



TESIS PROFESIONAL

MONITOREO Y VISUALIZACIÓN DEL TRÁFICO VIAL MEDIANTE DISPOSITIVOS CELULARES



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



INSTITUTO
TECNOLÓGICO
NACIONAL DE MÉXICO



VERACRUZ
GOBIERNO
DEL ESTADO



SEV
Secretaría
de Educación

SEMSys
Subsecretaría de Educación
Media Superior y Superior



DET
Departamento de Educación
Tecnológica del Estado
de Veracruz



TECNOLÓGICO
NACIONAL DE MÉXICO®



MONITOREO Y VISUALIZACIÓN DEL TRÁFICO VIAL MEDIANTE DISPOSITIVOS CELULARES

INSTITUTO TECNOLÓGICO SUPERIOR DE MARTÍNEZ DE LA
TORRE

TESIS

PRESENTADA COMO REQUISITO PARA OBTENER EL
TÍTULO DE:
LICENCIADA EN INGENIERÍA EN SISTEMAS
COMPUTACIONALES

Presenta:
TANIA AIDÉ VÁZQUEZ
BELTRÁN

Asesores:
M.C.A. SOFÍA ISABEL
FERNÁNDEZ GREGORIO

DR. LUIS ALBERTO
MORALES ROSALES

Martínez de la Torre, Veracruz [AGOSTO 2021]

FICHA TÉCNICA		
Estudiante		
Apellido paterno	Apellido materno	Nombre (s)
Vázquez Beltrán Tania Aidé		
No. de control: 160i0192		
Carrera: Ingeniería en sistemas computacionales		
Correo: 160i0192@tecmartinez.edu.mx		
Asesor(es) y/o colaboradores ITSMT		
M.C.A. Sofía Isabel Fernández Gregorio		
DR. Luis Alberto Morales Rosales		
Datos del lugar donde se desarrolló el proyecto		
Nombre o razón social		
Universidad Michoacana de San Nicolás de Hidalgo		
Dirección (calle, número, colonia, ciudad, código postal)		
Av. General Francisco J. Mejía s/n. Felicitas del Rio. Morelia Michoacán.		
Asesor externo: Dr. Luis Alberto Morales Rosales		
Departamento: Maestría de infraestructura del Transporte en la Rama de las Vías Terrestres.		
Cargo: Catedrático Conacyt.		
Correo: iamorales@conacyt.mx		
Teléfono y extensión: +52 (443) 3223500 (4346)		

FORMATO DE LIBERACIÓN DEL PROYECTO PARA TITULACIÓN INTEGRAL

	Liberación del Proyecto para Titulación Integral	Página 1 de 1
---	---	---------------

Martínez de la Torre, Ver. a 29 de Junio de 2021

C. DR FROYLAN ROSALES MARTÍNEZ
JEFE(A) DE DEPTO. SERVICIO SOCIAL Y
RESIDENCIAS PROFESIONALES
INSTITUTO TECNOLÓGICO SUPERIOR DE MARTÍNEZ DE LA TORRE
P R E S E N T E

Por este medio informo que ha sido liberado el siguiente proyecto para la Titulación Integral:

Nombre del estudiante:	Tania Aidé Vázquez Beltrán		
Carrera:	Ingeniería en Sistemas Computacionales	No. De control:	16010192
Nombre Proyecto:	Monitoreo y visualización del tráfico vial mediante dispositivos celulares.		
Opción Titulación:	Tesis		

Agradezco su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados(as).

A T E N T A M E N T E


M.C. JULIO CÉSAR MACÍAS GÓMEZ
JEFE DE DIVISIÓN DE CARRERA DE
INGENIERÍA EN SISTEMAS
COMPUTACIONALES



JEFATURA DE CARRERA
ING. SISTEMAS COMPUTACIONALES


M.C.A. Sofia Isabel
Fernández Gregorio
ASESOR


M.R. y S.L. Angel Salas
Martínez
REVISOR


M.S.E. Hugo Lucas
Alvarado
REVISOR

Cop. Estudiante.
Cop. Archivo.

F-11-07
Rev. 1

**CARTA DE AUTORIZACIÓN DE ENTREGA DE TESIS EN
SOPORTE DIGITAL**



Carta de Autorización de Entrega de Tesis en
Soporte Digital

Página 1 de 1

No. de Oficio: DET/ITSMT/DA/ISC/053/2021

ASUNTO: Autorización de entrega

Martínez de la Torre, Ver., a 14 de Julio de 2021.

C. TANIA AIDÉ VÁZQUEZ BELTRÁN
No DE CONTROL 16010192
EGRESADO (A) DE LA CARRERA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
P R E S E N T E

Por medio de la presente hago constar que ha cumplido satisfactoriamente con lo estipulado por el Lineamiento para la Titulación Integral.

Por tal motivo se autoriza la entrega de la Tesis en soporte digital titulada:

Monitoreo y visualización del tráfico vial mediante dispositivos celulares

Dándose un plazo máximo de 30 días naturales a partir de la fecha de la expedición de la presente para realizar la solicitud del Acta de Recepción para la obtención del Título Profesional.

ATENTAMENTE


M.C.C. y T. Keila Elena Ocaña Drouaillet
Presidenta de Academia de Ingeniería
en Sistemas Computacionales

C.p. División de Estudios Profesionales
C.p. Archivo


Ing. Hernet Bautista Ruiz
Jefe de Carrera de Ingeniería
en Sistemas Computacionales



JEFATURA DE CARRERA
ING. SISTEMAS COMPUTACIONALES

F-11-09
Rev. 1

DEDICATORIA

A mi familia por la paciencia que han tenido conmigo, a mis hijos y esposo, a mis padres Constanca y Atanasio, mis hermanos Edgar y Estefanía.

A mi esposo Eduardo por su paciencia y comprensión, y a mis hijos Leonardo Daniel y Héctor Eduardo por emocionarse y querer ver cómo funcionaba la aplicación.

Gracias a todos aquellos que me han impulsado a seguir adelante a pesar de ya no tener energía todos ustedes han sido mi energía.

A mis compañeros porque unidos somos más fuertes aunque cada quien luche su propia batalla. A mis maestros a aquellos que con palabras de aliento nos impulsaron a seguir adelante y a los que nos hicieron ver que nos hacía falta más compromiso y coraje para salir adelante, a los que nos mostraron el camino y nos soltaron de la mano para aprender a buscar la excelencia por cuenta propia. A todos ellos y a todos ustedes Gracias.

AGRADECIMIENTOS

Agradecer de primera mano a mis asesores la M.C.A. Sofía Isabel Fernández Gregorio Y al Dr. Luis Alberto Morales Rosales ya que sin ellos no habría tenido la oportunidad de conocer y participar en este proyecto.

A mis compañeras Seydi Sarai y Claudia Lizeth por su compañía y apoyo durante este periodo y por darnos ánimos cuando nos sentíamos perdidas.

Al Instituto Tecnológico Superior de Martínez de la Torre y a la Universidad Michoacana de San Nicolás de Hidalgo por la colaboración que permite a estudiantes como yo ser parte de proyectos de gran impacto.

RESUMEN

En el mundo de la conducción existen normas y reglas para conducir un vehículo. La mayoría de conductores desarrolla su propio estilo de conducción. Cada conductor tiene diferentes hábitos de manejo y esto se refleja en el comportamiento del tránsito urbano. Las acciones de los conductores al manejar su vehículo pueden generar situaciones en el tráfico; como lo son congestionamiento vial, accidentes causados por conducción acelerada entre otros. Obtener los datos del comportamiento de conducción por medio de un dispositivo celular contribuiría al análisis del tráfico de vial en zonas urbanas. Por lo que este trabajo de investigación y desarrollo se basa en visualizar y monitorear el comportamiento de conducción de los vehículos en movimiento mediante el uso de un dispositivo móvil, el cual es celular. En el que se desarrolla una aplicación móvil que es capaz de detectar la ubicación del vehículo por medio de asignación de servicios de localización, muestre el recorrido que va realizando el vehículo en movimiento y la trayectoria que realiza, mientras lo muestra en un mapa de Google Maps, el cual se implementa en la aplicación junto con sus credenciales API's y servicios de GoogleMaps, además de detectar la velocidad a la cual va dicho vehículo. Lo cual se pretende lograr por medio del acelerómetro que incluye el teléfono inteligente. Todos los datos que se obtienen se almacenaran en una base de tipo local SQLite para su análisis posterior. La aplicación de monitoreo y visualización se implementó en Android Studio debido a que el sistema operativo Android están posicionado como uno de los más comunes sistemas operativos en celulares actualmente.

Palabras Clave: *aplicación, monitoreo, GoogleMaps, trafico vial, comportamiento de conducción, acelerómetro.*

ABSTRACT

In the world of driving there are rules and regulations for driving a vehicle. Most drivers develop their own driving style. Each driver has different driving habits and this is reflected in the behavior of urban traffic. The actions of the drivers when driving their vehicle can generate situations in the traffic; such as traffic congestion, accidents caused by accelerated driving among others. Obtaining driving behavior data through a cellular device would contribute to the analysis of road traffic in urban areas. So this research and development work is based on visualizing and monitoring the driving behavior of moving vehicles through the use of a mobile device, which is cellular. In which a mobile application is developed that is capable of detecting the location of the vehicle by assigning location services, it shows the route that the moving vehicle is taking and the trajectory it takes while showing it on a Google Maps map. , which is implemented in the application together with its API's credentials and GoogleMaps services, in addition to detecting the speed at which said vehicle is going. Which is achieved through the accelerometer that includes the smartphone. All the data obtained will be stored in a local SQLite database for later analysis. The monitoring and visualization application was implemented in Android Studio because the Android operating system is positioned as one of the most common operating systems on cell phones today.

Keywords: application, monitoring, GoogleMaps, road traffic, driving behavior, accelerometer.

CONTENIDO

FICHA TÉCNICA	IV
FORMATO DE LIBERACIÓN DEL PROYECTO PARA TITULACIÓN INTEGRAL.....	V
.....	V
CARTA DE AUTORIZACIÓN DE ENTREGA DE TESIS EN SOPORTE DIGITAL	VI
.....	VII
DEDICATORIA	VIII
AGRADECIMIENTOS	IX
RESUMEN.....	X
ABSTRACT.....	XI
LISTA DE FIGURAS	XIV
LISTA DE CUADROS.....	XVI
1. INTRODUCCIÓN	1
2. PROBLEMA DE LA INVESTIGACIÓN	2
3. JUSTIFICACIÓN	2
4. OBJETIVOS DEL PROYECTO	3
4.1 OBJETIVO GENERAL	3
4.2 OBJETIVOS ESPECÍFICOS	3
5. HIPÓTESIS.....	3
6. MARCO TEÓRICO.....	4
7. MATERIALES Y MÉTODOS	27
8. RESULTADOS Y DISCUSIÓN	35
8.1 Resultados Obtenidos.....	35
9. CONCLUSIÓN.....	52
10. RECOMENDACIONES.....	53
11. REFERENCIAS BIBLIOGRÁFICAS.....	54
12. ANEXOS	56

LISTA DE FIGURAS

<i>Figura 1. Brújulas casera y convencional. Fuente: BBCNews Mundo Tecnología, 2015.....</i>	<i>7</i>
<i>Figura 2. Funciones extras de celular Fuente: BBCNews Mundo Tecnología, 2015.....</i>	<i>8</i>
<i>Figura 3. Sistemas Operativos más utilizados en el mundo. Fuente Statista. 2018.....</i>	<i>13</i>
<i>Figura 4. Logo de aplicación Maps. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>22</i>
<i>Figura 5. Logo de aplicacion TOMTOMEuropa. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>23</i>
<i>Figura 6. Logo de aplicación MapsQuest. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>23</i>
<i>Figura 7. Logo de aplicación BackCountry Navigator PRO GPS. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>24</i>
<i>Figura 8 Logo de aplicación Sygic. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>24</i>
<i>Figura 9. Logo de aplicación OsmAnd. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>25</i>
<i>Figura 10. Logo de aplicación Waze GPS social y tráfico. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>25</i>
<i>Figura 11. Logo de aplicación MapsWithMe. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>26</i>
<i>Figura 12. Logo de aplicación Navigon Europe. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>26</i>
<i>Figura 13. Logo de aplicación Backpackers GPS Trails. Fuente: ABCdesevilla, Mobility,2013.....</i>	<i>26</i>
<i>Figura 14. Pantalla de Android Studio. Fuente:elaboración propia.....</i>	<i>29</i>
<i>Figura 15. Clases. Fuente:elaboración propia.....</i>	<i>30</i>
<i>Figura 16. Clase SQLite. Fuente:elaboración propia.....</i>	<i>30</i>
<i>Figura 17. Clase Marcador 1ª parte. Fuente:elaboración propia.....</i>	<i>31</i>
<i>Figura 18. Clase Marcador 2ª parte. Fuente:elaboración propia.....</i>	<i>31</i>
<i>Figura 19. Clase Marcador 3ª parte. Fuente:elaboración propia.....</i>	<i>32</i>
<i>Figura 20. Clase MapsActivity. Fuente:elaboración propia.....</i>	<i>32</i>
<i>Figura 21. Método miUbicacion dentro de onMapReady. Fuente:elaboración propia.....</i>	<i>33</i>
<i>Figura 22. Métodos agregarMarcador y actualizarUbicacion. Fuente:elaboración propia.....</i>	<i>33</i>
<i>Figura 23. Método miUbicacion con permisos de localización.. Fuente:elaboración propia.....</i>	<i>34</i>
<i>Figura 24. Métodos guardar, mostrar y mostrarMarcador. Fuente:elaboración propia.....</i>	<i>34</i>

<i>Figura 25. Captura de pantalla de aplicación. Fuente:elaboración propia.</i>	35
<i>Figura 26. Captura de pantalla de mapa durante ejecución de aplicación. Fuente:elaboración propia.</i>	36
<i>Figura 27. Captura de pantalla de acercamiento a posición actual. Fuente:elaboración propia.</i>	38
<i>Figura 28. Captura de pantalla de recorrido de prueba en posición inicial de recorrido. Fuente:elaboración propia.</i>	39
<i>Figura 29. Captura de pantalla de recorrido total. Fuente:elaboración propia.</i>	40
<i>Figura 30. Captura de pantalla en inmediaciones del recorrido, con referencia a zona conocida. Fuente:elaboración propia.</i>	41
<i>Figura 31. Captura de pantalla durante recorrido en zona céntrica de Vega de Alatorre. Fuente:elaboración propia.</i>	42
<i>Figura 32. Captura de pantalla del final del recorrido. Fuente:elaboración propia.</i>	43
<i>Figura 33. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 1. Hora 2:13 p.m. Fuente:elaboración propia.</i>	45
<i>Figura 34. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 1. Hora 2:16 p.m. Fuente:elaboración propia.</i>	46
<i>Figura 35. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 1. Hora 2:17 p.m. Fuente:elaboración propia.</i>	47
<i>Figura 36. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 1. Hora 2:19 p.m. Fuente:elaboración propia.</i>	48
<i>Figura 37. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 2. Fuente:elaboración propia.</i>	50
<i>Figura 38. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 3. Fuente:elaboración propia.</i>	51
<i>Figura 39. Pantalla de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Eje Z. (Dispositivo móvil sobre una mesa). Fuente: Aliferi, 2014.</i>	57
<i>Figura 40. . Pantalla de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Eje Y. (Dispositivo móvil en posición vertical). Fuente: Aliferi, 2014.</i>	58
<i>Figura 41. . Pantalla de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Eje X.</i>	

(Dispositivo móvil en posición horizontal). Fuente: Aliferi, 2014. 59
Figura 42. . Pantalla Principal de ejemplo de Acelerómetro de Aliferi, con métrica de valor para
Ejes X, Y, Z. (Dispositivo móvil en movimiento). Fuente: Aliferi, 2014. 60

LISTA DE CUADROS

Cuadro 1. Tiempo empleado en viaje al trabajo. Fuente INE 2001 4
Cuadro 2. Medios de transporte empleados según la distancia. Fuente INE 2001. 5

1. INTRODUCCIÓN

La presente investigación se basa en el hecho de querer descubrir cuál es el comportamiento de conducción que los conductores tienen al momento de manejar sus vehículos. Si la forma de conducir está ligada a que ocurran ciertos accidentes de tráfico vial, que las vialidades se congestionen, o que exista algún retraso debido a la forma de conducir de ciertas personas. De ahí surge la inquietud; que se puede hacer para saber cuál es la forma en la que las personas conducen sus vehículos y si la forma de conducir en ciertas zonas afecta a que el tráfico se alente o se detenga.

La importancia de poder obtener datos reales de vehículos en movimiento dará mayor objetividad y certeza para poder analizar los datos, con la seguridad de que son verídicos y así poder utilizarlos en análisis posteriores.

El objetivo de esto es desarrollar un sistema que monitoree y visualice el comportamiento de conducción de un vehículo en movimiento por medio de un dispositivo móvil mediante las coordenadas de ubicación obtenidas a través de servicios de localización y la aceleración obtenida por medio del acelerómetro que incluye el dispositivo. Con ello se establecerán los datos y medidas de desplazamiento vial.

El desarrollo de una aplicación que sea capaz de obtener los datos que se requieren para el análisis del comportamiento de conducción vial será el alcance de esta investigación.

Cada conductor desarrolla su propio estilo de conducción, muchas de las acciones al momento de conducir un vehículo pueden generar situaciones en el tráfico, sean de embotellamiento o casos más graves como accidentes, que pueden causar la muerte tanto del conductor como de otras personas, además de la pérdida de bienes materiales.

2. PROBLEMA DE LA INVESTIGACIÓN

Aunque existen ciertas normas a la hora de conducir, la mayoría de conductores desarrolla su propio estilo de conducción. Es decir, cada conductor tiene diferentes hábitos de manejo y esto se refleja en el comportamiento del tránsito urbano.

Muchas de las acciones de los conductores al momento de conducir su vehículo pueden generar situaciones en el tráfico como congestión vial o conducción acelerada, que puede causar accidentes de tránsito. Obtener los datos del comportamiento de conducción vial por medio de un dispositivo celular contribuirá al análisis de la conducción vial en zonas urbanas. Así se podrá determinar con un análisis posterior si el comportamiento vial está relacionado con que ocurran estas situaciones, y determinar si existen otros factores que obstruyan, detengan el tráfico o generen accidentes, como podría ser un factor ambiental o de infraestructura.

Para este trabajo se pretende realizar una aplicación móvil que se apoye del acelerómetro y el GPS contenido en el dispositivo móvil para poder obtener los datos necesarios y así poder analizar el comportamiento del vehículo cuando se encuentre en movimiento.

3. JUSTIFICACIÓN

La creación de la aplicación apoyará con la obtención de información de la ubicación del vehículo, el desplazamiento, las coordenadas y la aceleración. Implementado la aplicación en diferentes vehículos en una zona determinada se podrá identificar el tipo de vialidad de la zona, mediante los datos que se obtengan.

Con el desarrollo de la aplicación se podrán obtener los datos de conducción de un vehículo en movimiento y así se podrá determinar su comportamiento. Los datos serán almacenados en una base de datos de tipo local. Con esto se guardaran los datos para su posterior análisis. Con la implementación de la aplicación en varios vehículos en una zona

se puede determinar el comportamiento de conducción que conlleven a ciertas situaciones de tráfico vial en una zona determinada.

4. OBJETIVOS DEL PROYECTO

4.1 OBJETIVO GENERAL

Desarrollar un sistema de monitoreo y visualización de tráfico vial a partir del seguimiento de un dispositivo móvil mediante las coordenadas de ubicación para establecer los datos y medidas de desplazamiento vial.

4.2 OBJETIVOS ESPECÍFICOS

- Analizar los parámetros para la recolección de datos del desplazamiento vial en un mapa para definir los requerimientos del sistema.
- Realizar el modelado de datos y diseño del sistema considerando una plataforma móvil con manejo de Google Maps, coordenadas y uso de acelerómetro para la obtención de datos.
- Desarrollar la interfaz gráfica que implemente el modelado establecido para el uso amigable del sistema.
- Implementar el API de Google Maps para obtener las coordenadas y el comportamiento del desplazamiento del dispositivo móvil.
- Realizar pruebas unitarias y de sistema para verificar el correcto funcionamiento de la aplicación.

5. HIPÓTESIS

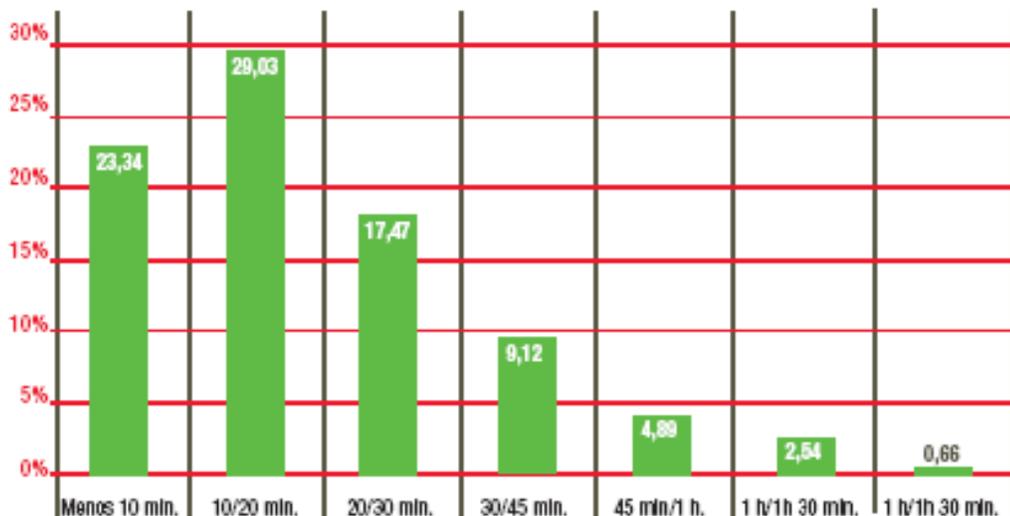
La implementación de una aplicación móvil para el monitoreo y visualización del tráfico vial mediante dispositivos celulares ayudará a conocer el tipo de comportamiento de manejo del conductor y con los datos obtenidos se podrá hacer un análisis del comportamiento vial en cierta zona.

6. MARCO TEÓRICO

6.1 Marco histórico

El tiempo destinado al transporte hasta los centros de trabajo o cualquier otro destino es un tiempo que reduce del tiempo de descanso y de las horas de sueño de los trabajadores. Además, al tener que realizar la tarea de conducir un automóvil no es posible relajarse, leer un libro, descansar o comunicarse con otros compañeros o amigos.

Según un estudio del Instituto Sindical de trabajo Ambiente y salud esta es la realidad laboral de una buena parte de los trabajadores, que superan diariamente los horarios contratados y de convenio. De hecho, el 17,21% de los trabajadores emplea más de 1 hora en ir y volver del trabajo. Ver, Cuadro 1. A ese tiempo de transporte que hay que añadir al tiempo de trabajo. (Instituto Sindical de trabajo Ambiente y salud [ISTAS], 2001), eso sin contar algún percance sobre las vialidades como lo son los atascos, tráfico lento o algún accidente además el medio de transporte que se utilice. Como se puede observar en la Cuadro 2. .Todo esto provoca acumular muchas horas de trabajo perdidas en el desplazamiento. Pero que es lo que provoca este tipo de situaciones en la vialidad.



Cuadro 1. Tiempo empleado en viaje al trabajo. Fuente INE 2001

	Vehículo privado	Transporte público	A pie	Bici	N/A
Menos 10'	51,80	1,51	44,15	0,89	1,65
10-20'	68,99	9,70	18,33	0,55	2,43
20-30'	65,35	24,85	7,06	0,26	2,48
30-45'	56,68	38,67	2,15	0,11	2,39
45' -1h	45,10	51,74	0,72	0,05	2,39
1h-1h 30'	39,28	57,94	0,43	0,04	2,31
Más de 1h 30'	46,56	47,11	1,77	0,18	4,38

Cuadro 2. Medios de transporte empleados según la distancia. Fuente INE 2001.

El vehículo privado es el transporte más utilizado hasta el umbral temporal comprendido entre 45 minutos y una hora; a partir de estos tiempos de viaje, los trabajadores eligen modos de desplazamiento más seguros y baratos como el transporte público, alcanzando al 51,74% de los viajes. (ISTAS, 2001),

Los desplazamientos no motorizados únicamente están presentes en los recorridos más próximos hasta el umbral de los 20 minutos, a partir de ese tiempo los modos motorizados captan la mayoría de los viajes.

La participación más importante del modo peatonal es en recorridos inferiores a 10 minutos, concentrando el 44,15% de los viajes (ISTAS, 2001). La bicicleta, a pesar de ser un modo marginal de desplazamientos, registra 70.000 trabajadores que utilizan diariamente este modo para acudir al trabajo, la mayoría se concentran en el umbral de la media hora.

Como se puede notar el transportarse es parte fundamental de las actividades humanas sin embargo hay conductas de manejo que aunado a alguna reparación o construcción en carretera provocan congestión vial o accidentes y aquí el punto de partida de esta investigación. Como determinar qué es lo que provoca congestión vial y cuales zonas son las que más presentan este tipo de problemas. Tomando en cuenta que lo que se califica como accidente vial se refiere a ciertos puntos como lo son exceso de velocidad,

Un accidente de tráfico o tránsito, entre otros términos, es un suceso que ocurre generalmente cuando un vehículo colisiona contra uno o más sectores de la vialidad (otro vehículo, peatón, animal, escombros del camino) u otra obstrucción estacionaria como

un poste, un edificio, un árbol, entre otros. Estos accidentes a menudo resultan en daños materiales (daños a los vehículos involucrados o al objeto investido), daños humanos (lesiones de diversa gravedad, discapacidad o muerte), así como costos financieros tanto para la sociedad como para las personas involucradas.

Estos no son aleatorios ni imprevisibles, y usualmente están acompañados por corresponsabilidades, como pueden ser ajenas al conductor (falta de señalización adecuada, carencia de iluminación en las calles, falla mecánica del vehículo, la mala construcción o el mal estado de una calle/avenida, etc.), así como propios del o los conductores en cuestión (no respetar las señales de tránsito, conducir en estado de ebriedad u otros efectos de estupefacientes, distracciones como utilizar el celular mientras se maneja, conducir a exceso de velocidad, realizar maniobras peligrosas, etc.).

Si bien, en la mayoría de los siniestros no se generaliza la culpabilidad, aunque no hay intención de lastimar, hay culpa. Por ejemplo, un conductor en estado de ebriedad atropella peatones por accidente, sin embargo sabe que es ilegal manejar en ese estado así como el hecho de que encontrarse en estado etílico reduce sus capacidades de maniobra, por lo que el hecho vial deja de ser impredecible o inevitable.

6.2 Marco conceptual

6.2.1 ¿Qué es el acelerómetro?

Los acelerómetros son dispositivos que miden la aceleración, que es la tasa de cambio de la velocidad de un objeto. Esto se mide en metros por segundo al cuadrado (m/s^2) o en las fuerzas G (g), (BBC Mundo Tecnología, 2015).

Los acelerómetros también se encuentran en muchos celulares inteligentes ya en el mercado. Estos sensores detectan las vibraciones generadas cuando se mueve el teléfono, y perciben cada sacudida o bamboleo para medir la inclinación del celular.

Las aplicaciones que utilizan el acelerómetro para dar lecturas de vibraciones pueden ser útiles, por ejemplo, para saber cuán suave es un carro durante una prueba de manejo. (BBC Mundo Tecnología, 2015).

El sensor de acelerómetro de Android es un sensor de movimiento basado en hardware, este puede medir la fuerza de aceleración en m/s^2 que se aplica a un dispositivo en los tres ejes físicos (x, y, z), incluida la fuerza de gravedad. (Aliferi, 2014).

Si se utilizará el acelerómetro como un sismógrafo, podría medir los temblores del suelo e incluso formar parte de un sistema de alerta temprana de terremotos, siempre y cuando haya un número suficiente de usuarios que recojan y envíen esta información.



Figura 1. Brújulas casera y convencional. Fuente: BBCNews Mundo Tecnología, 2015.

6.2.2 Receptor de GPS

Esta es, quizás la función más conocida. Mucha gente depende de sus teléfonos inteligentes para orientarse.

El chip de GPS dentro del celular detecta las señales de radio de varios satélites que orbitan la tierra para calcular su posición exacta.

Además de determinar tu ubicación en un mapa, algunas aplicaciones utilizan el receptor de GPS de tu teléfono para darte información precisa, como la ubicación de tus fotos por si se te olvida dónde las tomaste, o el lugar donde aparcaste el carro, particularmente útil para despistados al volante.

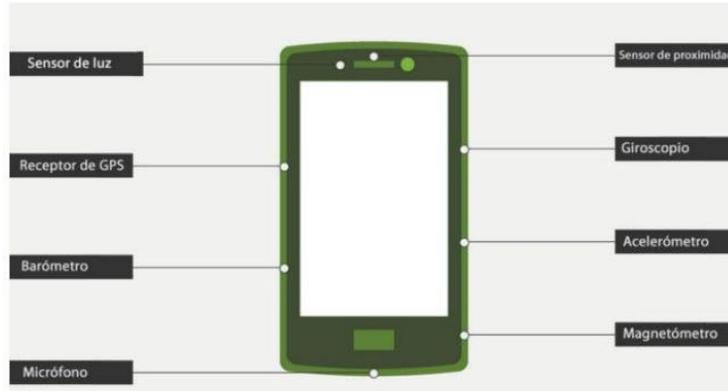


Figura 2. Funciones extras de celular Fuente: BBCNews Mundo Tecnología, 2015.

6.2.3 Aplicaciones Móviles

¿Qué es una aplicación móvil?

Una aplicación móvil es un programa que usted puede descargar y al que puede acceder directamente desde su teléfono o desde algún otro aparato móvil – como por ejemplo una Tablet o un reproductor MP3. (Comisión Federal De Comercio para Consumidores, 2011)

¿Qué se necesita para descargar y usar una aplicación?

Se necesita un Smartphone o algún otro aparato móvil con acceso a internet. No todas las aplicaciones funcionan en todos los aparatos móviles. Cuando se compra uno de estos aparatos debe usar el sistema operativo y el tipo de aplicaciones que corresponde a ese aparato. Los sistemas operativos móviles Android, Apple, Microsoft y BlackBerry tienen tiendas de aplicaciones que operan en línea en las cuales se puede buscar, descargar e instalar las aplicaciones. Algunos comerciantes minoristas también operan tiendas de aplicaciones en internet. Se tendrá que usar una tienda que ofrezca las aplicaciones que funcionen con el sistema operativo de su aparato. Para establecer una cuenta, es posible que se tenga que suministrar el número de una tarjeta de crédito, especialmente si se va a descargar una aplicación que no es gratuita.

Planes de servicios de datos y WiFi: Dos maneras de acceder a internet desde su teléfono

Usted puede acceder a internet usando un plan de datos relacionado con su servicio de teléfono, o a través de una conexión WiFi. Por lo general, las compañías de telefonía móvil cobran un cargo mensual por los planes de datos para conectarse a internet.

Usualmente, las conexiones WiFi son más rápidas, pero para poder usarlas es necesario estar dentro del rango de una red pública. La mayoría de los puntos de conexión WiFi de uso público – como los de las cafeterías, aeropuertos y hoteles – no codifican la información que usted envía a través de internet y no son conexiones seguras. Se puede consultar las recomendaciones para usar redes WiFi públicas.

Para establecer una conexión con una red WiFi, se tendrá que pagar el cargo de acceso a internet y el costo de un enrutador inalámbrico, y así mismo se necesitará tomar algunas medidas para proteger la red.

6.2.4 Tipos de aplicaciones móviles

Aplicación Nativa

Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un sistema diferente, por lo que si se quiere que una App esté disponible en todas las plataformas se deberán de crear varias Apps con el lenguaje del sistema operativo seleccionado.

Por ejemplo:

- Las Apps para iOS se desarrollan con lenguaje Objective-C.
- Las Apps para Android se desarrollan con lenguaje Java.
- Las Apps en Windows Phone se desarrollan en .Net.

Cuando se habla de desarrollo móvil casi siempre se está refiriendo a aplicaciones nativas. La principal ventaja con respecto a los otros dos tipos, es la posibilidad de acceder a todas las características del hardware del móvil: cámara, GPS, agenda, dispositivos de almacenamiento y otras muchas. Esto hace que la experiencia del usuario sea mucho más positiva que con otro tipo de Apps.

Además las aplicaciones nativas no necesitan conexión a internet para que funcionen. La descarga e instalación de estas Apps se realiza siempre a través de las tiendas de aplicaciones (App store de los fabricantes). Está claro que si el coste no es un obstáculo en la empresa, o se tiene la certeza de que la App será rentable, la mejor opción será siempre el desarrollo de una aplicación nativa para cada plataforma (iOS, Android y Windows Phone). Si el presupuesto es limitado, las aplicaciones web tienen también grandes ventajas para tu negocio.

Web App o aplicación Web

Una aplicación web o web App es la desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, JavaScript y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL. Por ejemplo en Safari, si se trata de la plataforma iOS. El contenido se adapta a la pantalla adquiriendo un aspecto de navegación APP.

¿Puede considerarse esto una APP? En realidad la gran diferencia con una aplicación nativa (además de los inconvenientes que se muestran en la tabla) es que no necesita instalación por lo que no pueden estar visibles en app store y la promoción y comercialización debe realizarse de forma independiente. De todas formas se puede crear un acceso directo que sería como “instalar” la aplicación en el dispositivo.

Las Apps web móviles son siempre una buena opción si nuestro objetivo es adaptar la web a formato móvil.

Aplicación Híbrida

Una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las Apps híbridas se desarrollan con lenguajes propios de las web App, es decir, HTML, JavaScript y CSS por lo que permite su uso en

diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que a pesar de estar desarrollada con HTML, Java o CSS, es posible agrupar los códigos y distribuirla en App store.

PhoneGap es uno de los frameworks más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas. Otro ejemplo de herramienta para desarrollar Apps híbridas es Cordova.

6.2.5 Sistemas operativos para móviles

Un sistema operativo móvil o es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él. Al igual que los PC que utilizan Windows, Linux o Mac OS, los dispositivos móviles tienen sus sistemas operativos como Android, iOS, Windows Phone o BlackBerry OS, entre otros. Los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. Algunos de los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.

IOS

IOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros.

La última versión del sistema operativo es iOS 13 (aparecida en junio de 2019) que sustituye a iOS 12 con el objetivo principal de mejorar la experiencia del usuario, integrando cambios relevantes, entre ellos el nuevo Modo Oscuro o la posibilidad de utilizar memorias externas con la App Archivos del sistema.

Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee una interfaz fluida. La interacción con el

sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

En el marco de las filtraciones acerca de los programas de vigilancia mundial de 2013-2014 de Edward Snowden, Der Spiegel publicó que la NSA estadounidense tiene grupos de trabajo dedicados a descifrar los sistemas de seguridad de iOS; además tiene pequeños programas conocidos como scripts que permiten a la agencia vigilar a los usuarios de las distintas versiones del sistema iOS su geolocalización, notas de voz, fotos y otras aplicaciones como Google Earth, Facebook o Yahoo! Messenger. iOS se deriva de macOS, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Tipo Unix. iOS cuenta con cuatro capas de abstracción.

Android

Android es un sistema operativo móvil basado en Kernel de Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes (Wear OS), automóviles (Android Auto) y televisores (Android TV).

Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y que adquirió en 2005.

Android fue presentado el 5 de noviembre del 2007 junto con la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El mismo día se anuncia la primera versión del sistema operativo: Android 1.0 Apple Pie. El código fuente principal de Android se conoce como Android Open Source Project (AOSP), que se licencia principalmente bajo la Licencia Apache.

Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 80 % al año 2017, muy por encima de iOS (Ver, Figura 3).

Casi 2.000 millones de dispositivos montan Android como sistema operativo.

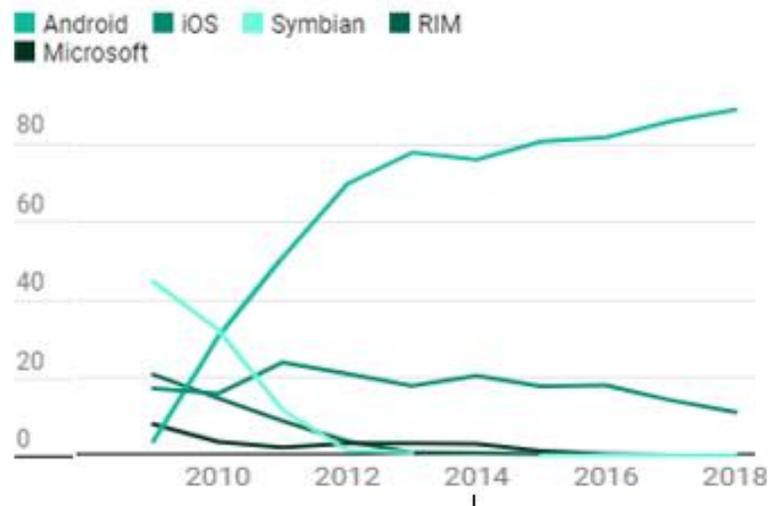


Figura 3. Sistemas Operativos más utilizados en el mundo. Fuente Statista. 2018

6.2.6 Entorno de desarrollo para aplicación móvil

Entorno de desarrollo integrado (IDE)

IDE (Integrated Development Environment) es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno de programación empaquetado como un programa o aplicación, que nos provee de un marco de trabajo agradable para la mayoría de los lenguajes de programación. (Villegas, 2014).

Los entornos integrados de desarrollo para Android. Constan entre sus características básicas con:

- Editor de código
- Compilador
- Depurador (debbuger)
- Constructor de interfaz gráfica

Cada día se publican nuevas aplicaciones para móviles y el crecimiento de estas es exponencial. Desarrollar una aplicación móvil requiere de conocimientos de programación y tiempo de dedicación.

Entre los IDE más populares para el desarrollo de aplicaciones Android podemos destacar los siguientes:

- Eclipse
- NetBeans
- IntelliJ
- Android Studio
- Aide

Eclipse

Villegas (2014), describe a Eclipse como una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el plug-in JDT que viene incluido en la distribución estándar del IDE.

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.

Principales características:

- Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una pre-configuración de ventanas y editores, relacionadas entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.
- Gestión de proyectos: el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios,... El IDE nos proporcionará asistentes y ayudas para la creación de proyectos. Por ejemplo,

cuando creamos uno, se abre la perspectiva adecuada al tipo de proyecto que estemos creando, con la colección de vistas, editores y ventanas pre-configurada por defecto.

- Depurador de código: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente nos ayuda a mejorar nuestro código. Para ello sólo debemos ejecutar el programa en modo depuración (con un simple botón). De nuevo, tenemos una perspectiva específica para la depuración de código, la perspectiva depuración, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.
- Extensa colección de plug-ins: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que nos imaginemos tenemos el plug-in adecuado.

NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. (Villegas, 2014). Existe además un número importante de módulos para extenderlo. NetBeans IDE1 es un producto libre y gratuito sin restricciones de uso.

Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (Villegas, 2014).

El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5.2, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.

Plataforma NetBeans

Es un framework que simplifica el desarrollo de aplicaciones para Java Swing. El paquete de NetBeans IDE para Java SE contiene lo que se necesita para empezar a desarrollar plugins y aplicaciones basadas en la plataforma NetBeans; no se requiere un SDK adicional.

Las aplicaciones pueden instalar módulos dinámicamente. Algunas aplicaciones pueden incluir un módulo de actualización para permitir a los usuarios descargar Actualizaciones de firma digital y nuevas características directamente dentro de la aplicación en ejecución. Reinstalando una actualización o una nueva versión, no obligando a los usuarios a descargar toda la aplicación de nuevo.

La plataforma ofrece servicios reusables comunes para las aplicaciones de escritorio,

permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones. Algunas de las características de la aplicación son:

- Gestión de la interfaz de usuario (menús y barras de herramientas)
- Gestión de configuración de usuario
- Gestión de almacenamiento (guardar o cargar algún tipo de dato)
- Gestión de ventana
- Marco Asistente (soporta diálogos paso a paso)
- Librería visual de NetBeans
- Herramientas de desarrollo integrado

NetBeans IDE es libre, código abierto, multiplataforma con soporte integrado para el lenguaje de programación Java.

Provee Soporte para la creación de aplicaciones orientadas a servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL para web services. Permite crear aplicaciones Web con PHP 7, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en PHP. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE

Android Studio

En la página oficial de desarrolladores de Android (Developers), se describe que Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan la productividad cuando se desarrollan apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para incluir nuevos cambios de código y recursos a la app en ejecución sin reiniciarla.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- Compatibilidad con C++ y NDK.
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.

Aliferi (2014) nos dice que la plataforma Android proporciona varios sensores que le permiten monitorear el movimiento de un dispositivo. Algunos de estos sensores están basados en hardware y otros están basados en software.

La plataforma Android admite tres amplias categorías de sensores:

- Sensores de movimiento.
- Termómetros de sensores ambientales.
- Sensores de posición.

Los sensores de movimiento son útiles para monitorear el movimiento del dispositivo,

como inclinación, sacudida, rotación o balanceo, el sensor de acelerómetro de Android es un sensor de movimiento basado en hardware. Este puede medir la fuerza de aceleración en m/s^2 que se aplica a un dispositivo en los tres ejes físicos (x, y, z), incluida la fuerza de gravedad. (Aliferi, 2014).

6.2.7 Base de datos para dispositivos móviles.

Las bases de datos han sido durante bastante tiempo la forma más común de almacenar y administrar datos. Por lo tanto, en un escenario típico, una aplicación móvil usa una base de datos alojada en la nube y se conecta remotamente a ella para acceder a sus datos. Esto, por supuesto, implica que la aplicación móvil necesita una conexión de red activa y bastante rápida.

La base de datos es la forma más común de almacenar y administrar datos. Desde hace bastante tiempo, las bases de datos se manejan en el lado del servidor o en la nube y los dispositivos móviles solo se comunican con ellos a través de la red. (Tecnologías información, 2018).

Sin embargo, para hacer que las aplicaciones sean más receptivas y menos dependientes de la conectividad de la red, la tendencia del uso fuera de línea o la menor dependencia de la red están ganando popularidad.

Hoy en día, las aplicaciones mantienen la base de datos localmente o hacen una copia de DB en la nube en el dispositivo local y se sincronizan con ella una vez al día o cada vez que hay una conectividad de red. Esto ayudará en aplicaciones más rápidas y receptivas que son funcionales incluso cuando no hay conectividad a Internet o es limitada.

La validación de datos también se conoce como validación de entrada.

Las bases de datos para Android deben ser:

- Ligeras ya que el almacenamiento es limitado en dispositivos móviles.
- Sin requisito de servidor
- En una forma de biblioteca con ninguna o muy limitada dependencia (incrustable) para que se pueda usar cuando sea necesario

- Rápido y seguro.
- Fácil de manejar mediante código y opción para hacerlo privado o compartido con otras aplicaciones.
- Poca memoria y consumo de energía

Las bases de datos incrustadas son bibliotecas livianas y autónomas sin componentes de servidor, sin necesidad de administración, una pequeña huella de código y requisitos de recursos limitados. Las aplicaciones móviles pueden vincularse (estáticamente o dinámicamente) con ellas y luego usarlas para crear y gestionar sus propias bases de datos privadas o compartidas localmente en el dispositivo.

FireBase

FireBase es una plataforma de Google, conocida por su servicio de base de datos en la nube. Este servicio permite conectar aplicaciones web y móvil con esa base de datos y actualizarse a tiempo real bidireccionalmente.

Pero FireBase no sólo es el backend sino que integra funcionalidades propias muy interesantes para el desarrollo (hosting, tests A/B, etc.) y también a nivel de crecimiento de negocio (analíticas y AD's).

Los datos se sincronizan a tiempo real con la aplicación. La comunicación es bidireccional, es decir que cuando modifico datos en la BD estos cambios se van reflejados directamente en la web y viceversa.

SQLite

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. Es un proyecto de dominio público creado por D. Richard Hipp. No necesita licencia y tiene soporte en Android, iOS, Windows Phone y Blackberry.

El motor de SQLite no es un proceso independiente con el que el programa principal se comunica.

La biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

Características principales de SQLite

- La base de datos completa se encuentra en un solo archivo.
- Puede funcionar enteramente en memoria, lo que la hace muy rápida.
- Tiene un footprint menor a 230KB.
- Es totalmente autocontenida (sin dependencias externas).
- Cuenta con librerías de acceso para muchos lenguajes de programación.
- Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.
- Soporta funciones SQL definidas por el usuario (UDF).
- El código fuente es de dominio público y se encuentra muy bien documentado.

6.3 Marco Referencial

Se buscó información relacionada con la aplicación que se realizará para usar de referencia y analizar las posibles aplicaciones ya existentes.

Se encontraron varias aplicaciones que nos sirven como antecedentes de la aplicación que se pretende realizar para el monitoreo del tráfico vial. Cada una tiene su aportación unas son ya muy conocidas como la ya bastante utilizada Google Maps (Figura, 4).

Las nuevas tecnologías ponen a disposición un montón de herramientas en los dispositivos móviles para localizar incluso al que quiera perderse. Estas aplicaciones entre las cuales se pueden elegir las funciones conforme a los intereses y actividades que

el usuario necesite, (ABCdesevilla, 2013). Como Maps, TOMTOM, MapQuest, BackCountry Navigator PRO GPS, Sygic, OsmAnd, Waze GPS social y tráfico, MapsWithMe, Navigon Europe y Backpackers GPS Trails.



Figura 4. Logo de aplicación Maps. Fuente: ABCdesevilla, Mobility,2013.

6.3.1 Maps

Comenzamos por analizar la aplicación que nos trae uno de los gigantes de internet, Google, con su famoso Maps (Figura, 4). Como todo lo que hace esta compañía, este navegador, si bien no es tan vistoso como otros, es sencillo, eficaz, y además, gratuito. Proporciona navegación vía GPS por voz, tanto si vas en coche, a pie o con transporte público, muestra mapas con edificios en 3D e información sobre el tráfico. Por supuesto, con Maps podrás acceder a Street View para ver ubicaciones con imagen real, e incluso tiene mapas de algunos edificios públicos, como aeropuertos y hoteles. El uso de esta aplicación es gratis.

Google Maps tiene una infinidad de funciones, desde decirte cómo llegar a un sitio a pie y los tiempos de paso del transporte público hasta crear mapas personalizados, comunicarte con empresas y valorarlas posteriormente. Hasta están probando una visualización usando realidad aumentada. Maps es una herramienta muy versátil y, como tal, no podía faltar la posibilidad de conocer tus coordenadas exactas.

Las coordenadas se componen de dos elementos. Por un lado tenemos la latitud, que es el ángulo entre cualquier punto y el ecuador, es decir, las líneas horizontales del mapa. La longitud es el ángulo a lo largo del ecuador desde cualquier punto del planeta, siendo el meridiano de referencia el que pasa por el Observatorio de Greenwich (longitud cero). En otras palabras, la longitud son los arcos verticales. Ambas se miden en grados

sexagesimales y permiten conocer la ubicación exacta de cualquier punto del globo.



Figura 5. Logo de aplicación TOMTOMEuropa. Fuente: ABCdesevilla, Mobility,2013.

6.3.2 TomTom Europa

TomTom ha sido desde los inicios de los navegadores GPS uno de los más conocidos, y ello es debido a la calidad de sus mapeados. (Figura, 5). Lo puedes utilizar sin conexión a internet, por lo que no debe importarte si estás perdido en medio de las estepas escocesas y sin cobertura en medio de una tormenta. Eso sí, ocupa más de 3 gigas de espacio en tu dispositivo y cuesta casi 60 euros, pero su calidad está fuera de toda duda. Además TomTom tiene información sobre radares, el tráfico, y ofrece visualizaciones en 2D o 3D, El precio de la aplicación está en 59,99 €.



Figura 6. Logo de aplicación MapsQuest. Fuente: ABCdesevilla, Mobility,2013.

6.3.3 MapQuest

Parecido al TomTom pero con una calidad no tan cuidada. No por nada es una aplicación gratuita. (Figura, 6). Activa el GPS guiado por voz para encontrar con MapQuest la ruta de destino en países de todo el mundo, con información sobre el tráfico, o sobre locales de tu interés, como tiendas o restaurantes. Es gratis.



Figura 7. Logo de aplicación BackCountry Navigator PRO GPS. Fuente: ABCdesevilla, Mobility,2013.

6.3.4 BackCountry Navigator PRO GPS

BackCountry Navigator PRO GPS (Figura, 7). Nos proporciona navegación GPS offline (es decir, una vez instalado no necesitamos acceso internet) que nos guiará en nuestras rutas de senderismo, rutas en canoa, montañismo...gracias a mapas topográficos detallados. Dispone de una versión gratuita que te servirá para evaluar si de verdad te interesa comprar la aplicación. El precio de esta aplicación está en 7,99 €.



Figura 8 Logo de aplicación Sygic. Fuente: ABCdesevilla, Mobility,2013.

6.3.5 Sygic

Sygic es un navegador GPS offline (Figura, 8) que utiliza mapas 2D o 3D, con rutas alternativas para que seleccionar la que más convenga y avisos en carretera, como sobre el tráfico, si ha habido algún accidente en tu ruta, o si viene alguna curva peligrosa. Además, la pantalla de navegación es personalizable. Esta aplicación es gratis.



Figura 9. Logo de aplicación OsmAnd. Fuente: ABCdesevilla, Mobility,2013.

6.3.6 OsmAnd

OsmAnd estaría a medio camino entre la aplicación TomTom y la de Maps, algo que viene reflejado por su precio. Trabaja como navegador GPS tanto online como offline, con mapas vectoriales compactos de países de todo el planeta, que puedes ir descargándose en función de los viajes que el usuario realiza. Existe una versión gratuita de esta App, y otra de pago que está en 59,99 €.



Figura 10. Logo de aplicación Waze GPS social y tráfico. Fuente: ABCdesevilla, Mobility,2013.

6.3.7 Waze GPS social y tráfico

Si buscas una forma diferente y más activa de navegación, Waze GPS (Figura, 10), aplicación que incorpora social y tráfico, puede ser una buena opción para estar informado de lo que ocurre en las vialidades. Y es que esta aplicación se basa en la información facilitada por todos los usuarios, ya sea de forma activa (informando de accidentes, rutas colapsadas o limpias, gasolineras baratas...), como de forma pasiva, ya que con el mero hecho de llevar la aplicación abierta, ésta ya recoge información sobre cuánto tardamos en circular y la hace pública. De esta forma, con Waze gozarás de una ruta con un montón de información actualizada. Es gratis.



Figura 11. Logo de aplicación MapsWithMe. Fuente: ABCdesevilla, Mobility,2013.

6.3.8 MapsWithMe

Además de ser un buen navegador GPS con búsquedas de rutas incluso sin conexión a internet, MapsWithMe (Fig., 11), te ofrece búsquedas por categorías, es decir, si quieres encontrar un buen restaurante, una tienda, una peluquería... esta aplicación detectará las más cercanas y con un solo clic te guiará directamente hasta ellas. Dispone de una versión gratuita. Y una de pago que está en 3,78 €.



Figura 12. Logo de aplicación Navigon Europe. Fuente: ABCdesevilla, Mobility,2013.

6.3.9 Navigon Europe

Navigon Europe (Figura, 12), es un desarrollado navegador GPS con múltiples conexiones con Google, de forma que puedes conectar una ubicación determinada con Google Street View o Google Local Search. También ofrece un sistema de cámaras que te permitirán ver el tráfico en tiempo real. Esta aplicación tiene un costo de 59,95 €.



Figura 13. Logo de aplicación Backpackers GPS Trails. Fuente: ABCdesevilla, Mobility,2013.

6.3.10 Backpackers GPS Trails

Otro navegador GPS esencial para los amantes de los deportes al aire libre es este Backpackers GPS Trails, (Figura, 13). Diseñado para los que les gusta hacer senderismo, montañismo, trial, o cualquier otro deporte de ruta en campo abierto. Ofrece la posibilidad de guardar rutas y compartirlas con otros usuarios, así como marcar puntos de interés en cualquier punto de la ruta, especificando los detalles del mismo. Precio de la aplicación: 3,82 €.

7. MATERIALES Y MÉTODOS

Actividad 1. Recabar información necesaria para el desarrollo del sistema: se buscó información relacionada al tema para analizar las posibles aplicaciones ya existentes. Encontrando 11 aplicaciones que usan de referencia el posicionamiento con GPS, siendo algunas de usó gratuito y otras con versión de pruebas y de pago.

Actividad 2. Analizar parámetros de recolección de datos del desplazamiento vial en un mapa para definir requerimientos. Se analizaron diferentes parámetros como lo son desplazamiento, distancia, velocidad promedio, velocidad instantánea y aceleración. Se obtuvieron fórmulas de cálculo de las anteriores y así se pudo determinar qué datos son los que se necesitan para la obtención de esas variables.

Actividad 3. Se definió el diseño para empezar a desarrollar el sistema.

Actividad 4. Se realizó modelado de datos y diseño de sistema considerando plataforma móvil con manejo de Google Maps, Coordenadas y acelerómetro para obtención de datos. La plataforma que es uso fue Android Studio, se realizó instalación y se incorporó el uso de Google Maps con un Activity.

Actividad 5. Se realizó el desarrollo de la aplicación de acuerdo con los requisitos y

diseños definidos. Los requisitos que se tomaron para el desarrollo de la aplicación fueron, obtención de coordenadas (latitud y longitud), intervalo de actualización y análisis de acelerómetro.

Actividad 6. Se implementaron en la API de Google Maps las siguientes funciones Maps SDK for Android, Geocoding API, Geolocation API y Places API. Con estos se puede implementar diferentes tipos de mapas, convertir coordenadas en direcciones, localizar ubicación por medio de GPS y ubicar lugares de referencia como tiendas, restaurantes, hoteles, gasolineras, etc.

Actividad 7. Se desarrolló la interfaz gráfica que implementó el modelado establecido con un diseño simple, sencillo pero funcional para uso amigable del sistema.

Actividad 8. Se implementó la API de Google Maps para obtener coordenadas y el comportamiento de desplazamiento del dispositivo móvil. Se obtuvieron coordenadas de ubicación actual del dispositivo móvil y se muestra el punto con un marcador. Se obtiene el recorrido de desplazamiento el cual se visualiza en el mapa por medio de un marcador que se actualiza cada 15 segundos.

Actividad 9. Realizar pruebas para verificar el correcto funcionamiento de la aplicación. Se realizó un primer recorrido a pie en la localidad del El Tacahuite, Vega de Alatorre Ver. (Figura, 27). Para prueba de funcionamiento en zona corta. Para la segunda prueba se realiza un recorrido en taxi de la localidad de Emilio Carranza a la localidad de El Tacahuite se monitorea recorrido y se obtiene la actualización de posición conforme el vehículo realiza el desplazamiento. (Figura, 29).

Codificación de aplicación en Android Studio

Clases

En la figura 14, se muestra la pantalla de Android Studio, y en la figura 15 se pueden observar a las diferentes clases que conforman el proyecto.

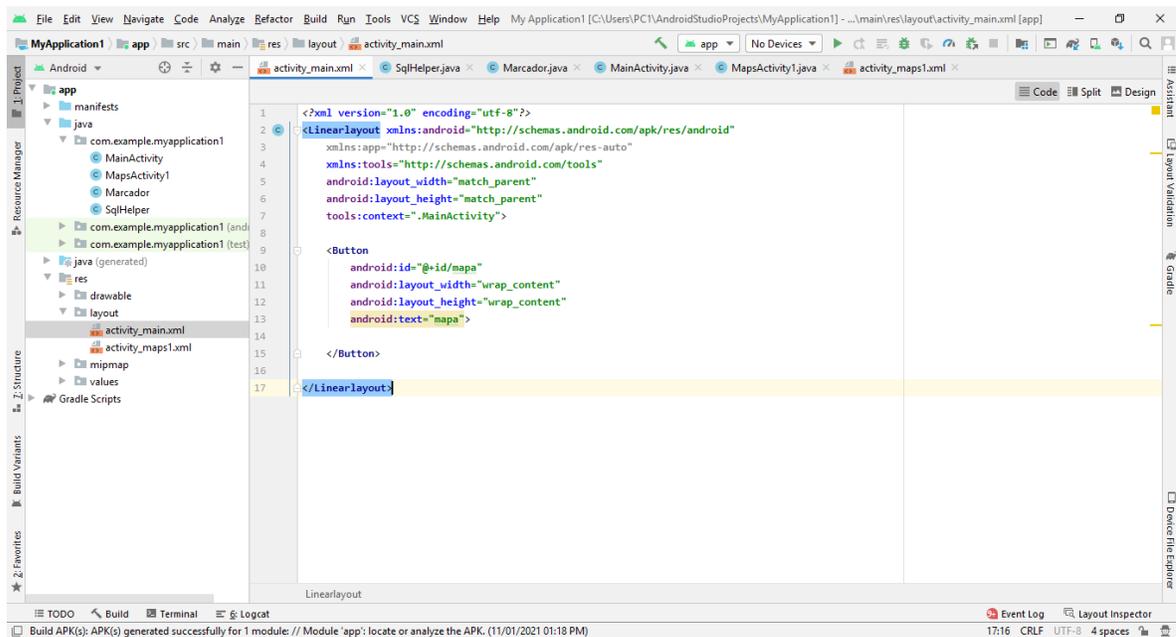


Figura 14. Pantalla de Android Studio. Fuente: elaboración propia.

Se creó el proyecto con las siguientes clases, un MainActivity, el MapsActivity1 que es el que muestra el mapa, una clase Marcador y una donde se crea la base de datos en SQLite.

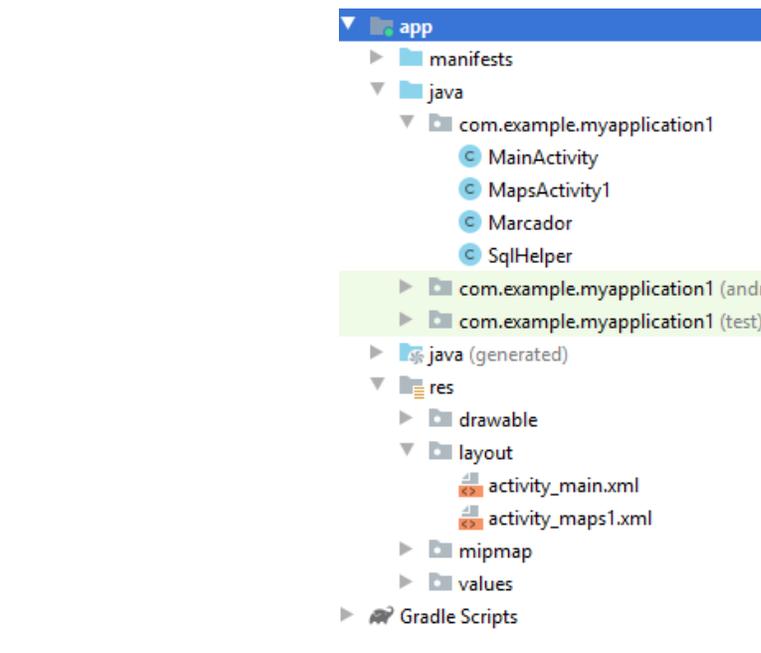


Figura 15. Clases. Fuente: elaboración propia.

Se crea estructura la base de datos, (Fig. 16)

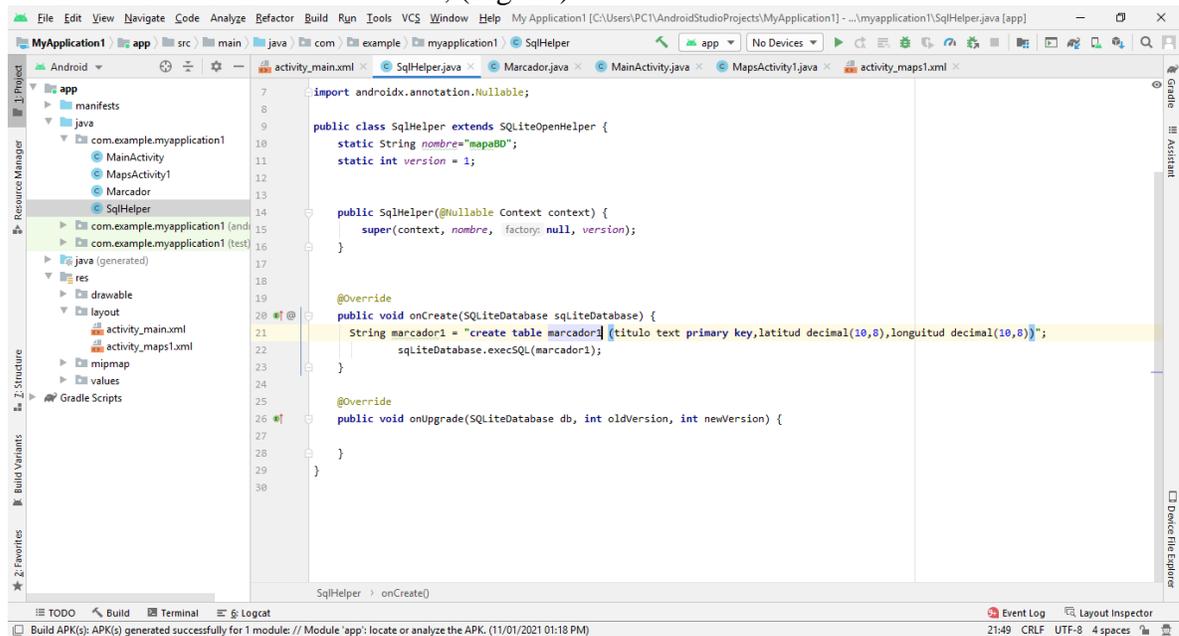


Figura 16. Clase SQLite. Fuente: elaboración propia.

La clase marcador se crea para obtener los datos que se guardarán, se implementan los métodos Getter and Setter, los cuales se utilizan para llenar la base de datos. (Figura, 17).

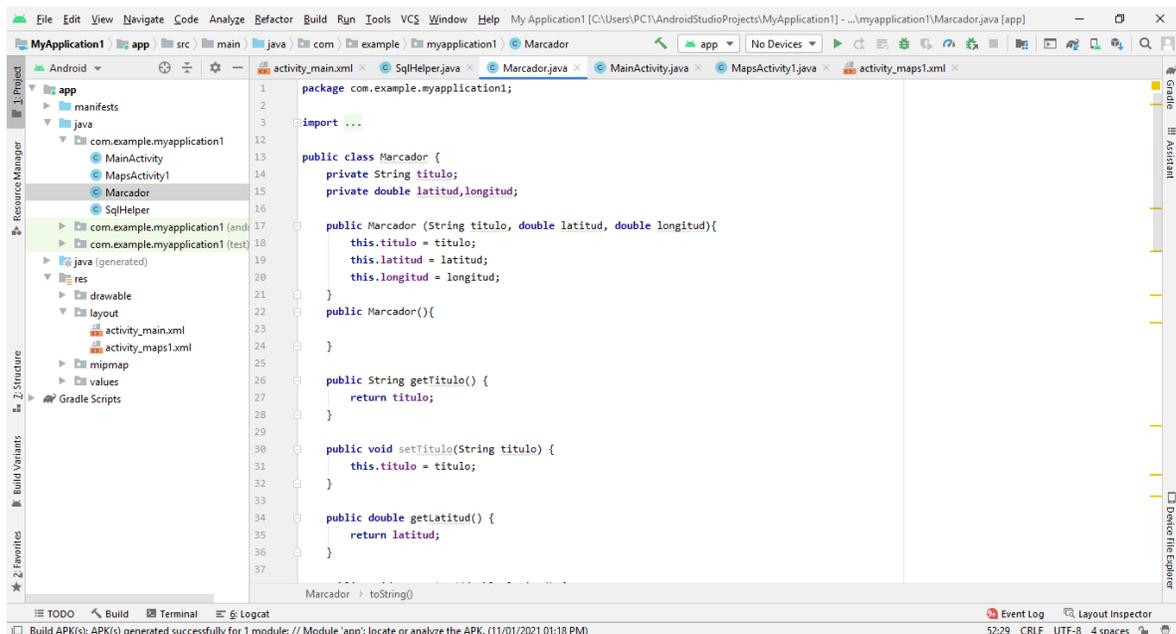


Figura 17. Clase Marcador 1ª parte. Fuente: elaboración propia.

Además se crean dos métodos, uno de lectura y otro de escritura para la base de datos de SQLite.

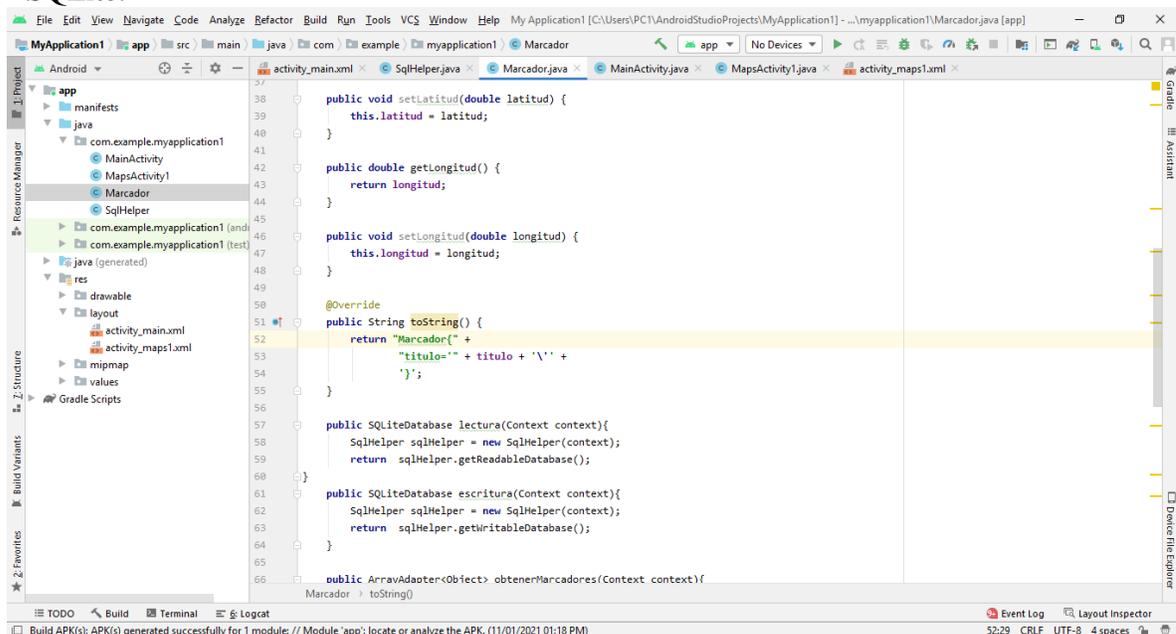


Figura 18. Clase Marcador 2ª parte. Fuente: elaboración propia.

Se crea un ArrayAdapter en el cual se guardarán los datos de los marcadores. (Figura, 19).

```

65
66
67 public ArrayAdapter<Object> obtenerMarcadores(Context context){
68     try{
69         ArrayAdapter<Object> adapter = new ArrayAdapter<>(context,android.R.layout.simple_list_item_1);
70         ArrayList<Object> arrayList = new ArrayList<>();
71         Cursor c = lectura(context).rawQuery( "select * from marcador1", selectionArgs: null);
72         while (c.moveToNext()){
73             arrayList.add(new Marcador(c.getString( c.getColumnIndex( 0),c.getDouble( c.getColumnIndex( 1),c.getDouble( c.getColumnIndex( 2))));
74         }
75         adapter.addAll(arrayList);
76         return adapter;
77     } catch (Exception e)
78     {
79         return new ArrayAdapter<Object>(context,android.R.layout.simple_list_item_1);
80     }
81 }
82
83 public long ingresar (Context context) {
84     ContentValues content = new ContentValues();
85     content.put("titulo", titulo);
86     content.put("latitud", latitud);
87     content.put("longitud", longitud);
88     return escritura(context).insert( table: "marcador1", nullColumnHack: null,content);
89 }
90
91
92
93 }

```

Figura 19. Clase Marcador 3ª parte. Fuente:elaboración propia.

La clase MapsActivity1 muestra el mapa. (Figura, 20).

```

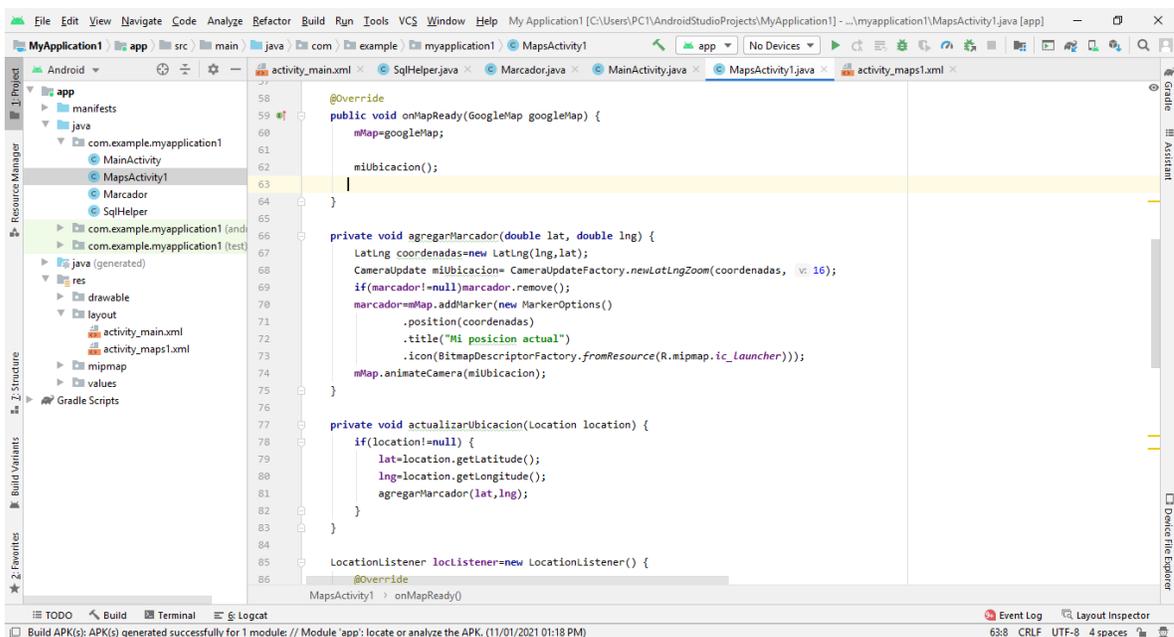
1 package com.example.myapplication1;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29 public class MapsActivity1 extends FragmentActivity implements OnMapReadyCallback {
30
31     EditText titulo;
32     Spinner marcadores;
33     Button guardar;
34     LatLng punto;
35     //se creo un marcador y 2 variables para latitud y longitud//
36     private GoogleMap mMap;
37     private Marker marcador;
38     double lat = 0.0;
39     double lng = 0.0;
40
41
42
43 @Override
44 protected void onCreate(Bundle savedInstanceState) {
45     super.onCreate(savedInstanceState);
46     setContentView(R.layout.activity_maps1);
47
48     SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
49         .findFragmentById(R.id.map);
50     mapFragment.getMapAsync( onMapReadyCallback: this);
51
52     //titulo = (EditText) findViewById(R.id.titulo);
53     // marcadores = (Spinner) findViewById(R.id.marcadores);
54     // guardar = (Button) findViewById(R.id.guardar);
55 }

```

Figura 20. Clase MapsActivity. Fuente:elaboración propia.

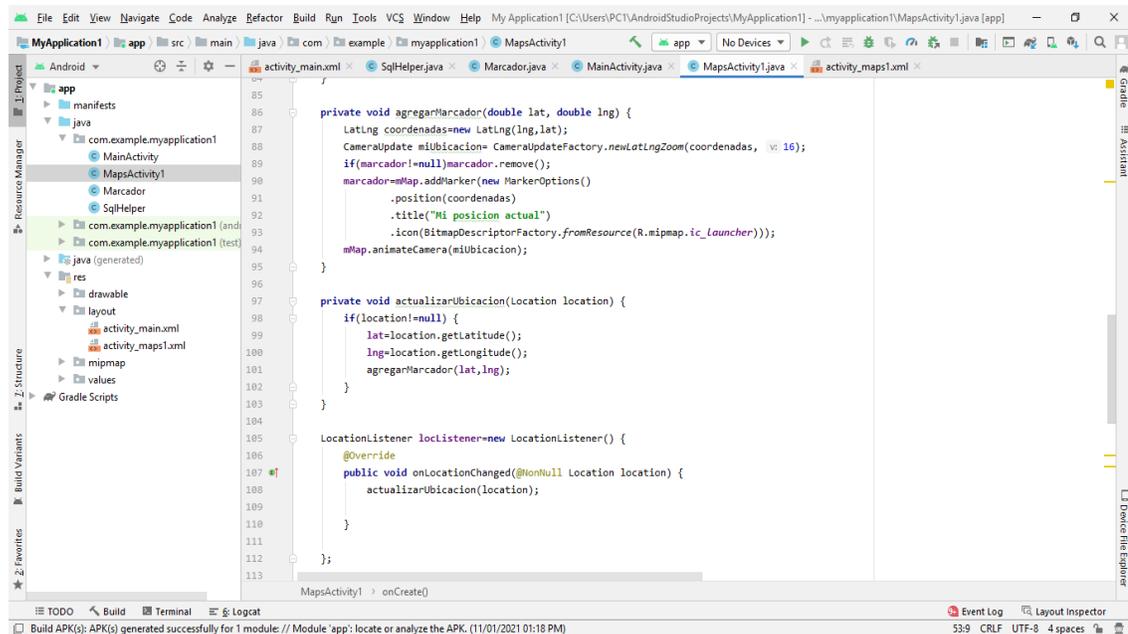
El método onMapReady. (Figura, 21). Este método indica lo que la aplicación realizará cuando el mapa este activo. Aquí se llama al método miUbicacion, el cual se asocia a

otros dos métodos que son el método actualizarubicacion y agregarMarcador.



```
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
@Override  
public void onMapReady(GoogleMap googleMap) {  
    mMap=googleMap;  
    miUbicacion();  
}  
  
private void agregarMarcador(double lat, double lng) {  
    LatLng coordenadas=new LatLng(lng,lat);  
    CameraUpdate miUbicacion= CameraUpdateFactory.newLatLngZoom(coordenadas, 16);  
    if(marcador!=null)marcador.remove();  
    marcador=mMap.addMarker(new MarkerOptions()  
        .position(coordenadas)  
        .title("Mi posicion actual")  
        .icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_launcher)));  
    mMap.animateCamera(miUbicacion);  
}  
  
private void actualizarUbicacion(Location location) {  
    if(location!=null) {  
        lat=location.getLatitude();  
        lng=location.getLongitude();  
        agregarMarcador(lat,lng);  
    }  
}  
  
LocationListener loListener=new LocationListener() {  
    @Override
```

Figura 21. Método miUbicacion dentro de onMapReady. Fuente:elaboración propia.



```
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
private void agregarMarcador(double lat, double lng) {  
    LatLng coordenadas=new LatLng(lng,lat);  
    CameraUpdate miUbicacion= CameraUpdateFactory.newLatLngZoom(coordenadas, 16);  
    if(marcador!=null)marcador.remove();  
    marcador=mMap.addMarker(new MarkerOptions()  
        .position(coordenadas)  
        .title("Mi posicion actual")  
        .icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_launcher)));  
    mMap.animateCamera(miUbicacion);  
}  
  
private void actualizarUbicacion(Location location) {  
    if(location!=null) {  
        lat=location.getLatitude();  
        lng=location.getLongitude();  
        agregarMarcador(lat,lng);  
    }  
}  
  
LocationListener loListener=new LocationListener() {  
    @Override  
    public void onLocationChanged(@NonNull Location location) {  
        actualizarUbicacion(location);  
    }  
};
```

Figura 22. Métodos agregarMarcador y actualizarUbicacion. Fuente:elaboración propia.

El método miUbicacion da permiso a los servicios de localización. (Ver, Figura, 23).

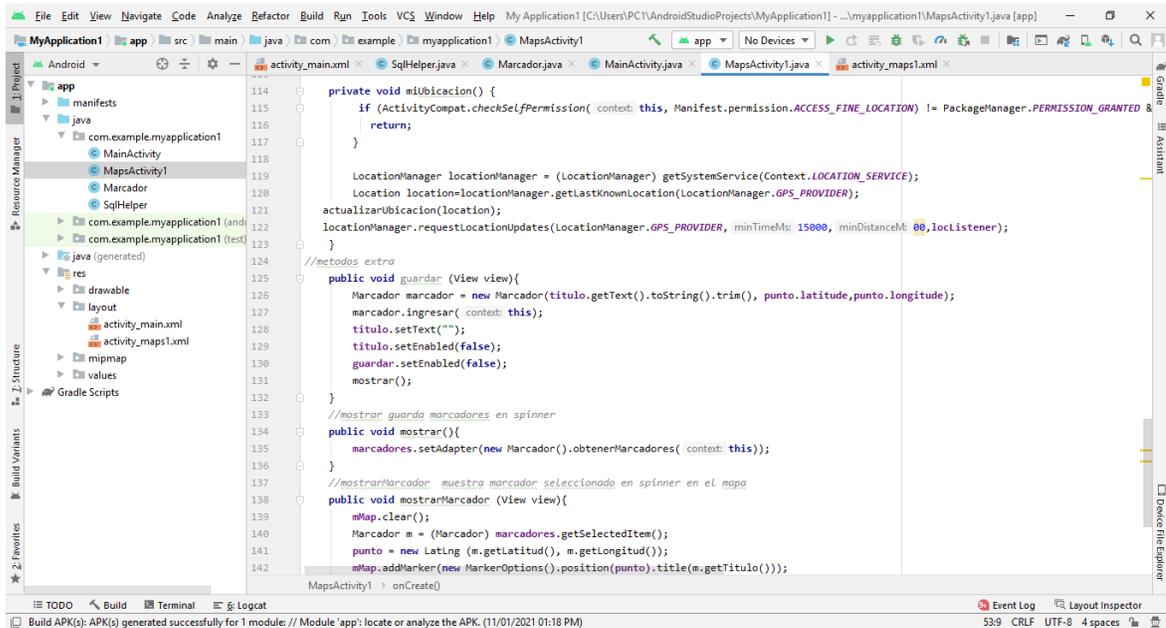


Figura 23. Método miUbicacion con permisos de localización.. Fuente:elaboración propia.

Métodos de guardado, mostrar en mapa y Spinner. Como se puede apreciar en la Figura 24. Se puede ver los métodos que se implementan para el guardado de datos.

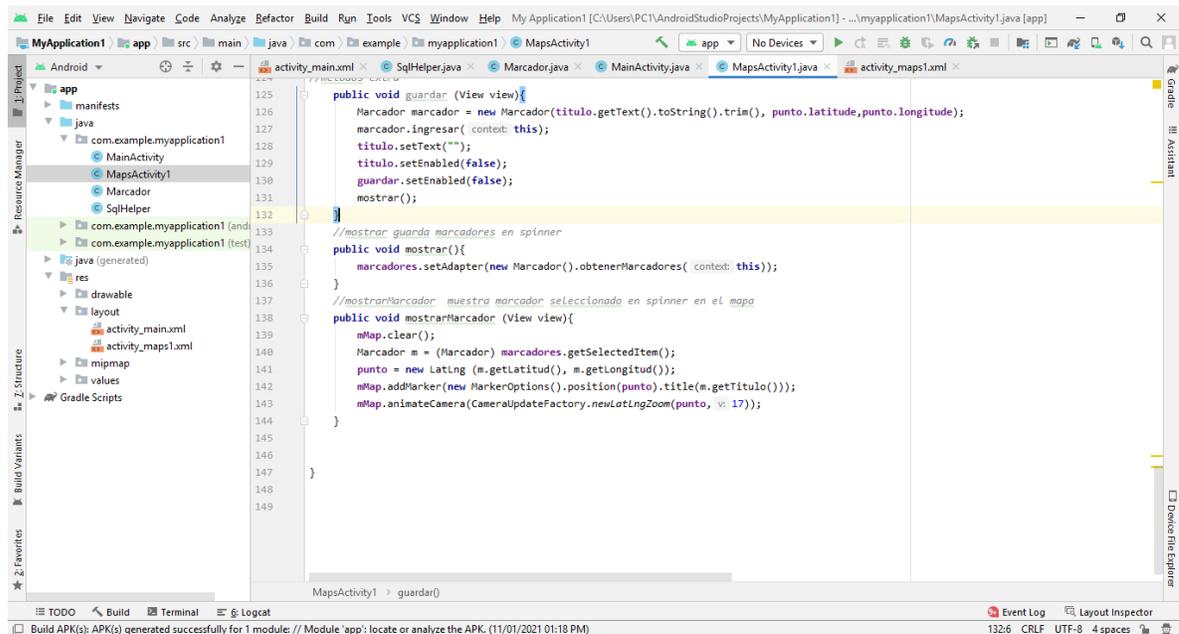


Figura 24. Métodos guardar, mostrar y mostrarMarcador. Fuente:elaboración propia.

8. RESULTADOS Y DISCUSIÓN

8.1 Resultados Obtenidos

Los resultados obtenidos del diseño y desarrollo de la aplicación móvil para el monitoreo y visualización del tráfico vial son los siguientes: La pantalla de inicio muestra un EditText con un Spinner y los botones de guardar y mostrar. (Figura, 25).

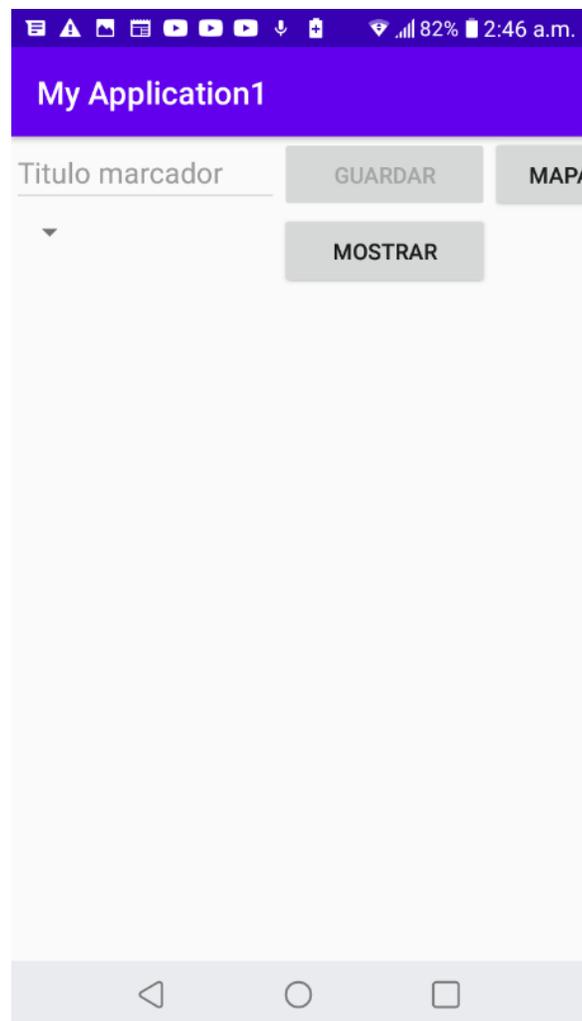


Figura 25. Captura de pantalla de aplicación. Fuente:elaboración propia.

Al abrir y pulsar sobre el botón con la leyenda “MAPA” se muestra el mapa que se incorporo dentro de las API’S de golocalizacion de la plataforma de Gogle Maps con la ubicación actual del dispositivo. (Figura, 26)



Figura 26. Captura de pantalla de mapa durante ejecución de aplicación.

Fuente:elaboración propia.

Al dar clic sobre el marcador que aparece en pantalla de la aplicación saldrá la información del marcador seleccionado, esta información solo muestra por el momento un texto con la leyenda “Mi posición actual”.(Figura, 26).

Al dar doble clic en cualquier parte de la pantalla donde se muestra el mapa, la aplicación realiza un zoom en la posición seleccionada, este acercamiento a el mapa solo tardara unos segundos en dicha posición, ya que la aplicación volverá a realizar un nuevo zoom en la posición donde se localize actualmente el dispositivo. Enfocando el marcador de posición actual. (Figura, 27).

Conforme el dispositivo avance, el marcador desaparecerá de la posición en la que se encontraba y cambiara a una nueva posición, la cual será donde ahora se ubique el dispositivo. La aplicación actualizará la nueva posición en el mapa, mostrando un nuevo marcador en la nueva posición del dispositivo. Esto se puede observar en las Figuras 29, 30, 31 y 32.



Figura 27. Captura de pantalla de acercamiento a posición actual. Fuente:elaboración propia.

Prueba de funcionalidad de la aplicación, se realizó un recorrido de la localidad de Emilio Carranza a la localidad de El Tacahuite. Vega de Alatorre, Ver.

Como se observa el marcador está ubicado en Emilio Carranza, el recorrido se inició en ese

lugar. (Figura, 28).

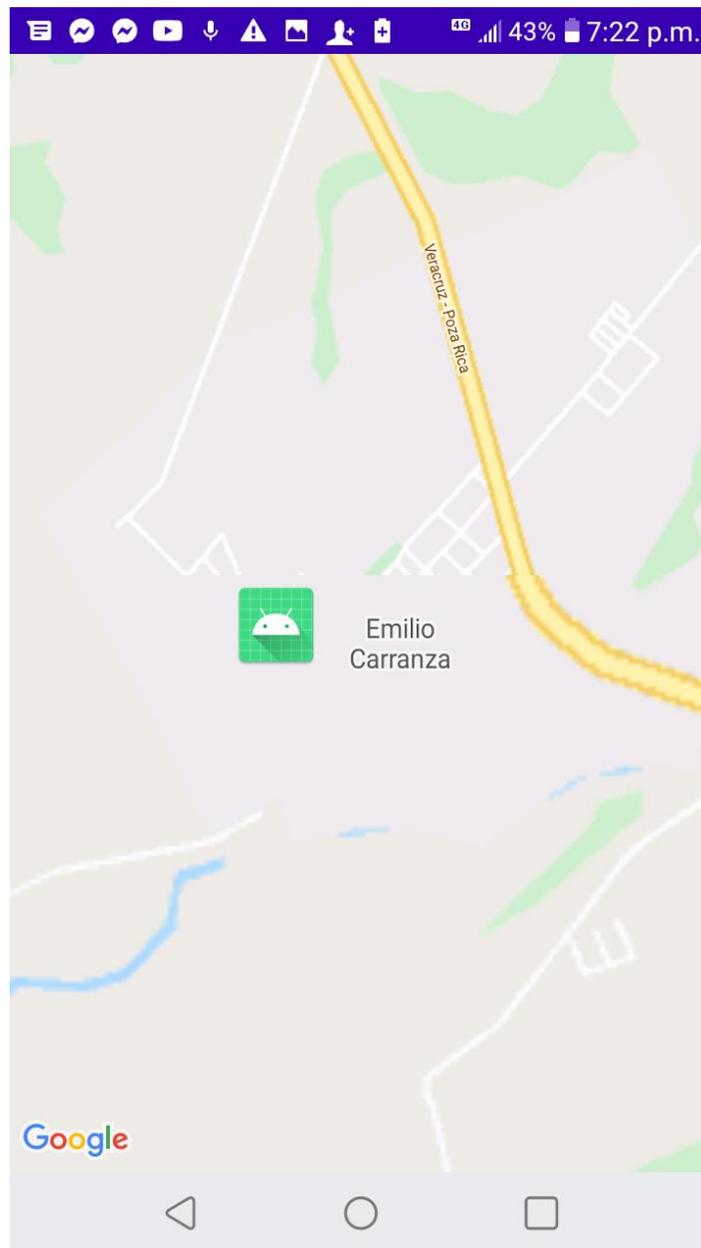


Figura 28. Captura de pantalla de recorrido de prueba en posición inicial de recorrido.

Fuente:elaboración propia.

Al acercar el mapa se aprecia un poco borroso debido a que no se estaba conectado a una red WIFI, ni a uso de datos, pero aun así la aplicación fue capaz de ubicar la posición, después

de eso se conectó a una red de datos y así se pudo cargar el mapa. (Figura, 29).

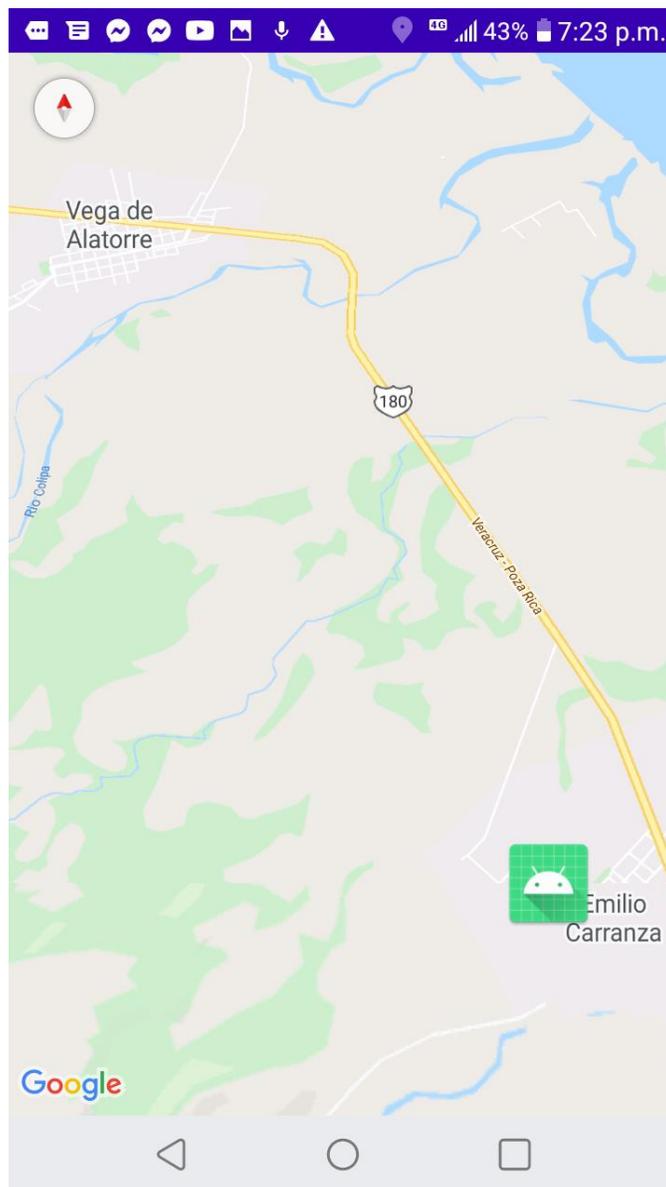


Figura 29. Captura de pantalla de recorrido total. Fuente:elaboración propia.

Se toma captura del recorrido cerca de Gasera La Reforma (Figura, 30), conocido punto entre las localidades de Emilio Carranza y Vega de Alatorre. El rio que se aprecia en la captura anterior es el que se muestra en la siguiente captura cuando el vehículo se encontraba en dicho punto siendo apreciado con el marcador de la aplicación.



Figura 30. Captura de pantalla en inmediaciones del recorrido, con referencia a zona conocida. Fuente: elaboración propia.

Entrando a Vega de Alatorre un poco después de haber pasado el palacio municipal, se toma la siguiente captura. (Figura, 31).



Figura 31. Captura de pantalla durante recorrido en zona céntrica de Vega de Alatorre.

Fuente: elaboración propia.

Y finalmente llegando a la localidad El Tacahuite al punto final del recorrido de prueba. Se aprecia el marcador en la zona final del recorrido. (Figura, 32).



Figura 32. Captura de pantalla del final del recorrido. Fuente:elaboración propia.
En la figura 32 se puede observar el punto final del recorrido de prueba, este lugar es la

localidad del Tacahuite, perteneciente al municipio de vega de Alatorre.

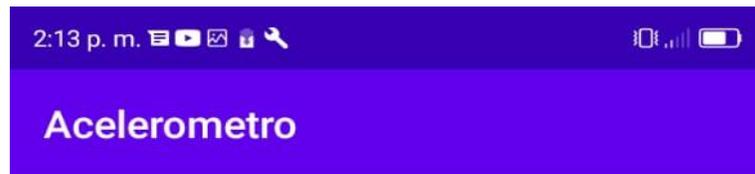
8.2 Acelerómetro

Se implementó código de acelerómetro para lo cual se creó una actividad para escuchar los cambios del acelerómetro de Android, además se creó un escuchador para mostrar los valores de aceleración actuales de los ejes x, y, z. De estos valores la actividad detecta los valores máximos que se registren durante la ejecución y los muestra en un Edit Text en pantalla, La implementación de esta actividad se realizó de forma externa para probar el correcto funcionamiento de esta, hacer pruebas pertinentes y posteriormente integrarla a la aplicación principal. Todo esto se basó en un ejemplo desarrollado (**Ver Anexo A**).

Los resultados que se obtuvieron fueron los siguientes:

Conforme avanza el vehículo se puede ir apreciando los cambios de aceleración que va teniendo el vehículo. Mostrando tanto la velocidad actual, como los valores máximos obtenidos durante el recorrido.

Las siguientes capturas de pantalla se tomaron a la aplicación durante un viaje en taxi, el cual partió de Vega de Alatorre a la localidad de Emilio Carranza. Se tomaron cuatro capturas de pantallas (Figura, 33, 34, 35, 36), las cuales demuestran los cambios de aceleración que se presentaron durante el recorrido.



Maxima Aceleracion:

Maxima Aceleracion Eje X:

22.6555

Maxima Aceleracion Eje Y:

3.8338997

Maxima Aceleracion Eje Z:

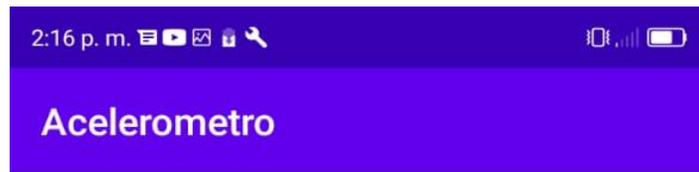
22.8204

Aceleracion actual

Eje X:	Eje Y	Eje Z
8.1775	0.0	9.444399

Figura 33. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm.

1. Hora 2:13 p.m. Fuente: elaboración propia.



Maxima Aceleracion:

Maxima Aceleracion Eje X:

22.6555

Maxima Aceleracion Eje Y:

5.6579

Maxima Aceleracion Eje Z:

22.8204

Aceleracion actual

Eje X:	Eje Y	Eje Z
10.8375	0.0	8.4564

Figura 34. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm.

1. Hora 2:16 p.m. Fuente: elaboración propia.



Maxima Aceleracion:

Maxima Aceleracion Eje X:

22.6555

Maxima Aceleracion Eje Y:

7.0259

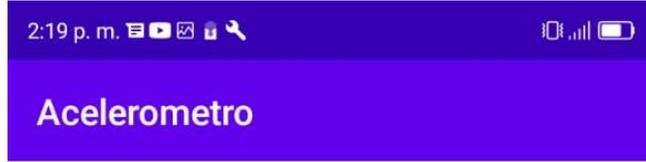
Maxima Aceleracion Eje Z:

22.8204

Aceleracion actual

Eje X:	Eje Y	Eje Z
6.9994993	0.0	10.0144

Figura 35. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 1. Hora 2:17 p.m. Fuente:elaboración propia.



Maxima Aceleracion:

Maxima Aceleracion Eje X:

22.6555

Maxima Aceleracion Eje Y:

7.0259

Maxima Aceleracion Eje Z:

23.6564

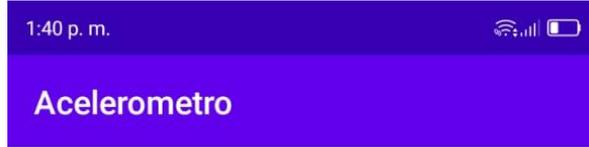
Aceleracion actual

Eje X:	Eje Y	Eje Z
7.6075	0.0	9.4064

Figura 36. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm.

1. Hora 2:19 p.m. Fuente: elaboración propia.

En un recorrido realizado otro día se obtuvieron los siguientes resultados. El vehículo fue nuevamente un taxi pero esta vez el conductor fue otro y estos fueron los datos obtenidos. Como se puede apreciar en la (Figura, 36) y la (Figura, 37) siendo el mismo recorrido realizado por ambos vehículos de la misma clase (ya que ambos son taxi modelo Tsuru) se aprecia que las lecturas de la segunda prueba (Figura, 37), son más altas. Lo que indica que el 2do conductor maneja un poco más rápido que el primero, haciendo notar que durante ambos recorridos no se tuvo ningún problema de tráfico, embotellamiento, estancamiento de vehículos, ni accidentes.



Maxima Aceleracion:

Maxima Aceleracion Eje X:

26.949501

Maxima Aceleracion Eje Y:

18.0079

Maxima Aceleracion Eje Z:

27.2284

Aceleracion actual

Eje X:	Eje Y	Eje Z
6.7715	5.6959	8.608399

Figura 37. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 2. Fuente:elaboración propia.

La siguiente prueba se toma de Vega de Alatorre a Martínez de la Torre con el mismo conductor de la segunda prueba (Figura, 37). Pudiendo apreciar que a pesar de ser el mismo conductor las lecturas obtenidas son diferentes. (Figura, 38).

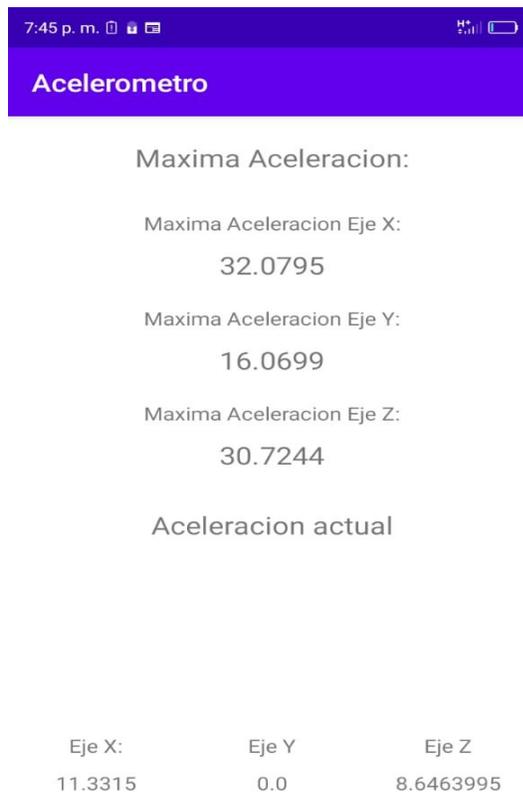


Figura 38. Captura de pantalla de Actividad de Acelerómetro durante recorrido Núm. 3. Fuente:elaboración propia.

Tanto de la prueba número 2, (Figura, 37) y la prueba número 3. (Figura, 38). Aunque es diferente trayecto, el conductor fue el mismo para ambos casos, y como se puede observar las lecturas tienen variaciones. Lo que nos muestra que a pesar de ser el mismo individuo la forma de conducción varía. Los conductores no solo desarrollan su propio estilo de conducción sino que también se adaptan a las situaciones de tráfico que se les presenten durante sus recorridos. Aunado a todo eso se debe de considerar la infraestructura de las carreteras y vialidades por donde transitan los vehículos.

9. CONCLUSIÓN

En conclusión la aplicación de monitoreo y visualización de tráfico vial realiza la función de ubicación del dispositivo por medio de localización GPS del dispositivo. Así mismo muestra el recorrido que realiza el vehículo al desplazarse, agregando un marcador que actualiza su posición cada 15 segundos en la pantalla. Se necesita un poco de más de codificación para lograr guardar los datos de posición del marcador en la base de datos, así como agregar otras características como lo son diferentes tipos de mapas y manipulación del acelerómetro para no solo obtener la aceleración a la cual se desplaza el vehículo, si no también poder determinar las paradas bruscas, el frenado, vueltas, disminución de velocidad etc.

La información que se obtenga de la aplicación servirá para su posterior análisis.

Los conductores no solo desarrollan su propio estilo de conducción sino que también se adaptan a las situaciones de tráfico que se les presenten durante los recorridos que realizan. Aunado a todo eso se debe de considerar la infraestructura de las carreteras y vialidades por donde transitan los vehículos, ya que en las pruebas realizadas se obtuvieron diferentes lecturas a pesar de ser en la prueba numero 1 el mismo recorrido con diferente conductor y en la prueba numero 2 el mismo conductor en dos distintos recorridos.

La experiencia laboral adquirida durante el proceso de desarrollo fue la codificación de Android Studio con Google Maps, ya se tenía un conocimiento de que se podían trabajar en conjunto, sin embargo no se había tenido la oportunidad de desarrollar una aplicación con Google Maps. Además de interactuar con el sensor de movimiento de Android, el cual fue el acelerómetro.

10. RECOMENDACIONES

Se necesita más codificación para lograr guardar los datos de posición del marcador en la base de datos, así como también integrar el acelerómetro a el proyecto principal y que los datos que se optengan durante el recorrido se puedan guardar a la base de datos, agregar características visuales como diferentes tipos de mapas, una interfaz mas llamativa e interactiva, manipulación del acelerómetro para obtener no solo la aceleración a la cual se desplaza el vehículo, si no también determinar las paradas bruscas, el frenado, vueltas, disminución de velocidad etc.

11. REFERENCIAS BIBLIOGRÁFICAS

ABCdesevilla, Mobility (21/01/2013). Las 10 mejores aplicaciones Android de mapas y navegación GPS. Información recuperada de: https://sevilla.abc.es/mobility/las_mejores_app/android/las-mejores-app-android/las-diez-mejores-aplicaciones-android-de-mapas-y-navegacion-gps-2/

Aliferi Cryssa (18/07/2014). Java Code Geeks. Ejemplo de acelerómetro en Android. Información recuperada de: <https://examples.javacodegeeks.com/android/core/hardware/sensor/android-accelerometer-example/>

BBCNews Mundo Tecnología. Ocho maneras de darle usos insospechados al celular (Enero, 2015). Información recuperada de: https://www.bbc.com/mundo/noticias/2015/01/150102_tecnologia_uso_distinto_de_sensores_celular_ig

Comisión Federal De Comercio para Consumidores (Septiembre 2011). Aplicación Móvil: ¿Que son y Cómo funcionan? Información recuperada de: <https://www.consumidor.ftc.gov/articulos/s0018-aplicaciones-moviles-que-son-y-como-funcionan>

Desarrolladores de Android. Introducción a Android Studio. Developers. Información recuperada de: <https://developer.android.com/studio/intro?hl=es-419>

Escuela pública digital universidad la punta. Modulo física. Información recuperada de:

<http://contenidosdigitales.ulp.edu.ar/exe/fisica/desplazamiento.html>.

Instituto Sindical de trabajo Ambiente y salud (2001). El tiempo de transporte es tiempo de trabajo. Información recuperada de: <https://istas.net/istas/guias-interactivas/planes-de-movilidad-sostenible-en-empresas/consecuencias-del-modelo-de-7>

Nieto García José (14/02/2019). XatacaMovil. Como saber tus coordenadas exactas en google Maps. Información recuperada de: <https://www.xatakamovil.com/aplicaciones/como-saber-tus-coordenadas-exactas-google-maps-compartirlas>.

Metodologías de desarrollo de Software. Información recuperada de: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>.

Villegas López David (21-05-2014) – Academia Android. Entornos integrados de desarrollo para Android. Información recuperada de: <https://academiaandroid.com/ide-entornos-integrados-de-desarrollo-para-android/>

12. ANEXOS

ANEXO A: Ejemplo de Acelerómetro.

Aliferi (2014) desarrolla un ejemplo de acelerómetro. En el cual se mide la aceleración en m/s^2 que se aplica a un dispositivo en los ejes (x, y, z), este ejemplo incluye la fuerza de gravedad en sus mediciones. Además de muestra valores actuales y máximos, cuando los valores del acelerómetro están por encima de un cierto umbral la aplicación vibra.

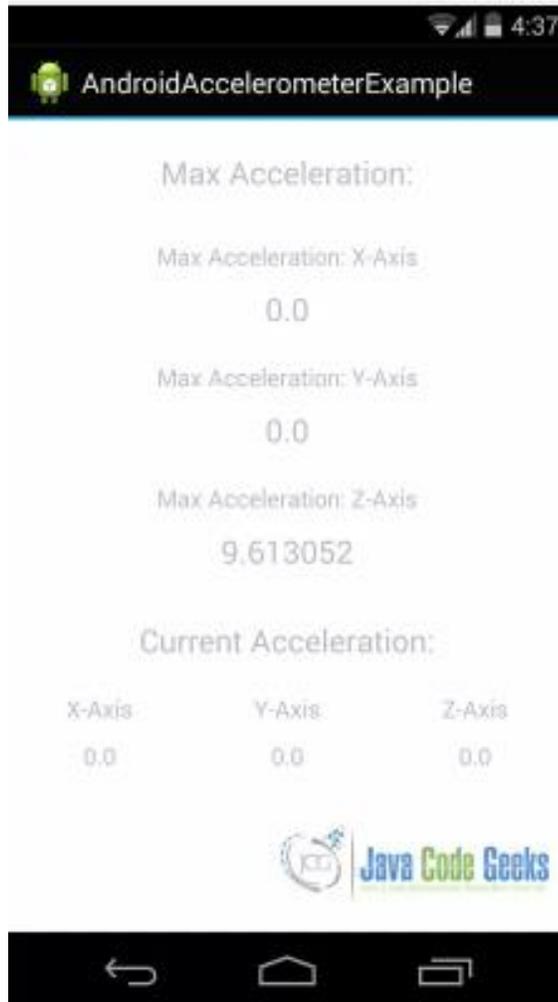


Figura 39. Pantalla de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Eje Z. (Dispositivo móvil sobre una mesa). Fuente: Aliferi, 2014.



Figura 40. . Pantalla de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Eje Y. (Dispositivo móvil en posición vertical). Fuente: Aliferi, 2014.



Figura 41. . Pantalla de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Eje X. (Dispositivo móvil en posición horizontal). Fuente: Aliferi, 2014.



Figura 42. . Pantalla Principal de ejemplo de Acelerómetro de Aliferi, con métrica de valor para Ejes X, Y, Z. (Dispositivo móvil en movimiento). Fuente: Aliferi, 2014.