



Certificación ISO 9001:2015 ‡

Modelo para el Registro Nacional de Infraestructura de Transporte Sostenible a través de una aplicación Android

Diana García Ramírez
Ricardo Eugenio Arredondo Ortiz
Elsa María Morales Bautista

**Publicación Técnica No. 494
San Fandila, Qro, 2017**

SECRETARÍA DE COMUNICACIONES Y TRANSPORTES
INSTITUTO MEXICANO DEL TRANSPORTE

**Modelo para el Registro Nacional de
Infraestructura de Transporte Sostenible a través
de una aplicación Android**

Publicación Técnica No. 494
San Fandila, Qro, 2017

Esta investigación fue realizada en la Coordinación Integración del Transporte del Instituto Mexicano del Transporte, por M.I/M.E Ricardo Eugenio Arredondo Ortiz, T.S.U. Diana García Ramírez y M.S.I. Elsa María Morales Bautista, de la Unidad de Sistemas de Información Geoespacial del IMT.

Se agradecen los comentarios y apoyo brindado por el **Dr. Carlos Daniel Martner Peyrelonge** Coordinador de Integración del Transporte, así como la asesoría técnica brindada por la Ing. Sandra Álvarez Granados, en la laboriosa tarea de reconciliación de formatos ODT, DOC y DOCX.

Contenido

Índice de tablas	vi
Índice de figuras	viii
Resumen	x
Abstract	xii
Introducción	1
1 Antecedentes	3
1.1 Transporte sostenible	3
1.2 Alineamiento con el Plan Nacional de Desarrollo 2013-2018	3
1.3 Alineamiento con el Programa Sectorial de Comunicaciones y Transportes 2013-2018	6
1.4 Siget Móvil	8
1.5 El Inventario Nacional de Infraestructura de Transporte	8
2 Planteamiento del Problema	9
2.1 Planteamiento del problema	9
2.2 Objetivos	10
2.3 Estrategias	10
2.4 Alcances	10
3 Propuesta Metodológica	11
3.1 Metodología SCRUM	11
4 Marco Teórico	15
4.1 Sistemas de Información Geográfica	15
4.2 SIGET-Móvil	16

4.3	Lenguajes de programación para desarrollo de aplicaciones móviles para Android.	19
4.3.1	Java.....	19
4.3.2	Open Source	20
4.4	Base de Datos.....	21
4.4.1	SQLite	21
4.4.2	MySql	21
4.4.3	SQL server	22
4.5	Mapas colaborativos	23
4.5.1	Sitios para creación de mapas Colaborativos.	23
4.5.2	Formatos de archivo para la exportación	24
4.5.2.1	GeoJson	24
5	Supuestos Técnicos	27
6	Descripción de la Propuesta de Solución	29
6.1	Requerimientos	29
6.2	Herramientas Informáticas	29
6.3	Características de los requerimientos.	29
6.4	Definición de los Datos de Entrada (tablas BD).	33
6.5	Diagramas UML.	35
6.5.1	Diagrama de Caso de Uso.....	35
6.5.2	. Descripción caso de uso.....	36
6.5.3	Diagrama de Clases	41
6.5.4	Diagrama de actividades	42
6.6	Diseño de Base de Datos.....	43
6.6.1	Script de Base de Datos	43
6.7	Diseño de pantallas.....	44
6.8	Codificación.....	49
6.8.1	Estructura del proyecto.....	49
7	Evaluación	51

7.1 Pruebas.....	51
7.2 Realización de Correcciones.....	53
Conclusiones.....	55
Anexo 1. Pantallas	57
Anexo 2. Codificación de Clases.....	72
Anexo 3. Clases de Controlador	82
Anexo 4. Clases del Modelo.	92
Referencias	96

Índice de tablas

Tabla 1-1. Principales elementos para el alineamiento del proyecto, con el Plan Nacional de Desarrollo (PND) 2013-2018 y con el Programa Sectorial de Comunicaciones y Transportes (PSCT) 2013-2018.....	7
Tabla 6-1. Característica de Pantalla Bienvenida.....	29
Tabla 6-2. Característica pantalla BiSiget-Movil.....	30
Tabla 6-3. Característica de Activar/Desactivar GPS.....	30
Tabla 6-4. Característica de Inicio de captura de ciclovía.....	30
Tabla 6-5. Característica de Terminar Captura de Ciclovía.....	31
Tabla 6-6. Característica de Infraestructura.....	31
Tabla 6-7. Característica de Consulta de Datos Capturados.....	31
Tabla 6-8. Característica de Exportar CVS.....	32
Tabla 6-9. Característica de Enviar Datos Al SIG.....	32
Tabla 6-10. Característica de Borrar datos locales.....	32
Tabla 6-11. Datos de entrada formulario Ciclovía (carretera).....	33
Tabla 6-12. Datos de entrada formulario Infraestructura.....	34
Tabla 6-13. Datos entrada Formulario Coordenadas.....	35
Tabla 6-14. Descripción Registrar Ciclovía.....	36
Tabla 6-15. Descripción Registrar Infraestructura.....	37
Tabla 6-16. Descripción Exporta Coordenadas.....	38
Tabla 6-17. Descripción Observar Datos.....	38
Tabla 6-18. Descripción Envía Datos.....	39
Tabla 6-19. Descripción Eliminar Datos.....	39
Tabla 6-20. Descripción Recepción Datos.....	40
Tabla 7-1. Tabla de control de pruebas.....	51
Tabla 7-2. Pruebas y errores encontrados.....	53

Índice de figuras

Figura 1-1. Esquema del Plan Nacional de Desarrollo 2013-2018;Error! Marcador no definido.	
Figura 2-1. SIGET 1.....	9
Figura 3-1. Proceso de la metodología Scrum.....	12
Figura 4-1 Sistema de Información Geográfica	16
Figura 4-2 Siget Móvil en Google Play	18
Figura 6-1. Diagrama de clases.....	35
Figura 6-3. Diagrama de clases.....	41
Figura 6-4. Diagrama de actividades.	42
Figura 6-5. Pantalla principal de la aplicación.	44
Figura 6-6. Pantalla principal de Funciones.....	44
Figura 6-7. Formulario de Registro de Ciclovía.	45
Figura 6-8. Formulario de registro de Infraestructura.....	46
Figura 6-9. Vista de datos.....	47
Figura 6-10. Pantalla de Exportar Datos.....	48
Figura 6-11. Ícono de Aplicación	49
Figura 6-12. Estructura de Proyecto.....	50

Resumen

El presente estudio presenta un modelo para el registro nacional de infraestructura de transporte sostenible mediante una aplicación móvil.

Como parte del aprovechamiento de los Sistemas de Información Geográficos, el Instituto Mexicano del Transporte ha desarrollado una aplicación móvil de nombre SIGET-Móvil, la cual utiliza los Sistemas de Posicionamiento Global, para realizar un inventario de la infraestructura carretera a nivel nacional, registrando todas las características con las que cuentan para facilitar el levantamiento y localización de los mismos desde un dispositivo móvil.

Derivada de esa iniciativa, el proyecto que se presenta propone el desarrollo de una aplicación similar que permita realizar el inventario de la infraestructura de transporte sostenible, como las ciclovías existentes a nivel nacional, buscando integrar a los usuarios en su levantamiento digital y facilitar la ubicación de las mismas, así como sus características físicas correspondientes.

El registro, envío y consulta de datos estará abierta a todo el público sin limitarlo a ningún sector específico.

El proyecto fue concluido en su primera fase y abre el espacio para continuar con la fase operativa a través de los períodos de prueba y error. Para ello, el actual proyecto presenta toda la documentación y datos correspondientes para la continuación de este proyecto, como son el proyecto en IDE de programación, codificación, estructura del proyecto, entre otros elementos principales.

Abstract

This study presents a model for the national registry of sustainable transport infrastructure through a mobile application.

The Mexican Institute of Transport has developed a mobile application using the Geographic Information System technology, for the national inventory of road infrastructure, named “SIGET-Móvil” which uses GPS function to locate the road infrastructure nationwide, recording all the features that they account by locating them easier from a mobile device.

Departing from that initiative, the current project proposes a similar approach to build a national inventory for sustainable transport, such as cycle roads, bike paths and so on, looking forward to integrate all users to participate in the inventory and to exploit the information derived, as location, surface characteristics, length and wide.

Such inventory would be open for consultation for everybody, through the same mobile application.

The project was accomplished at its first stage but more work would be needed to get a successful end. Therefore, all documents and corresponding data would be available to continue programming, such as project IDE programming, coding, project structure, etc., for the continuation of this Project.

Introducción

En la Secretaría de Comunicaciones y Transportes (SCT) y en particular en la Subsecretaría de Infraestructura se han realizado, con el liderazgo del Instituto Mexicano del Transporte (IMT), diversos proyectos y acciones para insertar organizacionalmente el uso y aprovechamiento de las geotecnologías en apoyo a sus procesos y funciones sustantivas.

Para la SCT resulta de gran importancia contar con información precisa y actualizada sobre la localización y condiciones de la infraestructura, lo que hace indispensable disponer de mecanismos que garanticen la generación y acopio de datos que contribuyan a mejorar los procesos de planeación, organización y gestión del sector transporte. Es posible lograr lo anterior si dichos datos son incorporados a sistemas de información geográfica eficientes en términos de velocidad de procesamiento, capacidad de almacenamiento y confiabilidad (Morales, 2015).

La unidad de Sistemas de Información Geoespacial desarrolla, coordina y supervisa investigación aplicada y desarrollo tecnológico para el aprovechamiento de sistemas de información georreferenciada en el transporte.

Por lo tanto, los servicios tecnológicos son los encargados de desarrollar herramientas para analizar y gestionar proyectos de accesibilidad y servicios de transporte, mediante la aplicación de tecnologías de manejo de información georreferenciada de vanguardia:

Inventarios de Infraestructura para el transporte.

Generación y actualización de información geoespacial de redes carreteras e infraestructura asociada, mediante el registro directo en campo con el uso del Sistema de Posicionamiento Global (GPS), para incorporarla a actividades operativas mediante su integración sistémica.

Accesibilidad espacial con base en la movilidad terrestre.

Evaluación de accesibilidad espacial, expresada en “tiempo de recorrido”, tanto a pie, como a través de distintos medios de desplazamiento o modos de transporte, considerando la pendiente del terreno y, en su caso, la superficie de rodamiento según tipo de camino (Morales, 2015).

Para aspirar a cumplir con los procesos de planeación, organización, gestión, evaluación y operación en el Sector Transporte, que exigen sistemas eficientes de manejo y análisis de información en términos de velocidad de procesamiento, capacidad de almacenamiento, versatilidad y confiabilidad, resulta indispensable, como elemento de partida, disponer de mecanismos que garanticen la generación y el acopio del insumo esencial para que funcione el sistema, esto es, de los datos. Ante esta realidad, el Instituto Mexicano del Transporte inició desde 1991 una línea

de investigación tendiente a evaluar las denominadas tecnologías de georreferenciación (SIG, GPS y procesamiento de imágenes de satélite), con el propósito ulterior de generar un inventario digital de datos geográficamente referenciados y diseñar, a partir de éste, un sistema de información geográfica especializado para su utilización en el Sector Transporte.

En este contexto y en respuesta a la demanda del Sector Transporte por obtener información precisa y actualizada sobre la localización de la infraestructura y de sus atributos asociados (características y condiciones de la misma), la Unidad de Sistemas de Información Espacial del Instituto Mexicano del Transporte con la colaboración de la Secretaría de Comunicaciones y Transportes (SCT), llevó a cabo el levantamiento del Inventario Nacional de Infraestructura para el Transporte (INIT), fase inicial y plataforma de partida del Sistema de Información Geoestadística para el Transporte (SIGET), mediante el empleo de receptores GPS (Sistema de Posicionamiento Global) para la captura en campo de información geográficamente referenciada y de un Sistema de Información Geográfica (SIG) –ArcInfo y ArcView– para el procesamiento posterior de los datos y programación de la interfaz para usuario final.

El SIGET es, en suma, la solución propuesta al problema de la carencia de un sistema integral de información en el Sector Transporte, que coadyuve a la toma de decisiones, con base en el manejo relacional de las bases de datos estadísticos en su expresión territorial, desde un ambiente gráfico de fácil manejo, con funciones diversas de consulta y despliegue visual, análisis espacial y representación cartográfica.

El SIGET es el resultado de un proyecto que eslabona distintas actividades, a lo largo de varios años, conjugadas desde el principio en el objetivo de proporcionar una herramienta útil que contribuya a la planeación, gestión y operación del sistema de transporte nacional. Entre las actividades realizadas con este propósito, destacan fundamentalmente dos, por un lado, la correspondiente a la construcción del cimiento del sistema en sí mismo, es decir, a la generación de la información georreferenciada de la infraestructura para el transporte, mediante el levantamiento del INIT con el empleo del GPS; la segunda actividad fundamental consistió en diseñar, integrar, estructurar y programar, con la plataforma del SIG, las funciones, operaciones e interfaz gráfica del SIGET (Morales, 2015).

1 Antecedentes

Uno de los elementos fundamentales del desarrollo del transporte sostenible, en México, estriba en el conocimiento de la infraestructura existente, que permita evaluar su disponibilidad, aprovechar su potencial, planificar su conservación, potenciar su crecimiento y organizar su conectividad en una red que impulse el progreso del país.

El desarrollo del proyecto de investigación propuesto transita por su alineamiento al Plan Nacional de Desarrollo 2013-2018 y al Programa Sectorial de Comunicaciones y Transportes 2013-2018, en los términos que se proponen en los siguientes párrafos.

1.1 Transporte sostenible

Se refiere al traslado de personas y/o mercancías, de un lugar a otro, que puede ejecutarse sin merma de los recursos existentes. Los principales ejemplos de este tipo de transporte son el ciclismo y la caminata. En este trabajo se busca desarrollar una aplicación informática que permita realizar el inventario nacional de infraestructura ciclista.

1.2 Alineamiento con el Plan Nacional de Desarrollo 2013-2018

El Plan Nacional de Desarrollo 2013-2018, establece como elemento central construir “Un México donde cada quien pueda escribir su propia historia de éxito y ser feliz” (PND) y como objetivo principal “llevar a México a su máximo potencial, a través de cinco metas nacionales y tres estrategias transversales” (PND), mismas que se despliegan en la figura 1.1.

FIGURA 1.1. ESQUEMA DEL PLAN NACIONAL DE DESARROLLO 2013-2018.



Fuente: PND, p.21

Figura 1 1. Esquema del Plan Nacional de Desarrollo 2013-2018

El proyecto que se propone se alinea a la cuarta meta que busca “Un México Próspero, que promueva el crecimiento sostenido de la productividad en un clima de estabilidad económica y mediante la generación de igualdad de oportunidades. Lo anterior considerando que una infraestructura adecuada y el acceso a insumos estratégicos fomentan la competencia y permiten mayores flujos de capital y conocimiento hacia individuos y empresas con el mayor potencial para aprovecharlo” (PND).

Adicionalmente, el proyecto se alinea a la quinta meta, que busca alcanzar “Un México con Responsabilidad Global que sea una fuerza positiva y propositiva en el mundo, una nación al servicio de las mejores causas de la humanidad. Nuestra actuación global debe incorporar la realidad nacional y las prioridades internas, enmarcadas en las otras cuatro Metas Nacionales, para que éstas sean un agente definitorio de la política exterior. Aspiramos a que nuestra nación fortalezca su voz y su presencia en la comunidad internacional, recobrando el liderazgo en beneficio de las grandes causas globales. Reafirmaremos nuestro compromiso con el libre comercio, la movilidad de capitales, la integración productiva, la movilidad segura de las personas y la atracción de talento e inversión al país. Ante los desafíos que enfrentamos tenemos la responsabilidad de trazar una ruta acorde con las nuevas realidades globales” (PND).

Por tanto, se plantea que este proyecto tiene una gran pertinencia para el Instituto Mexicano del Transporte, partiendo del principio de que el Plan Nacional de Desarrollo instruye a todas las dependencias de la Administración para alinear todos

los Programas Sectoriales, Institucionales, Regionales y Especiales en torno a conceptos tales como Democratizar la Productividad, un Gobierno Cercano y Moderno, así como Perspectiva de Género.

A esto se le agrega el interés del Plan Nacional de Desarrollo en reducir la dependencia que tiene el país de los combustibles fósiles, “impulsando el uso de fuentes de energía alternativas, fomentado la innovación y el mercado de tecnologías, tanto en el campo de la energía como en el aprovechamiento sustentable de los recursos naturales” (PND). También reconoce la participación de la sociedad en la conservación del capital natural y sus bienes y servicios ambientales, que son un elemento clave para el desarrollo nacional para el bienestar de la población.

En este sentido, el Plan Nacional refrenda el compromiso nacional con la agenda internacional de medio ambiente y desarrollo sustentable, en cuyo contexto han suscrito más de 90 acuerdos y protocolos vigentes, siendo líder en temas como cambio climático (PND).

Para lograrlo, se necesita contar con una infraestructura que facilite el flujo de productos, servicios y el tránsito de personas de una manera ágil, eficiente y a un bajo costo. Una infraestructura que potencie la capacidad productiva del país y abra nuevas oportunidades de desarrollo para la población.

Para crear un México Próspero, el Objetivo 4.4. (PND) busca impulsar y orientar un crecimiento verde incluyente y facilitador que preserve el patrimonio natural al mismo tiempo que genere riqueza, competitividad y empleo. Para ello se trata de fortalecer la política nacional de cambio climático y cuidado al medio ambiente para transitar hacia una economía competitiva, sustentable, resiliente y de bajo carbono (PND, Estrategia 4.4.3.).

En el Objetivo 4.9. se busca “Contar con una infraestructura de transporte que se refleje en menores costos para realizar la actividad económica”. El Programa propone varias líneas de acción que refuerzan la idea de comunicar a las localidades rurales (PND).

Líneas de acción:

- Fomentar que la construcción de nueva infraestructura favorezca la integración logística y aumente la competitividad derivada de una mayor interconectividad.
- Evaluar las necesidades de infraestructura a largo plazo para el desarrollo de la economía, considerando el desarrollo regional, las tendencias demográficas, las vocaciones económicas y la conectividad internacional, entre otros.

Sector carretero

- Consolidar y/o modernizar los ejes troncales transversales y longitudinales estratégicos, y concluir aquellos que se encuentren pendientes.
- Mejorar y modernizar la red de caminos rurales y alimentadores.
- Conservar y mantener en buenas condiciones los caminos rurales de las zonas más marginadas del país, a través del Programa de Empleo Temporal (PET).
- Modernizar las carreteras interestatales.
- Llevar a cabo la construcción de libramientos, incluyendo entronques, distribuidores y accesos.
- Ampliar y construir tramos carreteros mediante nuevos esquemas de financiamiento.
- Realizar obras de conexión y accesos a nodos logísticos que favorezcan el tránsito intermodal.
- Garantizar una mayor seguridad en las vías de comunicación, a través de mejores condiciones físicas de la red y sistemas inteligentes de transporte.

1.3 Alineamiento con el Programa Sectorial de Comunicaciones y Transportes 2013-2018

En su mensaje en el Programa Sectorial de Comunicaciones y Transportes 2013-2018, el Secretario de Comunicaciones y Transportes indica que se busca “conectar a las comunidades más alejadas, mejorar la productividad y elevar la competitividad global de México, así como acrecentar su potencial de desarrollo” (SCT).

También indica que “se construirán y modernizarán más carreteras, caminos rurales y alimentadores, libramientos y puentes, que fortalecerán la red troncal carretera, a los corredores longitudinales y transversales que unen al norte del país con el sur y al Océano Pacífico con el Golfo de México; infraestructura que se conectará con los otros modos de transporte” (SCT).

El Programa Sectorial describe que la red carretera del país suma 374,262 km. De ellos, 49,169 km conforman la red federal (8,459 km son autopistas de cuota y 40,710 km constituyen la red federal libre de peaje). Las redes troncal e intertronal de 24,308 km se consideran estratégicas, ya que conectan el 70% de las poblaciones del país. Dentro de los principales retos que enfrenta el sector transporte se encuentra el de elevar la seguridad vial, ya que cada año se suscitan entre 3.3 y 3.8 millones de accidentes de tránsito (PND).

Tabla 1-1. Principales elementos para el alineamiento del proyecto, con el Plan Nacional de Desarrollo (PND) 2013-2018 y con el Programa Sectorial de Comunicaciones y Transportes (PSCT) 2013-2018.

N°	CONCEPTO	ALINEAMIENTO
1	Impulsar y orientar un crecimiento verde incluyente y facilitador que preserve el patrimonio natural al mismo tiempo que genere riqueza, competitividad y empleo.	Objetivo 4.4. del PND
2	Implementar una política integral de desarrollo que vincule la sostenibilidad ambiental con costos y beneficios para la sociedad.	Estrategia 4.4.1. del PND
3	Fortalecer la política nacional de cambio climático y cuidado al medio ambiente para transitar hacia una economía competitiva, sustentable, resiliente y de bajo carbono.	Estrategia 4.4.3. del PND
4	Contar con una infraestructura de transporte que se refleje en menores costos para realizar la actividad económica.	Objetivo 4.9. del PND
5	Modernizar, ampliar y conservar la infraestructura de los diferentes modos de transporte, así como mejorar su conectividad bajo criterios estratégicos y de eficiencia.	Estrategia 4.9.1. del PND
6	Potenciar la inversión en proyectos de transporte sustentable, mediante una estrategia sólida de rentabilidad socio-económica y beneficios ambientales.	Estrategia 3.3 del PSCT
7	Generar entornos seguros y amigables de convivencia familiar y social, actividades de tiempo libre y movilidad segura para las mujeres y las niñas.	Objetivo 5 del PSCT
8	Generar condiciones para una movilidad de personas integral, ágil, segura, sustentable e incluyente, que incremente la calidad de vida.	Objetivo 3 del PSCT
9	Desarrollar proyectos de convivencia urbana que incrementen la velocidad del traslado de carga y fortalezcan la seguridad vial.	Línea de acción 1.2.3. de la estrategia 1.2. del PSCT
10	Establecer un programa dirigido a la promoción y defensa de los derechos ambientales.	Estrategia 1.5.1. del PND

1.4 Siget Móvil

En 2015 el Instituto Mexicano del Transporte desarrolló una aplicación móvil geoinformática, denominada Sistema de Información Geoestadística para el Transporte (SIGET Móvil), que busca ofrecer una herramienta tecnológica que contribuya a generar reportes y mapas de avance de obras en la red carretera nacional.

El SIGET Móvil es una aplicación para dispositivos Android que permite capturar la información de puentes, marca de kilometraje y bancos de material. También permite visualizar, procesar y transferir información de forma geográfica referenciada del transporte mediante dispositivos móviles.

La importancia de esta herramienta radica en la posibilidad de hacer una mejor gestión de los datos de la Secretaría de Comunicaciones y Transportes. Con SIGET móvil se puede capturar el nombre de las carreteras, si son de la administración federal, estatal o municipal, así como el número de carriles y el tipo de pavimento. Para registrar la ubicación de un puente se capturará un solo punto de posición y se integrarán los datos asociados como la ubicación, el nombre, el tipo de estructura y su tamaño.

Una vez que se procesa la información de la marca de kilómetros, se puede realizar análisis de seguridad, instalación, mantenimiento y conservación de la infraestructura. Otro aporte de esta tecnología es la generación de mapas e inventarios de la infraestructura, que podrá ser desplegada en oficina utilizando programas SIG de escritorio.

Para el diseño y programación de la aplicación se utiliza el programa *Eclipse*, mientras que para la creación de la base de datos se utiliza la plataforma *SQLite*.

El SIGET Móvil es una aplicación que fue desarrollada por la investigadora Elsa María Morales Bautista, de la Coordinación de Ingeniería Portuaria y Sistemas Geoespaciales, que participará en el proyecto propuesto, proporcionando asesoría y supervisión para la adaptación de esa tecnología, que permita la captura de los datos de la infraestructura para el transporte sostenible.

1.5 El Inventario Nacional de Infraestructura de Transporte

Durante varios años, el IMT ha realizado un importante esfuerzo para construir el inventario nacional de infraestructura de transporte, que ha permitido desarrollar una valiosa colección de archivos que detallan la composición de la red de transporte en líneas con sus respectivos atributos y múltiples capas de puntos de interés (e.g. puentes, alcantarillas, señalizaciones, etc.). Con éstos se inició la construcción de un Sistema de Información Geográfica del Transporte (SIGET) con la finalidad de hacer más versátiles y eficientes los procesos de visualización, consulta, mapeo temático y análisis espacial (Morales, 2015).

El proyecto propuesto permitirá incorporar la información de la infraestructura de transporte sostenible que el país ha ido construyendo en los últimos diez años, a través de inversiones de los tres niveles de gobierno (Arredondo, 2014).

2 Planteamiento del Problema

2.1 Planteamiento del problema

El Instituto Mexicano del Transporte (IMT) cuenta con un desarrollo y supervisión de las actividades científicas y tecnológicas de investigación básica y aplicada para la implementación de los sistemas de información georreferenciada al transporte.

En la actualidad, los procesos de planeación, organización, gestión, evaluación y operación en el Sector Transporte exigen sistemas eficientes de manejo y análisis de información en términos de velocidad de procesamiento, capacidad de almacenamiento, versatilidad y confiabilidad. Parte de estos procesos, implica el registro y control del inventario nacional de la infraestructura de transporte, como lo es la creciente construcción de infraestructura ciclista, que realizan los tres niveles de gobierno, a lo ancho y largo de la república mexicana.

Por tanto, derivado de la necesidad de Integrar un inventario nacional en formato digital y geográficamente referenciado de la infraestructura Ciclista, así como de la información estadística relacionada, se presenta este proyecto, el cual tiene como objetivo diseñar un modelo de gestión con base en una aplicación geoinformática en un dispositivo móvil habilitado para capturar, transferir y procesar información de la infraestructura de transporte ciclista geográficamente referenciada.



Fuente: (imt.mx, Micro sitios.)

Figura 2-1. SIGET 1.

Por lo tanto, el modelo propuesto es un aprovechamiento de la aplicación denominada SIGET-Móvil, cuyo modelo conceptual se muestra en la figura 2.1. y que fue desarrollada por la Investigadora Elsa María Morales Bautista (2015), para el proceso de captura de información de campo, sobre la ubicación y características de la infraestructura de transporte carretero y que será adaptada para el transporte sostenible como son ciclovías, ciclorrutas, ciclopuertos, biciestacionamientos, centros de transparencia e interconexión modal, entre otros.

El proyecto involucra el rediseño de la aplicación SIGET-Móvil, incluyendo el rediseño de las planillas de captura y el diseño de la base de datos interna de la aplicación (IMT, Sistemas de Información Geoespacial, 2016).

2.2 Objetivos.

Objetivo General.

Desarrollar una aplicación geoinformática para dispositivos móviles, que permita realizar un inventario nacional de infraestructura para el transporte sostenible, particularmente infraestructura ciclista.

Objetivos específicos.

- 1) Rediseñar la aplicación Siget Móvil, ya disponible para dispositivos móviles, para adaptarla a la captura de datos de infraestructura para el transporte sostenible.
- 2) Desarrollar una planilla para la captura de información de campo.
- 3) Diseñar una Base de datos Interna para la aplicación.

2.3 Estrategias

Para poder lograr una solución satisfactoria en el desarrollo del proyecto, se implementará una metodología ágil con la que se puede lograr avances mucho más rápidos y eficientes, la cual permita trabajar mucho más rápido y para ello la metodología de la que se hará uso es SCRUM, ya que es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, y donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

2.4 Alcances

Este trabajo se limitará exclusivamente a la realización de la aplicación para la captura de información sobre infraestructura ciclista y se dejará la realización del inventario mismo para una segunda etapa, en la que se buscará la colaboración de los gobiernos estatales, municipales y federal, así como agrupaciones civiles que quieran participar en la gran tarea de hacer el levantamiento nacional de la información.

3 Propuesta Metodológica

3.1 Metodología SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas, que se apoyan unas a otras y su selección tiene origen en los trabajos de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

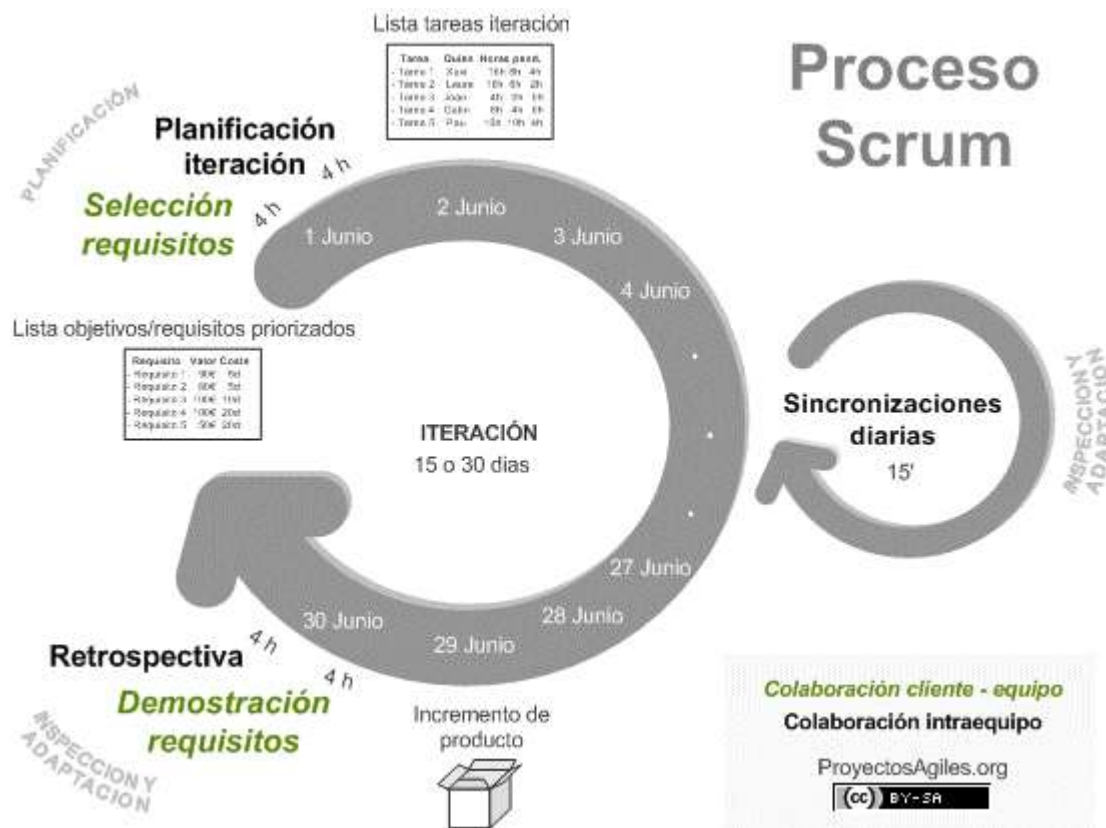
Es por ello que para este proyecto se decidió manejar Scrum, ya que la forma de trabajo se maneja por medio de reuniones semanales, en las cuales se hace entrega y revisión del entregable que se pidió; así mismo, si existe alguna corrección se hace mención y se planifica nuevamente, pero esto no afecta en el siguiente entregable, por lo tanto se puede continuar con el siguiente producto y realizar la corrección del anterior.

➤ El proceso

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

En este caso se manejaron iteraciones de una semana, en las cuales se daba revisión al entregable planificado, con el fin de mostrar avance sobre el proyecto y, una vez revisado dicho avance, se planificaba el siguiente entregable, se especificaban los nuevos requerimientos y nueva fecha de revisión.

A continuación, se visualiza una figura que muestra el flujo del proceso de la metodología Scrum.



Fuente: (<http://www.proyectosagiles.org/>,2015)

Figura 3-1. Proceso de la metodología Scrum.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto.

En esta lista, el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y la inversión mediante la replanificación de objetivos del producto, la cual realiza durante la iteración con vista a las siguientes iteraciones.

Cabe mencionar que Scrum comúnmente se maneja en equipos de trabajo, pero en este caso es una sola persona la encargada de llevar a cabo el proyecto, pero esto no afecta al flujo del proceso, ya que todo se lleva como se estipula en dicha metodología.

Las actividades que se llevan a cabo en Scrum son las siguientes:

1) Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Esta consiste en dos partes:

- **Selección de requisitos** (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
- **Planificación de la iteración** (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.

2) Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido.

En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador (*Scrum Master*) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

Elimina los obstáculos que el equipo no puede resolver por sí mismo.

Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

3) Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- a) **Demostración** (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo.

En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto.

- b) Retrospectiva (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

4 Marco Teórico

4.1 Sistemas de Información Geográfica

En la actualidad, los procesos de planeación, organización, gestión, evaluación y operación en el Sector Transporte exigen sistemas eficientes de manejo y análisis de información, en términos de velocidad de procesamiento, capacidad de almacenamiento, versatilidad y confiabilidad. Para aspirar a cumplir con lo anterior, resulta indispensable, como elemento de partida, disponer de mecanismos que garanticen la generación y el acopio del insumo esencial para que funcione el sistema, esto es, de los datos. Ante esta realidad, el Instituto Mexicano del Transporte inició desde 1991 una línea de investigación tendiente a evaluar las denominadas tecnologías de georreferenciación (SIG, GPS y procesamiento de imágenes de satélite), con el propósito ulterior de generar un inventario digital de datos geográficamente referenciados y diseñar, a partir de éste, un sistema de información geográfica especializado para su utilización en el Sector Transporte.

De acuerdo con IGN (2015), un Sistema de Información Geográfica (SIG) permite ver el mundo y lo que hay en éste, con una perspectiva distinta.

Las definiciones tradicionales describen a los SIG como un conjunto de hardware, software, datos personas y procedimientos; organizados para capturar, almacenar, actualizar, manejar, analizar y desplegar eficientemente rasgos de información referenciados geográficamente. Una definición más actual y puntual es que un SIG es un sistema que, por medio de computadoras y datos geográficos, ayuda a nuestro mejor entendimiento del mundo en que vivimos y nos permite resolver los problemas que se afrontan a diario.

A través de un SIG los mapas pueden ser integrados y correlacionados fácilmente con múltiples datos. De hecho, mediante un campo común de referencia, cualquier información en una tabla puede visualizarse en un mapa instantáneamente y cualquier problema representado en un mapa puede analizarse varias veces. Al contrario de lo que sucede con mapas tradicionales, los mapas en un SIG cambian dinámicamente en la medida que los datos alfanuméricos son actualizados. En la práctica, un SIG puede mapear cualquier información que está almacenada en bases de datos o tablas que tengan un componente geográfico, lo cual posibilita visualizar patrones, relaciones y tendencias. Con el SIG se tiene una perspectiva nueva y dinámica en el manejo de la información con el fin de ayudar a tomar mejores decisiones.

La información geográfica es importante, inclusive en situaciones como las de búsqueda de una nueva casa, ya que se tienen que tomar en cuenta una gran variedad de aspectos geográficos, tales como: sectores de estratificación, precio del terreno por colonia, cercanía a lugares como escuelas, supermercados e iglesias, vías de acceso, zonas de riesgo, etc. No es pues, una decisión simple, pero con la

ayuda de un SIG se puede facilitar, de manera que se acomode a las especificaciones deseadas.



Fuente: (La ciudad Viva, 2012)

Figura 4-1 Sistema de Información Geográfica

4.2 SIGET-Móvil

El SIGET es, en suma, la solución propuesta al problema de la carencia de un sistema integral de información en el Sector Transporte, que coadyuve a la toma de decisiones, con base en el manejo relacional de las bases de datos estadísticos en su expresión territorial, desde un ambiente gráfico de fácil manejo, con funciones diversas de consulta y despliegue visual, análisis espacial y representación cartográfica.

El SIGET es el resultado de un proyecto que eslabona distintas actividades, a lo largo de varios años, conjugadas desde el principio en el objetivo de proporcionar una herramienta útil que contribuya a la planeación, gestión y operación del sistema de transporte nacional. Entre las actividades realizadas con este propósito, destacan fundamentalmente dos, por un lado, la correspondiente a la construcción del cimiento del sistema en sí mismo, es decir, a la generación de la información georreferenciada de la infraestructura para el transporte, mediante el levantamiento del INIT con el empleo del GPS; la segunda actividad fundamental consistió en diseñar, integrar, estructurar y programar, con la plataforma del SIG, las funciones, operaciones e interfaz gráfica del SIGET.

Para alcanzar el objetivo central del proyecto SIGET consistente en conformar un sistema informático eficiente, versátil y sencillo para el registro, análisis y representación de la información geográfica y estadística asociada al transporte, fue

necesario que a partir de la plataforma cimentada con el inventario de información georreferenciada de la infraestructura para el transporte, generado con el GPS, se procediera de manera paralela a la identificación, acopio e integración de las bases de datos provenientes de fuentes diversas y al diseño de una interfaz para usuario final del sistema.

El SIGET cumple con los objetivos de ser un mecanismo de acceso, consulta, despliegue visual, análisis espacial y representación cartográfica de la información generada por otras fuentes y medios relacionados con el Sector Transporte. Al mismo tiempo, la consecución del SIGET permitió disponer, por primera vez en el país, de la información georreferenciada relativa a los distintos modos de transporte y los componentes infraestructurales asociados, con un adecuado nivel de precisión con base en los levantamientos realizados con GPS.

Por último, debe señalarse que el SIGET exige, para mantenerse vigente, un proceso continuo de mejora que, conforme a sus objetivos, incorpore y adapte la innovación constante de la plataforma tecnológica que lo sustenta, tanto en términos de actualización y ampliación de la información que lo integra, como de la programación y el desarrollo de funciones y operaciones, que respondan a las necesidades planteadas por los usuarios, razón de ser del Sistema de Información Geo estadística para el Transporte.

Un teléfono celular equipado con GPS puede ser utilizado para llevar a cabo la captura y transferencia de datos georreferenciada para ser integrados y visualizados a través de un SIG vía Web, prácticamente en tiempo real, cuando se cuenta con red de datos celular o WI-FI.

Mediante la instalación de la aplicación Siget Móvil en un teléfono celular, el personal en campo de cualquier organismo responsable de la operación de una red de caminos, tendrá la posibilidad de llevar a cabo la georreferenciación de tramos carreteros y otros puntos de infraestructura de transporte y enviar estos datos a una base de datos instalada en un servidor vía Web, mismos que podrán ser visualizados por el personal de oficina a través de un navegador Web o software SIG de escritorio, con la posibilidad de generar reportes y mapas de avance.

La aplicación Siget Móvil se encuentra a disposición del personal de la SCT mediante su descarga en línea a través de la plataforma de distribución de aplicaciones móviles Google Play (véase figura 5).

La información registrada mediante la aplicación Siget Móvil representa una ventaja principalmente para los Centros SCT en cuanto a que apoyará actividades como la conservación y mantenimiento de la infraestructura carretera lo que a su vez impacta en la mejora de actividades tales como:

- El análisis espacial y representación gráfica de las variables técnicas y económicas.

- La diferenciación territorial de las redes de acuerdo con el estado de deterioro de los pavimentos.
- El trazo de metas e itinerarios del personal en campo.
- La calendarización y distribución territorial de las acciones, maquinaria y equipos.
- La priorización de las labores de conservación con base en el criterio del valor de la carga transportada.



Fuente: (Tesis MSI, Elsa Morales, 2015)

Figura 4-2 Siget Móvil en Google Play

Siget Móvil representa una herramienta para el personal en campo que requiere georreferenciar una carretera o tramo en particular de la misma, así como registrar la ubicación precisa de otros elementos de infraestructura para el transporte como lo son puentes, marcas de kilómetro y bancos de material. El usuario de la aplicación Siget Móvil, después de realizar el registro de datos, tiene la posibilidad de enviarlos para su almacenamiento a una base de datos alojada en un servidor vía internet donde son procesados para su visualización en una aplicación en Web.

Para realizar el proceso de georreferenciar una carretera o un tramo de ésta, la aplicación utiliza el GPS del dispositivo para registrar la posición o coordenadas “X” y “Y” en una frecuencia de tiempo de un segundo mientras se realiza el

recorrido de la carretera a bordo de un vehículo. Al finalizar el recorrido, se obtiene un conjunto de registros de posición o puntos que deben ser enviados al servidor para ser procesados y se conviertan en una línea que representará el trazo de la carretera. La aplicación también permite registrar la información que identifica y describe a la carretera georreferenciada.

Para llevar a cabo el registro de la ubicación de elementos de infraestructura puntuales, tales como un puente o un banco de material, la aplicación cuenta con un formulario de captura en el cual el usuario ingresa la información que describe al elemento, misma que es asociada a las coordenadas de ubicación que se toman del receptor GPS del dispositivo móvil.

(Morales, 2015)

4.3 Lenguajes de programación para desarrollo de aplicaciones móviles para Android.

4.3.1 Java.

Java es un lenguaje de programación orientado a objetos el cual se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo. Con la programación en Java, se pueden realizar distintos aplicativos, como son applets, que son aplicaciones especiales, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor WEB. Por lo general los applets son programas pequeños y de propósitos específicos.



La programación en Java, permite tanto el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente y, de esta forma, logra distribuir el trabajo a realizar.

Java ha sido probado, ajustado, ampliado y probado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Java está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible. Al poner a disposición de todo el mundo aplicaciones en entornos heterogéneos, las empresas pueden proporcionar más servicios y mejorar la productividad, las comunicaciones y colaboración del usuario final y reducir drásticamente el costo de propiedad tanto para aplicaciones de usuario como de empresa. Java se ha convertido en un valor impagable para los desarrolladores, ya que les permite:

- Escribir software en una plataforma y ejecutarla virtualmente en otra
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles

- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, micro controladores, módulos inalámbricos, sensores, Gateway, productos de consumo y prácticamente cualquier otro dispositivo electrónico (Programación, 2016)

4.3.2 Open Source

Es una expresión de la lengua inglesa que pertenece al ámbito de la informática. Aunque puede traducirse como “fuente abierta”, suele emplearse en español directamente en su versión original, sin su traducción correspondiente.



Se califica como *open source*, por lo tanto, a los programas informáticos que permiten el acceso a su código de programación, lo que facilita modificaciones por parte de otros programadores ajenos a los creadores originales del software en cuestión.

Es importante distinguir entre el software *open source*, que dispone de la mencionada característica de presentar su código abierto, y el software libre. Existe software libre que no brinda acceso al código, y programas open source que se distribuyen de manera comercial o que requieren de una autorización para ser modificados.

Según Richard Stallman, que es como el padre del software libre, - el "Software Libre" es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en "libre" como en "libertad de expresión", no como en "cerveza gratis" [en inglés una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión]. "Software Libre" se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

1. La libertad de usar el programa, con cualquier propósito (Libertad 1).
2. La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (Libertad 2). El acceso al código fuente es una condición previa para esto.
3. La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (Libertad 3).
4. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (Libertad 4). El acceso al código fuente es un requisito previo para esto.

(Definición, 2015)

4.4 Base de Datos

4.4.1 SQLite



Es un motor de base de datos de SQL incorporado. A diferencia de la mayoría de las otras bases de datos SQL, SQLite no tiene un proceso servidor independiente, lee y escribe directamente a archivos de disco ordinarios. Una base de datos completa de SQL con varias tablas, índices, triggers y vistas, está contenida en un archivo de disco único. La base de datos de formato de archivo es multiplataforma - puede copiar libremente una base de datos entre los sistemas de 32 bits y de 64 bits o entre big-endian y little-endian arquitecturas. Estas características hacen de SQLite una opción popular como un formato de archivo de aplicación. Piense en SQLite no como un reemplazo para Oracle sino como un reemplazo para Fopen.

SQLite es una biblioteca compacta. Con todas las características habilitadas, el tamaño de la biblioteca puede ser inferior a 500KiB, dependiendo de la configuración de la plataforma de destino y de optimización del compilador (El Código de 64 bits es más grande. y algunas optimizaciones del compilador tales como inlining función agresivo y desenrollamiento del bucle puede provocar que el código objeto sea mucho más grande.) Si se omiten las características opcionales, el tamaño de la biblioteca SQLite puede reducirse por debajo 300KiB. SQLite también se puede hacer para funcionar en un espacio mínimo de pila (4KiB) y muy poco montón (100KiB), haciendo SQLite una opción popular motor de base de datos en memoria limitado aparatos tales como teléfonos móviles, PDAs y reproductores MP3. Hay un equilibrio entre el uso de memoria y velocidad. SQLite se ejecuta generalmente más rápido, entre más memoria se le dé. Sin embargo, el rendimiento suele ser bastante bueno, incluso en entornos de poca memoria.

La base de código SQLite es apoyado por un equipo internacional de desarrolladores que trabajan en SQLite a tiempo completo. Los desarrolladores siguen ampliando las capacidades de SQLite y mejoran su fiabilidad y rendimiento al tiempo que mantienen la compatibilidad con la especificación publicada interfaz, sintaxis SQL y base de datos de formato de archivo. El código fuente es totalmente gratuito para cualquier persona que quiera, pero el apoyo profesional también está disponible. (Rómmel, 2007)

4.4.2 MySql

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.



MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre es en el soporte técnico que se ofrece y la posibilidad

de integrar este gestor en un software propietario ya que, de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco, estos elementos faltantes están siendo incorporados, tanto por desarrolladores internos, como por desarrolladores de software libre (Bravo, 2007).

4.4.3 SQL server



Es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

SQL Server usa la arquitectura Cliente / Servidor para separar la carga de trabajo en tareas que corran en computadoras tipo Servidor y tareas que corran en computadoras tipo Cliente:

- El Cliente es responsable de la parte lógica y de presentar la información al usuario. Generalmente, el cliente corre en una o más computadoras Cliente, aunque también puede correr en una computadora Servidor con SQL Server.
- SQL Server administra Bases de Datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc.) entre las múltiples peticiones. La arquitectura Cliente /Servidor permite desarrollar aplicaciones para realizar en una variedad de ambientes.

El motor de base de datos relacional de SQL Server 2000 admite las características necesarias para satisfacer los exigentes entornos de procesamiento de datos, además de proteger la integridad de los datos a la vez que minimiza la carga de trabajo que supone la administración de miles de usuarios modificando la base de datos simultáneamente. Las consultas distribuidas de SQL Server 2000 permiten hacer referencia a datos de varios orígenes como si fuesen parte de una base de datos de SQL Server 2000. Al mismo tiempo, el soporte para transacciones distribuidas protege la integridad de las actualizaciones de los datos distribuidos. La duplicación permite también mantener varias copias de datos a la vez que garantiza que las distintas copias permanezcan sincronizadas. Puede duplicar un conjunto de datos en varios usuarios desconectados móviles, tenerlos trabajando de forma

autónoma y mezclar a continuación sus modificaciones con el publicador (EcuRed, 2009).

4.5 Mapas colaborativos

Los mapas colaborativos son una herramienta poderosa. Antes, quienes tenían el poder político o económico encomendaban la elaboración de mapas para controlar y modificar un territorio de acuerdo con sus intereses. Ahora, la tecnología permite a las personas retratar a su comunidad con libertad, lo que democratiza la generación y acceso a la información sobre un territorio, ampliando el espectro de actores con poder de transformación.

Google Mapas. Permite crear mapas colaborativos de dos formas: por invitación expresa mediante correo electrónico o marcando la opción para que cualquier persona pueda editarlo. Sin embargo, no tiene un sistema de seguimiento y control de cambios, por lo que la edición masiva es posible pero poco eficaz. Permite marcar puntos, rutas y polígonos añadiendo una etiqueta de texto enriquecido. Los datos son exportables en un fichero KML completo, conteniendo los datos de las marcas de posición, polígonos, etiquetas..., o bien como KML de enlace para visualizar en Google Earth. También permite la importación de ficheros KML-KMZ (Mapaton, 2016).

4.5.1 Sitios para creación de mapas Colaborativos.

MapatónCDMX es un esfuerzo conjunto para generar una base de datos abiertos de los recorridos de microbús, autobús y vagoneta de la Ciudad de México a partir del juego y la colaboración ciudadana. Funciona como un juego de ciudad con dos semanas de duración en el que los usuarios trazan recorridos completos del transporte público concesionado para sumar puntos y ganar premios. Es una iniciativa impulsada por 13 organizaciones de los sectores público, privado y social con el objetivo compartido de generar una base de datos abiertos del transporte más usado en la megalópolis.

CÍVICO es una plataforma digital que permite a sus usuarios mapear la ciudad, sumar puntos y ganar premios. Después de experiencias exitosas en Bogotá y Santiago de Chile, CÍVICO llegó a la Ciudad de México. La plataforma lanza misiones a sus usuarios para motivarlos a generar información específica de la ciudad.

Map Maker for Bikes es una herramienta para mapear los carriles de bicicleta de la Ciudad de México. Google LatAm en colaboración con Bicitekas A.C. y otras organizaciones promotoras de la movilidad no motorizada invitan a los ciclistas a mapear, tanto la infraestructura ya existente como la que es posible (Mapaton, 2016).

Map Maker

En Map Maker se puede añadir toda esta información en mapas y ayuda a motos y entusiastas de la caminata fácilmente a descubrir caminos nuevos y frescos para utilizar en un viaje saludable o simplemente para relajarse y descansar. En esta guía rápida, se explica cómo asignar rutas ciclistas de senderos y caminos y el punto de que a la información más detallada que se necesita en el camino. En Map Maker, se puede utilizar el Sendero / Camino entidad lineal para marcar caminos fuera de carretera y crear rutas de caminata en carretera a partir, directamente, la edición de los atributos de la carretera. También puede completar los detalles adicionales sobre los carriles bici urbanos y fuera de la carretera como la idoneidad para bicicletas/peatones, el estado de la ruta, el tipo de superficie, etc. Llenar estos datos puede ayudar a generar mapas de Google correspondientes a direcciones o recorridos para caminar por estos caminos.

4.5.2 Formatos de archivo para la exportación

Se pueden exportar las rutas con estos formatos:

- **KML**: contiene ubicaciones, marcadores, fotos y datos de sensores.
- **GPX**: contiene ubicaciones y marcadores, pero no datos de sensores, como la frecuencia cardíaca, la cadencia en bicicleta, la cadencia corriendo y la energía.
- **CSV**: contiene ubicaciones, marcadores y datos de sensores.
- **TCX**: contiene ubicaciones y algunos datos de sensores (Google, 2016).

4.5.2.1 GeoJson

Es un formato para la codificación de una variedad de estructuras de datos geográficos. Un objeto GeoJSON puede representar una geometría, una característica o un conjunto de características. GeoJSON soporta los siguientes tipos de geometría: Point, cadena lineal, polígono, multipunto, MultiLineString, MultiPolygon, y GeometryCollection. Características en GeoJSON contienen un objeto de geometría y propiedades adicionales y una colección de fenómenos representa una lista de características.

Una estructura de datos GeoJSON completa es siempre un objeto (en términos JSON). En GeoJSON, un objeto consiste en una colección de pares nombre / valor - también llamados miembros. Para cada miembro, el nombre es siempre una cadena. Valores de los miembros son o bien una cadena, número, objeto, matriz o uno de los literales: verdadero, falso y nula. Una matriz se compone de elementos, donde cada elemento es un valor como se describe anteriormente.

GeoJSON siempre consta de un solo objeto. Este objeto (en adelante, el objeto GeoJSON abajo) representa una geometría, característica, o una colección de características.

El objeto GeoJSON puede tener cualquier número de miembros (pares de nombre / valor).

El objeto GeoJSON debe tener un miembro con el nombre de "tipo". El valor de este elemento es una cadena que determina el tipo de objeto GeoJSON.

El valor del elemento de tipo debe ser uno o "*Feature Collection*". El caso de los valores de los miembros de tipo debe ser como se muestra aquí.

Un objeto puede tener un GeoJSON opcionales "CRS" miembro, el valor de los cuales debe ser un objeto del sistema de referencia de coordenadas (ver figura/tabla 3. Coordinar Objetos de sistema de referencia).

Un objeto GeoJSON puede tener un "Bbox" miembro, cuyo valor debe ser una matriz cuadro delimitador (GeoJson, 2008).

5 Supuestos Técnicos

El proyecto elaborado, denominado “Modelo para el registro Nacional de Infraestructura ciclista a través de una aplicación Android”, plantea la necesidad de recopilar datos de manera real sobre las ciclovías, puentes, ciclo puertos, bici estacionamientos, centros de transferencia e interconexión modal para localizarlos de una manera más sencilla y rápida.

Para ello se comenzó el prototipo de una aplicación móvil de nombre BiSiget-Movil que es una continuación de un proyecto elaborado por la MSI Elsa Morales Bautista, la cual es una aplicación diseñada para el registro de carreteras a nivel Nacional de nombre Siget Móvil.

Para poder poner en marcha el prototipo es necesario un dispositivo móvil (celular) que cuente con un sistema Android cuya versión mínima sea la v2.3.3 (GingerBread).

Para elaborar el nuevo proyecto fue necesario rediseñar la Aplicación ya antes mencionada con el IDE de programación Android Studio con una API menor a la versión 22 y 23 puesto que la compatibilidad en codificación es diferente a las versiones anteriores, el proyecto está programado con la API versión 19 así como el SDK mínimo del Android 2.3.3. También se utilizó el SGBD SQLite creando una base de datos corta de solo 4 tablas.

Es importante mencionar que los datos registrados en la aplicación se guardarán en el dispositivo móvil en el que se encuentre instalada y se deberán exportar al formato GeoJson antes de enviar la información a la base de datos del IMT.

6 Descripción de la Propuesta de Solución

6.1 Requerimientos

- › Realizar una adaptación de la aplicación Siget-Móvil de la MSI Elsa Morales Bautista.
- › Desarrollar una plantilla para la captura de Información de campo.
- › Diseñar y Crear base de datos interna para la aplicación en SQLite.
- › Exportar los datos capturados a formato GeoJSON.
- › Registrar Ciclovias e infraestructura Ciclista.

6.2 Herramientas Informáticas

- Android Studio.
- SQLite.
- Mozilla Firefox.

6.3 Características de los requerimientos.

Tabla 6.1.Característica de Pantalla Bienvenida

Nombre	Bienvenido
Descripción	Pantalla principal de la Aplicación donde se accede a la pantalla principal de funciones.
Estado	Aprobado.
Nivel de Prioridad	Medio, es una pantalla de inicio y Bienvenida.
Nivel de Riesgo	Medio, al programar de nuevo la aplicación puede ser retirada.

Tabla 6.2. Característica pantalla BiSiget-Movil

Nombre	BiSiget-Movil
Descripción	Pantalla Principal de funciones para la aplicación.
Estado	Aprobado
Nivel de Prioridad	Alta, en esta pantalla se acceden la mayoría de las funciones que contiene la aplicación.
Nivel de Riesgo	Crítico, si no existiese no se podría acceder a ninguna función.

Tabla 6-3. Característica de Activar/Desactivar GPS

Nombre	Activar GPS
Descripción	Aquí se activa el GPS para capturar las coordenadas utilizadas para el registro de la localización.
Estado	Aprobado.
Nivel de Prioridad	Alta. Con esto se localiza la ciclovías.
Nivel de Riesgo	Crítico. Si no existiese no se podría tomar el registro de las coordenadas y localización de las ciclovías o infraestructura.

Tabla 6-4. Característica de Inicio de captura de ciclovía

Nombre	Inicio Captura Ciclovía
Descripción	En esta función se inicia el registro de las coordenadas para la ubicación de la ciclovía.
Estado	Aprobado.
Nivel de Prioridad	Alto, proporciona coordenadas.
Nivel de Riesgo	Crítico, las coordenadas son necesarias.

Tabla 6-5. Característica de Terminar Captura de Ciclovía

Nombre	Terminar captura de ciclovía
Descripción	Registro de la ciclovía.
Estado	Aprobado.
Nivel de Prioridad	Alto, Aquí se realiza el registro de la ciclovía o cicloruta.
Nivel de Riesgo	Crítico, sin esta pantalla no se puede realizar el registro.

Tabla 6-6. Característica de Infraestructura

Nombre	Infraestructura
Descripción	Registro de Infraestructura.
Estado	Aprobado.
Nivel de Prioridad	Alto. Aquí se realiza el registro de infraestructura como puentes, ciclo puentes, etc.
Nivel de Riesgo	Crítico, si no existe no se puede registrar la infraestructura.

Tabla 6-7. Característica de Consulta de Datos Capturados.

Nombre	Exporta Datos
Descripción	Exporta datos de coordenadas al formato GeoJson para poder interpretarlos en google maps.
Estado	Aprobado.
Nivel de Prioridad	Alto, es necesario para localización de las rutas.
Nivel de Riesgo	Crítico si no funciona no puede enviar los datos.

Tabla 6-8. Característica de Exportar CVS

Nombre	Enviar Datos al SIG
Descripción	Envía datos al sistema del SIG en el IMT.
Estado	Incompleto.
Nivel de Prioridad	Alta. Con esto se puede enviar los datos capturados a un servidor del IMT
Nivel de Riesgo	Crítico. Sin esto no se envía nada.

Tabla 6-9. Característica de Enviar Datos Al SIG

Nombre	Consulta de Datos Capturados
Descripción	Consulta de coordenadas, Infraestructura, Banco de material, Ciclovías, etc.
Estado	Aprobado.
Nivel de Prioridad	Medio. Aquí se visualiza los datos ya registrados.
Nivel de Riesgo	Significativo. Aquí se puede apreciar los datos que se han capturado.

Tabla 6-10. Característica de Borrar datos locales.

Nombre	Borrar datos locales
Descripción	Borra los datos guardados en el dispositivo móvil.
Estado	Aprobado.
Nivel de Prioridad	Bajo. La aplicación necesita borrar datos para evitar saturación.
Nivel de Riesgo	Normal.

6.4 Definición de los Datos de Entrada (tablas BD).

Carretera. Captura de Datos para la ciclovía, descripción de los campos.

Tabla 6-11. Datos de entrada formulario Ciclovía (carretera)

Campo	Nombre en la BD	Tipo
Id	_id	Numérico. Campo que requiere Android para acceder a los datos almacenados en la tabla.
Código de Carretera	codigoC	Alfanumérico. Clave alfanumérica de la carretera señalizada en la misma carretera
Nombre de la Carretera	nombre	Texto. Se refiere al nombre oficial que le asigna la SCT.
Administración	administracion	Texto. Responsabilidad de mantenimiento. (Opciones: Federal, Estatal, Municipal, Particular)
Número de carriles	nCarriles	Numérico. Número total de carriles
Tipo de pavimento	tipoPav	Texto. Material de la superficie de rodamiento
Longitud (x)	longX	Decimal. Coordenada x (del punto donde se realiza el llenado de datos de la carretera)
Latitud (y)	latY	Decimal. Coordenada y (del punto donde se realiza el llenado de datos de la carretera)

Infraestructura. Descripción de campos que integran la tabla.

Tabla 6-12. Datos de entrada formulario Infraestructura

Campo	Nombre en la BD	Tipo
Id	_id	Campo que requiere Android para acceder a los datos almacenados en la tabla.
Tipo de la Infraestructura	tipol	Tipo de la infraestructura, Alcantarilla, puente, ciclo puerto, bici estacionamiento, centros de transferencia interconexión Modal
Nombre del Puente	noml	Sustantivo propio con el cual se conoce el objeto espacial
Tamaño del puente	longl	Longitud total del puente. (Valores: Chico 6-30m, Mediano 30-100m, Grande más de 100m)
Estructura del puente	estrl	Tipo de material con el que se encuentra construida la estructura
Marca de kilómetro	mkm	Punto en donde la SCT marca el kilómetro en determinado tramo de la red carretera.
Señalética	senaletica	Señalización para anunciar que es una ciclovía. Vertical u Horizontal
Longitud (x)	longX	Coordenada x
Latitud (y)	latY	Coordenada y

Coordenadas. Descripción de los campos que integran la tabla de coordenadas.

Tabla 6-13. Datos entrada Formulario Coordenadas

Campo	Nombre en la BD	Tipo
Longitud	longX	Coordenada x
latitud	latY	Coordenada y

6.5 Diagramas UML.

6.5.1 Diagrama de Caso de Uso

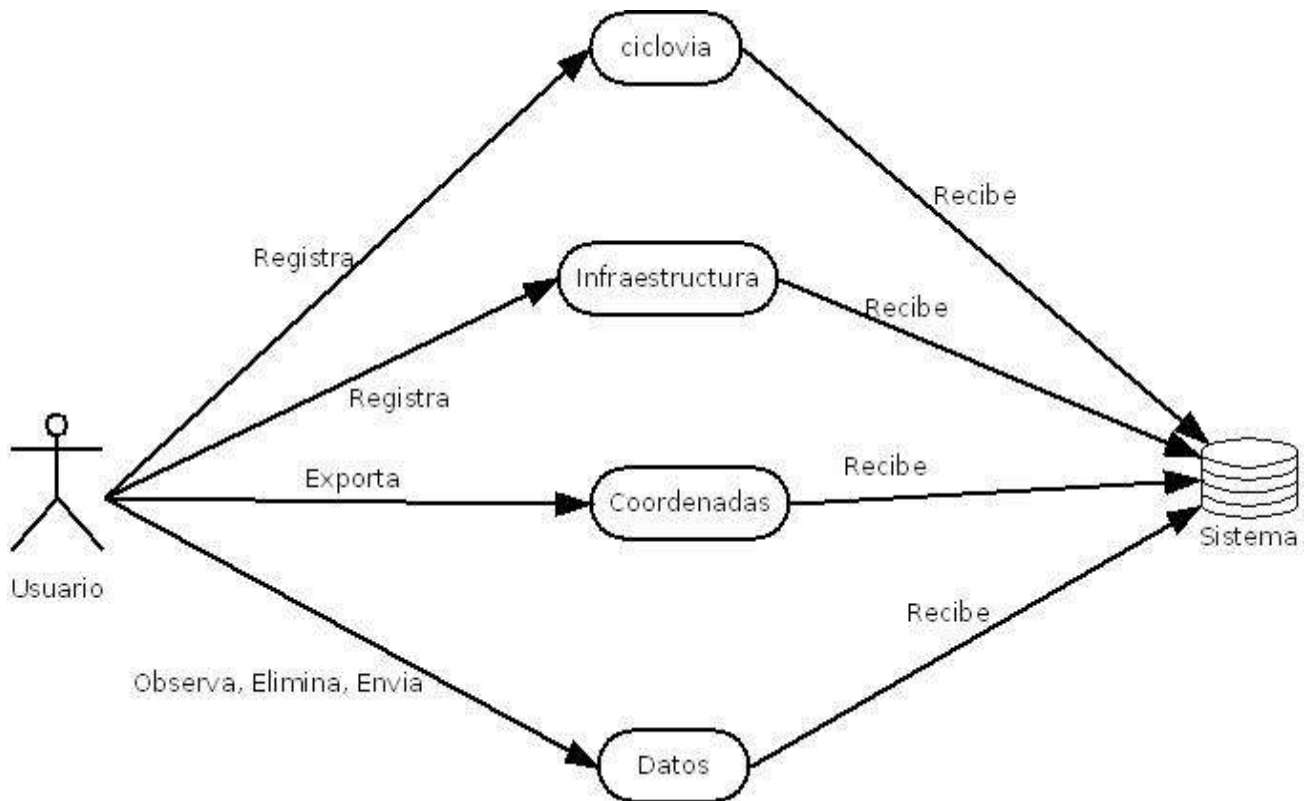


Figura 6-1. Diagrama de clases

6.5.2 . Descripción caso de uso

Tabla 6-14. Descripción Registrar Ciclovía

Caso de Uso	Registrar Ciclovía
Actores	Usuario
Tipo	Ventana de Sistema
Propósito	Agregar características que componen la ciclovía así como su ubicación.
Resumen	Lo registra el Usuario y queda almacenado en los datos locales.
Precondiciones	Se tiene que iniciar el registro de coordenadas de la ciclovía.
Flujo principal	Registro de datos no necesita autenticación.
Excepciones	Se registran las coordenadas.

Tabla 6-15. Descripción Registrar Infraestructura

Caso de Uso	Registrar Infraestructura
Actores	Usuario
Tipo	Ventana de Sistema
Propósito	Agregar características que componen la infraestructura así como su ubicación.
Resumen	Lo registra el Usuario y queda almacenado en los datos locales.
Precondiciones	Se tiene que iniciar el registro de coordenadas de la Infraestructura
Flujo principal	Registro de datos no necesita autenticación.
Excepciones	Se registran las coordenadas. Es necesario elegir el tipo de Infraestructura.

Tabla 6-16. Descripción Exporta Coordenadas

Caso de Uso	Exporta Coordenadas
Actores	Usuario
Tipo	Ventana de Sistema
Propósito	Exporta Datos a formato CSV.
Resumen	Los datos registrados anteriormente son convertidos a otro formato.
Precondiciones	Se necesitan datos registrados.
Flujo principal	Registro de datos no necesita autenticación.
Excepciones	Se registran las coordenadas.

Tabla 6-17. Descripción Observar Datos

Caso de Uso	Observa datos
Actores	Usuario, Sistema
Tipo	Ventana de Sistema
Propósito	Observa datos ya registrados.
Resumen	Los datos registrados anteriormente se pueden observar.
Precondiciones	Se necesitan datos registrados.
Flujo principal	No necesita autenticación.
Excepciones	Solo es una lista.

Tabla 6-18. Descripción Envía Datos

Caso de Uso	Envía datos
Actores	Usuario
Tipo	Envío de datos
Propósito	Observa datos ya registrados.
Resumen	Los datos registrados anteriormente son enviados.
Precondiciones	Se necesitan datos registrados.
Flujo principal	Necesita una autenticación del usuario.
Excepciones	Solo es una lista.

Tabla 6-19, Descripción Eliminar Datos

Caso de Uso	Eliminar datos
Actores	Usuario
Tipo	Función de Sistema
Propósito	Elimina todos los datos registrados en la base de datos local.
Resumen	Los datos registrados anteriormente son los eliminados.
Precondiciones	Se necesitan datos registrados.
Flujo principal	No necesita autenticación.
Excepciones	Solo es una lista.

Tabla 6-20. Descripción Recepción Datos

Caso de Uso	Recibe datos
Actores	Sistema IMT.
Tipo	Recepción de datos
Propósito	Observa datos ya registrados.
Resumen	Los datos son enviados al sistema de un servidor externo.
Precondiciones	Se necesitan datos registrados.
Flujo principal	Se necesita autenticación.
Excepciones	Solo es una lista.

6.5.3 Diagrama de Clases

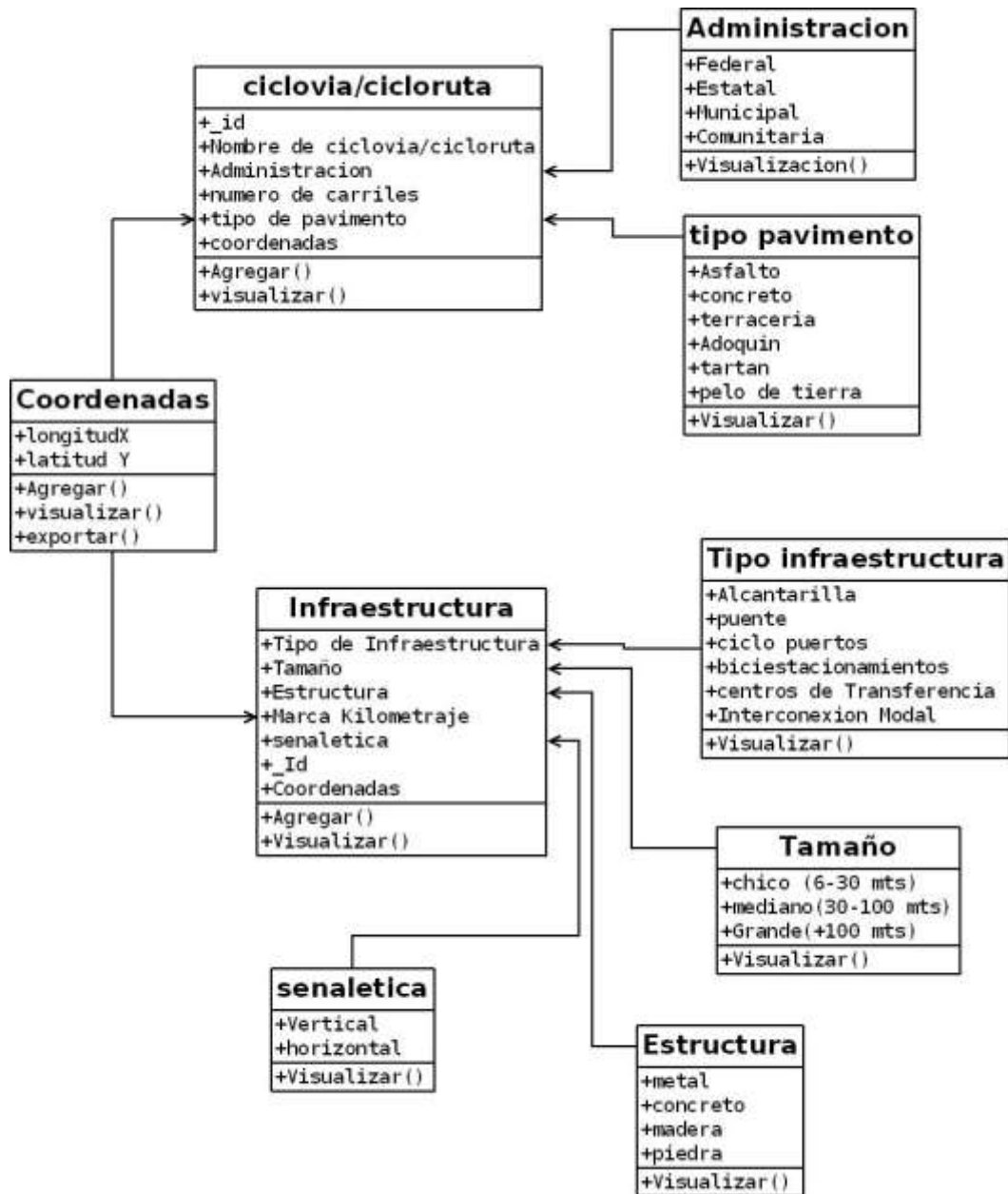


Figura 6-2. Diagrama de clases

6.5.4 Diagrama de actividades

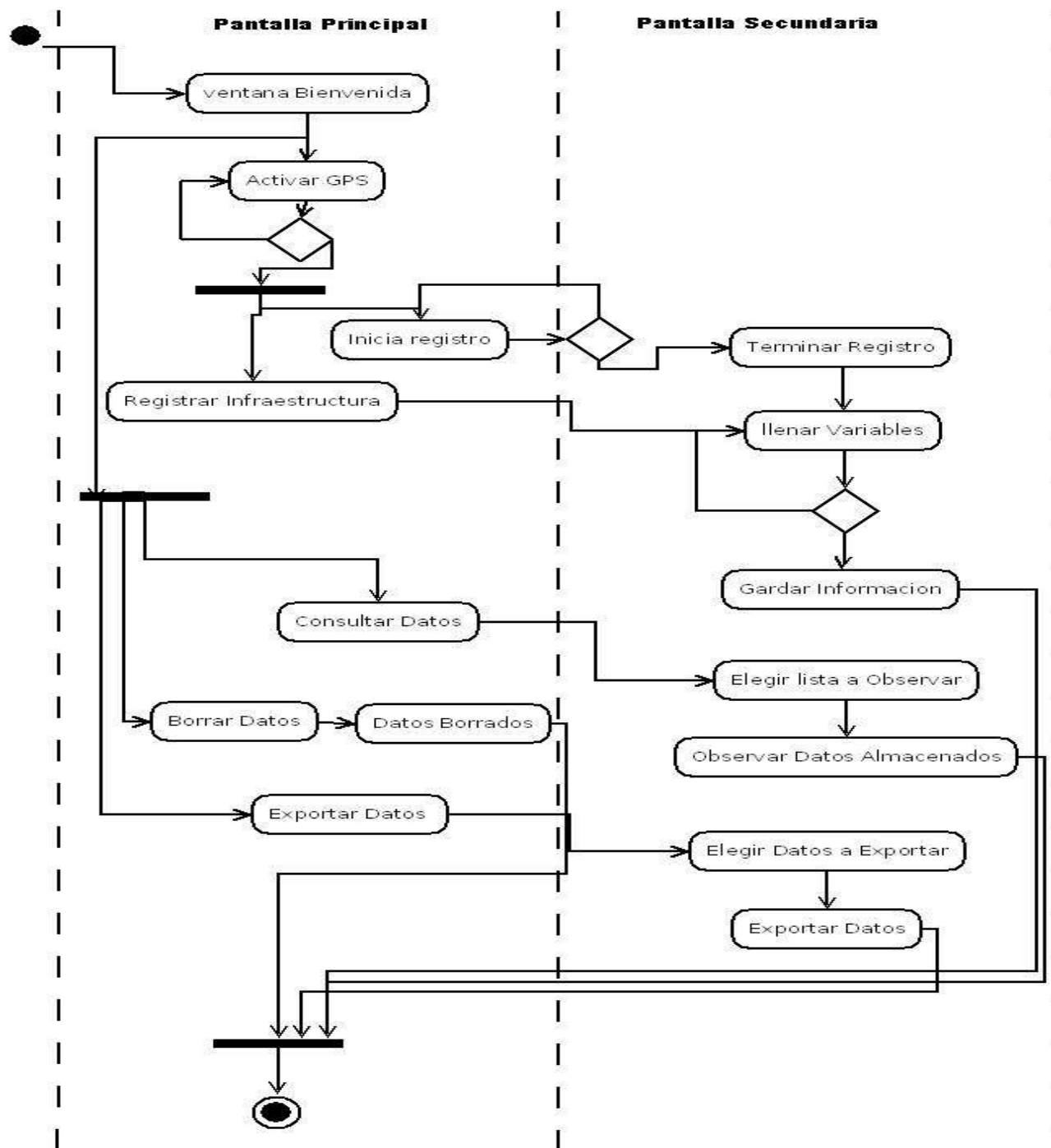


Figura 6-3. Diagrama de actividades.

6.6 Diseño de Base de Datos

6.6.1 Script de Base de Datos

Para la creación de la BD en SQLite se requirió elaborarla con características para Android utilizando las siguientes instrucciones:

- CREATE TABLE “android_metadata”(“locale” TEXT DEFAULT ‘en_US’)
- INSERT INTO “android_metadata” VALUES (‘en_US’)

Nombre de Base de datos: **Bi-siget**

```
CREATE TABLE ‘carretera’ (  
  `_id`    INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
  `codigoC`  VARCHAR,  
  `nombre`   VARCHAR,  
  `administracion`  VARCHAR,  
  `nCarriles` INTEGER,  
  `tipoPav`  NUMERIC,  
  `longX`    DOUBLE,  
  `latY`     DOUBLE  
);
```

```
CREATE TABLE ‘coordenadas’ (  
  `_id`    INTEGER NOT NULL,  
  `latY`   DOUBLE,  
  `longX`  DOUBLE,  
  PRIMARY KEY(_id)  
);
```

```
CREATE TABLE ‘infraestructura’ (  
  `_id`    INTEGER NOT NULL,  
  `Tipol`  VARCHAR,  
  `noml`   VARCHAR,  
  `longl`  VARCHAR,  
  `estrl`  VARCHAR,  
  `mKm`    INTEGER,  
  `bMat`   VARCHAR,  
  `longX`  DOUBLE,  
  `latY`   DOUBLE,  
  PRIMARY KEY(_id)  
);
```

6.7 Diseño de pantallas



Figura 6-4. Pantalla principal de la aplicación.

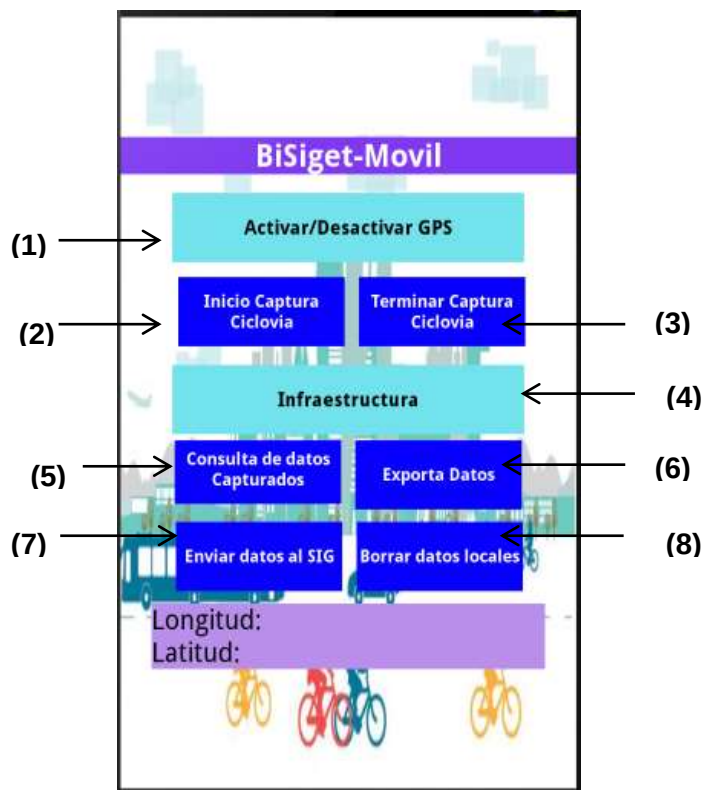


Figura 6-5. Pantalla principal de Funciones

1. Botón para Activar o Desactivar el GPS
2. Botón para comenzar el inicio de la captura de las coordenadas de la ciclovía.
3. Botón para Terminar la captura de Coordenadas y registrar ciclovía dirige a la pantalla de registro (véase figura 8).
4. Botón para Registrar Infraestructura dirige a la pantalla de registro Infraestructura (véase figura 9).
5. Botón para la consulta de Datos dirige a pantalla Consulta de Datos (véase figura 10)
6. Botón para exportar datos de coordenadas a GeoJSON dirige a pantalla de Exportar (véase figura 11)
7. Botón para ingresar a la pantalla login y enviar datos (véase figura 12)
8. Botón para Borrar datos locales alojados en el dispositivo móvil.

5554:prueba

3:45

Registro

(1) → Codigo de la Ciclovía/cicloruta

Nombre Ciclovía/Cicloruta ← (2)

(3) → Seleccione el tipo de Administracion

Seleccione el Numero de Carriles ← (4)

(5) → Seleccione el tipo de Pavimento

Registrar ← (6)

Figura 6-6. Formulario de Registro de Ciclovía.

1. Caja de texto para ingresar el código de la ciclovía
2. Caja de texto para ingresar el nombre de la ciclovía.
3. Lista desplegable para seleccionar el tipo de Administración de la Ciclovía.
4. Lista desplegable para seleccionar el Número de carriles con las que cuenta de la Ciclovía.
5. Lista desplegable para seleccionar el tipo de Pavimento de la Ciclovía
6. Botón para Registrar todos los datos Ingresados a la base de datos.

Figura 6-7. Formulario de registro de Infraestructura

1. Lista desplegable para seleccionar el Tipo de Infraestructura a la que pertenece.
2. Caja de texto para ingresar el nombre de la Infraestructura.
3. Lista desplegable para seleccionar el tipo de Administración de la Ciclovía.
4. Lista desplegable para seleccionar el Tipo de estructura con la que cuenta.
5. Caja de Texto para ingresar el Kilometraje de la Infraestructura.

6. Lista desplegable para seleccionar el Tipo de Señalética.
7. Botón para Registrar todos los datos Ingresados a la base de datos.

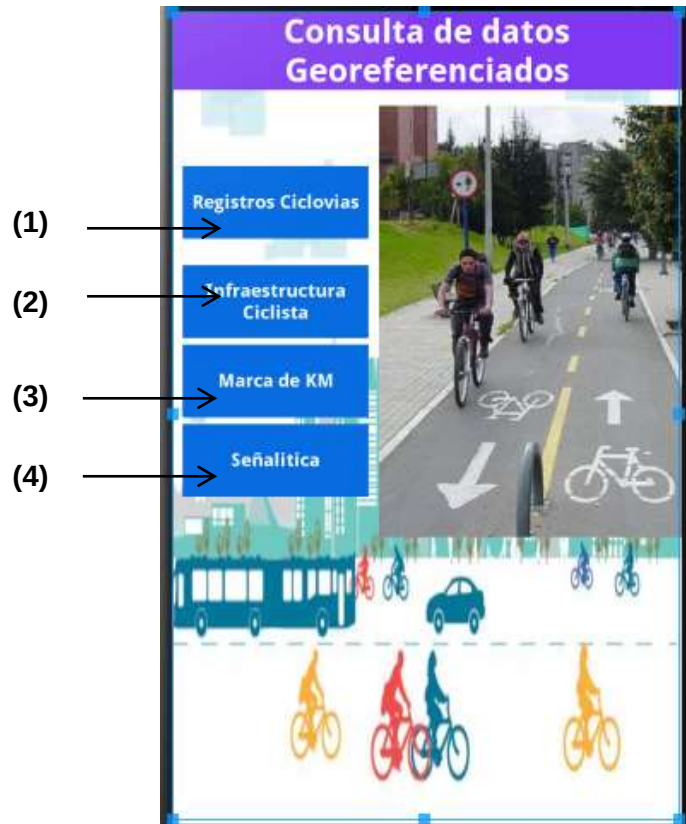


Figura 6-8. Vista de datos

1. Botón que muestra Datos locales Registrados de Ciclovías.
2. Botón que muestra Datos locales Registrados de Infraestructura.
3. Botón que muestra Datos locales Registrados de marcas de kilometraje.
4. Botón que muestra Datos locales Registrados de las ciclovías que cuentan con Señalización.

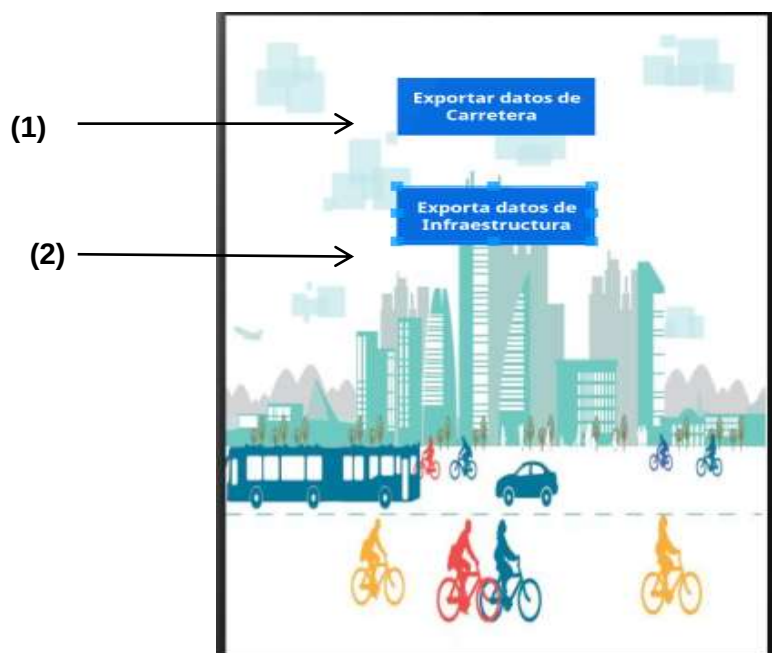


Figura 6-9. Pantalla de Exportar Datos

1. Botón que Exporta Registrados de Ciclovías a formato GeoJson.
2. Botón que Exporta Registros de Infraestructura a GeoJson.

6.8 Codificación.

6.8.1 Estructura del proyecto



Figura 6-10. Icono de Aplicación

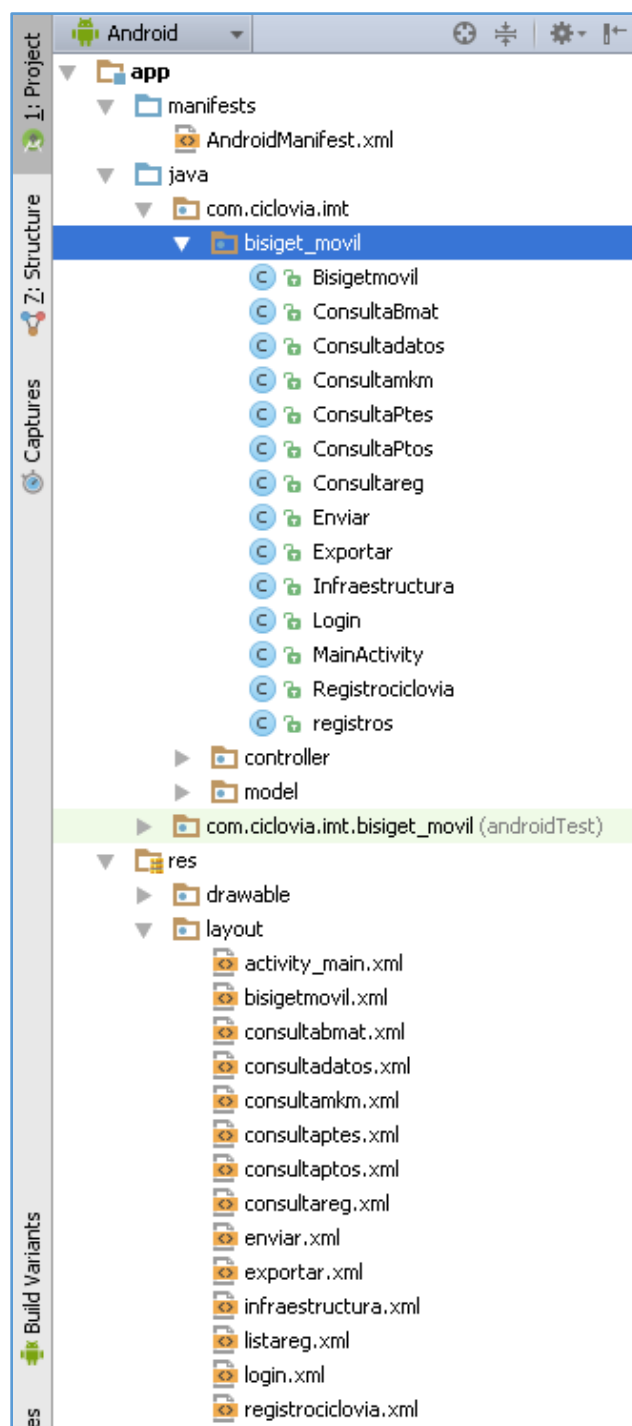


Figura 6-11. Estructura de Proyecto

7 Evaluación

7.1 Pruebas

Con la finalidad de poder localizar errores que afecten el uso y rendimiento del programa o que puedan presentar un probable riesgo para las decisiones de las funciones para la cual fue diseñada la herramienta es necesario llevar a cabo pruebas de funcionalidad. Las pruebas que se realizan son las siguientes:

- A. Interfaces de Usuario (Vista y carga de datos)
- B. Módulos de Codificación

Tabla 7-1. Tabla de control de pruebas.

Fecha	Lugar	Modulo	Prueba	Responsable	Observaciones
	IMT	Bienvenida	A	T.S.U. Diana García Ramírez	Se necesita diseño responsivo.
	IMT	Bisiget	A	T.S.U. Diana García Ramírez	Algunos botones no cumplen función de ligamiento de pantallas.
	IMT	Terminar Captura	A	T.S.U. Diana García Ramírez	Cajas de texto no encajan
	IMT	Infraestructura	A	T.S.U. Diana García Ramírez	Listas desplegables no funcionan, validar listas desplegables.
	IMT	Consulta de Datos	A	T.S.U. Diana García Ramírez	Imagen no se acopla la pantalla

	IMT	Exportar datos	A	T.S.U. Diana García Ramírez	Legibilidad de texto
	IMT	Enviar Datos al SIG	A	T.S.U. Diana García Ramírez	Posicionamiento de botones
	IMT	Login	A	T.S.U. Diana García Ramírez	Validar cajas de texto
	IMT	Activar/ Desactivar GPS	B	T.S.U. Diana García Ramírez	No activa el GPS
	IMT	Inicio Captura Ciclovía	B	T.S.U. Diana García Ramírez	No inicia la captura
	IMT	Terminar Captura Ciclovía	B	T.S.U. Diana García Ramírez	No registra los datos en la base de datos
	IMT	Infraestructura	B	T.S.U. Diana García Ramírez	No registra Nada, cierra la aplicación
	IMT	Consulta de Datos capturados	B	T.S.U. Diana García Ramírez	No existen datos
	IMT	Exporta Datos	B	T.S.U. Diana García Ramírez	No convierte los datos
	IMT	Enviar datos al SIG	B	T.S.U. Diana García Ramírez	No está completo
	IMT	Borrar datos Locales	B	T.S.U. Diana García Ramírez	Incompleto

	IMT	Registros	B	T.S.U. Diana García Ramírez	No se observan los datos
	IMT	Puntos X, Y	B	T.S.U. Diana García Ramírez	No se observan los datos
	IMT	Puentes	B	T.S.U. Diana García Ramírez	No se observan los datos
	IMT	Marca de km	B	T.S.U. Diana García Ramírez	No se observan los datos
	IMT	Señalética	B	T.S.U. Diana García Ramírez	No se observan los datos

Tabla 7-2. Pruebas y errores encontrados

7.2 Realización de Correcciones.

Después de realizar las pruebas anteriores y tomar nota de los errores encontrados se procede a corregir los errores encontrados, con el objetivo de mejorar el rendimiento de la herramienta y su capacidad de respuesta en los aspectos de programación.

Se puede afirmar que se realizaron las siguientes correcciones:

Mejoramiento el diseño responsivo de cada pantalla para su adaptabilidad en diferentes tamaños de dispositivos móviles.

Mejoramiento de cajas de texto y listas desplegables, así como la función y validación de cada uno de ellos.

Mejoramiento de la posición de cada elemento de las pantallas con una mejor ubicación para el agrado de los usuarios.

Una mejor opacidad de los elementos de color.

Con base en esto se puede decir que hubo una mejora notable en la calidad de la aplicación, así como la personalización del icono de la aplicación acorde a la función y temática de la aplicación.

Conclusiones

Es evidente que la tecnología está siempre en un estado cambiante, por tanto, es necesario adaptarse cuanto antes a ella. En este proyecto se decidió que es necesario aprovechar las posibilidades que brinda un sistema de Información geográfica y lo que se puede hacer con ella.

Este trabajo permite registrar y localizar, de una manera más sencilla la infraestructura ciclista que existe en el país para así poder hacer un uso más provechoso de ella.

Este trabajo se inspira en la aplicación creada por la M.S.I. Elsa morales Bautista denominada “SIGET Móvil” que cumple con los objetivos de ser un mecanismo de acceso, consulta, despliegue visual, análisis espacial y representación cartográfica de la información generada por otras fuentes y medios relacionados con el Sector Transporte. Alcanzando el objetivo central del proyecto SIGET consistente en conformar un sistema informático eficiente, versátil y sencillo para el registro, análisis y representación de la información geográfica y estadística asociada al transporte.

Anexo 1. Pantallas

Nombre en proyecto: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.ciclovia.imt.bi_siget.MainActivity"
    android:background="@drawable/fondo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentTop="true">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:orientation="vertical">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="BiSiget-Movil"
                android:id="@+id/tvTitulo"
                android:textSize="@dimen/activity_vertical_margin"
                android:textStyle="bold"
                android:textColor="#6532a8" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Bienvenido"
                android:id="@+id/tvBienvenido"
                android:textColor="#000000"
                android:textStyle="bold"
                android:textSize="@dimen/abc_action_bar_stacked_max_height"
                android:layout_marginTop="60dp" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Comenzar"
                android:id="@+id/btnComenzar"
                android:textColor="#f9fafc"
                android:textStyle="bold"
```

```
        android:background="#066ce0"
        android:width="@dimen/size_buton_wheigth"
        android:height="@dimen/size_buton"
        android:textSize="@dimen/activity_horizontal_margin"
        android:layout_marginTop="60dp" />
    </LinearLayout>
</ScrollView>
</RelativeLayout>
```

Nombre en proyecto: bisigetmovil.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo"
    android:weightSum="1">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollView2">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center"
            android:gravity="center">
            <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center">

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:textAppearance="?android:attr/textAppearanceLarge"
                    android:text="BiSiget-Movil"
                    android:id="@+id/textView"
                    android:textStyle="bold"
                    android:background="#8039f1"
                    android:textSize="25dp"
                    android:layout_gravity="center_horizontal"
                    android:width="@dimen/abc_config_prefDialogWidth"
                    android:textColor="#ffffff"
                    android:gravity="center"
                    android:layout_marginTop="0dp" />

                <Button android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Activar/Desactivar GPS"
                    android:id="@+id/btnActivar"
                    android:layout_gravity="center_horizontal"
                    android:gravity="center"
                    android:textSize="16dp"
                    android:textStyle="bold"
                    android:background="#72e2ed"
                    android:width="@dimen/abc_panel_menu_list_width"
```

```

        android:height="@dimen/size_buton"
        android:singleLine="false"
        android:layout_marginTop="15dp" />
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginTop="12dp"
    android:gravity="center"
    android:layout_marginLeft="-5dp">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Inicio Captura Ciclovía"
        android:id="@+id/btnInicio"
        android:width="@dimen/size_buton_wheigth"
        android:height="@dimen/size_buton"
        android:background="#0800ff"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:layout_marginLeft="15dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Terminar Captura Ciclovía"
        android:id="@+id/btnFin"
        android:width="@dimen/size_buton_wheigth"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:background="#0800ff"
        android:layout_gravity="top"
        android:layout_marginLeft="12dp"
        android:height="@dimen/size_buton" />
</LinearLayout>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Infraestructura"
        android:id="@+id/btninfraest"
        android:width="@dimen/abc_panel_menu_list_width"
        android:height="@dimen/size_buton"
        android:background="#72e2ed"
        android:textSize="16dp"
        android:textStyle="bold"
        android:layout_marginTop="15dp"
        android:layout_gravity="center_horizontal" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="5dp"
    android:gravity="center">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Consulta de datos Capturados"
        android:id="@+id/btnConsulta"

```

```
        android:background="#0800ff"
        android:textStyle="bold"
        android:textColor="#ffffff"
        android:width="@dimen/size_buton_wheigth"
        android:height="@dimen/size_buton"
        android:layout_marginTop="-5dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Exporta CVS"
    android:id="@+id/btnExport"
    android:width="@dimen/size_buton_wheigth"
    android:height="@dimen/size_buton"
    android:textColor="#ffffff"
    android:textStyle="bold"
    android:background="#0800ff"
    android:layout_marginLeft="12dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:layout_marginLeft="-7dp">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enviar datos al SIG"
        android:id="@+id/btnEnviar"
        android:width="@dimen/size_buton_wheigth"
        android:height="@dimen/size_buton"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:background="#0800ff"
        android:layout_marginLeft="15dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Borrar datos locales"
        android:id="@+id/btnBorrar"
        android:width="@dimen/size_buton_wheigth"
        android:height="@dimen/size_buton"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:background="#0800ff"
        android:layout_marginLeft="12dp" />

</LinearLayout>
<LinearLayout
    android:orientation="vertical"
    android:layout_width="331dp"
    android:layout_height="match_parent"
    android:layout_marginTop="10dp"
    android:gravity="center">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
```

```

        android:text="Longitud:"
        android:id="@+id/txtlongitud"
        android:background="#b98feb"
        android:width="@dimen/size_buton_wheigth_Large"
    />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Latitud:"
        android:id="@+id/txtlatitud"
        android:background="#b98feb"
        android:width="@dimen/size_buton_wheigth_Large"/>

    </LinearLayout>
</LinearLayout>
</ScrollView>
</RelativeLayout>

```

Nombre en proyecto: registrociclovía.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollView3" >

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:textAppearance="?android:attr/textAppearanceLarge"
                    android:text="Registro"
                    android:id="@+id/txtRegistro"
                    android:textStyle="bold"
                    android:background="#8039f1"
                    android:textSize="25dp"
                    android:width="@dimen/abc_config_prefDialogWidth"
                    android:textColor="#ffffff"
                    android:gravity="center" />

                <EditText
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/txtCodigo"

```

```
        android:hint="Codigo de la Ciclovia/cicloruta"
        android:width="@dimen/size_buton_wheigth_Large"
        android:gravity="center"
        android:layout_gravity="center_horizontal" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/txtNombre"
    android:width="@dimen/size_buton_wheigth_Large"
    android:hint="Nombre Ciclovia/Cicloruta"
    android:gravity="center"
    android:layout_gravity="center_horizontal" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spAdministracion"
    android:layout_gravity="center_horizontal" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spCarriles"
    android:layout_gravity="center_horizontal" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/sptipoPavimento"
    android:layout_gravity="center_horizontal" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Registrar"
    android:id="@+id/btnRegistro"
    android:background="#066ce0"
    android:textColor="#f9fafc"
    android:width="@dimen/abc_action_bar_stacked_tab_max_width"
    android:textStyle="bold"
    android:textSize="16dp"
    android:layout_centerHorizontal="true"
    android:height="@dimen/size_buton"
    android:layout_gravity="center_horizontal" />
</LinearLayout>

</LinearLayout>
</ScrollView>
</RelativeLayout>
```

Nombre en proyecto: infraestructura.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo">

    <ScrollView
```



```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/scrollView7"
    android:layout_alignParentTop="true"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Infraestructura"
    android:id="@+id/tvInfraestructura"
    android:background="#8039f1"
    android:textColor="#ffffff"
    android:width="@dimen/abc_config_prefDialogWidth"
    android:textStyle="bold"
    android:gravity="center"
    android:textSize="25dp" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spTipoI"
    android:layout_gravity="center_horizontal" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/txtNombreI"
    android:width="@dimen/size_buton_wheigth_Large"
    android:hint="Nombre de la Infraestructura"
    android:textStyle="bold"
    android:layout_gravity="center_horizontal" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tvDetalle"
    android:width="@dimen/size_buton_wheigth_Large"
    android:layout_gravity="center_horizontal" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spTamanio"
    android:layout_gravity="center_horizontal" />

```

```
<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spEstructura"
    android:layout_gravity="center_horizontal" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/txtKilometraje"
    android:width="@dimen/size_buton_wheigth_Large"
    android:hint="Marca Kilometraje"
    android:textStyle="bold"
    android:layout_gravity="center_horizontal" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spMaterial"
    android:layout_gravity="center_horizontal" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Latitud:"
    android:id="@+id/tvlatitudI"
    android:width="@dimen/abc_config_prefDialogWidth"
    android:background="#b98feb"
    android:textColor="#000000"
/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Longitud:"
    android:id="@+id/tvLongitudI"
    android:width="@dimen/abc_config_prefDialogWidth"
    android:background="#b98feb"
    android:textColor="#000000" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Registrar"
    android:id="@+id/btnRegistroInf"
    android:textStyle="bold|normal"
    android:textColor="#f9fafc"
    android:gravity="center"
    android:background="#066ce0"
    android:textSize="16dp"
    android:width="@dimen/abc_action_bar_stacked_tab_max_width"
    android:height="@dimen/size_buton"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="5dp" />
</LinearLayout>
</LinearLayout>
</ScrollView>
</RelativeLayout>
```

Nombre en proyecto: consultadatos.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollView4">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Consulta de datos Georeferenciados"
                android:id="@+id/tvTitulo"
                android:layout_gravity="center_horizontal"
                android:width="320dp"
                android:textSize="25dp"
                android:textColor="#ffffff"
                android:textStyle="bold"
                android:background="#8039f1"
                android:gravity="center" />

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@+id/tvTitulo"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true">

                <LinearLayout
                    android:orientation="vertical"
                    android:layout_width="156dp"
                    android:layout_height="match_parent"
                    android:gravity="center">

                    <Button
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Registros"
                        android:id="@+id/btnRegistros"
                        android:background="#066ce0"
                        android:gravity="center"
                        android:textColor="#ffffff"
                        android:textStyle="bold"
                        android:width="@dimen/size_buton_wheigth"
                        android:height="@dimen/size_buton"
                        android:layout_marginTop="15dp" />

                    <Button
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Puntos X,Y"
        android:id="@+id/btnPuntos"
        android:background="#066ce0"
        android:textStyle="bold"
        android:textColor="#ffffff"
        android:layout_below="@+id/btnRegistros"
        android:width="@dimen/size_buton_wheigth"
        android:height="@dimen/size_buton"
        android:layout_marginTop="5dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Puentes"
    android:id="@+id/btnPuentes"
    android:background="#066ce0"
    android:gravity="center"
    android:textColor="#ffffff"
    android:textStyle="bold"
    android:width="@dimen/size_buton_wheigth"
    android:height="@dimen/size_buton"
    android:layout_marginTop="20dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Marca de KM"
    android:id="@+id/btnKM"
    android:textStyle="bold"
    android:textColor="#ffffff"
    android:gravity="center"
    android:background="#066ce0"
    android:width="@dimen/size_buton_wheigth"
    android:height="@dimen/size_buton"
    android:layout_marginTop="5dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Banco de Material"
    android:id="@+id/btnBMaterial"
    android:background="#066ce0"
    android:gravity="center"
    android:textColor="#ffffff"
    android:textStyle="bold"
    android:width="@dimen/size_buton_wheigth"
    android:height="@dimen/size_buton"
    android:layout_marginTop="5dp" />
</LinearLayout>

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:src="@drawable/nuevo" />
</LinearLayout>
```

```

        </LinearLayout>
    </LinearLayout>
</ScrollView>
</RelativeLayout>

```

Nombre en proyecto: exportar.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollView6"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:gravity="center"
            android:baselineAligned="false">
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Exportar datos de Carretera a CSV "
                android:id="@+id/btnExpCarretera"
                android:layout_gravity="center_horizontal"
                android:layout_alignParentTop="true"
                android:layout_centerHorizontal="true"
                android:width="@dimen/size_buton_wheigth"
                android:height="@dimen/size_buton"
                android:textColor="#ffffff"
                android:textStyle="bold"
                android:gravity="center"
                android:background="#066ce0"
                android:layout_marginTop="60dp" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Exporta datos de Infraestructura CSV"
                android:id="@+id/btnExpInfraestructura"
                android:layout_centerVertical="true"
                android:layout_centerHorizontal="true"
                android:background="#066ce0"
                android:textColor="#ffffff"
                android:textStyle="bold"
                android:width="@dimen/size_buton_wheigth"

```

```
        android:height="@dimen/size_buton"
        android:layout_marginTop="50dp"
        android:gravity="center" />
    </LinearLayout>
</ScrollView>

</RelativeLayout>
```

Nombre en proyecto: enviar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollView5"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Bienvenido"
                android:id="@+id/tvBienvenido"
                android:layout_gravity="center_horizontal"
                android:gravity="center"
                android:textColor="#000000"
                android:textSize="@dimen/abc_action_bar_stacked_max_height"
                android:layout_marginTop="25dp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="&quot;&quot;"
                android:id="@+id/tvNombreU"
                android:textColor="#000000"
                android:layout_gravity="center_horizontal" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Enviar archivos de Carreteras e Infraestructura al
SIG"
                android:id="@+id/tvDetalles"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="44dp"
                android:gravity="center"
                android:textColor="#000000"
```

```

        android:textStyle="bold"
        android:textSize="20dp" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Enviar"
            android:id="@+id/button3"
            android:layout_marginTop="47dp"
            android:textSize="20dp"
            android:textColor="#ffffff"
            android:textStyle="bold"
            android:background="#066ce0"
            android:height="@dimen/size_buton"
            android:width="@dimen/size_buton_wheigth"
            android:layout_gravity="center_horizontal" />
    </LinearLayout>
</ScrollView>
</RelativeLayout>

```

Nombre en proyecto: login.xml

Es importante modificar el Archivo **AndroidManifest.xml**

```

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

    <uses-sdk android:minSdkVersion="9"/>
    <uses-library android:name="com.google.android.maps" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Bisigetmovil">
            <intent-filter>
                <action android:name="com.ciclovia.imt.bi_siget.BISIGET" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>

        <activity android:name=".Consultadatos">
            <intent-filter>
                <action android:name="com.ciclovia.imt.bi_siget.CONULTADATOS" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

```
</activity>

<activity android:name=".Enviar">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.ENVIAR" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".Exportar">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.EXPORTAR" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".Infraestructura">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.INFRAESTRUCTURA" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".Login">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.LOGIN" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".Registrociclovia">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.REGISTROCICLOVIA" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".ConsultaBmat">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.CONSULTABMAT" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".Consultamkm">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.CONSULTAMKM" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".ConsultaPtes">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.CONSULTAPTES" />
```



```
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".ConsultaPtos">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.CONSULTAPTOS" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity android:name=".Consultareg">
    <intent-filter>
        <action android:name="com.ciclovia.imt.bi_siget.CONSULTAREG" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

</application>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>

</manifest>
```

Anexo 2. Codificación de Clases

Nombre de la clase: MainActivity

```
package com.ciclovia.imt.bisiget_movil;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity implements OnClickListener {
    Button comenzar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        comenzar = (Button) findViewById(R.id.btnComenzar);
        comenzar.setOnClickListener(this);
    }

    public void onClick(View v) {
        switch (v.getId()){
            case R.id.btnComenzar:
                Intent otraP=new Intent("com.ciclovia.imt.bi_siget.BISIGET");
                startActivity(otraP);
                break;
        }
    }
}
```

Método del Botón

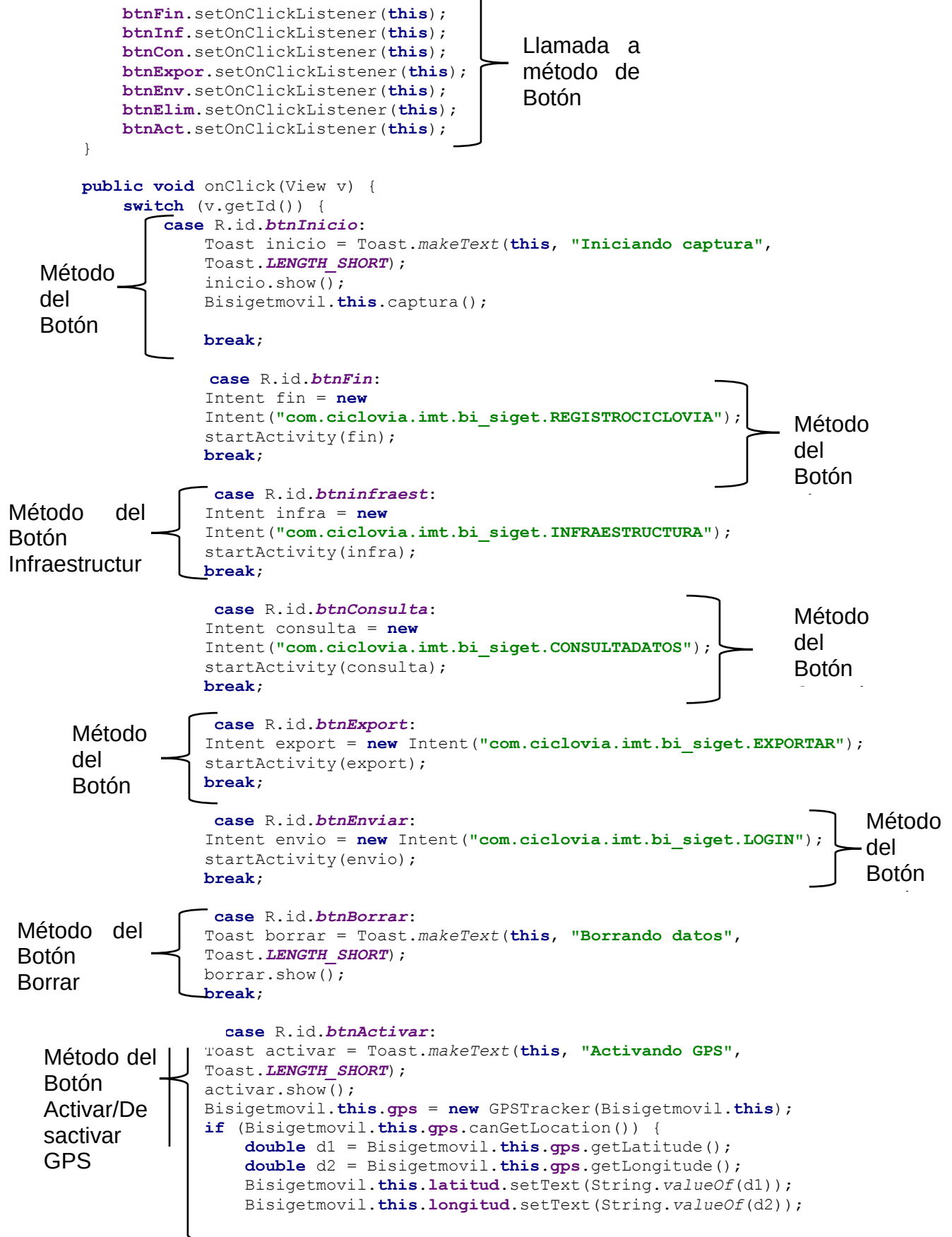
Nombre de la clase: Bisiget

```
public class Bisigetmovil extends Activity implements OnClickListener {

    Button btnIni, btnFin, btnInf, btnCon, btnExpor, btnEnv, btnElim, btnAct;
    GPSTracker gps;
    TextView longitud, latitud;
    final Handler mHandler = new Handler();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.bisigetmovil);
        btnIni = (Button) findViewById(R.id.btnInicio);
        btnFin = (Button) findViewById(R.id.btnFin);
        btnInf = (Button) findViewById(R.id.btninfraest);
        btnCon = (Button) findViewById(R.id.btnConsulta);
        btnExpor = (Button) findViewById(R.id.btnExport);
        btnEnv = (Button) findViewById(R.id.btnEnviar);
        btnElim = (Button) findViewById(R.id.btnBorrar);
        btnAct = (Button) findViewById(R.id.btnActivar);
        latitud = (TextView) findViewById(R.id.txtlatitud);
        longitud = (TextView) findViewById(R.id.txtlongitud);

        btnIni.setOnClickListener(this);
    }
}
```



```
        return;
    }
    Bisigetmovil.this.gps.showSettingsAlert();

    break;
}

}

public static Thread hilo;
public static boolean status;

protected void captura() {
    if (hilo == null) {
        setStatus(true);
        hilo = new Thread(new Hilo());
        hilo.start();
    }
}

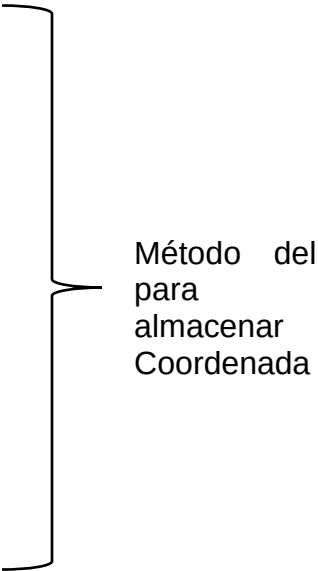
public List<coordenadas> datosc = new ArrayList<coordenadas>();
ProgressDialog dialog = null;
public void guarda()
{
    double d1 = 0.0D;
    double d2 = 0.0D;
    this.gps = new GPSTracker(this);
    if (this.gps.canGetLocation())
    {
        d1 = this.gps.getLatitude();
        d2 = this.gps.getLongitude();
        this.latitud.setText(String.valueOf(d1));
        this.longitud.setText(String.valueOf(d2));
    }
    for (;;)
    {
        this.datosc.add(new coordenadas(d1, d2));
        return;
        // this.gps.showSettingsAlert();
    }
}

public void setStatus(boolean paramBoolean) {
    status = paramBoolean;
}

class Hilo
    implements Runnable
{
    final Runnable iniciaCaptura = new Runnable()
    {
        public void run()
        {
            Toast.makeText(Bisigetmovil.this, "Captura de puntos GPS de la
ciclovia", 0).show();
            Bisigetmovil.this.guarda();
        }
    };

    Hilo() {}

    public void run()
    {
        for (;;)
        {
```



Método del
para
almacenar
Coordenada

```

        if (!Bisigetmovil.status) {
            return;
        }
        try
        {
            Thread.sleep(3000L);
            Bisigetmovil.this.mHandler.post(this.iniciaCaptura);
            if (Bisigetmovil.status) {
                continue;
            }
            return;
        }
        catch (Exception localException)
        {
            throw new Error("Error en el hilo");
        }
    }
}
}
}

```

Nombre de la clase: RegistroCiclovia

```

public class Registrociclovia extends Activity implements OnClickListener {
    private Spinner spAdmin, spCarriles, spTipo;
    private TextView codigoCv, nombreC;
    private Button agregar;
    private List<String> listaAdm, listaCarr, listaTipo;
    private String codigoC, nombre, administracion, nCarriles, tipoPav;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.registrociclovia);
        codigoCv=(TextView) findViewById(R.id.txtCodigo);
        nombreC=(TextView) findViewById(R.id.txtNombre);
        agregar=(Button) findViewById(R.id.btnRegistro);
        this.SpinerAdministracion();
        this.SpinerCarriles();
        this.SpinerTipo();
        agregar.setOnClickListener(this);
    }

    public void SpinerAdministracion()
    {
        spAdmin = (Spinner) findViewById(R.id.spAdministracion);
        listaAdm=new ArrayList<String>();
        listaAdm.add("Seleccione el tipo de Administracion");
        listaAdm.add("Federal");
        listaAdm.add("Estatal");
        listaAdm.add("Municipal");
        listaAdm.add("Comunitaria");
        ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, listaAdm);

        adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spAdmin.setAdapter(adaptador);
        spAdmin.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int

```

```
position, long id) {
    administracion=parent.getItemAtPosition(position).toString();
}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}

});

}

public void SpinnerCarriles()
{
    spCarriles = (Spinner) findViewById(R.id.spCarriles);
    listaCarr=new ArrayList<String>();
    listaCarr.add("Seleccione el Numero de Carriles");
    listaCarr.add("1");
    listaCarr.add("2");
    ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, listaCarr);

    adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spCarriles.setAdapter(adaptador);
    spCarriles.setOnItemClickListener(new OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
            nCarriles=parent.getItemAtPosition(position).toString();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
}

public void SpinnerTipo()
{
    spTipo = (Spinner) findViewById(R.id.sptipoPavimento);
    listaTipo=new ArrayList<String>();
    listaTipo.add("Seleccione el tipo de Pavimento");
    listaTipo.add("Asfalto");
    listaTipo.add("Concreto");
    listaTipo.add("Adoquin");
    listaTipo.add("Tartan");
    listaTipo.add("Pelo de Tierra");
    ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, listaTipo);

    adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spTipo.setAdapter(adaptador);
    spTipo.setOnItemClickListener(new OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
            tipoPav=parent.getItemAtPosition(position).toString();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
}
```

```

    });
}
**Método de Botón**
public void onClick(View v)
{
    DatabaseHelper mydbHelper =new DatabaseHelper(getApplicationContext());
    nombre=nombreC.getText().toString();
    codigoC=codigoCv.getText().toString();

    try{
        mydbHelper.createDataBase();
    }catch (Exception e){
        throw new Error("No se creo la BD");
    }
    try{
        mydbHelper.openDataBase();
        SQLiteDatabase db = mydbHelper.getWritableDatabase();
        ContentValues valores=new ContentValues();
        valores.put("codigoC",codigoC);
        valores.put("nombre",nombre);
        valores.put("administracion",administracion);
        valores.put("nCarriles",nCarriles);
        valores.put("tipoPav",tipoPav);

        db.insert("carretera", null, valores);
        db.close();

        mydbHelper.closeDataBase();
        mydbHelper.close();

    }catch (SQLException sqle){
        throw sqle;
    }

    Intent inatras=new Intent(Registrociclovía.this,MainActivity.class);
    startActivity(inatras);
}
}

```

Nombre de la clase: Infraestructura

```

public class Infraestructura extends Activity implements OnClickListener {

    private Spinner spTip, spTam, spEstr,spBMat;
    private List<String> listaTip, listaTam, listaEstr,listaBMat;
    private Button regist;
    private EditText nomb, km;
    private TextView latitud,longitud;
    private String tipoI,nomI,longI,estrI,mkm,bMat,lonX,latiY;
    public GPSTracker gps;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.infraestructura);

        regist=(Button) findViewById(R.id.btnRegistroInf);
        nomb=(EditText) findViewById(R.id.txtNombreI);
        km=(EditText) findViewById(R.id.txtKilometraje);
        this.SpinerTipoInfraestructura();
    }
}

```

```
this.SpinnerTamanoInf();
this.SpinnerEstructura();
this.SpinnerBancoMaterial();

regist.setOnClickListener(this);
longitud=(TextView) findViewById(R.id.tvLongitudI);
latitud=(TextView) findViewById(R.id.tvlatitudI);

gps=new GPSTracker(Infraestructura.this);
if(gps.canGetLocation()){
    double latitude=(double) (gps.getLatitude());
    double longitude=(double) (gps.getLongitude());

    latitud.setText(String.valueOf(latitude));
    longitud.setText(String.valueOf(longitude));

}else{
    gps.showSettingsAlert();
}

}

public void SpinnerTipoInfraestructura()
{
    spTip = (Spinner) findViewById(R.id.spTipoI);
    listaTip=new ArrayList<String>();
    listaTip.add("Seleccione el tipo de Infraestructura");
    listaTip.add("Puente");
    listaTip.add("Ciclopuerto");
    listaTip.add("Biciestacionamiento");
    listaTip.add("Centro de Transferencia e Interconexion Modal");
    listaTip.add("Alcantarilla");

    ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, listaTip);

    adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spTip.setAdapter(adaptador);

    spTip.setOnItemClickListener(new OnItemSelectedListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            tipoI=parent.getItemAtPosition(position).toString();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
}

public void SpinnerTamanoInf()
{
    spTam = (Spinner) findViewById(R.id.spTamano);
    listaTam=new ArrayList<String>();
    listaTam.add("Seleccione el tamaño del Puente");
    listaTam.add("Chico (6-30 mts)");
    listaTam.add("Mediano (30-100 mts)");
    listaTam.add("Grande(100-)");
    ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
```



```

android.R.layout.simple_spinner_item, listaTam);

adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spTam.setAdapter(adaptador);

spTam.setOnItemSelectedListener(new OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
        longI=parent.getItemAtPosition(position).toString();
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

});
}
public void SpinnerEstructura()
{
    spEstr = (Spinner) findViewById(R.id.spEstructura);
    listaEstr=new ArrayList<String>();
    listaEstr.add("Seleccione el tipo de Estructura");
    listaEstr.add("Metal");
    listaEstr.add("Concreto");
    listaEstr.add("Madera");
    listaEstr.add("Piedra");
    ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, listaEstr);

    adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spEstr.setAdapter(adaptador);
    spEstr.setOnItemSelectedListener(new OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
            estrI=parent.getItemAtPosition(position).toString();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }

    });
}
public void SpinnerBancoMaterial()
{
    spBMat = (Spinner) findViewById(R.id.spMaterial);
    listaBMat=new ArrayList<String>();
    listaBMat.add("Seleccione el tipo de Señaletica");
    listaBMat.add("Horizontal");
    listaBMat.add("Vertical");
    ArrayAdapter<String> adaptador =new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, listaBMat);

    adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spBMat.setAdapter(adaptador);
    spBMat.setOnItemSelectedListener(new OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
            bMat=parent.getItemAtPosition(position).toString();
        }
    }
}

```

```
        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
}

public void onClick(View v) {
    DatabaseHelper mydbHelper =new DatabaseHelper(getApplicationContext());
    nomI=nomb.getText().toString();
    mkm=km.getText().toString();
    lonX=longitud.getText().toString();
    latiY=latitud.getText().toString();

    try{
        mydbHelper.createDataBase();
    }catch (Exception e){
        throw new Error("No se creo la BD");
    }
    try{
        mydbHelper.openDataBase();
        SQLiteDatabase db = mydbHelper.getWritableDatabase();
        ContentValues valores=new ContentValues();
        valores.put("tipoI", tipoI);
        valores.put("nomI", nomI);
        valores.put("longI", longI);
        valores.put("estrI", estrI);
        valores.put("mkm", mkm);
        valores.put("bMat", bMat);
        valores.put("longX", lonX);
        valores.put("latY", latiY);

        db.insert("infraestructura", null, valores);
        db.close();

        mydbHelper.closeDataBase();
        mydbHelper.close();

    }catch (SQLException sqle){
        throw sqle;
    }

    Intent inatras=new Intent(Infraestructura.this,MainActivity.class);
    startActivity(inatras);
}

}
```

Nombre de la clase: ConsultaReg

```
public class Consultareg extends Activity {

    ListView lvDetail;
    Context context = Consultareg.this;
    DatabaseHelper dbh=new DatabaseHelper(this);

    ArrayList myList = new ArrayList();
    String[] nombre = new String[dbh.consultaregistro().size()];
    String[] codigo = new String[dbh.consultaregistro().size()];
    String[] admin = new String[dbh.consultaregistro().size()];
    int[] carril = new int[dbh.consultaregistro().size()];
}
```

```
String[] pav = new String[dbh.consultaregistro().size()];

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.consultareg);

    lvDetail = (ListView) findViewById(R.id.listreg);
    // insert data into the list before setting the adapter
    // otherwise it will generate NullPointerException - Obviously
    getDataInList();
    lvDetail.setAdapter(new AdpterReg(context, myList));
}

private void getDataInList() {
    int i = 0;
    for (;;)
    {
        if (i >= dbh.consultaregistro().size())
        {
            carretera ld = new carretera();
            ld.setNombre(nombre[i]);
            ld.setCodigoC(codigo[i]);
            ld.setAdministracion(admin[i]);
            ld.setnCarriles(carril[i]);
            ld.setTipoPav(pav[i]);
            // Add this object into the ArrayList myList
            myList.add(ld);
        }
    }
}
```

Anexo 3. Clases de Controlador

Nombre de la clase: DataBaseHelper

```
public class DatabaseHelper extends SQLiteOpenHelper{
//-----cadena--de--conexion-----
-----

    private static String DB_PATH = "/data/data/mx.imt.bisigetmovil/databases/";
    private static String DB_NAME = "dbBisiget.sqlite";

    private SQLiteDatabase myDataBase;
    private final Context myContext;

    public DatabaseHelper(Context context) {

        super(context, DB_NAME, null,1);
        this.myContext = context;
    }

    public void createDataBase() throws IOException {

        boolean dbExist = checkDataBase();
        SQLiteDatabase db_Read= null;

        if (dbExist) {
            // Si existe, no hacemos nada!
        } else {
            db_Read=this.getReadableDatabase();
            db_Read.close();
            try {

                copyDataBase();

            } catch (IOException e) {

                throw new Error("Error copiando Base de Datos");
            }
        }
    }

    private void copyDataBase() throws IOException {

        String outFileName=DB_PATH+DB_NAME;
        OutputStream databaseOutput = new FileOutputStream(outFileName);
        InputStream databaseInput =myContext.getAssets().open(DB_NAME);

        byte[] buffer = new byte[1024];
        int length;

        while ((length = databaseInput.read(buffer))!= -1) {
            if (length>0){
                databaseOutput.write(buffer,0,length);
            }
        }

        databaseInput.close();
    }
}
```

```

        databaseOutput.flush();
        databaseOutput.close();
    }

    private boolean checkDataBase() {

        SQLiteDatabase checkDB = null;

        try {
            String myPath = DB_PATH + DB_NAME;
            checkDB = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READWRITE);

        } catch (SQLException e) {
            // Base de datos no creada todavia
            File dbFile=new File(DB_PATH+DB_NAME);
            return dbFile.exists();
        }

        if (checkDB != null) {

            checkDB.close();
        }

        return checkDB != null ? true : false;
    }

    public void openDataBase() {

        // Open the database
        String myPath = DB_PATH + DB_NAME;
        myDataBase = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READWRITE);
    }

    public void closeDataBase() {

        myDataBase.close();
    }

    public void onCreate(SQLiteDatabase db) {

    }

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // En caso de una nueva versión habría que actualizar las tablas
    }

    //-----consultas-----
    public List<coordenadas> consultaCoordenadas() {
        openDataBase();
        List <coordenadas> total_coordenadas=new ArrayList<coordenadas>();
        String[] valores_coord={"lat","longit"};
        Cursor
qcursor=myDataBase.query("coordenadas",valores_coord,null,null,null,null,null);
        if (qcursor.getCount()>0)
        {
            qcursor.moveToFirst();
            do{
                coordenadas coord=new
coordenadas(qcursor.getFloat(0),qcursor.getFloat(1));
                total_coordenadas.add(coord);
            }while (qcursor.moveToNext());
            myDataBase.close();
        }
    }

```

```
        qcursor.close();
    }
    return total_coordenadas;
}

public ArrayList<infraestructura> consultainfraestructura() {
    openDataBase();
    ArrayList <infraestructura> total_Infra=new ArrayList<infraestructura>();
    String[] valores_Infra={"tipoI", "nomI", "estrI"};
    String [] select={"ninguno"};

    Cursor
    qcursor=myDataBase.query("infraestructura",valores_Infra,"tipoI!=?",select,null,null,
null,null);
    if (qcursor.getCount()>0)
    {
        qcursor.moveToFirst();
        do{
            infraestructura infra=new infraestructura
(qcursor.getString(0),qcursor.getString(1),qcursor.getString(2));
            total_Infra.add(infra);
        }while (qcursor.moveToNext());
        myDataBase.close();
        qcursor.close();
    }
    return total_Infra;
}

public ArrayList<marcasKm> consultamkm() {
    openDataBase();
    ArrayList <marcasKm> total_mkm=new ArrayList<marcasKm>();
    String[] valores_mkm={"mkm"};
    String [] select={"MK"};

    Cursor
    qcursor=myDataBase.query("infraestructura",valores_mkm,"mkm!=?",select,null,null,
null);
    if (qcursor.getCount()>0)
    {
        qcursor.moveToFirst();
        do{
            marcasKm m=new marcasKm (qcursor.getString(0));
            total_mkm.add(m);
        }while (qcursor.moveToNext());
        myDataBase.close();
        qcursor.close();
    }
    return total_mkm;
}

public ArrayList<bancoMaterial> consultabm() {
    openDataBase();
    ArrayList <bancoMaterial> total_bm=new ArrayList<bancoMaterial>();
    String[] valores_bm={"bMat"};
    String [] select={"ninguno"};

    Cursor
    qcursor=myDataBase.query("infraestructura",valores_bm,"bMat!=?",select,null,null,
null);
    if (qcursor.getCount()>0)
    {
        qcursor.moveToFirst();
        do{
```

```

        bancoMaterial bm=new bancoMaterial (qcursor.getString(0));
        total_bm.add(bm);
    }while (qcursor.moveToNext());
    myDataBase.close();
    qcursor.close();
}
return total_bm;
}

public ArrayList<carretera> consultaregistro() {
    openDataBase();
    ArrayList <carretera> total_reg=new ArrayList<carretera>();
    String[]
valores_carr={"codigoC", "nombre", "administracion", "nCarriles", "tipoPav"};

    Cursor
qcursor=myDataBase.query("carretera",valores_carr,null,null,null,null,null);

    qcursor.moveToFirst();
    do{
        carretera regis=new carretera
(qcursor.getString(0),qcursor.getString(1),qcursor.getString(2),qcursor.getInt(3)
,qcursor.getString(4));
        total_reg.add(regis);
    }while (qcursor.moveToNext());
    myDataBase.close();
    qcursor.close();

    return total_reg;
}
//-----exportar-----
public ArrayList<coordenadasexp> exportCoordenadas() {
    openDataBase();
    ArrayList <coordenadasexp> total_coordenadasexp=new
ArrayList<coordenadasexp>();
    String[] valores_coordenadasexp={"longitudX", "latitudY"};

    Cursor qcursor=myDataBase.query("coordenadas",valores_coordenadasexp
,null,null,null,null,null);

    qcursor.moveToFirst();
    do{
        coordenadasexp coord=new coordenadasexp
(qcursor.getFloat(0),qcursor.getFloat(1));
        total_coordenadasexp.add(coord);
    }while (qcursor.moveToNext());
    myDataBase.close();
    qcursor.close();

    return total_coordenadasexp;
}
public ArrayList<registroexp> exportRegistros() {
    openDataBase();
    ArrayList <registroexp> total_registroexp=new
ArrayList<registroexp>();
    String[]
valores_registroexp={"codigoC", "nombre", "administracion", "nCarriles", "tipoPav"};

    Cursor qcursor=myDataBase.query("carretera",valores_registroexp
,null,null,null,null,null);

    qcursor.moveToFirst();

```

```
        do{
            registrosexp coord=new registrosexp
            (qcursor.getString(0),qcursor.getString(1),qcursor.getString(2),qcursor.getInt(3)
            ,qcursor.getString(4));
            total_registrosexp.add(coord);
        }while (qcursor.moveToNext());
        myDataBase.close();
        qcursor.close();

        return total_registrosexp;
    }

    public ArrayList<infraestructuraexp> exportInfraestructura () {
        openDataBase();
        ArrayList <infraestructuraexp> total_infraestructuraexp=new
        ArrayList<infraestructuraexp>();
        String[]
        valores_infraestructuraexp={"tipoI","nomI","longI","estrI","mkm","bMat","lonX","latY"};

        Cursor qcursor=myDataBase.query("carretera",valores_infraestructuraexp
        ,null,null,null,null,null);

        qcursor.moveToFirst();
        do{
            infraestructuraexp coord=new infraestructuraexp
            (qcursor.getString(0),qcursor.getString(1),qcursor.getString(2),qcursor.getString
            (3),qcursor.getString(4),qcursor.getString(5),qcursor.getFloat(6),qcursor.getFloa
            t(7));
            total_infraestructuraexp.add(coord);
        }while (qcursor.moveToNext());
        myDataBase.close();
        qcursor.close();

        return total_infraestructuraexp;
    }
}
```

Nombre de la clase: GPSTracker

Nota: esta clase se puede encontrar en la web. Es la clase que permite Activar el GPS y también captura las coordenadas.

```
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;

public class GPSTracker extends Service implements LocationListener {

    private final Context mContext;
```

```

// flag for GPS status
boolean isGPSEnabled = false;

// flag for network status
boolean isNetworkEnabled = false;

// flag for GPS status
boolean canGetLocation = false;

Location location; // location
double latitude; // latitude
double longitude; // longitude

// The minimum distance to change Updates in meters
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10L; // 10 meters

// The minimum time between updates in milliseconds
private static final long MIN_TIME_BW_UPDATES = 3000L; // 1 minute

// Declaring a Location Manager
protected LocationManager locationManager;

public GPSTracker(Context context) {
    this.mContext = context;
    getLocation();
}

public Location getLocation() {
    try {
        locationManager = (LocationManager) mContext
            .getSystemService(LOCATION_SERVICE);

        // getting GPS status
        isGPSEnabled = locationManager
            .isProviderEnabled(LocationManager.GPS_PROVIDER);

        // getting network status
        isNetworkEnabled = locationManager
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnabled && !isNetworkEnabled) {
            // no network provider is enabled
        } else {
            this.canGetLocation = true;
            if (isNetworkEnabled) {
                locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_TIME_BW_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                Log.d("Network", "Network");
                if (locationManager != null) {
                    location = locationManager
                        .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
            // if GPS Enabled get lat/long using GPS Services
            if (isGPSEnabled) {
                if (location == null) {

```

```
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
    Log.d("GPS Enabled", "GPS Enabled");
    if (locationManager != null) {
        location = locationManager

.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (location != null) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    }
}

}
}
}
}
} catch (Exception e) {
    e.printStackTrace();
}

return location;
}

/**
 * Stop using GPS listener
 * Calling this function will stop using GPS in your app
 * */
public void stopUsingGPS() {
    if(locationManager != null){
        locationManager.removeUpdates(GPSTracker.this);
    }
}

/**
 * Function to get latitude
 * */
public double getLatitude() {
    if(location != null){
        latitude = location.getLatitude();
    }

    // return latitude
    return latitude;
}

/**
 * Function to get longitude
 * */
public double getLongitude() {
    if(location != null){
        longitude = location.getLongitude();
    }

    // return longitude
    return longitude;
}

/**
 * Function to check GPS/wifi enabled
 * @return boolean
 * */
public boolean canGetLocation() {
```

```

        return this.canGetLocation;
    }

    /**
     * Function to show settings alert dialog
     * On pressing Settings button will launch Settings Options
     */
    public void showSettingsAlert(){
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

        // Setting Dialog Title
        alertDialog.setTitle("GPS is settings");

        // Setting Dialog Message
        alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");

        // On pressing Settings button
        alertDialog.setPositiveButton("Settings", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,int which) {
                Intent intent = new
                Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                mContext.startActivity(intent);
            }
        });

        // on pressing cancel button
        alertDialog.setNegativeButton("Cancel", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });

        // Showing Alert Message
        alertDialog.show();
    }

    @Override
    public void onLocationChanged(Location location) {
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
    }

    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }
} (AndroidHive, 2012)

```

Nombre de la clase: AdpaterReg

```
public class AdpterReg extends BaseAdapter {
    ArrayList<carretera> myList = new ArrayList<carretera>();
    LayoutInflater inflater;
    Context context;

    public AdpterReg(Context context, ArrayList myList) {
        this.myList = myList;
        this.context = context;
        inflater = LayoutInflater.from(this.context);
    }

    @Override
    public int getCount() {
        return myList.size();
    }

    @Override
    public carretera getItem(int position) {
        return myList.get(position);
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        MyViewHolder mViewHolder;

        if (convertView == null) {
            convertView = inflater.inflate(R.layout.listareg, parent, false);
            mViewHolder = new MyViewHolder(convertView);
            convertView.setTag(mViewHolder);
        } else {
            mViewHolder = (MyViewHolder) convertView.getTag();
        }

        carretera currentListData = getItem(position);

        mViewHolder.tvadmin.setText(currentListData.getAdministracion());
        mViewHolder.tvcarril.setText(currentListData.getnCarriles());
        mViewHolder.tvcodigo.setText(currentListData.getCodigoC());
        mViewHolder.tvnombrec.setText(currentListData.getNombre());
        mViewHolder.tvpav.setText(currentListData.getTipoPav());

        return convertView;
    }

    private class MyViewHolder {
        TextView tvadmin;
        TextView tvcarril;
        TextView tvcodigo;
        TextView tvnombrec;
        TextView tvpav;

        public MyViewHolder(View item) {
            tvadmin = (TextView) item.findViewById(R.id.tvAdmin);
            tvcarril = (TextView) item.findViewById(R.id.tvCarril);
            tvcodigo = (TextView) item.findViewById(R.id.tvCodigo);
            tvnombrec = (TextView) item.findViewById(R.id.tvNombreC);
            tvpav = (TextView) item.findViewById(R.id.tvPav);
        }
    }
}
```

```
}  
  }  
}
```

Anexo 4. Clases del Modelo.

Nombre de la clase: carretera

```
public class carretera {
    private String codigoC;
    private String nombre;
    private String administracion;
    private int nCarriles;
    private String tipoPav;

    public carretera() { }

    public carretera(String codigo,String nom,String admin,int nCar,String tipop)
    {
        this.codigoC=codigo;
        this.nombre=nom;
        this.administracion=admin;
        this.nCarriles=nCar;
        this.tipoPav=tipop;
    }
    public String getCodigoC() {
        return codigoC;
    }

    public void setCodigoC(String codigoC) {
        this.codigoC = codigoC;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getAdministracion() {
        return administracion;
    }

    public void setAdministracion(String administracion) {
        this.administracion = administracion;
    }

    public int getnCarriles() {
        return nCarriles;
    }

    public void setnCarriles(int nCarriles) {
        this.nCarriles = nCarriles;
    }

    public String getTipoPav() {
        return tipoPav;
    }

    public void setTipoPav(String tipoPav) {
```

```

        this.tipoPav = tipoPav;
    }
}

```

Nombre de la Clase: infraestructura

```

public class infraestructura {
    private String tipoI;
    private String nomI;
    private String longI;
    private String estrI;
    private String mkm;
    private String bMat;
    private float longX;
    private float latY;

    public infraestructura(String tipo,String nom,String estruct)
    {
        this.tipoI=tipo;
        this.nomI=nom;
        this.estrI=estruct;
    }

    public infraestructura(String tipo,String nom,String longitud,String
    estruct,String marca, String bancom,float lx,float ly)
    {
        this.tipoI=tipo;
        this.nomI=nom;
        this.longI=longitud;
        this.estrI=estruct;
        this.mkm=marca;
        this.bMat=bancom;
        this.longX=lx;
        this.latY=ly;
    }

    public String getNomI() {
        return nomI;
    }

    public void setNomI(String nomI) {
        this.nomI = nomI;
    }

    public String getLongI() {
        return longI;
    }

    public void setLongI(String longI) {
        this.longI = longI;
    }

    public String getEstrI() {
        return estrI;
    }

    public void setEstrI(String estrI) {
        this.estrI = estrI;
    }

    public String getMkm() {

```

```
        return mkm;
    }

    public void setMkm(String mkm) {
        this.mkm = mkm;
    }

    public String getbMat() {
        return bMat;
    }

    public void setbMat(String bMat) {
        this.bMat = bMat;
    }

    public float getLongX() {
        return longX;
    }

    public void setLongX(float longX) {
        this.longX = longX;
    }

    public float getLatY() {
        return latY;
    }

    public void setLatY(float latY) {
        this.latY = latY;
    }

    public String getTipoI() {
        return tipoI;
    }

    public void setTipoI(String tipoI) {
        this.tipoI = tipoI;
    }
}
```

Nombre de la Clase: coordenadas

```
public class coordenadas {
    private double longX;
    private double latY;

    public coordenadas(double lx, double ly)
    {
        this.setLongX(lx);
        this.setLatY(ly);
    }

    public double getLongX() {
        return longX;
    }

    public void setLongX(double longX) {
        this.longX = longX;
    }
}
```



```
public double getLatY() {  
    return latY;  
}  
  
public void setLatY(double latY) {  
    this.latY = latY;  
}  
}
```

Referencias

- AndroidHive. (july de 2012). *Android GPS, Location Manager Tutorial*. Recuperado el Marzo de 2016, de Android GPS, Location Manager Tutorial: <http://www.androidhive.info/2012/07/android-gps-location-manager-tutorial/>
- Bravo, I. M. (2007). *Informatica*. Recuperado el Febrero de 2016, de Que es MySql: <http://indira-informatica.blogspot.mx/2007/09/qu-es-mysql.html>
- definicion, d. (2015). *definicion de open source*. Recuperado el 05 de Febrero de 2016, de definicion de open source: <http://definicion.de/open-source/>
- EcuRed. (2009). *Microsoft SQL Server*. Recuperado el febrero de 2016, de Microsoft SQL Server.: http://www.ecured.cu/Microsoft_SQL_Server
- Google, M. (2016). *MyTraks*. Recuperado el Febrero de 2016, de MyTraks: <https://support.google.com/gmm/answer/6333516>
- <http://www.proyectosagiles.org/>. (7 de Abril de 2015). <http://www.proyectosagiles.org/>.
Obtenido de <http://www.proyectosagiles.org/>:
<http://www.proyectosagiles.org/sites/default/files/image/diagrama-proceso-scrum.gif>
- IGN. (2015). *Sistemas de Información Geográfica*. Recuperado el 04 de febrero de 2016, de Sistemas de Información Geográfica: <https://www.ign.es/ign/layoutIn/actividadesSistemaInfoGeografica.do>
- IMT. (Enero de 2016). *IMT*. Recuperado el 26 de Enero de 2016, de IMT: <http://www.imt.mx/images/files/gral/USIG.pdf>
- IMT. (Enero de 2016). *IMT*. Recuperado el 02 de Febrero de 2016, de Presentación general: <http://www.imt.mx/informacion-general/quienes-somos/presentacion-general.html>
- IMT. (Enero de 2016). *Instalaciones*. Recuperado el 03 de Febrero de 2016, de Instalaciones: <http://www.imt.mx/informacion-general/quienes-somos/instalaciones.html>
- IMT. (Enero de 2016). *Misión, visión y manual de organización*. Recuperado el 02 de Febrero de 2016, de Misión, visión y manual de organización: <http://www.imt.mx/informacion-general/quienes-somos/mision-y-vision.html>
- IMT. (Enero de 2016). *Sistemas de Información Geoespacial*. Recuperado el 03 de Febrero de 2016, de Sistemas de Información Geoespacial: <http://www.imt.mx/micrositios/sistemas-de-informacion-geoespacial/servicios-tecnologicos/sistema-de-informacion-geostatistica-para-el-transporte-siget/introduccion.html>

- IMT, S. d. (Enero de 2016). *IMT*. Recuperado el 02 de febrero de 2016, de Sistemas de información geográfica: <http://www.imt.mx/micrositios/sistemas-de-informacion-geoespacial/servicios-tecnologicos/sistema-de-informacion-geostatistica-para-el-transporte-siget/tecnologias-de-georreferenciacion/sistemas-de-informacion-geografica.html>
- INEGI. (2014). *Sistema de Información*. Recuperado el 04 de Abril de 2016, de Sistema de Información: <http://www.inegi.org.mx/inegi/SPC/doc/internet/sistemainformaciongeografica.pdf>
- Mapaton. (Febrero de 2016). *El poder de los mapas colaborativos*. Recuperado el Febrero de 2016, de El poder de los mapas colaborativos: <http://blog.mapatoncd.mx/los-mapas-colaborativos/>
- Morales Bautista, Elsa María (2015). Modelo de gestión de información geoespacial de la infraestructura para el transporte en México mediante dispositivos móviles. Tesis UAQ.
- programacion, I. d. (2016). *java*. Recuperado el 05 de febrero de 2016, de java: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>
- proyectosagiles.org. (2016). *proyectos agiles*. Recuperado el 04 de Febrero de 2016, de Qué es SCRUM: <http://proyectosagiles.org/que-es-scrum/>
- Rómmel, F. (septiembre de 2007). *SG Buzz*. Recuperado el febrero de 2016, de SQLite: La Base de Datos Embebida: <http://sg.com.mx/revista/17/sqlite-la-base-datos-embebida#.VwPqDpzhDbQ>



Carretera Querétaro-Galindo km 12+000
CP 76700, Sanfandila
Pedro Escobedo, Querétaro, México
Tel +52 (442) 216 9777 ext. 2610
Fax +52 (442) 216 9671

publicaciones@imt.mx

<http://www.imt.mx/>