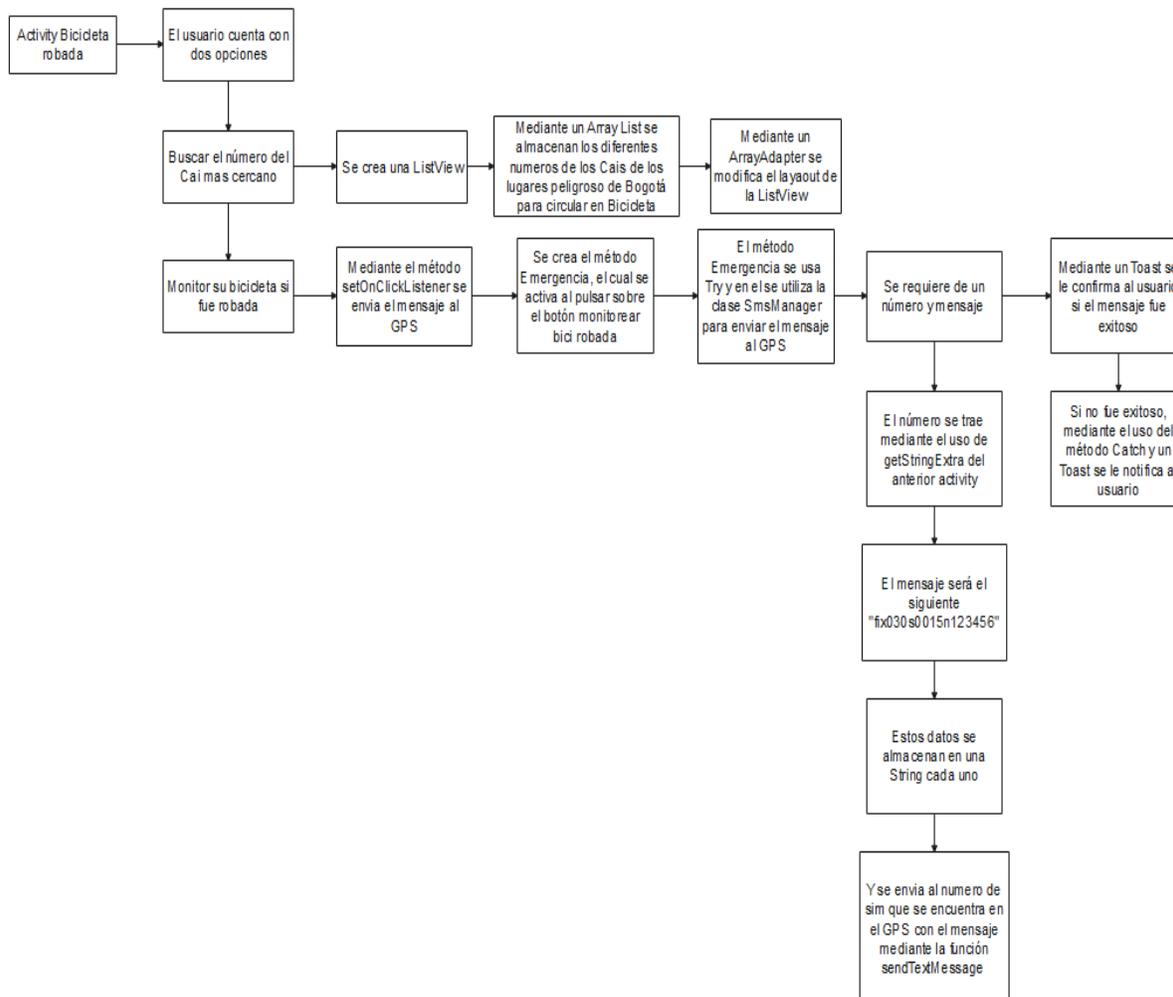


Figura 21. Diagrama de flujo de datos Activity Bicicleta robada. Autoría propia (2020).

En la tercera opción del menú principal, es donde ocurre el más alto flujo de datos puesto que, se debe primero enviar un mensaje como se mencionó anteriormente utilizando la clase `sendTextMessage`, pero también se requiere de capturar el mensaje proveniente del GPS, para esto se utiliza el método `BroadcastReceiver` para capturar el mensaje, también se utiliza la clase `getMessageBody` para esto, los datos obtenidos deben ser leídos internamente por la aplicación para luego mostrar la ubicación al usuario, los datos recibidos del GPS se almacenan en una `String`, es decir el mensaje, el cual se debe separar para sustraer la información necesaria, para esto se utiliza la clase `Split`. Solamente se necesita la URL que viene en el mensaje, la cual debe ser

separada con el método anteriormente mencionado, se guarda en un String y posteriormente se envía a un WebView para que el usuario visualice por medio de Google Maps la ubicación de su bicicleta.

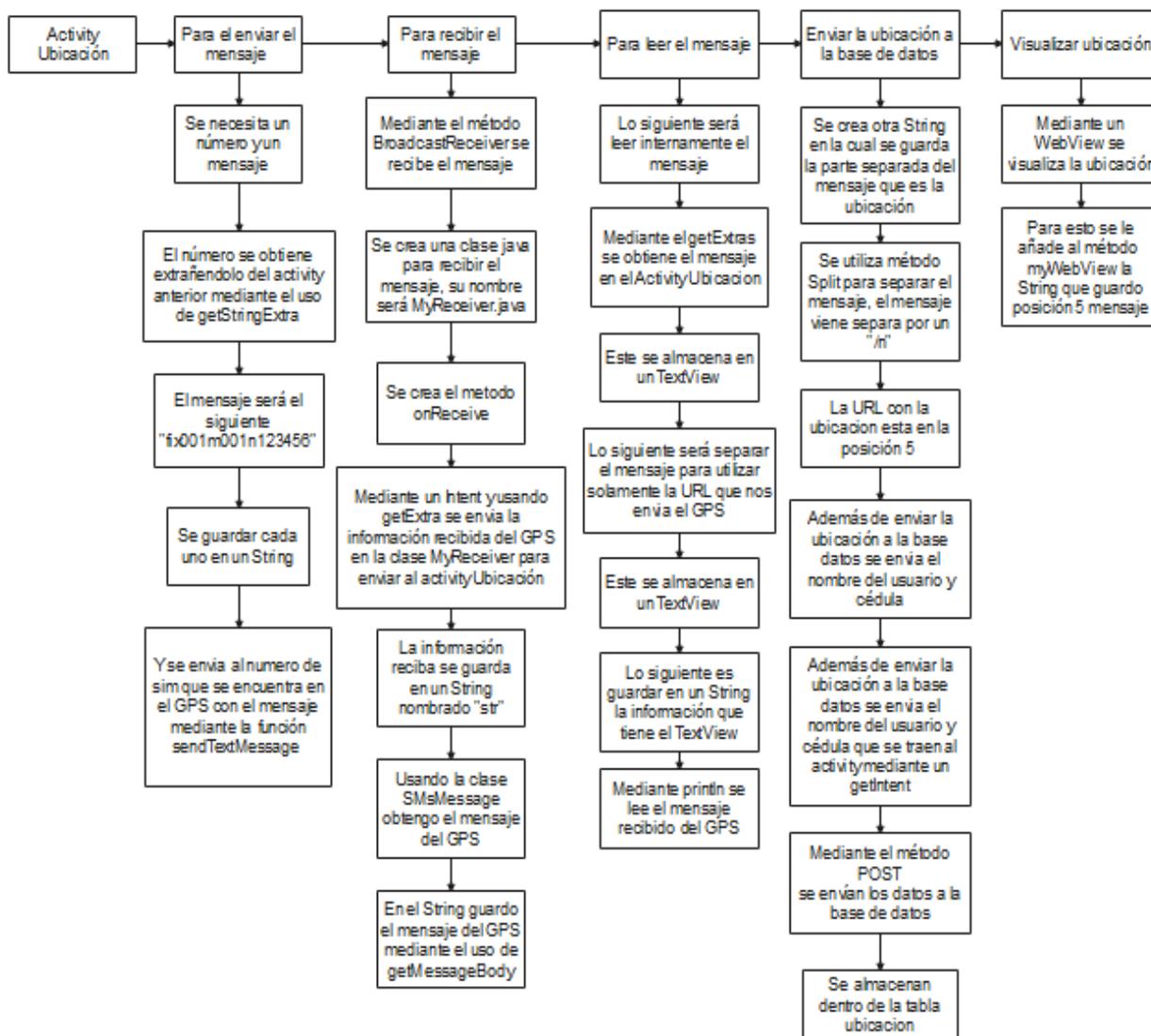


Figura 22. Diagrama de flujo de datos Activity Ubicación. Autoría propia (2020).

La última opción de la aplicación es configuración, el flujo de datos que ocurre en esta opción del menú principal se puede observar en la **figura 23**, dentro de la opción configuración tiene el usuario dos opciones en las cuales se realiza el envío de un mensaje de texto al GPS, nuevamente se utiliza el método sendTextMessage para el envío de mensajes, pero en este caso una de sus dos opciones (Añadir contacto de emergencia), requiere que el propio usuario digite una parte del

mensaje, que será almacenado con el resto del comando AT en una String y posteriormente enviarlo con el método `sendTextMessage`, la parte que requiere el método, es número de la Sim-Card que está en el GPS, esta se trae del activity `MenUser`. Por otra parte, se le informa también al usuario si el mensaje fue enviado con éxito al GPS o no, mediante un `Toast`.

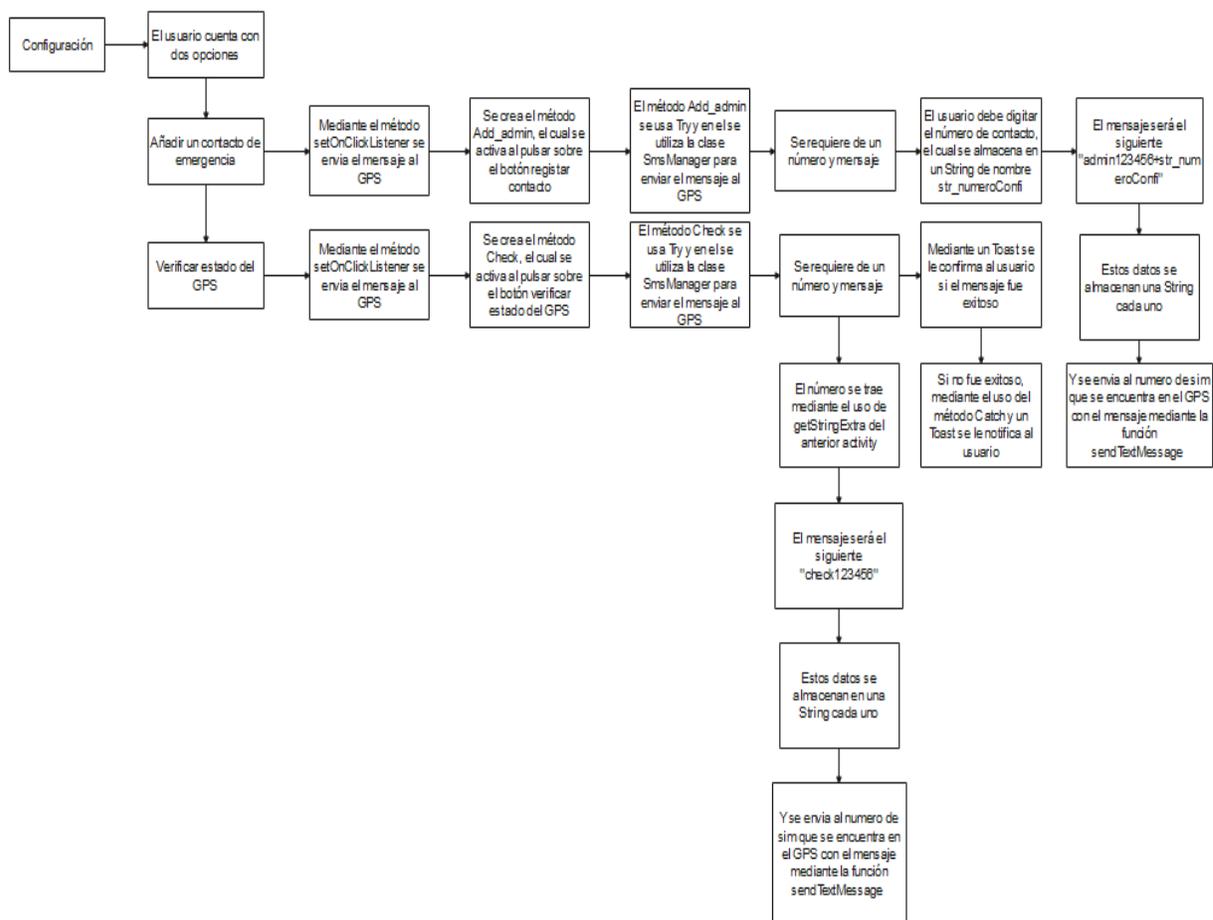


Figura 23. Diagrama de flujo de datos Activity configuración. Autoría propia (2020).

Programación de activities

En este caso, el desarrollo de la aplicación móvil comenzará con la creación del activity login y el activity registro, en una primera instancia no diseña la parte gráfica solamente se agregará los botones necesarios, los EditText que son los elementos en donde el usuario digitar los campos que requiere el registro como también los campos que requiere el login. Por otra estos EditText deberán tener un ID de referencia para poder llamarlos en el archivo JAVA y utilizarlos para guardar la información que el usuario digite. El activity registro tendrá la función de capturar los datos que

proporcione el usuario y enviarlos a la base de datos y luego pasar al siguiente activity que es login, en el cual el usuario podrá iniciar sesión. Todo esto mediante la programación de los botones, es decir, cuando el usuario pulse sobre el botón **REGISTRO** la aplicación enviará los datos a la base de datos, para poder enviar estos datos se utiliza la librería “**Volley**”, la cual se usa para el envío de peticiones HTTP desde la aplicación hacia un servidor. Básicamente la función del Volley es crear la petición, guardarla en el elemento “**cache dispatcher**” que comprueba si la respuesta de la petición actual puede ser obtenida de resultados previos guardados en caché. Si es así, entonces se pasa a **parsear** la respuesta guardada y luego se presenta al hilo principal. En caso negativo, se envía la petición a la cola de conexiones pendientes, donde reposan todas aquellas peticiones que están por ejecutarse (Revelo, 2015). Por último, la petición pasa al componente “**network dispatcher**” el cual envía la petición HTTP hacia el servidor. En la siguiente ilustración se observa el proceso que realiza Volley cuando tiene una petición. Así mismo, para el correcto funcionamiento de la librería Volley se debe incluir la dependencia de dicha librería, en el archivo “**build.gradle (:app)**”.

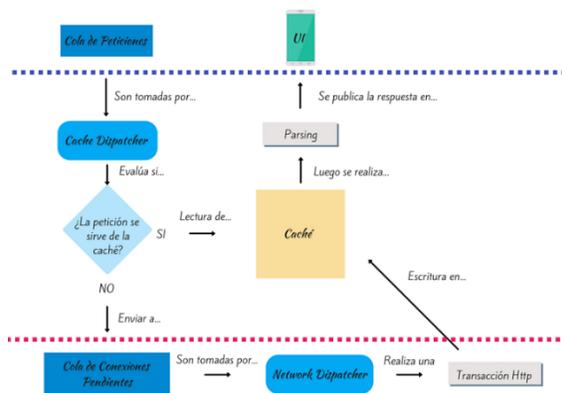


Figura 24. Funcionamiento librería Volley. (Revelo, 2015).

Esta librería se utilizará tanto como en el activity registro como en el activity login para enviar el registro del usuario, así como también para el inicio de sesión de un usuario dado que, en el activity login se verifica la información proporcionada por el usuario para su inicio de sesión lo cual, se realiza de igual manera enviado una petición con los datos ingresados y se comprueban si existen y que sean correctos en el caso los datos serán email y contraseña. De igual manera esta petición surge cuando el usuario presión el botón **INICIAR SESIÓN**, en caso que el usuario no se

encuentre registrado al momento de iniciar sesión, podrá ir al apartado de registro mediante otro botón el cual se llama **REGISTRARSE** que se encuentra en el activity login.

Otro aspecto a tener en cuenta es el de no permitir campos vacíos, es decir en caso que el usuario no digite algún campo al momento de registrarse o iniciar sesión la aplicación deberá notificar que algún campo no fue diligenciado y por tanto no podrá iniciar sesión o registrarse, como puede observar en la figura 11. También se incluye la clase “ProgressDialog” cuya función es notificar mediante un recuadro de carga que la aplicación está realizando el registro, para evitar que usuario pulse varias veces el botón de REGISTRO.



Figura 25. ProgressDialog. Autoría propia (2020).

Otro aspecto a tener en cuenta es el de no permitir campos vacíos, es decir en caso de que el usuario no digite algún campo, al momento de registrarse o iniciar sesión la aplicación deberá notificar que algún campo no fue diligenciado y por tanto no podrá iniciar sesión o registrarse.

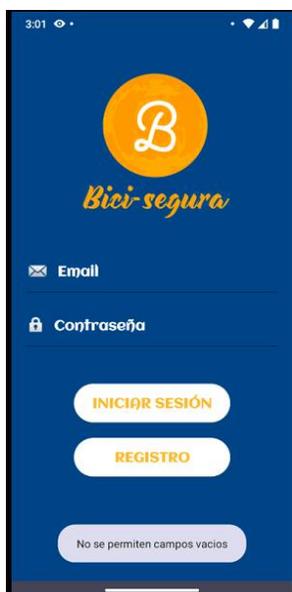


Figura 26. Comprobación campos vacíos. Autoría propia (2020).

Por otra se debió cambiar el hosting local a un web hosting, ya que presentaba inconvenientes al momento de realizar un registro o iniciar sesión, uno de estos inconvenientes es al intentar registrarse por medio de una red diferente a la que está conectada el servidor local, por ello se utiliza 000webhost, principalmente porque es gratuito y su interfaz es sencilla de interpretar, al momento de crear la base datos, así como también al momento de subir los archivos PHP que ya tenían en el servidor local, además al utilizar un servidor web permite que cualquier usuario pueda

Por otro lado, la aplicación debe tener permisos de Internet, para poder enviar la información al servidor, estos permisos se deben incluir en archivo “**AndroidManifest.XML**”. Mediante la siguiente línea de código se activan el uso de Internet en la aplicación.

```
<uses-permission android:name="android.permission.INTERNET" />
```

Así mismo se deben mencionan los restantes permisos que requiere la aplicación, los cuales son para recibir mensajes, enviar mensajes, escribir mensajes, leer mensajes y tener la ubicación del celular. Y serán las siguientes líneas de código:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<uses-permission android:name="android.permission.WRITE_SMS" />
```

```
<uses-permission android:name="android.permission.SEND_SMS" />
```

```
<uses-permission android:name="android.permission.READ_SMS" />
```

```
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

Al igual que el permiso de Internet estos permisos deben estar en el archivo “**AndroidManifest.XML**”, si estos permisos no se agregan la aplicación no funcionaría puesto que son de obligatorios, para que la aplicación tenga acceso a los mensajes de texto, escribir mensajes desde la aplicación, así como también leer los mensajes que envía el GPS.

Al siguiente activity que pasa el usuario después de iniciar sesión es el menú principal, el cual cuenta con cinco funciones, las cuales tiene acceso el usuario por medio de botones, cada botón está programado para realizar un evento, el cual comienza cuando el usuario pulse sobre alguno de estos.

Botones menú principal

El primer botón esta nombrado como “**Sincronizar GPS**”, el uso de este botón es para conectar el GPS con la aplicación, mediante él envió de un mensaje desde la aplicación hacia el GPS, para poder enviar mensajes de texto se utiliza la clase “**SmsManager**” la cual permite envió de datos, texto y mensajes, en este caso se enviará un mensaje con el texto “**begin123456**” y el número que se utilizará es el de la sim card que se encuentra en el GPS. Además, mediante un “**try**” y un “**catch**” cuya función es capturar el dato que se envía, en caso que el dato se envié correctamente por medio del “**try**” enviará un mensaje en la pantalla del dispositivo móvil, que dirá “**Mensaje enviado**” en caso contrario se usa el catch para mostrar un mensaje de error el cual será “**Mensaje no enviado, datos incorrectos.**” Estos mensajes me programan mediante un “**toast**”, que se utiliza para mostrar mensajes en una aplicación. Por otra parte, el GPS nos dará responderá con el siguiente mensaje de texto: “**begin ok!**”, al recibir este mensaje ya se podrá consultar la ubicación del GPS.

El segundo botón le permite al usuario visualizar los sitios inseguros con los que cuenta la ciudad de Bogotá, previamente nombrados en la investigación de zonas inseguras de la ciudad de Bogotá, para poder visualizar estas zonas fue necesario crear un nuevo activity de tipo “**Google Maps Activity**” para poder tener acceso a los mapas que usa Google Maps, otro aspecto a tener en cuenta es que se debe tener una credencial. Para esto se debe ingresar a la consola de desarrolladores de

Google y crear un nuevo proyecto, este nuevo proyecto necesita unos servicios, que mediante ellos se podrá visualizar los mapas o el tipo de mapa que se dese utilizar, los servicios que se utilizan en este caso son los siguientes:

Maps SDK for Android (añade mapas basados en datos de Google Maps a la aplicación Android con el SDK de Maps para Android).

Geocoding API (convierte las direcciones en coordenadas geográficas (geocodificación), que se pueden usar para colocar marcadores el mapa).

Geolocation API (encuentra una ubicación y un radio de precisión basado en la información de cosas como torres celulares y puntos de acceso WiFi que un cliente móvil puede detectar con el Geolocation API).

Luego de añadir estos servicios se procede a crear la credencial que requiere el activity de Google Maps, y copiar la credencial dentro del archivo “**Google_maps_api.xml**”. Al tener ya la credencial se procede a asignar los marcadores o los puntos más inseguros de la ciudad de Bogotá, esto realiza por medio de una coordenada la cual debe contar con latitud y longitud. En la tabla se observar el nombre del marcador y su respectiva latitud y longitud, para un total de nueve puntos críticos respecto a la seguridad de los bici-usuarios, estos marcadores se tomaron en cuenta según la investigación realizada previamente y explicada en *página 22*, en la *tabla 4* se puede ver los puntos que se toman de referencia con sus respectivas coordenadas.

Tabla 4.
Sitios inseguros

Lugar	Latitud	Longitud
Avenida Caracas	4.6052994	-74.0810054
Calle 80 con carrera 68	4.7083584	-74.1126969
Entrada a Suba (conejera)	4.7624409	-74.1218454
Park Way	4.6294205	-74.0750569
Centro de Bogotá	4.6045094	-74.0764948
Calle 134 con Autopista Norte	4.7185623	-74.0521461
Calle 170 con Autopista Norte	4.7603085	-74.066333

Humedal Juan Amarillo	4.719556	-74.0977137
Cali con calle 13	4.65758	-74.1375998
Patio Bonito	4.6345706	-74.1638007
Subida a Patios	4.675638	-74.030137
Puente Guadua	4.7270855	-74.1272127

Nota. Autoría propia, 2020.

Después de los añadir los marcadores en el activity Google Maps, se escribe de igual manera un mensaje cuando el usuario pulse sobre alguno de los nueve marcadores, el mensaje será **“Cuidado cuando circule por esta zona”**, en este activity también se debe configurar el zoom inicial del mapa que verá el usuario al iniciar esta activity, como también el punto inicial. En siguiente figura se aprecia que el punto de referencia es **“park_way”** y el zoom es de **“11”**.

```

// Calle 170
LatLng calle_170 = new LatLng( 4.7603085,  -74.066333);
mMap.addMarker(new MarkerOptions().position(calle_170).title("Calle 170 con Autopista Norte").snippet("Cuidado cuando circule en esta zona")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.alarma)));

// Humedal Juan Amarillo
LatLng juan_amarillo = new LatLng( 4.719556,  -74.0977137);
mMap.addMarker(new MarkerOptions().position(juan_amarillo).title("Humedal Juan Amarillo").snippet("Cuidado cuando circule en esta zona")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.alarma)));

// Cali con calle 13
LatLng cali_calle_13 = new LatLng( 4.65758,  -74.1375998);
mMap.addMarker(new MarkerOptions().position(cali_calle_13).title("Av. Cali con Calle 13").snippet("Cuidado cuando circule en esta zona")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.alarma)));

// Patio Bonito
LatLng patio_bonito = new LatLng( 4.6345706,  -74.1638007);
mMap.addMarker(new MarkerOptions().position(patio_bonito).title("Patio Bonito").snippet("Cuidado cuando circule en esta zona")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.alarma)));

// Subida a Patios
LatLng patios = new LatLng( 4.675638,  -74.030137);
mMap.addMarker(new MarkerOptions().position(patios).title("Subida a Patios").snippet("Cuidado cuando circule en esta zona")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.alarma)));

// Puente Guadua
LatLng guadua = new LatLng( 4.7270855,  -74.1272127);
mMap.addMarker(new MarkerOptions().position(guadua).title("Puente Guadua").snippet("Cuidado cuando circule en esta zona")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.alarma)));

mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(park_way, 11));

```

Figura 27. Activity Google Maps.java. Autoría propia (2020).

En la **figura 28** se puede observar la interfaz que tiene el Activity Maps, luego de añadir de añadir los correspondientes marcadores, además se observa la leyenda con un mensaje de precaución para alertar al usuario cuando este procesada a salir de su casa o apartamento, y este alerta al pasar por alguno de estos sitios.

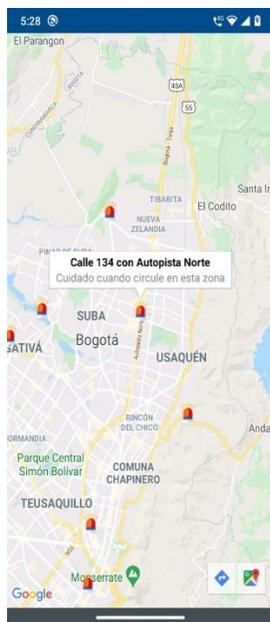


Figura 28. Activity Google Maps. Autoría propia (2020).

El tercer botón tiene la función de enviar a un siguiente activity “**Ubicación**” con el cual el usuario puede ver la ubicación de la bicicleta mediante pulsar el botón “**RASTREAR UBICACIÓN**” y este lo redirige a la aplicación Google Maps del dispositivo y podrá visualizar la ubicación de la bicicleta. Para poder realizar el visualizar la ubicación el botón “**RASTREAR UBICACIÓN**” lo que realiza es enviar el siguiente mensaje de texto "fix001m001n123456", a lo que GPS responde con la siguiente información:

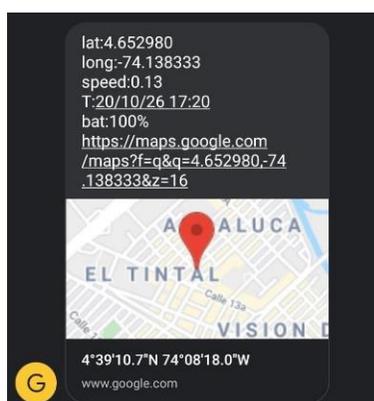


Figura 29. Respuesta GPS. Autoría propia (2020).

Esta información debe pasar por la aplicación de manera interna, es decir, el usuario debe verla en la pantalla de aplicación, para lograr capturar estos datos se utiliza la librería “**BroadcastReceiver**” que permite enviar y recibir eventos mediante **intents** (intenciones), en este caso los eventos serán el envío y recepción de mensajes que vienen del GPS, luego de recibir el mensaje se extrae la **URL** que es la última parte del mensaje, se guarda en una variable de tipo string, esta se envía a la base de datos con el nombre y cédula del usuario. Por otra parte, esa variable se le hace un llamado en **WebView**, el cual permite visualizar una **URL**, en este caso al enviar la variable al **WebView** abre la aplicación Google Maps del teléfono Android. En la siguiente ilustración se observa como por medio del **Logcat** de Android Studio se imprime la **URL**:

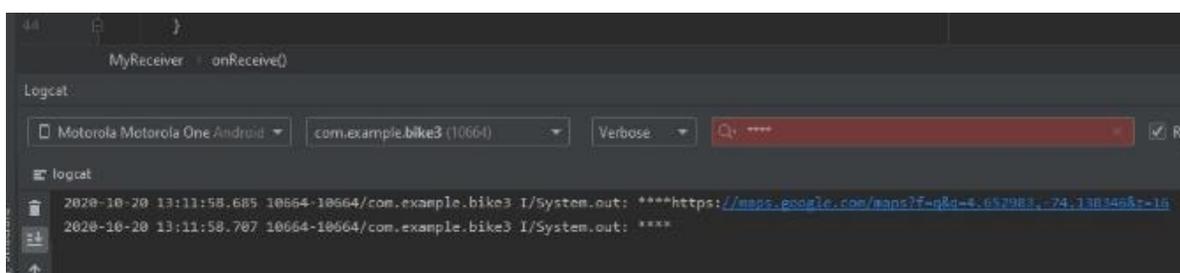


Figura 30. Logcat. Autoría propia (2020).

El cuarto botón le permite al usuario enviar un mensaje al GPS para que este le envíe cada tres minutos la ubicación, en caso de un posible robo, así mismo tiene los números de los CAÍIS más cercanos a los sitios más inseguros que sean configurado en el anterior activity (Maps Activity).



Figura 31. Activity Emergencia. Autoría propia (2020).

Se puede realizar un monitor al pulsar el botón bici robado, la aplicación confirmará el envío del mensaje mediante un toast.



Figura 32. Confirmación de monitoreo. Autoría propia (2020).

Se puede observar las figuras 34, 35 y 37 como el GPS envía la ubicación, la hora, el porcentaje de batería entre otra información, cada minuto.



Figura 33. Monitoreo 1. Autoría propia (2020).



Figura 34. Monitoreo 2. Autoría propia (2020).

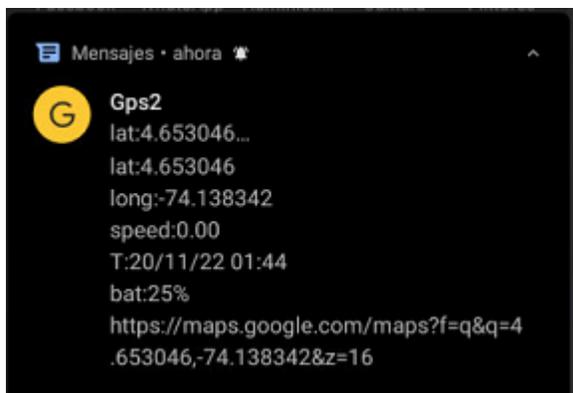


Figura 35. Monitoreo 3. Autoría propia (2020).

El quinto y último botón “**Configuraciones**” este activity donde el usuario podrá ingresar cinco contactos más lo cuales tendrán acceso a la ubicación del GPS mediante una llamada que se realice al GPS, este responderá con la ubicación, la hora actual, la batería del GPS y la latitud y longitud. En la siguiente figura se observa el diseño que tiene el activity configuración.



Figura 36. Activity Configuración. Autoría propia (2020).

Por último, quedaría mencionar que le deben pedir unos permisos al usuario que el usuario pueda utilizar la aplicación sin ningún inconveniente. Estos permisos son los de ubicación, leer mensajes, escribir mensajes y enviar mensajes. Los cuales se le piden al usuario al comenzar la

aplicación y se puede comprobar que la aplicación los tiene revisando en la información de la aplicación.

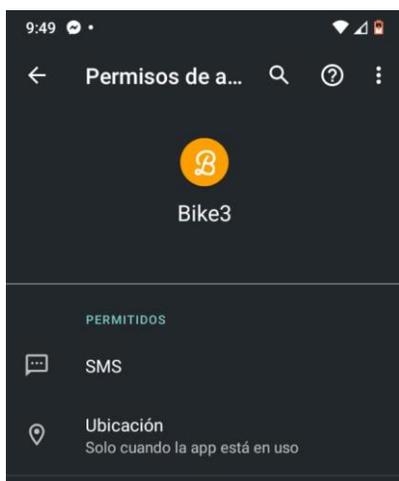


Figura 37. Permisos activos. Autoría propia (2020).

Funcionamiento del sistema de geolocalización

Después del desarrollo de la aplicación se procede a la instalación del GPS en la bicicleta para realizar las respectivas pruebas, lo primero que se debe hacer es observar que la sim card que se encuentra en el GPS tenga señal, para ello se comprueba observando que led que se halla en la parte lateral derecha este parpadeando, esto quiere decir que el GPS ya cuenta con señal y se le puede enviar mensajes para saber en la ubicación de la bicicleta, por otra parte, se observa que el GPS se encuentra por debajo del sillín de la bicicleta, intentando pasar por desapercibido y aparentando ser una luz led trasera.



Figura 38. Vista del GPS en la bicicleta. Autoría propia (2020).

El siguiente paso será iniciar sesión en la aplicación, si el usuario esta registra deberá ir a la parte de registro de la aplicación, la cual se puede acceder mediante el botón registro que ese encuentra en la parte inferior de la primera activity o ventana de aplicación que el usuario encuentra al entrar en la aplicación, y proceder a realizar el registro.



Figura 39. Registro de nuevo usuario. Autoría propia (2020).

En la base de datos se puede comprobar que el registro de este usuario fue exitoso, como se observa en la siguiente figura:

+ Opciones										
			id	nombre	email	contrasena	cedula	marcobici	simbici	
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	3	edisson	edisson@mail.com	qwerty	147852	N567	3164545570
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	7	karen	karen@mail.com	qwertyu	124846	N73737	3147016815
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	8	Andrés	andres@mail.com	1234567	31646100	k73737	3616151714
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	9	Ana	ana@mail.com	asdfgh	12121212	j7373	3195104469
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	11	Jorge Jiménez	jorge.jimenezl@uniagustiniana.edu.co	12345678	1097556051	Z575798	3503984472

Figura 40. Visualización del registro de usuario. Autoría propia (2020).

Lo siguiente será ingresar que el usuario ingrese su correo y contraseña.



Figura 41. Inicio de sesión. Autoría propia (2020).

Al iniciar sesión el usuario tendrá las funciones que sean programado para el uso del GPS en la bicicleta, además en la siguiente figura, la aplicación en la parte superior le da la bienvenida al usuario.



Figura 42. Menú principal. Autoría propia (2020).

Para ubicar la bicicleta el usuario debe ingresar a la tercera opción del menú principal, la cual es **Ubicación**, la cual se visualiza de la siguiente manera.

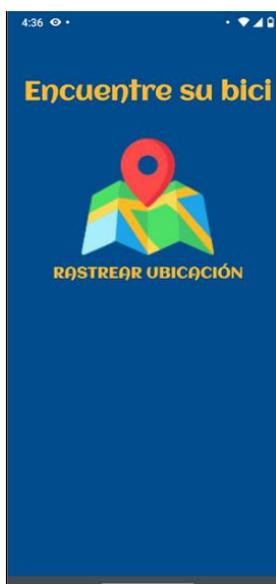


Figura 43. Opción ubicación. Autoría propia (2020).

Lo siguiente que tendrá que hacer el usuario será pulsa sobre el botón **RASTREAR UBICACIÓN**, y en cuestión de segundos el GPS recibe el mensaje y por medio de Google Maps el usuario obtendrá la ubicación de su bicicleta.

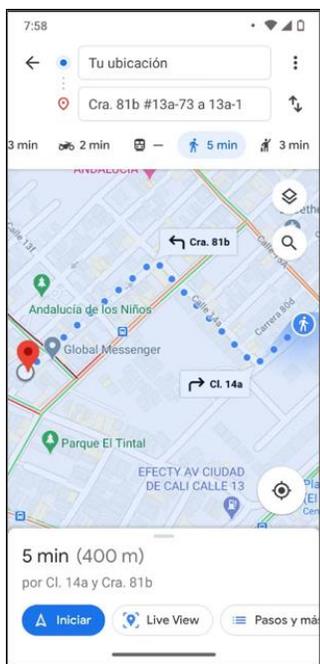


Figura 44. Ubicación de la bicicleta. Autoría propia (2020).

Como se puede observar en la figura anterior la bicicleta será el punto de color rojo que se encuentra a 400 metros de distancia de la ubicación donde se encuentra el usuario.

Por otra en se puede confirmar que el envío de dicha ubicación fue éxito, si se revisa la base de datos donde se almacena la ubicación del usuario, la cual se envía a la aplicación y permite que se visualice por medio de la aplicación de Google Maps del teléfono.

<input type="checkbox"/>		Editar		Copiar		Borrar	4	edisson	147852	https://maps.google.com/maps?f=q&q=4.652923,-74.13...
<input type="checkbox"/>		Editar		Copiar		Borrar	5	edisson	147852	https://maps.google.com/maps?f=q&q=4.652928,-74.13...
<input type="checkbox"/>		Editar		Copiar		Borrar	6	edisson	147852	https://maps.google.com/maps?f=q&q=4.652923,-74.13...
<input type="checkbox"/>		Editar		Copiar		Borrar	7	edisson	147852	https://maps.google.com/maps?f=q&q=4.652923,-74.13...
<input type="checkbox"/>		Editar		Copiar		Borrar	8	Jorge Jiménez	1097556051	https://maps.google.com/maps?f=q&q=4.653048,-74.13...

Figura 45. Verificación de envío de ubicación. Autoría propia (2020).

Esta ubicación que se almacena en la base de datos se puede verificar que sea la misma, copiando la URL que se almacenó en la base de datos y abriéndola en cualquier navegador de un computador.

Otra de las funciones con las que cuenta la aplicación es registrar cinco números de teléfono los cuales pueden acceder a la ubicación del GPS mediante una llamada, la aplicación le confirma al usuario que se le envió al GPS, así mismo el GPS responde con un mensaje al usuario.



Figura 47. Registro de contacto. Autoría propia (2020).



Figura 48. Confirmación. Autoría propia (2020).

En la parte de configuración se agregó una opción más para el usuario, en la cual puede verificar o chequear el estado del GPS respecto al porcentaje de batería, el estado de la señal GSM, GPS y GPRS, además de la APN y la dirección IP. Como se observa en las figuras 50,51 y 52, en la cual se aprecia el nuevo botón en el activity configuración (verificación estado del GPS), el toast de mensaje enviado y mensaje de respuesta del GPS.



Figura 49. Activity configuración. Autoría propia (2020).



Figura 50. Toast de confirmación del mensaje enviado. Autoría propia (2020).



Figura 51. Respuesta del GPS. Autoría propia (2020).

Tiempos de retardo del sistema de geolocalización

Después de terminado el sistema de geolocalización se procede a realizar una serie de pruebas respecto la latencia de la aplicación y el GPS Tracker, es decir al momento en que la aplicación recibe un mensaje proveniente del GPS, además de cuánto tiempo le toma al GPS encontrar la señal GSM. Para esto se realizó diez pruebas para verificar cuánto tiempo le toma al sistema estar en completo funcionamiento.

La primera prueba fue el tiempo de encendido del GPS, el cual tiene un promedio de 14,08 segundos, desde que se pulsa el botón de encendido hasta que el GPS tiene señal GSM. En la tabla 5 se puede observar los tiempos que le toma la GPS encender y su correspondiente promedio. Se puede observar que el tiempo de encontrar señal es relativamente bajo, este puede cambiar dependiendo si el GPS Tracker se encuentra en un lugar abierto o un lugar cerrado como puede ser un sótano o un parqueadero en estos casos es posible que el GPS no encuentre señal o le tome 1 minuto en encontrarla.

Tabla 5.
Tiempo de encendido GPS Tracker

Tiempo de encendido GPS (segundos)
15,76
15,18
7,97
14,21
14,24
14,69

	15,18
	14,29
	15,41
	13,89
Promedio	14,08

Nota. Autoría propia, 2020.

La segunda prueba realizada fue, saber cuánto tiempo le toma al GPS responder a la aplicación móvil cuando se envía el mensaje de sincronización, cabe resaltar que para esta prueba se realizó en varios móviles Android, dado la sincronización solamente debe realizarse una vez. El promedio de tiempo de respuesta fue 7,74 segundos, en general la mayoría de veces el GPS respondió en rápido, solamente un dato de prueba que supero casi los 10 segundos, este tiempo es obtuvo cuando la bicicleta se encontraba dentro de la casa, como se resalta si el GPS Tracker no está en un espacio libre de edificios o se encuentra dentro de una casa, puede demorarse un poco más, ya que cuenta con el campo de visión libre entre los satélites y el lugar donde está el GPS.

Tabla 6.

Sincronización GPS

	Tiempo de sincronización (segundos)
	8,50
	6,43
	7,14
	6,89
	10,85
	7,56
	7,87
	7,90
	7,50
	6,79
Promedio	7,74

Nota. Autoría propia, 2020.

Otra prueba realizada el tiempo de retardo al solicitar la ubicación de la bicicleta desde la aplicación móvil, que se realiza mediante el botón rastrear ubicación, en general el tiempo de retardo estuvo entre 7 a 8 segundos, pero en dos puntuales el tiempo de respuesta subió a 10 y a 18 segundos, esto ocurrió porque se realizó la prueba cuando la bicicleta se encontraba en un sitio cerrado, puntualmente estaba en un sótano. El tiempo promedio fue 9,14, los demás datos de prueba se pueden consultar en la tabla 7.

También cabe señalar que en el momento que la aplicación recibe el mensaje y abre la aplicación de Google Maps debe tomarse en cuenta. Además, si ocurriera un robo cada segundo es fundamental al momento de intentar recupera la bicicleta por esto, esta fue la última prueba que se realizó para determinar qué tan eficaz es el sistema de geolocalización. El promedio de respuesta en abrir la aplicación de Google Maps fue de 2,51 segundos, se debe tener en cuenta la velocidad del móvil y la conexión a Internet con la que cuenta el móvil.

Tabla 7.

Tiempo de retardo de ubicación

	Latencia de ubicación (segundos)
	18,30
	10,11
	7,33
	8,68
	7,76
	7,77
	7,87
	7,90
	7,50
	8,15
Promedio	9,14

Nota. Autoría propia, 2020.

Tabla 8.

Tiempo de carga de Google Maps

	Tiempo de carga de Google Maps (segundos)
	4,40
	0,91
	4,10
	1,01
	1,84
	3,24
	1,50
	2,07
	2,36
	3,68
Promedio	2,51

Nota. Autoría propia, 2020.

Comportamiento de la batería

Respecto a la batería del GPS Tracker, aunque el manual especifica que tienen una duración máxima de 80 horas, aproximadamente 3 días y 8 horas, esto no se cumple si, se mantiene en constante uso, es decir realizando peticiones de ubicación, chequeo de estado de la batería o monitoreo constante de la ubicación del sistema de geolocalización la batería no dura más de días, la ventaja con la cual cuenta el GPS Tracker son dos, la primera incluye dos baterías, la duración de carga de una batería es de 2 horas con 40 minutos aproximadamente, y la segunda es que si el GPS Tracker no se encuentra en modo activo, el automáticamente se apagara y pues esto trae una ventaja respecto al ahorro de la batería, pero eso si debe tenerse en cuenta que si no se le realiza una petición de ubicación o cualquier otra el GPS Tracker, este se apagara alrededor de 20 a 15 minutos es decir no enviará información acerca de la ubicación. Este dato se debe tener en cuenta cuando se circule por la ciudad de Bogotá, se recomienda, avisar a un contacto de emergencia previamente que este enviado mensajes de ubicación la GPS, no solo para vigilar la ubicación de la bicicleta y del usuario si no para evitar un posible apagado automático por parte del GPS Tracker.

Conclusiones

Dentro del análisis expuesto sobre de los diferentes dispositivos de GPS del mercado se evidencia que la elección del GPS Tracker fue la correcta debido a que utiliza la tecnología GPRS, es decir mediante el uso de la red móvil envía la información para realizar peticiones a la unidad, mientras que las otras tecnologías Bluetooth, Wi-Fi o LBS presentan una deficiencia de cobertura en caso de presentarse un hurto.

Con la realización de este proyecto se logró con éxito el proceso de comunicación de una unidad de GPS enviando y recibiendo los datos de la unidad en tiempo real.

Mediante el uso de servicios gratuitos de 000webhost, es viable implementar una base de datos para almacenar, registrar y monitorear las diferentes variables obtenidas de la unidad de GPS y de la misma aplicación móvil.

Se demostró que es viable la transmisión de los datos obtenidos de la unidad de GPS a un dispositivo móvil Android, permitiendo almacenar los datos para posteriormente ser procesados.

Este proyecto evidencia que es posible diseñar una red de monitoreo con hardware de bajo costo y software libre, logrando obtener parámetros de ubicación, hora, tiempo de reporte, nivel de batería y velocidad en tiempo real, permitiendo visualizarlos mediante el uso de una aplicación Android en caso de hurto.

La unidad de GPS Tracker al utilizar una red móvil para el envío de la ubicación puede llegar a presentar inconvenientes, cuando la unidad de GPS se encuentre en sitios subterráneos dado que, no obtendría cobertura móvil para recibir la petición enviada desde la aplicación móvil y así mismo no podría responder a la petición enviada.

Recomendaciones

El GPS que se compró permite el envío de la ubicación hacia la aplicación, pero el tamaño del dispositivo GPS no es el ideal para instalar en las bicicletas, puesto que su tamaño es relativamente grande lo cual no le permitiría pasar desapercibido ante un posible hurto y los ladrones podrían encontrar y deshacerse de él, por tanto, se recomienda comprar un GPS de menor tamaño para evitar esto.

Por otra parte, hay que tener en cuenta las condiciones meteorológicas para el correcto funcionamiento del GPS, dado que si se desea conocer la ubicación de la bicicleta el GPS y el cielo se encuentra nublado o lloviendo el GPS no funcionará correctamente, dado que no tiene el punto de mira despejado y en algunas ocasiones no hay respuesta por parte del GPS.

Hay que tener en cuenta también, que al realizar alta cantidad de peticiones la empresa prestadora del servicio móvil, podría bloquear momentáneamente el envío en mensajes de texto en la sim card que se encuentra en GPS, se deberá llamar a la empresa y solicitar el desbloqueo del envío de mensajes. Y que esta sim card este al nombre de la persona que envía las peticiones desde el teléfono, si no es así, tendrá que ubicar a la persona que pertenezca esta sim card para que ella proceda a realizar dicho desbloqueo.

Referencias

- Apasionados. (2015). *Apasionados*. Obtenido de <https://apasionados.es/blog/nodejs-4430/>
- Appsmovilescavucm. (2020). *Appsmovilescavucm*. Obtenido de <https://appsmovilescavucm.wordpress.com/que-es-una-app-movil/>
- Aprendiendo Arduino. (2019). *Aprendiendo Arduino*. Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/2016/12/11/que-es-arduino-2/>
- Arduino.cl. (2019). *Arduino.cl*. Obtenido de Arduino.cl: <https://arduino.cl/que-es-arduino/>
- AutosOnline. (2020). *AutosOnline*. Obtenido de <https://www.autosonline.cl/tips-y-consejos/equipamiento-automotriz/un-completo-sistema-de-seguridad.html>
- Ciampagna, J. M. (2013). *El blog del profe José*. Obtenido de <https://elprofejose.com/2013/03/04/lbs-servicios-basados-en-ubicacion/>
- Comisión de Regulación de Comunicaciones. (2020). *Comisión de Regulación de Comunicaciones*. Obtenido de https://www.crcm.gov.co/recursos_user/Normatividad/Resoluciones_Generales/00003501.pdf
- El Localizador GPS. (2020). *El Localizador GPS*. Obtenido de <https://ellocalizadorgps.com/>
- Ferreño, E. (2020). *El androide libre*. Obtenido de <https://elandroidelibre.lespanol.com/2020/01/android-studio-que-es-y-para-que-se-utiliza.html>
- fundación Aquae. (2019). *Aquae Fundación*. Obtenido de Aquae Fundación: <https://www.fundacionaquae.org/consejos-del-agua/cinco-ventajas-utilizar-la-bicicleta-medio-transporte/>
- Godfrey, D., & Song, M.-H. (2016). Safe Bike : Secure your Bicycle with this smart Arduino based GPS device. *Computer Science*, 11.
- Gómez, J. D. (2019). *La FM*. Obtenido de La FM: <https://www.lafm.com.co/bogota/robo-de-bicicletas-en-bogota-han-aumentado-un-400-en-tres-anos>
- GPS.GOV. (2019). *GPS.GOV*. Obtenido de GPS.GOV: <https://www.gps.gov/systems/gps/spanish.php>
- Hosting Red. (2020). *Hosting Red*. Obtenido de <https://www.hostingred.com/hosting/informacion-hosting/>

- Hostinger tutoriales. (2020). *Hostinger tutoriales*. Obtenido de <https://www.hostinger.co/tutoriales/que-es-un-hosting>
- IAC. (2018). Obtenido de <https://www.iac.com.co/que-es-iot/>
- IEBS. (2010). *itop*. Obtenido de <https://www.itop.es/blog/item/que-es-iot-internet-of-things.html>
- Maikii. (20 de Noviembre de 2020). *Maikii*. Obtenido de [https://www.maikii.com/es/faq/que-es-el-tracker-bluetooth-como-funciona#:~:text=Una%20vez%20que%20smartphone%20y,\(por%20ejemplo%20las%20llaves\)](https://www.maikii.com/es/faq/que-es-el-tracker-bluetooth-como-funciona#:~:text=Una%20vez%20que%20smartphone%20y,(por%20ejemplo%20las%20llaves)).
- Ministerio de Tecnologías de la Información y las Comunicaciones. (2020). *Ministerio de Tecnologías de la Información y las Comunicaciones*. Obtenido de <https://www.mintic.gov.co/portal/inicio/3707:Ley-1341-de-2009>
- Murillo, Y. F. (2016). *Diseño de un Sistema de Seguridad y Monitoreo Satelital para una Bicicleta por Medio de Redes GSM/GPRS*. Obtenido de Repositorio Institucional Universidad Distrital: <http://hdl.handle.net/11349/5969>
- Neoattack. (2019). *Neoattack*. Obtenido de Neoattack: <https://neoattack.com/neowiki/arquitectura-web/>
- P. Qiu, X. Liu, S. Wen, Y. Zhang, K. N. Winfree, & C. Ho. (2018). *The Development of an IoT Instrumented Bike: for Assessment of Road and Bike Trail Conditions*. Obtenido de IEEEExplore: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8538268&isnumber=8538019>
- Patel, P., Rauniyar, S. K., Singh, T., Dwivedi, B., & Tripathi, P. H. (2018). Arduino Based Child Tracking System Using GPS and GSM. *International Research Journal of Engineering and Technology (IRJET)*, 4.
- Penagos, J. G. (2018). *El Espectador*. Obtenido de El Espectador: <https://www.elespectador.com/noticias/bogota/radiografia-del-hurto-de-bicicletas-en-bogota-articulo-813609>
- Penagos, J. G. (2018). *El Espectador*. Obtenido de <https://www.elespectador.com/noticias/bogota/radiografia-del-hurto-de-bicicletas-en-bogota-articulo-813609>

- Redacción Bogotá. (2018). *El Espectador*. Obtenido de El Espectador: <https://www.elespectador.com/noticias/bogota/el-49-de-las-ciclorutas-estan-en-mal-estado-personeria-de-bogota-articulo-750954>
- Revelo, J. (2015). *Hersomoa Programación*. Obtenido de <http://www.hermosaprogramacion.com/2015/02/android-volley-peticiones-http/>
- Sampieri, R. H. (2014). *Metodología de la investigación sexta edición*. Ciudad de México: McGraw Hill.
- Samsung. (2018). Obtenido de <https://www.samsung.com/latin/support/mobile-devices/what-is-gps/>
- Tailit. (2020). *Tailit*. Obtenido de <https://www.tailit.com/es/rastreador-gps/>
- Torres, E. (2017). Control de Asistencia y Monitoreo GPS para Trabajadores y Vehículos de la Empresa Sistemas de Automatización y Control Radicada en la Ciudad de Ibarra. *Political Science*, 12.
- Velasco, C. (2018). *lawebera.es*. Obtenido de <https://www.lawebera.es/posicionamiento/que-es-arquitectura-web-efecto-en-posicionamiento.php>
- Villagómez, C. (2017). *CCM*. Obtenido de <https://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>
- Yun-Yao Chen, & Yueh-Yun Wang. (2010). *zDesign and implementation of a web services-based system for bike positioning*. Obtenido de IEEEExplore: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5533316&isnumber=5533315>
- Zepeda, R. (2015). *EL INFORMÁTICO.COM* . Obtenido de EL INFORMÁTICO.COM : <http://blogdelinformatico-reizer.blogspot.com/2015/11/que-es-xampp.html>