



# Universidad Autónoma del Estado de México

## Facultad de Geografía

*“Geolocalización en dispositivos móviles,  
su integración a plataformas Android con  
framework PhoneGap. Caso de uso: App  
para reportar fallas de servicios públicos  
en el Municipio de Toluca.”*

QUE PARA OBTENER EL TÍTULO DE:  
Licenciado en Geoinformática

P R E S E N T A:  
Edgar Alejandro Martínez Sánchez

Asesor de Tesis  
Ing. Esperanza Palma Salgado

Revisores  
Ing. Sandra Lucía Hernández Zetina  
MTRO. Leonardo Alfonso Ramos Corona



Toluca, Estado de México

# CONTENIDO

INTRODUCCIÓN.....	5
PLANTEAMIENTO DEL PROBLEMA.....	7
JUSTIFICACION.....	9
OBJETIVOS .....	11
Objetivo general .....	11
Objetivos específicos.....	11
CAPITULO 1. MARCO TEÓRICO.....	12
1.1. Aplicaciones para dispositivos móviles: Visión general .....	12
1.1.1. Sistemas Operativos móviles, situación actual.....	14
1.1.2. Tipos de aplicaciones .....	16
1.2. Mapas en dispositivos móviles .....	19
1.2.1. GPS y Geolocalización .....	19
1.2.2. W3C Geolocation API Specification .....	21
1.2.3. Google Maps .....	21
1.3. Aplicaciones híbridas con PhoneGap en Android .....	23
1.3.1. PhoneGap .....	23
1.3.2. Características .....	24
1.3.3. Datos en PhoneGap .....	26
1.3.4. Servicios REST .....	27
1.3.5. JSON como protocolo de transferencia de datos .....	27
CAPITULO 2. METODOLOGÍA.....	29
2.1. Caso: “Aplicación para reportar fallas en servicios públicos en el Municipio de Toluca” .....	31
2.2. Obtención y representación de posición geográfica obtenida con aplicaciones híbridas .....	35
2.2.1. Geolocalización en PhoneGap y su integración con Google Maps .....	37
2.3. Conexión a bases de datos externa en aplicaciones en PhoneGap.....	48
2.3.1. Servicios REST: Del espacio geográfico al espacio virtual.....	51
2.3.2. Servicios REST: Transacción de datos georreferenciados con aplicaciones híbridas .....	53

2.3.3. Visualización de resultado en otras interfaces .....	67
3. RESULTADOS .....	77
3.1 Costos aproximados .....	78
3.2. FaSeP aplicación híbrida para Android .....	80
4. CONCLUSIONES .....	87
Bibliografía .....	<b>¡Error! Marcador no definido.</b>

## Ilustraciones

<i>Ilustración 1. Número promedio de aplicaciones publicadas diariamente. StarDust. Agosto 2013.....</i>	13
<i>Ilustración 2. Número de aplicaciones disponibles en Google Play desde el 2009 hasta 2015. Fuente: IDC &amp; Statista.....</i>	14
<i>Ilustración 3. Predicciones del mercado global por Sistema Operativo. Fuente: IDC &amp; Statista .....</i>	15
<i>Ilustración 4. Incremento de desarrolladores en plataformas Android y iOS. Fuente: Developer Economics &amp; Statista.....</i>	16
<i>Ilustración 5. Arquitectura de una aplicación híbrida en PhoneGap. Fuente: Curso iOS y Android con PhoneGap y jQuery. Maximiliano Firtman .....</i>	25
<i>Ilustración 6. Arquitectura de Servicio REST. Fuente: www.video2brain.com .....</i>	27
<i>Ilustración 7. Diagrama de funcionamiento de aplicación para reportar fallas en servicios públicos en el municipio de Toluca. Elaboración propia.....</i>	34
<i>Ilustración 8. Representación de interacción entre GPS de aplicación y API de Google Maps. Elaboración propia .....</i>	36
<i>Ilustración 9. Diagrama de obtención de localización por la aplicación móvil y su representación en el mapa. Elaboración propia.....</i>	40
<i>Ilustración 10. Pasos generales del proceso de representación de ubicación con Google Maps y PhoneGap en la aplicación generada .....</i>	43
<i>Ilustración 11. Funcionamiento de proceso de geolocalización en aplicación para reportar fallas en servicios públicos.....</i>	45
<i>Ilustración 12. Proceso de geocodificación inversa en la aplicación desarrollada. Elaboración propia.....</i>	47
<i>Ilustración 13. Sistema de Bases de Datos Móvil. Fuente: Evolución de las Bases de Datos: de Fijas a Móviles (Delgado Fernández &amp; Gama Moreno, 2009).....</i>	48
<i>Ilustración 14. Tasa de crecimiento de tráfico de datos móviles 2013- 2018. Fuente: IDC &amp; Statista .....</i>	50
<i>Ilustración 15. Dispositivos móviles y espacio geográfico.....</i>	51
<i>Ilustración 16. Usuarios de dispositivos móviles y servicios REST .....</i>	52
<i>Ilustración 17. Visualización de Manhattan en primero de mayo, 2012, representando el momento y lugar de los tweets que contengan las palabras clave, MayDay y Occupy. Fuente: <a href="http://www.socialancer.com/">http://www.socialancer.com/</a> .....</i>	52
<i>Ilustración 18. Conexión de aplicación móvil a servicio REST. ....</i>	54
<i>Ilustración 19. Peticiones y respuestas REST. Fuente: Video2brain.com.....</i>	54
<i>Ilustración 20. Capas de información en un mapa. Fuente: Dreamstime.com .....</i>	55
<i>Ilustración 21. Ejemplo Hola Mundo servicio REST con Slim PHP.....</i>	57
<i>Ilustración 22. Método POST. Fuente: Elaboración propia .....</i>	58
<i>Ilustración 23. Formulario de aplicación FaSeP. Fuente: Elaboración propia.....</i>	59

Ilustración 24. Método POST envió de objeto JSON a base de datos. FaSeP .....	61
Ilustración 25. El usuario manda. Fuente: ESRI España .....	62
Ilustración 26. Método GET. Fuente: Elaboración propia .....	63
Ilustración 27 Resultados JSON del servicio REST para aplicación FaSeP .....	65
Ilustración 28. Decodificación de resultado JSON a marcadores en el mapa .....	66
Ilustración 29. Transformación de objeto JSON a marcadores en aplicación FaSeP.....	67
Ilustración 30. Vista de tablas con datos JSON.....	72
Ilustración 31. Visualizador de datos JSON de fallas en servicios públicos en Google Maps .....	72
Ilustración 32. Visualizador de datos GeoJSON de fallas en servicios públicos en un mapa Leaflet.....	75
Ilustración 33. Agregación de capa vectorial GeoJSON desde servicio REST en QGis ..	75
Ilustración 34. Visualización de capa de datos GeoJSON en QGis.....	76
Ilustración 35. Diagrama de funcionamiento de aplicación para reportar fallas en servicios públicos en el municipio de Toluca. Elaboración propia .....	77
Ilustración 36. Comparación de datos gastados entre primera ejecución y primer reporte en la aplicación FaSeP.....	78
Ilustración 37. Aplicación FaSeP. ....	82
Ilustración 38. Árbol de decisiones para el desarrollo de aplicaciones nativas, híbridas y web. Fuente: OutSystems.....	84
Ilustración 39. Actividades y procesos realizados en la aplicación FaSeP.....	85

## Tablas

Tabla 1. SDK y Leguajes de Programación de principales plataformas móviles nativas. 17	
Tabla 2. Comparación de funciones entre API Google Maps y API Google Maps for Business. Fuente: Developers Google Maps.....	23
Tabla 3. Métodos de la API de Geolocalización W3C para obtener la posición geográfica. Fuente: Mozilla Developer Network.....	39
Tabla 4. Preferencias por acción entre aplicaciones y webs móviles. Fuente: Keynote Systems, Inc.....	49
Tabla 5. Métodos HTTP y su significado. Fuente: Video2brain.com .....	54
Tabla 6. Ejemplo de objetos geometry GeoJSON.....	69
Tabla 7. Calculo de gasto de internet al momento de realizar la investigación en primera ejecución y primer reporte de falla con la aplicación FaSeP según servicio de telefonía. Tabla elaborada con datos de la IFT. ....	79
Tabla 8. Calculo bruto aproximado de costo de veinte reportes considerando el costo del primer reporte .....	80
Tabla 9. Comparación entre tiempo de respuesta y peso de archivo respuesta JSON del servicio REST .....	80
Tabla 10. Comparación entre aplicación móviles. Fuente: Aplicacionesmovil.com.....	81

## INTRODUCCIÓN

La evolución de la tecnología ha dado pie en la actualidad a la mejora y optimización de artefactos que se utilizan en la vida cotidiana, este es el caso de los teléfonos inteligentes, llamados así porque llevaron más allá el enfoque del funcionamiento común de un teléfono móvil, que básicamente era, llamada de voz y SMS. La integraciones de funciones a los equipos móviles (cámara fotográfica, conexión a internet vía WIFI™ o red móvil, video llamadas, navegador GPS, etc.) y el adelanto tecnológico en sistemas operativos y plataformas de desarrollo, ha dado paso a que un gran colectivo de desarrolladores generen aplicaciones para optimizar funciones de los Smartphone.

Analizando los adelantos de la última década en tecnología móvil, se puede observar el cambio de perspectiva en cuanto a la comunicación e intercambio de información, lo cual ha beneficiado en gran parte al público que hace uso de los servicios brindados para dispositivos móviles e incluso también a campos de investigación, por ejemplo en el portal web "TICbeat" se menciona que las tecnologías móviles pueden ser utilizadas como "herramientas para el estudio de la movilidad humana y la demografía espacial, además para la educación en general son herramientas extremadamente poderosas y útiles para la investigación en ciencias sociales" (TICbeat, 2013). Sin embargo el área de estudio se globaliza debido a la implementación de datos espaciales en la información que puede ser procesada en un teléfono inteligente, potenciado aún más por la menor temporalidad en la generación de los datos.

Como tal las características mencionadas acerca de la generación de datos georreferenciados comprenden un campo de oportunidad e innovación en el estudio del espacio geográfico con herramientas como un teléfono móvil inteligente. En este sentido la tendencia del desarrollo en dispositivos móviles en especial las aplicaciones híbridas dan paso a un la generación de herramientas móviles multiplataforma aprovechando servicios web. Por la misma razón se existe la necesidad de formular herramientas que se utilicen para el estudio del espacio, utilizando las tecnologías móviles para generar información.

A raíz de lo mencionado se comprende que la integración entre hardware, servicios en internet, combinado herramientas de publicación de mapas, utilizando lenguajes de programación web en un framework de aplicaciones híbridas (PhoneGap), genera un campo de oportunidad para distribuir aplicaciones basadas en el estudio del espacio geográfico en multiplataforma.

Dentro de las necesidades que en la actualidad pueden percibirse en cuanto a la generación de datos georreferenciados, la integración de funciones geoespaciales como la geolocalización y geocodificación utilizando los sensores que los dispositivos móviles tienen integrados (GPS) llevaron a la generación de una aplicación (FaSeP) para ejemplificar la manera en que pueden utilizarse las problemáticas en el espacio para generar datos con un *Smartphone* o *Tablet* con Android.

Para integrar las herramientas mencionadas para obtener datos de un fenómeno en el espacio, se eligió para llevar a cabo la ejemplificación es la temática de las fallas en servicios públicos en la ciudad de Toluca, Estado de México. Utilizando la ubicación del sensor GPS del dispositivo móvil se creó una aplicación móvil híbrida, con la cual se puede capturar y almacenar en una base de datos registros de fallos en servicios públicos haciendo uso de tecnologías y técnicas con lenguajes de programación como HTML5, CSS3, JavaScript, JSON, y PHP.

## PLANTEAMIENTO DEL PROBLEMA

La tendencia en la actualidad en cuanto a los datos, apunta hacia compartir información en tiempo real, esto ha sido soportado por una parte por el internet y por otro lado gracias a la llegada de los dispositivos móviles que cuentan con hardware necesario para que se puedan optimizar tareas cotidianas de organizaciones principalmente aquellas que tienen que ver con el envío de información en tiempo real y el uso de sensores de los teléfonos inteligentes, como envío de ubicaciones con el GPS, fotografías, etc. todo esto a los entornos de desarrollo actuales en cuanto a sistemas operativos para dispositivos móviles.

*“Ha llegado a su fin la época en la que un departamento de desarrollo debía concentrarse, exclusivamente, en crear aplicaciones para PC. “* (Delgado, 2013). Esto principalmente porque en la actualidad la interacción entre la información que requiere un cliente debe de ser fácil de consultar desde cualquier lugar, generando esto la necesidad de crear un entorno más amigable al cliente que hace uso de las aplicaciones.

En los SIG este enfoque se ajusta perfectamente a la interacción de todos sus elementos, ya que las tecnologías en dispositivos móviles pueden ser un apoyo fundamental en la recolecta, manejo y consulta de datos geográficos en tiempo real. Además de que el trabajo de escritorio se ve beneficiado ya que pueden analizarse datos con menor temporalidad y en algunos casos exactitud. Pero antes de avanzar en el desarrollo de aplicaciones para dispositivos móviles incluyendo a aquellas con enfoque de SIG, se debe tomar en cuenta el equilibrio entre funcionalidad, experiencia de uso, rendimiento, mantenimiento y sobre todo costo.

Lo anterior nos dirige a una rama actual de desarrollo móvil conocida como aplicaciones híbridas, su funcionamiento en general consta de tres pasos.

1. Crear un núcleo de código común utilizando tecnologías web (HTML5, JavaScript, CSS).
2. complementarlo de forma opcional con elementos desarrollados con tecnologías nativas de cada plataforma (Objective C para iOS, Android SDK, etc.);
3. Finalmente, encapsularlo en un contenedor específico para cada plataforma, de manera que la aplicación pueda interactuar con el sistema operativo subyacente, acceder a las capacidades del hardware.

En tal caso las aplicaciones híbridas tienen gran potencial por la posibilidad de hacer uso de las capacidades del sistema operativo o también del hardware del dispositivo (cámara, GPS, acelerómetro, etc.).

Principalmente esto beneficia al desarrollo SIG, en el caso del GPS de un dispositivo móvil, ya que se puede utilizar de diferentes maneras para generar información.

Enfocándose más en el desarrollo de aplicaciones SIG móviles, se hace obligatorio tener una visión de que representa en la actualidad los servicios y herramientas de publicación de mapas en la web y como estos pueden ser aprovechados para generar aplicaciones híbridas mediante un framework, en este caso PhoneGap del cual solo se hará uso para generación de aplicaciones en plataformas Android.

Como primer caso de servicios de mapas que se ofrecen en la web se tiene ejemplo como Google Maps, Bing Map, Yahoo Maps, OpenLayer, Open Street, etc.; y que gracias a esto se han podido generar aplicaciones útiles para la vida cotidiana que pueden ser descargadas en los marketplaces como GPS, visores de información geográfica (Lugares turísticos, negocios, rutas, estaciones de medios de transporte, etc.) Y la razón de que estos recursos sean utilizados es que *“tienen una masa de usuarios que sobrepasa en mucho a la que pueden tener las principales firmas de programas de SIG. Estas herramientas son un punto de entrada sencillo y carente de barreras para el procesamiento y consulta de información geográfica por parte del gran público, que por el dinamismo y versatilidad incorporados en sus programas, hacen parecer estáticos y obsoletos a los principales proveedores de plataformas SIG comerciales.”* (Metternicht, 2006). Por otro lado tenemos también a herramientas de publicación de mapas como servidores de mapas y base de datos espaciales que representan una opción de mayor personalización para las aplicaciones geoespaciales.

Si se observa el panorama actual de las aplicaciones híbridas y los SIG, tomando en cuenta el uso un framework para desarrollo de aplicaciones móviles multiplataforma como lo es Phonegap, y la publicación y manejos de mapas en la web; teóricamente una integración sería posible, gracias a que el entorno de desarrollo se basa en lenguajes de para construir páginas web (HTML5, CSS3, Javascript) y lenguajes de servidor (PHP, Java, C#, etc.), y esto es complementado con tecnologías nativas de Android como plataforma contenedora, razón por la cual es posible hacer uso de características de hardware como el sensor GPS. Y por lo tanto ser una alternativa en la recolección, manejo, consulta, etc. de datos geoespaciales en tiempo real.

## JUSTIFICACION

En el ámbito mundial el espacio geográfico ha representado en los últimos años una tendencia de estudio para la toma de decisiones en una organización de cualquier tipo (gobierno, educación, agricultura, negocios, salud, etc.), más sin embargo también ha se ha vuelto un elemento para la toma de decisiones en la vida diaria para todo público "Incluso las personas que nunca han utilizado los mapas para analizar los datos encuentran que los mapas hacen que procesar la información se mucho más fácil y más eficaz." (ESRI, 2012).

"En todo el mundo, la gente está trabajando más eficientemente a causa de la información geográfica" (ESRI, 2012), y la causante de esta realidad es la tecnología; hoy en día se observa a casi cualquier persona accediendo a información geográfica desde su PC o dispositivos móviles (Smartphone o tablet) para tomar una decisión, ya se consultando una dirección, marcando una ruta para llegar a un sitio, etc., la información geográfica ha llegado a ser indispensable.

Si bien lo anterior es tendencia, se dio gracias al avance tecnológico que ha sido aprovechado por los SIG, como a lo largo de este documento se ha mencionado los el análisis de la información geográfica en los SIG depende del trabajo de escritorio y el de campo, el cual se ha visto beneficiado gracias a los dispositivos móviles como los GPS.

*"Las apps corporativas híbridas y HTML5 son las dos tendencias que están marcando el avance en el escenario de las aplicaciones móviles. Así lo constata la empresa experta en desarrollo web y móvil, Tecnológica, que pronostica un importante crecimiento de este mercado a la vista de la configuración del nuevo usuario móvil y multidispositivo. "* (Gutiérrez-Rubí, 2014)

De tal manera que la innovación en dispositivos móviles es el futuro, incluso posiblemente en el campo de la Geoinformática, ya sea que las funciones sean para trabajo de campo o para consulta de información para público en general, los *smartphones* o *tablets* con acceso a internet, son un medio para intercambiar y compartir información con mayor facilidad y de manera más rápida.

Pero enfocándose de una manera puntual en el futuro de las aplicaciones para dispositivos móviles, existe una vertiente que es una combinación entre las aplicaciones nativas, y aplicaciones web, estas son llamadas aplicaciones híbridas, estas tienen una base de codificación en ambiente web (HTML5, CSS y Javascript), pero están encapsuladas en una plataforma de desarrollo (Android, iOS, etc.) lo que

permite a la aplicación hacer uso de las funciones de software y hardware (GPS, Acelerómetro, Cámara, etc.) del dispositivo; lo cual hace a la aplicación resultante sea más dinámica; todas estas funciones logradas gracias a un framework .

*“No cabe considerar a las aplicaciones híbridas como una fórmula de compromiso, a medio camino entre las aplicaciones HTML5 y las 100% nativas, sino como una opción superior a ambas alternativas” (Delgado, 2013).*

Mirando entonces al entorno de los SIG y la datos geoespaciales, las aplicaciones híbridas podrían ser una herramienta muy útil en tareas de análisis de la información y compartición de la misma.

Se pretende una integración de servicio y herramientas de publicación de mapas en la web, con un framework llamado PhoneGap, para generar aplicaciones híbridas para dispositivos móviles con funciones geotecnologías para plataformas Android, *“el método de creación de aplicaciones híbridas será muy popular para integrar no sólo servicios SIG distribuidos sino también SIG con otras aplicaciones corporativas de tecnologías de la información” (Dangermond, 2008)*, se supone en primera instancia de esta integración podría surgir una relación estrecha entre las tareas las potencialidades de las geotecnologías y la practicidad que en ocasiones puede dar un aplicación en dispositivos móviles, lo cual podría llevar a una mejor interacción en el intercambio de datos geográficos en menor tiempo. Entonces en cuanto al campo de las aplicaciones híbridas es válido que se siga experimentando para de esta forma poder vislumbrar las posibilidades del desarrollo de aplicaciones híbridas en plataformas móviles.

# OBJETIVOS

## Objetivo general

- Integrar servicios y herramientas de publicación de mapas a partir de la geolocalización en dispositivos móviles con un framework llamado PhoneGap para generar aplicaciones geoinformáticas para dispositivos móviles en plataforma Android.

## Objetivos específicos

- Generar un proyecto PhoneGap e instalar los plugins para los sensores que se utilizaran (GPS, Cámara, transferencia de archivos, notificaciones).
- Obtener coordenadas X, Y con el sensor GPS de un *smarthphone* y mapear la posición por medio del API v3 de Google Maps.
- Transferir las coordenadas obtenidas a una base de datos MySQL.
- Diseñar aplicación web para reportar fallas en servicios públicos utilizando HTML5, CSS3, Javascript y PHP, además de librerías como JQUERY Mobile.
- Integración de diseño y funciones generadas para enviar y obtener coordenadas X,Y utilizando PhoneGap.
- Compilar y probar aplicación en emuladores y dispositivos reales

# CAPITULO 1. MARCO TEÓRICO

## 1.1. Aplicaciones para dispositivos móviles: Visión general

Un segmento que ha mejorado a raíz de la evolución en la tecnología en el equipamiento de los teléfonos inteligentes es el que se refiere a las aplicaciones móviles, *“entendemos como aplicación móvil, cualquier aplicación software creada por terceros y destinada a su instalación y ejecución en un dispositivo móvil, esto es de tamaño reducido e ideado para ser utilizado de manera inalámbrica.”* (Soto Aboal, 2012).

Gracias a estas aplicaciones se ha logrado que los *Smartphone, tablets* y dispositivos móviles tengan funciones más allá de aquellas que se consideran básicas ej. Llamadas, mensajes, administrador de contactos, navegar en internet, reproducción de música etc.

Entre las funciones adicionales más populares que brindan las aplicaciones para dispositivos móviles según el *“Mobile Consumer Report 2013 de Nielsen”* están:

- Juegos
- Redes Sociales (Facebook, Twitter, Instagram, etc.)
- Videos y Películas
- Noticias
- Navegación y búsqueda en Mapas
- Clima
- Finanzas y transferencias bancarias
- Compras
- Navegación Web
- Herramientas de productividad
- Reproducción de música online
- Fotografía y grabación de video
- E-mail

Pero más allá de las funciones que las aplicaciones pueden llevar a cabo se encuentran los beneficios que los Sistemas Operativos (SO) para dispositivos móviles han brindado a este mercado. Un ejemplo de estos SO son Android, iOS y Windows Phone, además estos son las plataformas de desarrollo en la que los desarrolladores se enfocan al momento de crear y publicar sus productos en las tiendas de aplicaciones móviles, esta tendencia se puede observar claramente en la siguiente gráfica.



Ilustración 1. Número promedio de aplicaciones publicadas diariamente. StarDust. Agosto 2013

Se puede observar que en primer plano el número total de aplicaciones existentes para cada plataforma antes mencionada (iOS, Android y Windows Phone) hasta Agosto del 2013, y en la parte inferior un promedio de aplicaciones publicadas diariamente, claramente se aprecia que para ese momento el mayor número de aplicaciones móviles estaban desarrolladas para iOS sin embargo el incremento de popularidad para la plataforma Android puede observarse al superar a sus competidores en número de aplicaciones publicadas diariamente. Se puede también asegurar la popularidad las plataformas graficadas, razón por la cual los desarrolladores están concentrados en generar aplicaciones para dispositivos móviles que funcionen con estos SO.

En el caso particular de Android el aumento de su popularidad puede integrarse al hecho de que diversos fabricantes de telefonía celular optaron desde el año 2009 instalar Android a sus productos, por esta razón el número de aplicaciones publicadas en Google Play<sup>1</sup> creció estrepitosamente, dicha afirmación puede corroborarse en la siguiente gráfica donde se aprecia como el número de aplicaciones disponibles para su descarga en la tienda Google Play en un lapso de seis años desde finales de 2009 hasta inicios de 2015.

<sup>1</sup> **Google Play Store** (anteriormente Android Market): es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google.

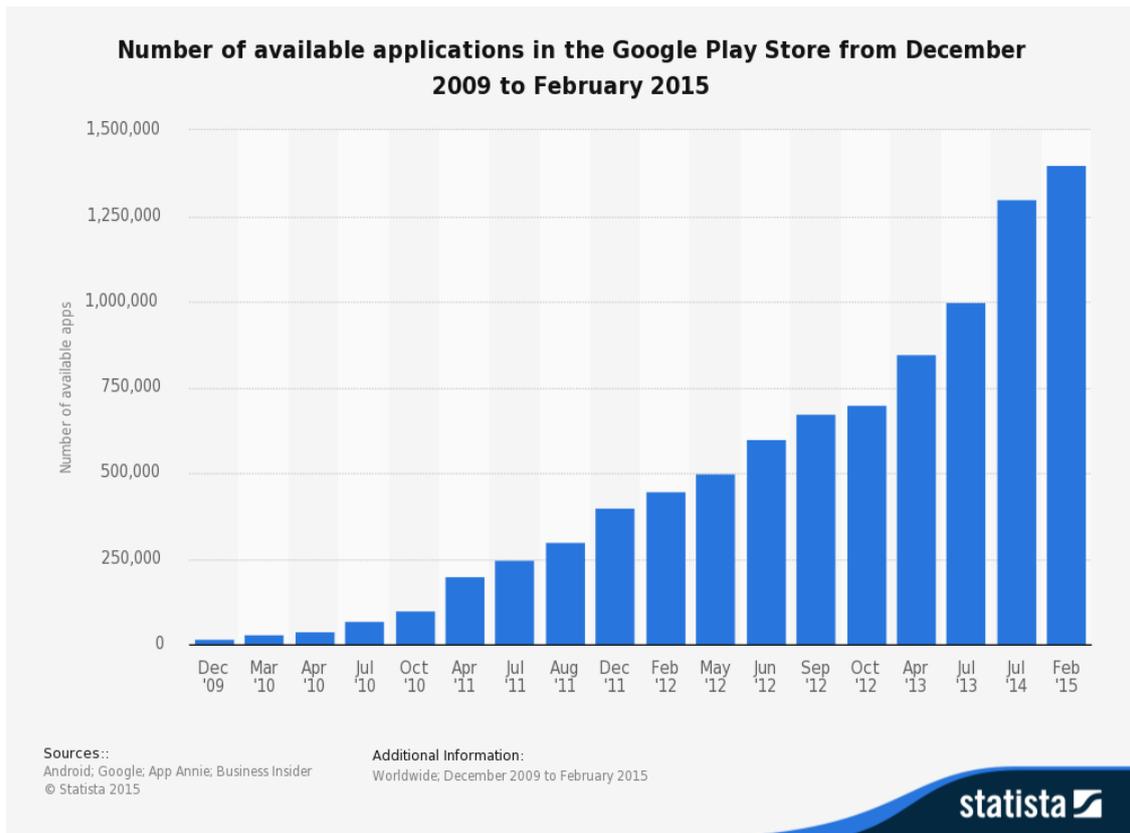


Ilustración 2. Número de aplicaciones disponibles en Google Play desde el 2009 hasta 2015.  
Fuente: IDC & Statista

### 1.1.1. Sistemas Operativos móviles, situación actual

En la actualidad existen una gama diversa de SO para dispositivos móviles que abarcan aquellos que son casi obsoletos (Symbian, WebOS, Windows Mobile, etc.), los más recientes (Firefox OS, Tizen y Ubuntu Touch) y además aquellos que lideran el mercado actual en dispositivos móviles (Android, iOS, BlackBerry y Windows Phone).

*“El Sistema Operativo (SO) móvil de un teléfono o tableta significa la interacción real con lo que se puede hacer a partir de las capacidades del hardware que conforman un equipo.” (Rivera, 2012)*

Los dispositivos móviles recientes de acuerdo al hardware que tienen integrado requieren entre sus funciones más atractivas la rapidez y fluidez de las tareas que se llevan a cabo en cada equipo. Los sistemas que podrían presumir de tener esta característica se han posicionado en el mercado rápido y de una manera sólida. En este caso se puede hablar de Android y iOS, de acuerdo con la consultora IDC (International Data Corporation) estos sistemas combinados lograron el 91.1% de las ventas de teléfonos móviles en el cuarto trimestre del 2012. *“los vendedores de*

teléfonos inteligentes Android y Apple vendieron un total de 207,6 millones de unidades en todo el mundo durante el cuatro trimestre del 2012, un aumento de 70.2% respecto a los 122,0 millones de unidades vendidas durante el cuarto trimestre del 2011” (IDC, 2013)

Con lo anterior se puede observar el liderazgo de los dos SO Android y iOS, en contraste en el informe de IDC también se menciona que Windows Phone subió un 58%, BlackBerry se desplomó un 81% con respecto a sus ventas en el mismo periodo 2011-2012 del estudio.

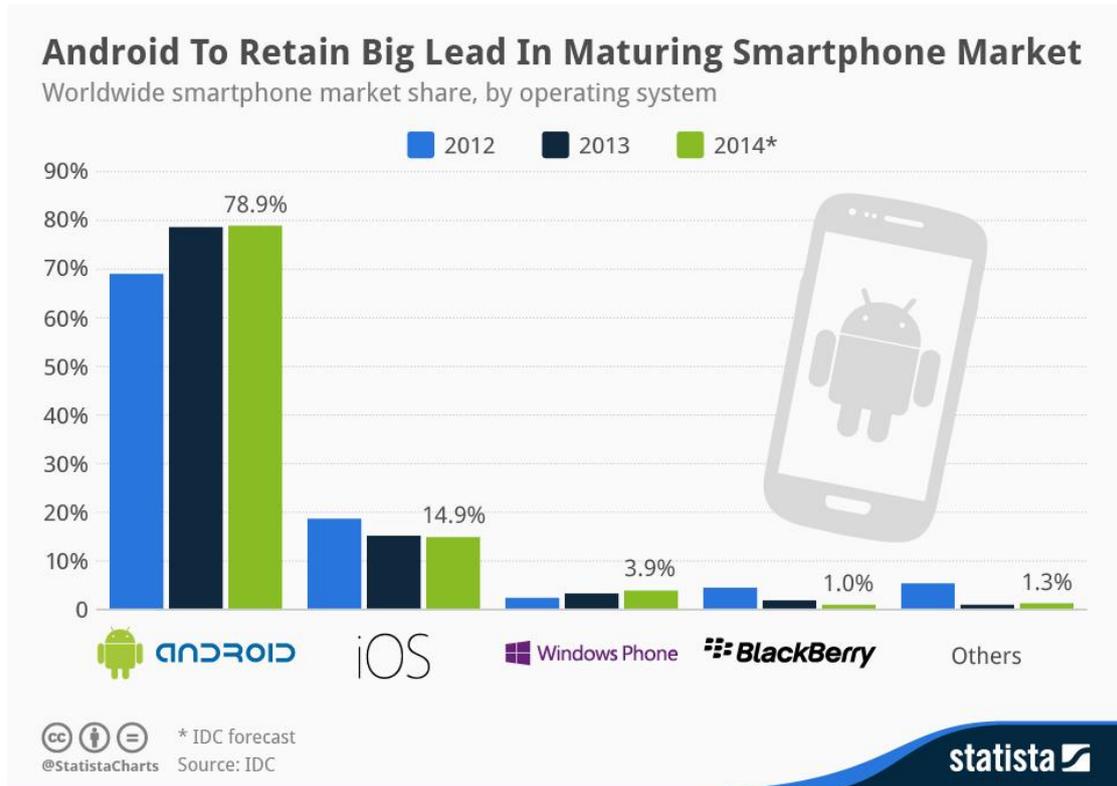


Ilustración 3. Predicciones del mercado global por Sistema Operativo. Fuente: IDC & Statista

En la anterior infografía se refleja el dominio que ha tenido Android en el mercado actual a comparación de otros SO móviles que se han quedado rezagados.

Las estadísticas, ventas, y características como rapidez y fluidez de los SO móviles, son importantes pero un elemento clave para que una plataforma o sistema operativo triunfe, son los desarrolladores. “Los desarrolladores son pieza clave en el posicionamiento de cualquier sistema operativo. El hecho de que proporcionen más opciones de contenido y uso de un teléfono hacen que una plataforma sea más apetecible puesto que permiten democratizar los gustos y las necesidades de un por demás fluctuante público usuario.” (Rivera, 2012)

Sistemas Operativos como Android y iOS cuentan con paquetes de herramientas de desarrollo (SDK), para construir, probar y depurar

aplicaciones nativas (pág. Siguiendo 1.1.2.1). Gracias a esto se ha podido ver un gran incremento en el número de desarrolladores en estas plataformas.

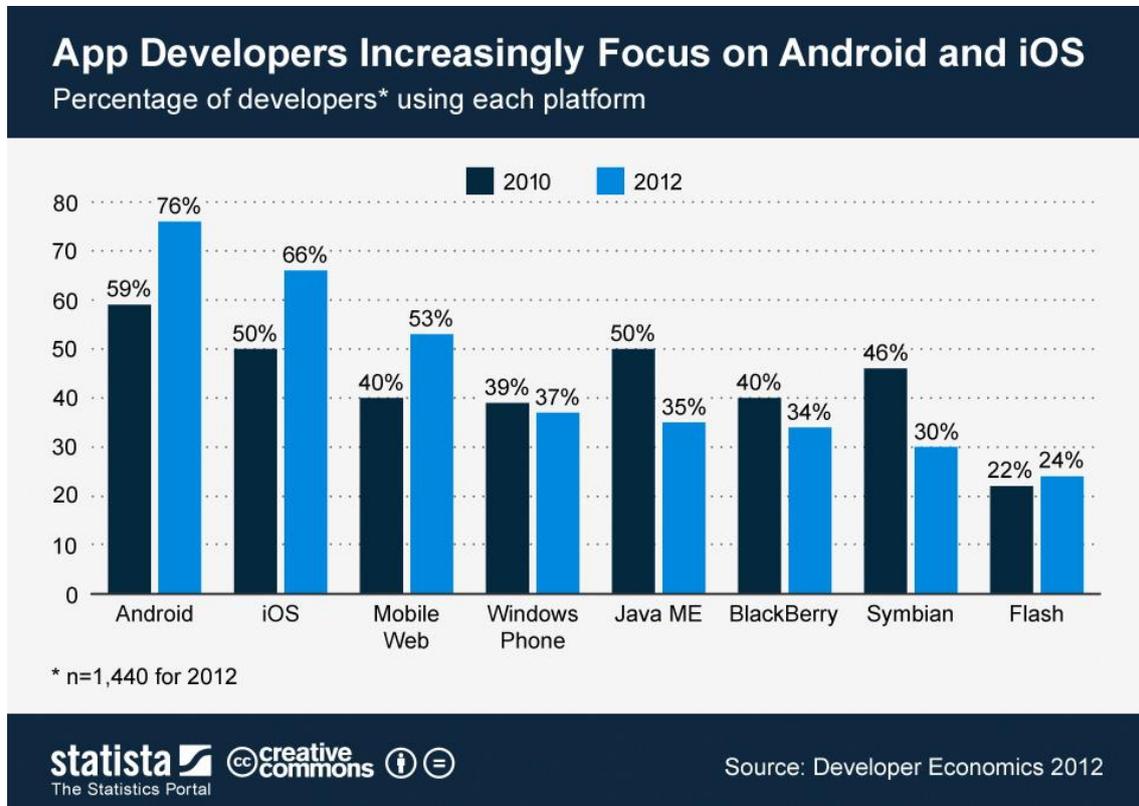


Ilustración 4. Incremento de desarrolladores en plataformas Android y iOS. Fuente: Developer Economics & Statista

Esta gráfica muestra que los desarrolladores de aplicaciones se centran cada vez más en el desarrollo para Android y iOS, mientras que otras plataformas están perdiendo poco a poco el apoyo de desarrolladores.

Y si bien ahora se conocen las plataformas más populares resta conocer el tipo de aplicaciones que son desarrolladas para estas plataformas.

### 1.1.2. Tipos de aplicaciones

Con el objetivo de cubrir las necesidades actuales, los desarrolladores generan aplicaciones con funciones varias para los dispositivos móviles, lo anterior gracias a las herramientas de desarrollo brindadas por los SO móviles. Pero muy aparte de estas herramientas existen otras alternativas para generar aplicaciones y además existen diferentes tipos de aplicaciones.

Se puede nombrar entre los tipos de aplicaciones las siguientes.

- Nativas
- Aplicaciones web
- Híbridas

El tipo de aplicación que se esté dispuesto a desarrollar siempre dependerá de las necesidades de cada solución, es decir la problemática y su complejidad, el hardware, software utilizado, etc.

### 1.1.2.1. Aplicaciones nativas

Las aplicaciones nativas son aquellas que están específicamente diseñadas para ejecutarse en un sistema operativo en específico. Con estas aplicaciones se pueden utilizar los sensores y funcionalidades del hardware de cada dispositivo

*“Una aplicación nativa se escribe en un lenguaje de programación y una interfaz para un sistema operativo y dispositivo específicos. De este modo, puede ofrecer el mejor rendimiento, pero también requiere una versión diferente (y es más costoso) para cada sistema operativo.” (IBM España, 2011)*

Gracias a la naturaleza de estas aplicaciones, se puede lograr una mejor experiencia de usuario, mejores gráficos, seguridad, funcionalidad sin conexión web, hacer uso de utilidades del dispositivo, entre otras. Sin embargo, algunas de las principales desventajas que se pueden presentar para los desarrolladores son el costo, mantenimiento y el tiempo en desarrollo de la aplicación.

Algunas herramientas nativas y lenguajes de desarrollo que se utilizan actualmente de acuerdo a las estadísticas mostradas anteriormente son:

SDK y Lenguajes de Programación de principales plataformas móviles nativas.		
Plataforma	SDK	Lenguaje
Android	Android SDK <ul style="list-style-type: none"><li>• Eclipse</li><li>• Android Studio</li></ul>	Java
iOS	XCode	Objetive C
Windows Phone	Visual Studio	C# y Visual Basic .NET

Tabla 1. SDK y Lenguajes de Programación de principales plataformas móviles nativas.

### 1.1.2.2. Aplicaciones WEB móviles

Estas aplicaciones son aquellas que están diseñadas para web browser de celulares. Es necesario tener acceso a internet para que estas puedan funcionar.

Las aplicaciones web pueden ser más compatibles con todos los dispositivos, SO y navegadores web, dependiendo esta compatibilidad con que tanto los desarrolladores sigan los estándares de desarrollo web. Las tecnologías recientes como HTML5, CSS3 combinadas con JavaScript permiten crear aplicaciones web adaptadas al dispositivo móvil de forma que parezcan aplicaciones nativas.

### 1.1.2.3. Aplicaciones híbridas

Las aplicaciones híbridas son por su naturaleza multiplataforma por diversas razones y a comparación de las aplicaciones web, las aplicaciones híbridas no necesitan una conexión a internet a pesar de que suelen estar programadas con lenguajes web es decir HTML5, CSS, JavaScript; cabe detallar que no son un navegador web ya pueden ser compiladas para diferentes plataformas y hacer uso del hardware de los dispositivos móviles (Cámara, GPS, Acelerómetro, etc.)

Algunas características de las aplicaciones híbridas para móviles son las siguientes:

*“1. Crear un núcleo de código común utilizando tecnologías web (HTML5, JavaScript, CSS);*

*2. Complementarlo de forma opcional con elementos desarrollados con tecnologías nativas de cada plataforma (Objective C para iOS, Android SDK, etc.);*

*3. Y, finalmente, encapsularlo en un contenedor específico para cada plataforma, de manera que la aplicación pueda interactuar con el sistema operativo subyacente, acceder a las capacidades del hardware y pueda distribuirse a través de los markets oficiales de cada fabricante (p. ej. – Apple Store, Google Play, etc.)” (Delgado, 2013)*

En la encuesta *Global Developer Survey*<sup>2</sup> llevada a cabo por Telerik se menciona que empresas grandes y pequeñas han decidido adoptar

---

<sup>2</sup> Encuesta a nivel mundial por la compañía Telerik a 2,200 desarrolladores y ejecutivos de en el ramo de tecnologías de la información con el fin de determinar las prioridades y puntos débiles que rodean al desarrollo de aplicaciones desde escritorio hasta móviles, la fecha de recusación de datos fue entre el 15 de Julio y el 1 de Agosto del 2014

HTML5 para desarrollar aplicaciones móviles ya que es menos costoso y más flexible. HTML5 suele ser por lo general el núcleo de código de gran parte de las aplicaciones híbridas.

## 1.2. Mapas en dispositivos móviles

Los mapas en dispositivos móviles son una herramienta que la mayoría de los *smartphones* y *tablets* incluyen gracias a que casi todos estos dispositivos cuentan con sensor GPS (Global Positioning System). La ubicación geográfica se ha vuelto importante para los usuarios de teléfonos inteligentes, *“Un uso común en las aplicaciones es mostrar que permitan al usuario situar rápidamente los lugares, los eventos, las tiendas...Explotando las funcionalidades de la localización geográfica, es posible geolocalizarse”* (Pérochon, 2012).

Los mapas y la localización geográfica han tomado un papel importante en las tareas de los usuarios de *smartphones*, tan solo en Estados Unidos el 48% de los usuarios de *smartphones* dicen usar los servicios de ubicación basados en la geolocalización con GPS, esto según el estudio *“The Mobile Consumer Global Snapshot 2013”*.

### 1.2.1. GPS y Geolocalización

Se conoce que GPS es una tecnología de posicionamiento eficiente. *“El GPS es un sistema basado en satélites artificiales, dispuestos en un constelación de 24 de ellos, para brindar al usuario una posición precisa”* (Leica Geosystems Inc, 2000)

Para que el hardware GPS de los dispositivos móviles funcione correctamente depende de un conjunto de segmentos que se mencionan a continuación:

- Segmento espacial: Conformado por la constelación de satélites GPS y los mismos satélites GPS
- Segmentos de control: Consiste en estaciones de control maestro, consiste en cinco estaciones de observación y cuatro antenas de tierra distribuidas en cinco puntos muy cercanos al ecuador terrestre.
- Segmento de usuario: comprende cualquiera que reciba señales GPS con un receptor, determinando su posición y hora.

El conocimiento de la ubicación geográfica y la navegación se ha convertido en una de las características más importantes de los teléfonos inteligentes y dispositivos móviles, más sin embargo existen desventajas en los sensores GPS de estos dispositivos, *“los teléfonos inteligentes por lo general tienen chips GPS de relativamente bajo costo, el desempeño de la localización y precisión es altamente dependiente de los factores ambientales”* (Hwang & Yu, 2012)

Por otro lado la exactitud de GPS varía también en función del número de satélites GPS y se reduce con lugares que interfieran con la señal, como en un bosque o alrededor de los edificios.

El GPS es importante para la geolocalización en la mayoría de los casos el término geolocalizar se entiende como la capacidad de asignar coordenadas geográficas a la información por medio de sensores de localización.

El poder obtener la posición geográfica permite a los usuarios de dispositivos móviles usar esta información para buscar lugares, rutas para llegar a un punto, ubicar direcciones, etc. Los datos obtenidos por la geolocalización es decir las coordenadas geográficas (para los GPS son “X, Y, Z” y para los sensores GPS en dispositivos móviles es decir *smartphones* y *tablets* son solo “X, Y”) son de gran ayuda al momento de obtener la localización inmediata de algún elemento de estudio, ya sea el comportamiento en las redes sociales, patrones de venta de productos, localización de fallas en servicios públicos, tránsito, etc.

Una pieza clave para que la geolocalización en dispositivos móviles funcione o sea tan popular han sido los servicios de mapas como Google Maps, que son librerías de cartografía online para que los usuarios de teléfonos inteligentes y demás dispositivos con SO móviles puedan utilizar, usando los sensores o chips GPS que cada dispositivo tiene integrado.

*“La generalización de la tecnología GPS en dispositivos de uso personal como los teléfonos móviles y ordenadores personales ha permitido que esta capacidad esté al alcance de cualquier ciudadano, y como consecuencia, el desarrollo de aplicaciones a distintos campos”* (Rodríguez, 2013)

## 1.2.2. W3C Geolocation API Specification

La API<sup>3</sup> de geolocalización fue diseñada para proporcionar acceso a la información de ubicación geográfica de un dispositivo conectado a internet, funciona normalmente en las versiones actuales de navegadores como Chrome, Firefox, Safari y Opera. La W3C Geolocation API Specification son las reglas a seguir ya que en esta especificación se describe la forma estándar para utilizar los servicios de geolocalización desde dentro de las aplicaciones Web

En PhoneGap el uso de esta API dependerá mucho de la compatibilidad del dispositivo móvil con HTML5, ya que cuando es usado el *plugin* "geolocation" la aplicación generada tiene permiso de usar el hardware de localización, optimizando el rendimiento de la API de geolocalización en comparación con una aplicación web.

## 1.2.3. Google Maps

Actualmente existen aplicaciones famosas específicamente de navegación geográfica en mapas para dispositivos web entre las aplicaciones más famosas están *Google Maps*, *Apple Maps*, *Here de Nokia* o *Maps de Bing exclusivo para dispositivos Nokia Lumia*.

Para esta investigación se hará solo referencia a Google Maps ya que se utilizará la API para desarrolladores y de esta manera acceder a funciones de navegación y geolocalización en dispositivos Android.

Se podría decir que Google Maps es una aplicación de las más importantes o si no la más importante en cuanto a navegación en mapas y localización geográfica, está disponible para las tres plataformas más importantes de la actualidad Android, iOS, Windows Phone y además funciona en Navegadores Web de dispositivos móviles. En los dispositivos Android se encuentra instalado casi en todos los casos de fábrica, pero aun así está disponible en la Google Play Store y ha sido instalada más de quinientos millones de veces.

Algunas de las características importantes de la aplicación son:

- Google Maps muestra dónde está el usuario gracias al GPS de cada dispositivo.

---

<sup>3</sup> **API:** es el conjunto de subrutinas, funciones y que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

- Muestra rutas para llegar a un punto deseado y las formas de llegar, además dar indicaciones a la aplicación por medio de la voz.
- Se puede conocer el tiempo aproximado para llegar a tu destino en función del estado real del tráfico.
- Se pueden buscar lugares (Restaurantes, Museos, empresas) y la información acerca de estos. Además se puede usar la posición actual para identificar los lugares que se encuentran alrededor.

### 1.2.3.1. *API de Google Maps*

Google Maps para desarrolladores es un conjunto amplio de APIs que permiten utilizar las funciones y utilidades de Google Maps en sitios web, aplicaciones web, aplicaciones móviles, etc.

Con las APIs de Google Maps los desarrolladores pueden crear aplicaciones basadas la ubicación geográfica, se pueden utilizar funciones específicas como conectar personas con lugares, formar rutas de un punto A al punto B, además se puede acceder a la base de datos de lugares y referencias de Google Maps, por otro lado ofrece la función de generar mapas con datos geoespaciales en la nube y marcadores personalizados, y visualización StreetView y oportunidad de añadir imágenes en 3D a las aplicaciones generadas.

En cuantas aplicaciones móviles, con la versión 3 de API de Google Maps se pueden crear aplicaciones HTML5 y JavaScript que se pueden utilizar en plataformas móviles y también en aquellas de escritorio, ya que la API funciona bien en la mayoría de tamaños de pantalla.

En cuanto al licenciamiento Google Maps API cuenta con dos versiones, una gratuita y otra que requiere pago (API Google Maps for Business). EN la tabla 2 se presentan los servicios que ofrece cada una.

Comparación de funciones entre API Google Maps y API Google Maps for Business.		
Funciones	API de Google Maps	API de Google Maps for Business
<b>Street View</b>	✓	✓
<b>Servicio web de codificación geográfica</b>	2.500 solicitudes diarias	100.000 solicitudes diarias
<b>Servicio web de rutas</b>	2.500 solicitudes diarias con 10 hitos por solicitud	100.000 solicitudes diarias con 23 hitos por solicitud
<b>Servicio web de matriz de distancia</b>	100 elementos por consulta 100 elementos cada 10	625 elementos por consulta 1.000 elementos cada

	segundos 2.500 elementos diarios	10 segundos 100.000 elementos diarios
<b>Servicio web de elevación</b>	2.500 solicitudes diarias con 25.000 muestras diarias	100.000 solicitudes diarias con 1.000.000 muestras diarias
<b>Resolución máxima de la API de Google Static Maps</b>	640 x 640	2.048 x 2.048
<b>Escala máxima de la API de Google Static Maps</b>	2X	4X
<b>Resolución máxima de la API de imágenes de Street View</b>	640 x 640	2.048 x 2.048

Tabla 2. Comparación de funciones entre API Google Maps y API Google Maps for Business.  
Fuente: Developers Google Maps

Complementariamente la versión Business ofrece Asistencia técnica, y un portal de informase de usos, distribución de servicios con Google Maps, control de publicidad Seguimiento de activos privados, etc.

### 1.3. Aplicaciones híbridas con PhoneGap en Android

El potencial de las aplicaciones híbridas ha crecido a raíz de las funcionalidades que frameworks de desarrollo como PhoneGap ofrecen, ya que cada versión libera nuevas características en cuanto al control del dispositivo móvil tratando de evadir con esto el lenguaje nativo de cada plataforma lo que permite que el código que se desarrolle pueda ser ocupado en diferentes plataformas móviles y de esta manera no se necesita un conocimiento específico de las herramientas de desarrollo de software asociados a cada plataforma móvil.

#### 1.3.1. PhoneGap

PhoneGap es un framework de código abierto de Adobe para desarrollar aplicaciones móviles con tecnologías estándar (HTML, CSS y JavaScript) y compilarlas para distintas plataformas móviles. En principio PhoneGap fue creado por una empresa canadiense llamada *nitobi*, Misma que creó una forma de crear aplicaciones web y empaquetarlas de manera nativa para SO móviles (Android, iOS, Windows Phone, Blackberry, etc.), posteriormente esta misma empresa liberó el código de PhoneGap. En

octubre del 2011 el “nitobi” fue adquirido por Adobe, sin embargo PhoneGap sigue siendo de código abierto ya que este proyecto fue donado a la fundación Apache, dentro de esta fundación el proyecto PhoneGap lleva un el cual es Apache Cordova, esto porque el nombre comercial PhoneGap es propiedad de Adobe.

### 1.3.2. Características

El framework PhoneGap es un conjunto de elementos y utilidades, como tal es una plataforma que permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo de aplicaciones multiplataforma, evitando el lenguaje de desarrollo nativo de cada plataforma móvil.

Las aplicaciones se ejecutan empaquetadas para cada plataforma y dependen de enlaces estándar con APIs que se instalan a través de líneas de comandos para acceder a las funciones de cada dispositivo es decir a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc.

Las aplicaciones desarrolladas en PhoneGap se pueden compilar y publicar en los mercados de aplicaciones móviles (Google Play, Apple Store, etc.)

*“Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado<sup>4</sup> es realizado mediante vistas web), pero no se tratan tampoco de aplicaciones web.”* (Video2Brain, 2011)

#### 1.3.2.1. Arquitectura

La arquitectura de una aplicación PhoneGap se integra de distintos elementos compuestos por procesos de desarrollo web que involucran tanto al dispositivo móvil como la programación en lenguajes como HTML, CSS y JavaScript. En la ilustración 5 se muestran las partes que conforman la arquitectura que compone una aplicación PhoneGap. (Ver p. 25)

---

<sup>4</sup> Término usado en para referirse al proceso de generar una imagen desde un modelo en dos o tres dimensiones.



Ilustración 5. Arquitectura de una aplicación híbrida en PhoneGap. Fuente: Curso iOS y Android con PhoneGap y jQuery. Maximiliano Firtman

Como se mencionó anteriormente una aplicación de PhoneGap está compuesta de distintos elementos, estos elementos son específicos y cumplen con una función particular, en el diagrama en la parte inferior se puede ver la interfaz de usuario (UI) que es la apariencia y la interacción con el usuario, los toques y desplazamientos en la pantalla, etc. La UI está compuesta de HTML, CSS, algunas librerías de PhoneGap para notificaciones y eventos, por otro lado existen frameworks de interfaz de usuario como jQuery y jQuery Mobile, que brindan código para que las aplicaciones luzcan como nativas gracias a los efectos del CSS3 y HTML5.

La *lógica* depende puramente de JavaScript, algunas características y extensiones de HTML5 y optativamente jQuery. Todo lo anterior para generar las funciones de la aplicación.

Los *datos* y la administración de datos (*Data* en la ilustración 5) se llevan a cabo con tecnología como AJAX, aunque se puede hacer uso de librerías de HTML5 que permiten almacenar datos en el dispositivo móvil y que estas persistan y se conserven guardadas aunque se cierre la aplicación. Adicionalmente PhoneGap tiene librerías que permiten el acceso a contactos y archivos que el dispositivo móvil almacene en su memoria interna.

Como se había mencionado antes, por medio de PhoneGap es posible acceder a funciones del hardware del dispositivo gracias a que provee de un conjunto de APIs para acceder a funciones de los sensores como GPS, acelerómetro, cámara, antena WIFI, sensor de movimiento, etc.

En el diagrama se hace referencia en la barra interior a los Metadatos, estos Metadatos deben ser generados para cada plataforma para la que se vaya a generar la aplicación y contienen información extra como el icono de la aplicación, el nombre que va a tener, los *plugins* y librerías de PhoneGap que han sido integrados a la aplicación que se va a generar.

Al tener todo los elementos anteriores desarrollados, se puede proceder a empaquetar la aplicación para que esta funcione en las plataformas como Android, iOS, Windows Phone, etc.

Se debe considerar que las plataformas soportadas por PhoneGap son las siguientes:

- Android
- Blackberry
- iOS
- Windows Phone
- Tizen
- Webos
- Symbian

También se debe de tomar en cuenta que algunas funciones no están disponibles para todas las plataformas, para información más específica se debe consultar el apartado de soporte de PhoneGap.

### 1.3.3. Datos en PhoneGap

En PhoneGap cuenta con una API de almacenamiento de datos local lo cual permite almacenar un limitado volumen de datos, el uso de esta API es relativamente sencilla, además se pueden generar consultas en SQL pero no consultas complejas, por otro lado en PhoneGap existe el plugin "File" este plugin contiene funciones que permiten a los usuarios el control de archivos en el dispositivo y además la descargar y carga de archivos a almacenamientos externos más sin embargo solo son archivos es por ello que en esta investigación se utilizó JSON que es un protocolo de intercambio de datos.

### 1.3.4. Servicios REST

Un servicio que se ha convertido en un recurso muy usado como alternativa a arquitecturas como SOAP para la transferencia de datos de una forma más ligera, debido al núcleo web de las aplicaciones híbridas que incluye un lenguaje como JavaScript el uso e integración de servicios REST se ve potenciado a su vez si se hace uso de plataformas como PhoneGap. REST (REpresentational State Transfer), es un tipo de arquitectura de desarrollo web que nos permite generar servicios que pueden ser consumidos por cualquier dispositivo que sea capaz de interpretar el protocolo HTTP por medio de un lenguaje de transferencia que puede ser JSON o XML.

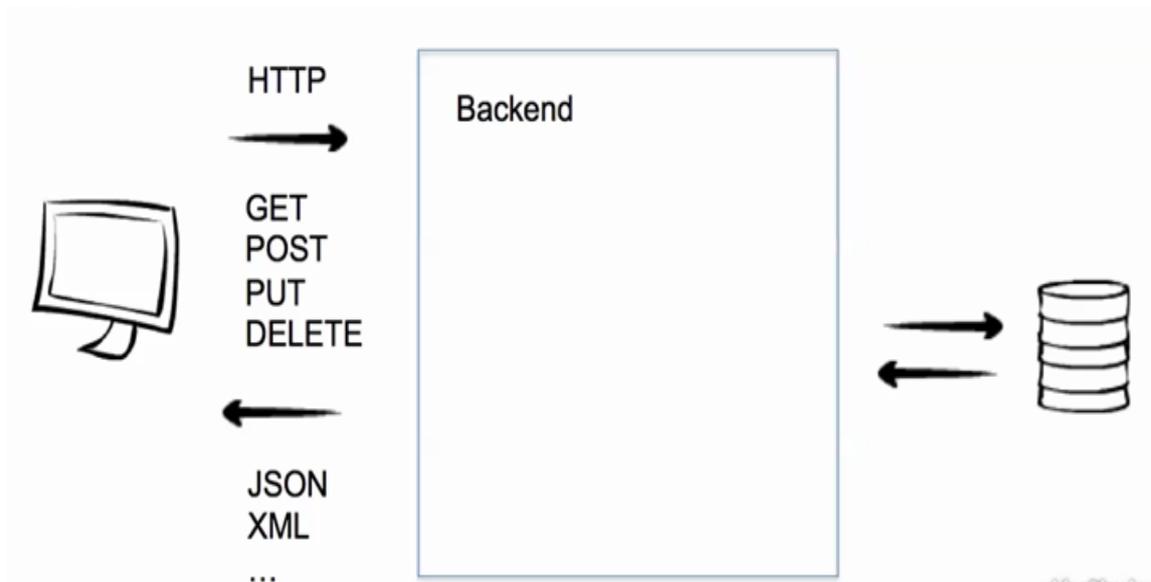


Ilustración 6. Arquitectura de Servicio REST. Fuente: [www.video2brain.com](http://www.video2brain.com)

### 1.3.5. JSON como protocolo de transferencia de datos

JSON es un protocolo ligero de transferencia e intercambio de datos basado en JavaScript y además una alternativa a XML. El formato JSON hace posible que se puedan compartir datos entre cliente y servidor y si bien está basado en JavaScript es interoperable y funciona para varios lenguajes de programación.

“Da igual si es móvil, escritorio, web, etc. Prácticamente JSON estará como protagonista en todos los lenguajes más conocidos y habituales usado en este momento” (Berganza, 2014)

Una estructura JSON está compuesta por objetos, un ejemplo es el siguiente:

```
{
  "Nombre" : "Edgar",
  "Edad": "23",
  "Aficiones": ["Música", "Futbol", "Nadar"],
  "Residencia": "Madrid"
}
```

Como se puede observar existen dos tipos de estructuras:

- Un conjunto de pares (clave, valor) encerrado entre los caracteres "{" y "}", separando la clave del valor por el símbolo ":", y separando cada par del siguiente con el carácter ","
- Un conjunto ordenado de valores encerrado entre los caracteres "[" y "]", y separando cada valor del siguiente con el carácter ",".

En este proyecto el lenguaje JSON se utilizara para comunicar y transferir datos recuperados por GPS del dispositivo móvil con aplicaciones híbridas a una base de datos en servidor externo.

## CAPITULO 2. METODOLOGÍA

Una aplicación híbrida como se ha mencionado anteriormente tiene una base de programación Web y de la misma forma comparte atributos de una arquitectura de desarrollo nativa, en esta investigación se realiza la integración de aplicaciones híbridas con otras tecnologías como conexión a base de datos, representación cartográfica en la web utilizando geolocalización de los dispositivos móviles para obtener coordenadas y generar datos que puedan ser consumidos posteriormente por medio de servicios web en *smartphones*, *tablets* o incluso computadoras personales o de escritorio.

Es importante mencionar que la generación de datos a partir del uso del GPS de los dispositivos móviles va de la mano con los servicios web a los que se conecta para almacenar, manejar o consumir datos debido a que si están dirigidas a que múltiples usuarios utilicen la aplicación para alimentar una base de datos, la interacción entre la aplicación *Front-end*<sup>5</sup> y *Back-end*<sup>6</sup> será frecuente.

Conforme a lo que se ha podido observar en los últimos años la obtención y almacenamiento de datos se ha convertido en una constante en las aplicaciones web y móviles ya que las formas de obtener datos se resumen en combinaciones de técnicas que generan automáticamente datos a partir de la detección del navegador, dirección IP, conexión a internet, incluso la localización como ya se comentó en esta investigación. Dichas técnicas mencionadas tienen mucho que ver con el auge de la geolocalización y la información generada a través del uso del GPS, ya sea detección de idioma, estado del tiempo, generación de rutas, etc. El poder utilizar la posición del dispositivo han disminuido considerablemente haciendo a veces hasta imperceptible la recopilación de datos haciendo en ocasiones imperceptible cuando las aplicaciones capturan la información del usuario.

Un ejemplo muy claro acerca de que a veces la recopilación de datos por parte de las aplicaciones móviles es el *Geofencing*<sup>7</sup> que ofrece la

---

<sup>5</sup> **Front-end:** El front-end en diseño de software y desarrollo web hace referencia a la visualización del usuario navegante o, dicho de otra manera, es la parte que interactúa con los usuarios

<sup>6</sup> **Back-end:** back-end es la parte que procesa la entrada de datos que se efectuó desde el front-end es decir, son los procesos que utiliza el administrador del sitio con sus respectivos sistemas para resolver las peticiones de los usuarios.

<sup>7</sup> **Geofencing:** es una tecnología que define un límite virtual alrededor de un área geográfica en el mundo real. De este modo, se establece un radio de interés que puede generar una acción en un teléfono geo- habilitado u otro dispositivo electrónico portátil por anuncios publicitarios.

posibilidad de generar anuncios publicitarios acorde a la ubicación exacta del usuario para segmentar a los consumidores y potenciar de este modo las ganancias por publicidad al público indicado.

*“Con el poder que han cobrado los smartphones en el día a día de los usuarios, los anunciantes están impacientes por encontrar formas de segmentar a sus usuarios, especialmente ante las oportunidades que pueden ofrecer los datos de localización para segmentar a los usuarios sobre la marcha.”* (MD: Marketing Directo, 2013)

Y aunque en la actualidad el uso de datos de localización obtenidos desde dispositivos móviles ha tenido un gran avance en el marketing debido a que se desea segmentar la publicidad dependiendo de la región de los consumidores, el área de oportunidad de análisis en otros campos lo que da pie a una forma diferente y más dinámica de obtener datos de localización en tiempo real, para dar respuesta a necesidades de los usuarios de dispositivos móviles que en este caso son una pieza fundamental, ya que desde un punto de vista ellos son los que alimentan las bases de datos, o si es visto desde otra perspectiva las aplicaciones en dispositivos móviles que hacen uso de los datos de localización bien pueden funcionar como una herramienta personalizada para otro tipo de tareas como el trabajo de campo, en fin como ya se había mencionado el campo de acción es amplio.

La importancia de la recolección de datos ha dado como resultado diversos fenómenos en la red y diferentes puntos de vista acerca de los datos que pueden ser obtenidos de los usuarios por medio de las aplicaciones en los dispositivos móviles, un ejemplo que puede mencionarse acerca de lo señalado es que recientemente han ocurrido casos donde las políticas de privacidad de redes sociales como Facebook o Twitter han tenido que ser modificadas para proteger datos personales de los usuarios de dichas aplicaciones.

En el caso de los datos que tienen que ver con la localización de los dispositivos móviles se puede mencionar los citados en capítulos anteriores que señala lo siguiente *“La recopilación y uso de datos de geolocalización plantea cuestiones de privacidad importante. La política de privacidad una aplicación debe discutir cómo la aplicación utiliza los datos de geolocalización, si se comparte con cualquiera de las partes y el nivel de precisión de los datos”* (Adobe Systems, 2014)

Lo anteriormente mencionado no es un impedimento para la generación de datos a partir de la localización basta con implementar un anuncio de privacidad sin embargo esta acción se realiza cuando la aplicación es distribuida sin embargo en esta investigación solo se ejemplificara el proceso de obtención de datos a partir de la geolocalización con

dispositivos móviles por lo que no es necesario implementar política de privacidad aunque como ya se ha mencionado son importantes si la aplicación se distribuye.

Observando el marco de referencia alrededor de la geolocalización y la generación de datos a partir de la obtención de la posición geográfica, desde este punto y en adelante se tratará de explicar el proceso de desarrollo de la “Aplicación para reportar fallas en servicios públicos en el Municipio de Toluca” donde se intenta la integración de distintas técnicas para lograr el proceso, y cabe mencionar que es un prototipo, ya que las herramientas utilizadas se ajustan sólo a las generalidades planteadas en el proyecto, sin embargo sirve como base de referencia para completar la generación de datos a partir de las coordenadas obtenidas por el GPS de un dispositivo con Android y la transferencia de la información generada a un servidor externo.

## 2.1. Caso: “Aplicación para reportar fallas en servicios públicos en el Municipio de Toluca”

Entrando al marco geoinformático que corresponde a esta investigación a partir de este capítulo se explica el proceso de desarrollo y como se integró a distintas herramientas geotecnológicas para generar la *aplicación para reportar fallas en servicios públicos en el Municipio de Toluca* se integraron funciones que van desde la captura de los datos de localización hasta su recuperación y visualización en la aplicaciones híbrida.

Si bien la generación de la aplicación representó la integración de diferentes herramientas geoinformáticas, el desarrollo de aplicaciones híbridas que tienen que ver con generación de datos a partir multiusuarios y que a partir del almacenamiento esta información pueda ser utilizada para distintos fines en plataformas móviles tiene un trasfondo al que puede hacerse referencia desde el enfoque del uso de estas aplicaciones en la geoinformática, lo mencionado se puede entender como una base de dos corrientes del desarrollo una que está muy arraigada a las geotecnologías que es la interoperabilidad y otra que tiene que ver con la web 2.0<sup>8</sup> que también integra términos de interoperabilidad y que además que puede ser perfectamente adoptada en las aplicaciones híbridas por su núcleo web.

---

<sup>8</sup> El término Web 2.0 comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web.

Claramente se tienen argumentos para colocar a las aplicaciones híbridas como área de oportunidad en el campo de las geotecnologías, y las razones no son tan complicadas, basta con entender que en primer plano las aplicaciones híbridas pueden ser multiplataforma, lo cual no significa interoperabilidad como tal, la interoperabilidad viene este caso de los datos e intercambio de estos. Desde ese punto de vista consumo de datos desde servicios web por medio de aplicaciones híbridas en dispositivos móviles es el tema rescatable, con lenguajes de transferencia como JSON y los que pueden interpretar los datos transferidos desde la aplicación híbrida el caso de JavaScript las limitaciones para lograr aplicaciones con datos interoperables son menores dado que conforme a lo que ya se conoce los dispositivos con plataformas como Android y iOS incluso otras son capaces de interpretar el núcleo web de las aplicaciones híbridas. Desde otro enfoque ya relacionado con las herramientas de análisis del espacio geográfico *“el SIG siempre ha sido una aplicación tecnológica sustentada por datos”* (Levinsohn , 2001), y por esta razón cuando se generó la aplicación es este trabajo no se dejó de lado el pensamiento SIG y más bien se integró y adaptó para poder utilizarlo en el desarrollo.

Por otro lado si se habla de desarrollo de aplicaciones móviles se usarán por distintos usuarios y su vez estos compartirán información, la mira debe ser dirigida hacia cierto principio que son: *“el fortalecimiento de la inteligencia colectiva, la gestión de las bases de datos como competencia básica, el fin del ciclo de las actualizaciones de versiones del software, los modelos de programación ligera junto a la búsqueda de la simplicidad, el software no limitado a un solo dispositivo y las experiencias enriquecedoras de los usuarios”* (Cobo Romaní & Pardo Kuklinski, 2007). Ya que en años recientes cualquier persona tiene acceso a un *Smartphone* o *Tablet* y también la mayoría de los usuarios de estas tecnologías hacen uso de aplicaciones para compartir información de distintos tipos.

Es momento de explicar la *aplicación para reportar fallas en servicio públicos en el municipio de Toluca*, que si bien sirve como prototipo del eje principal de esta investigación (generación de datos espaciales a partir del uso de dispositivos móviles y su almacenamiento), el fin fue tratar de seguir los principios mencionados anteriormente con el objetivo planteado.

Bajo el principio antes mencionado la aplicación híbrida desarrollada funciona para recolectar datos de localización, y es capaz de almacenar la información obtenida en una base de datos destino alimentada usuarios de dispositivos móviles Android, y aunque la aplicación puede

ser distribuida a mas plataformas en esta investigación se realizó solo para una plataforma.

Con lo que se mencionó anteriormente acerca de los datos y la generación de estos por medio de la participación de los usuarios se hace razonable la razón por la que se eligió la temática de fallas en servicios públicos en esta ejemplificación, si se comprende antes que en la mayoría de ocasiones los usuarios con dispositivos móviles están más cerca de los fenómenos que ocurren en el espacio geográfico debido a que cada vez más personas cuentan con teléfonos inteligentes, *tablets*, etc.

En el momento en que se pensó acerca de las fallas en servicios públicos como temática de ejemplificación para la aplicación realizada en esta investigación se definieron como aquellos servicios que el Estado presta a su población y que deben entregarse sin errores, sin fallas, sin lentitud, sin desperdicios y sin duplicidades, mediante el uso eficiente de los recursos materiales. Además se eligió el municipio de Toluca como área de prueba, debido a la cercanía al momento de desarrollar la investigación, y la facilidad de movimiento entre sus calles y los servicios de conexión a red que se disponen en la ciudad.

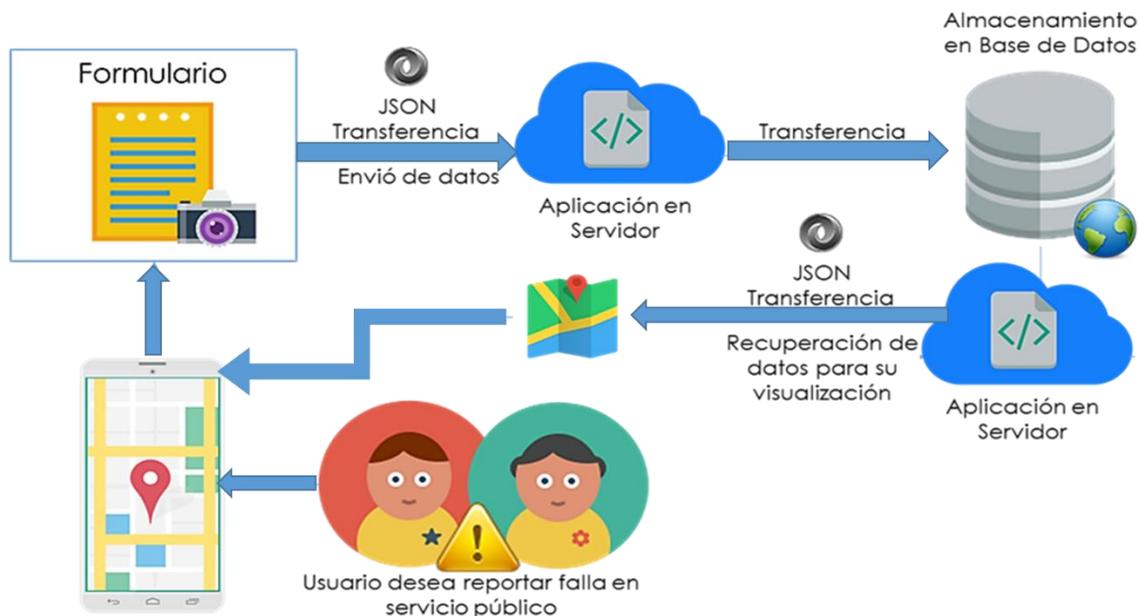
Una vez mencionado lo anterior cabe recordar el catálogo de fallas públicas que la aplicación podrá reportar y es el siguiente listado que está en basado en el servicio de reporte de fallas públicas de del sitio web del estado del Nuevo León, México:

- Aguas negras
- Alcantarillado
- Alumbrado público
- Baches
- Banquetas (destrozadas por reparación de servicios)
- Basura (falla en la recolección)
- Drenaje pluvial
- Contaminación ambiental
- Escombro
- Fuga de agua
- Fuga de gas
- Limpia (parques y vías)
- Obras públicas
- Lotes baldíos
- Arboles (plagas, por caer, poda ilegal)
- Pavimentación
- Postes tirados en la vía pública
- Regulación sanitaria
- Salud pública
- Semáforos descompuestos
- Vehículos abandonados en la vía pública

De este modo los usuarios pueden utilizar las opciones brindadas por la aplicación para reportar las fallas y es una forma de categorizar las problemáticas residuales de una avería en un servicio público en sí.

Sin embargo, dependiendo del fenómeno que se atienda con una aplicación híbrida las categorizaciones de los datos y funciones que se generan varían según las necesidades de cada proyecto, ya que existen elementos que se pueden automatizar más que otros.

Entonces el funcionamiento de la aplicación generada se planteó de manera general de la siguiente forma:



*Ilustración 7. Diagrama de funcionamiento de aplicación para reportar fallas en servicios públicos en el municipio de Toluca. Elaboración propia*

El usuario podrá reportar fallas en servicios públicos que encuentre en la calle directamente en el momento que la observe obteniendo la posición del dispositivo móvil si cuenta con una conexión a internet o posteriormente si no tiene acceso a una, ya que la aplicación es capaz de permitir desplazar el marcador de posición a otra ubicación. Dicho reporte se realizará como ya se había escrito marcando la posición de la falla y posteriormente se hace un llenado de un formulario simple el cual se envía al servidor, el servidor lo almacena y por medio de una aplicación *Back-end* regresa los datos capturados a la aplicación para observar que dicho reporte ha sido almacenado.

Se ha explicado de una forma general procedimiento que se sigue para llevar el proceso de reporte de fallas, a continuación, hablara de las herramientas específicas que hacen cada una de las tareas mencionadas en la descripción general, al ser una ejemplificación del tema se trató de simplificar cada uno de los procesos específicos para que se comprenda la connotación del objetivo en perseguido sin dejar de lado la esencia de cada pieza de la metodología.

## 2.2. Obtención y representación de posición geográfica obtenida con aplicaciones híbridas

La obtención de la posición geográfica del dispositivo móvil se consigue a partir del proceso de geolocalización por medio del GPS del *Smartphone* o *Tablet*, en general el proceso en cuanto al sensor se refiere es igual a cualquier GPS sin embargo en las aplicaciones híbridas y las web la geolocalización se maneja o controla con la API de geolocalización W3C utilizando JavaScript y HTML5 para configurar las opciones de reconocimiento de la posición en coordenadas decimales.

El uso de la posición geográfica en dispositivos móviles se ha vuelto una característica muy común que los usuarios de teléfonos inteligentes utilizan casi a diario para distintas actividades, ya sea trazar una ruta desde la posición en que se encuentra el usuario, consultar el tráfico, localizar sitios de interés cercanos, etc.

La geolocalización se ha vuelto una herramienta para los usuarios de teléfonos móviles para sus actividades cotidianas, pero más allá de esto las posibilidades que se pueden generar a raíz del uso de esta tecnología tienen un gran potencial en cuanto a la localización de eventos clave en el espacio geográfico.

EROSKI Consumer (2009) estudia y publica contenidos conforman un medio de comunicación de referencia en el ámbito del consumo y en cuanto a la geolocalización en dispositivos móviles afirman que:

*Las razones por las que se augura el éxito de los servicios geolocalizados son varias: La primera es que el teléfono móvil es el dispositivo tecnológico que más tiempo pasa con el usuario. La segunda se basa en que la tendencia en la informática es el acceso ubicuo a la Red, en cualquier lugar.*

En el caso de la "Aplicación para reportar fallas en servicios públicos en el Municipio de Toluca" la geolocalización se utilizó para ubicar aquellos problemas e servicios públicos ya mencionados anteriormente en un listado, haciendo uso del Plugin Geolocation de PhoneGap se obtuvo el permiso del dispositivo móvil para obtener la posición con la API estándar de geolocalización W3C

Lo anterior únicamente para la detección de las coordenadas decimales de la posición geográfica, posteriormente para representar los datos se configuró una aplicación JavaScript donde se integraron funciones de la API de Google Maps para representar la latitud y longitud obtenida por el GPS del dispositivo móvil, cabe resaltar que la API de Google Maps

puede ser configurada para recuperar la posición automáticamente sin tener que utilizar PhoneGap sin embargo las configuraciones que pueden hacerse con el Framework de desarrollo de aplicaciones híbridas son de gran ayuda para limitar tiempos de conexión y uso del sensor en el dispositivo móvil, y aumentar la exactitud de la posición geográfica.

Se habla tiempos de conexión por las configuraciones del sensor GPS en Android que contienen una opción para valerse de la conexión a internet para aumentar la detección de las coordenadas es decir obtener la posición deduciendo las distancias de las señales de red (IP, direcciones de RFID, WiFi y Bluetooth MAC y celulares GSM/CDMA)



*Ilustración 8. Representación de interacción entre GPS de aplicación y API de Google Maps. Elaboración propia*

En la ilustración se muestra el proceso que se desarrolló en la aplicación para obtener la posición del dispositivo y la representación en un mapa de Google con un marcador que indica las coordenadas detectadas de una falla en un servicio público esto hasta el momento solo en concepto, posteriormente se integró un formulario en el que se definen las opciones de transacción de datos con información de la falla.

## 2.2.1. Geolocalización en PhoneGap y su integración con Google Maps

PhoneGap como ya se ha mencionado antes es un framework de desarrollo de aplicaciones híbridas para dispositivos móviles, cuando se construye un nuevo proyecto PhoneGap la aplicación no es capaz de realizar tareas tales como el uso del GPS o la cámara, así que es necesario agregar características para lograr una interacción verdadera con el dispositivo a través de *Plugins* que proporcionan acceso al núcleo de Apache Cordova<sup>9</sup> APIs.

Antes de la versión 3.0 de PhoneGap todos los “*plugins*”<sup>10</sup> se agregaban automáticamente, a partir de la versión 3.0 pueden ser agregados conforme se requieran, esto para aligerar el peso y la fluidez del funcionamiento de las app.

Para agregar las funciones que se requerían en la aplicación desarrollada en este trabajo de investigación, el Plugin de geolocalización de PhoneGap sirvió para proporcionar el acceso a los datos de localización del sensor GPS para los dispositivos móviles donde se instalara la aplicación. La integración de la localización en el proyecto PhoneGap siguió una serie de pasos de configuración del marco de trabajo que es la creación del proyecto y la instalación de los Plugin para añadir características de acceso a sensores de un *Smartphone* o *Tablet*. (Ver Anexo 1 y 2).

Solo como referencia el comando de símbolo de sistema utilizado para integrar el Plugin para acceder a las funciones de localización en un proyecto para una aplicación PhoneGap es:

```
cordova plugin add org.apache.cordova.geolocation
```

Si antes el proyecto tiene instaladas plataformas de compilación simultáneamente por ejemplo Windows Phone, iOS y Android. El comando antes mencionado configura las opciones de localización para cada plataforma si es que está dentro de la compatibilidad del mismo sistema operativo móvil o si el dispositivo tiene instaladas esas características en hardware.

---

<sup>9</sup> **Apache Cordova:** PhoneGap es una distribución de Apache Cordova. Ambos sistemas tienen funciones casi idénticas, la diferencia principal entre Apache Cordova y PhoneGap es que el segundo tiene acceso a servicios de compilación en la nube proporcionados por Adobe Creative Cloud.

<sup>10</sup> **Plugin PhoneGap:** es el puente entre una aplicación web (cuyo contexto de ejecución es un WebView) y el dispositivo móvil. Se compone de dos partes: la implementación nativa (ya sea en Android, iOS, etc.) y la implementación Javascript.

En cuanto a esta investigación se refiere en el proyecto generado para la aplicación para reportar fallas en servicios públicos solo se instaló como plataforma de compilación Android y para este sistema operativo tanto para los demás las configuraciones específicas de instalación de características se hallan en el archivo "app/res/xml/config.xml", en Android se añaden las siguientes líneas cuando se instala el Plugin Geolocation:

```
<feature name="Geolocation">
    <param name="android-package"
value="org.apache.cordova.GeoBroker" />
</feature>
```

Adicionalmente cuando se instaló Android como plataforma de compilación en PhoneGap se generan también ciertas configuraciones una de estas es la generación del "AndroidManifest.xml" que es un archivo de configuración donde podemos aplicar las configuraciones básicas de una app como el nombre, versión, actividades, pero principalmente se agregan líneas en XML para añadir permisos al hardware del dispositivo, por ejemplo cuando se añade la característica de localización al proyecto de la aplicación híbrida en el archivo "app/AndroidManifest.xml" se introducen automáticamente las siguientes líneas para dar acceso a la Geolocalización en dispositivos con Android.

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

#### 2.2.1.1. Métodos de la API de geolocalización

Una de las principales novedades del HTML5 en su momento fueron las APIs JavaScript que potenciaron las funcionalidades en aplicaciones web haciendo más sencillas ciertas tareas una de estas la obtención de la posición geográfica de un usuario en un navegador.

Al ser PhoneGap un framework de desarrollo de aplicaciones híbridas que se vale en parte de HTML5 y JavaScript para hacer uso de las funciones del dispositivo entonces también al igual que en los navegadores las aplicaciones PhoneGap pueden hacer uso de las APIs de HTML5 siempre y cuando el *SmartPhone* o *Tablet* soporten este lenguaje. Entonces cuando se instala el Plugin Geolocation de PhoneGap se está otorgando

permiso a la aplicación para hacer uso de los sensores muchas veces a través de las APIs de HTML5.

El caso entonces es que PhoneGap soporta el uso de la API de geolocalización W3C que se puede utilizar también en aplicaciones web que se ejecutan desde navegadores como Chrome, Firefox, Opera, etc.

Para obtener la posición geográfica de un dispositivo móvil entonces la aplicación deberá integrar la API de geolocalización W3C a través de los métodos brindados que se requieran, en tal caso se cuenta con tres métodos para obtener la posición del dispositivo móvil. Dos de ellos que determina el uso del sensor GPS y de qué forma pueden obtener la posición si el sensor no se encuentra disponible y otro detiene la búsqueda de la localización para observar cambios en los resultados obtenidos del método *geolocation.watchPosition*. Los métodos se muestran a continuación:

Métodos de la API de geolocalización W3C	
Método	Descripción
<b>geolocation.getCurrentPosition</b>	Devuelve la posición actual del dispositivo.
<b>geolocation.watchPosition</b>	Devuelve la posición actual del dispositivo cuando se detecta un cambio de posición.
<b>geolocation.clearWatch</b>	Se utiliza para eliminar el registro de controladores de localización lugar/error previamente instalados usando <i>geolocation.watchPosition()</i>

Tabla 3. Métodos de la API de Geolocalización W3C para obtener la posición geográfica.  
Fuente: Mozilla Developer Network

Cuando se usan estos métodos regularmente existen tres parámetros o argumentos para ejecutar la obtención de la posición que se recupera en tres objetos de lectura (Position, PositionError, Coordinates). Los argumentos mencionados son:

- **geolocationSuccess:** Es la devolución de llamada que se pasa a la posición actual obtenida por el GPS o la red, es el único parámetro obligatorio.
- **geolocationError:** Es un parámetro opcional que devuelve la de llamada al sensor de geolocalización que se ejecuta si se produce un error.
- **geolocationOptions:** Es opcional también y aquí se establecen las opciones de geolocalización correspondientes a los algoritmos de obtención de coordenadas y tiempo de espera del sensor GPS.

Si se observan la APIs de Google Maps u OpenLayers se utiliza como base los métodos antes mencionados propios del estándar W3C. En la aplicación para reportar fallas en servicios públicos se utilizó únicamente

uno de los métodos `geolocation.getCurrentPosition` debido a la que las se desea obtener las coordenadas de un fenómeno fijo que no tiene un desplazamiento en una temporalidad corta.

### 2.2.1.2. Implementación de geolocalización en Google Maps

Con referencia a la temática de la aplicación de ejemplo para reportar fallas, la georreferenciación de eventos puntuales como lo es un desperfecto o avería en un servicio público es en este caso el eje de la temática espacial, es decir a partir de la obtención de la posición de un dispositivo ubicado en el lugar donde está ocurriendo el fenómeno se pueden generar datos aprovechando otras tecnologías para transferir, almacenar incluso complementar a esos mismos datos.

De manera más específica lo que se planteó lograr con la aplicación de ejemplo para reportar fallas en servicios públicos fue reflejar como los usuarios a través de la georreferenciación de fenómenos pueden generar de manera colectiva información acerca de sucesos reportados en el momento en que se detectan a través de la geolocalización y su integración con otras tecnologías que en suma sirven como una herramienta potente a la vez sencilla de desarrollar y multiplataforma por la naturaleza web de las aplicaciones híbridas.



Ilustración 9. Diagrama de obtención de localización por la aplicación móvil y su representación en el mapa. Elaboración propia

Como parte de lo que involucra la localización de fallas en servicios públicos en la aplicación que se desarrolló se planteó que a partir de la obtención de coordenadas de un reporte de una avería en un servicio se generaran datos a partir de la latitud y longitud conseguidas, en este caso la dirección del lugar donde se encuentra la falla (calle, colonia, municipio, etc.), estos datos son obtenidos de la función de geocodificación inversa de Google Maps y representados en un mapa de la misma API. Lo que se trata de reflejar en con lo mencionado anteriormente es aprovechar la localización para complementar la información obtenida utilizando librerías que brindan información por ejemplo tiempo atmosférico.

Para obtener la posición del dispositivo en este proyecto se utilizó el método *getCurrentPosition* este método de la API de geolocalización recibe los datos de geoposicionamiento del dispositivo de móvil por medio de un objeto llamado "*position*" o "*pos*" el cual contiene los atributos de localización que obtiene del dispositivo móvil y son los siguientes:

Atributos del objeto position	
Atributo	Descripción
<b>coords.latitude</b>	Obtiene la latitud en grados decimales
<b>coords.longitude</b>	Obtiene la longitud en grados decimales
<b>coords.altitude</b>	Altitud en metros
<b>coords.altitudeAccuracy</b>	Es la exactitud de la altitud obtenida en metros
<b>coords.accuracy</b>	La exactitud en metros de la posición obtenida a través de API
<b>coords.heading</b>	Dirección en la que se desplaza el dispositivo al momento de obtener la posición en grados a partir del norte.
<b>coords.speed</b>	Velocidad de desplazamiento en metros sobre segundo al momento de obtener la posición.
<b>Timestamp</b>	Fecha obtenida en timestamp del momento en que se obtuvo la posición.

Tabla 4. Atributos del objeto position.

Sin embargo los datos que puedan recuperarse dependerán del hardware que el dispositivo tenga instalado de fábrica, por ejemplo en algunos dispositivos con Android los datos de altura de la posición no pueden ser capturados debido al sensor GPS que tienen instalados, otro factor será la exactitud en parte por el hardware y también por el método de detección de la posición con apoyo de la triangulación de redes.

Para integrar la localización en la aplicación desarrollada se utilizó el método "*getCurrentPosition*" para representar la posición geográfica con la API Google Maps, existe otro método "*watchPosition*" que no fue usado debido a que este obtiene la posición y el cambio de posición si el dispositivo se mueve, y lo que se requirió en el proyecto era solo el obtener

las coordenadas de las fallas en servicios públicos para ubicar de manera puntual los registros. Sin embargo un ejemplo de en qué tipo de proyectos se puede utilizar el método “*watchPosition*” son aquellos que tienen que ver con la navegación GPS en tiempo real cuando se requiere una guía para llegar de un lugar a otro.

Un ejemplo de la base de la estructura de la codificación en JavaScript para obtener la localización por medio de la API Geolocation en PhoneGap con el método “*getCurrentPosition*” para la aplicación desarrollada se muestra a continuación.

```
//Método geolocation.geolocation.getCurrentPosition----->
function getCurrent() {

    if ( navigator.geolocation ) {
//Se establecen los datos que se obtendrán si funciona la geolocalización
        function success(position) {

document.getElementById('titulo').innerHTML="getCurrentPosition";
            document.getElementById('geolocation').innerHTML ="<br>"+
                'Latitud: '+ position.coords.latitude + '<br />' +
                'Longitud: '+ position.coords.longitude + '<br />' +
                'Precisión: '+ position.coords.accuracy + '<br />' +
                'Dirección: '+ position.coords.heading + '<br />' +
                'Velocidad: '+ position.coords.speed + '<br />' +
                'Timestamp: '+ position.timestamp + '<br />';

        }

//Se establece que pasara si la geolocalización falla
        function fail(error) {
            alert('Error en servicio Geolocalizador');
        }

//Se establecen las opciones de geolocalización
        var options = {
            maximumAge: 500000,
            enableHighAccuracy:true,
            timeout: 10000};
        navigator.geolocation.getCurrentPosition(success, fail,
options );

    }else {

//Se establece que pasara si el dispositivo no tiene
servicio geolocalizador o no responde a los argumentos success o error
        alert('Servicio Geolocalizador no responde');
    }

}
}
```

Código 1. Ejemplo del método *getCurrentPosition*

En el código anterior se muestran los parámetros y configuraciones básicas para la obtención de las coordenadas de la posición de un dispositivo móvil, la base del código anterior se ocupó para integrar, representar y obtener la geocodificación de la posición obtenida en un mapa con la API de Google Maps, lo que sirvió posteriormente para la aplicación que se generó para reportar fallas en servicios públicos.

La integración de Google Maps para generar una vista de la localización obtenida por un dispositivo móvil fue un proceso relativamente sencillo ya que solo había que insertar en un mapa la posición obtenida, los datos que se mapean son los obtenidos directamente del sensor del *Smartphone* o *Tablet* utilizando el Plugin de PhoneGap por lo que desde el principio en el código de la aplicación se desactivó la opción de geolocalización de la API de Google Maps que se puede observar en el llamado de la librería desde el HTML5 de la siguiente manera en la expresión `".exp&sensor=false"`:

```
<script
src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></s
cript>
```



Ilustración 10. Pasos generales del proceso de representación de ubicación con Google Maps y PhoneGap en la aplicación generada

En la ilustración anterior se puede apreciar el proceso de forma gráfica de representación de las coordenadas obtenidas con la aplicación utilizando PhoneGap y la API de Google Maps el cual resulta sencillo de apreciar sin embargo la codificación en JavaScript tampoco resultó tan complejo al momento de generarla, ya que solo es una adaptación del código mostrado anteriormente del método `"getCurrentPosition"` a una vista de mapa. De tal forma que la codificación base para el proyecto de aplicación para reportar fallas en servicios públicos que se desarrolló quedó de la siguiente manera:

```
// AL MOMENTO DE QUE LA APP SE INICIE SE EJECUTA LO SIGUIENTE
$( document ).on( "pageinit", "#map-page", function(e,data) {
//Se define la posición por defecto, Toluca, Estado de México
var defaultLatLng = new google.maps.LatLng(19.289168, -99.653440);
//Se definen las condiciones de localización
```

```

    if ( navigator.geolocation ) {
        function success(position) {

            // Si la geolocalización funciona se definen como
            // coordenadas que se dibujaran en el mapa
            // Las coordinas son definidas en un atributo LatLng de la
            // API Google Maps
            DibujaMapa(new google.maps.LatLng(
                position.coords.latitude,
                position.coords.longitude));
        }
        function fail(error) {

            //Si la geolocalización falla el mapa se dibuja en la
            // posición por defecto
            alert('Error en servicio Geolocalizador');
            DibujaMapa(defaultLatLng);
        }

        //Se definen opciones de geolocalización
        var options = {maximumAge: 500000,
            enableHighAccuracy:true, timeout: 10000};
        navigator.geolocation.getCurrentPosition(success, fail,
            options );

    } else {

        //Si el dispositivo no soporta la geolocalización dibuja
        // el mapa en posición por defecto
        alert('Geolocalizacion no soportada');
        DibujaMapa(defaultLatLng);
    }

    //Se define configuración de la API de Google Maps
    function DibujaMapa(latlng) {

        //Se definen opciones de mapa
        var myOptions = {
            zoom: 16,
            disableDefaultUI: true,
            center: latlng,
            mapTypeId: google.maps.MapTypeId.ROADMAP};

        //se envía un ID del objeto mapa
        var map = new google.maps.Map(document.getElementById("map-
            canvas"), myOptions);

        //Se define un infowindows con la información de las
        // coordenadas
        var infowindow = new google.maps.InfoWindow({
            position: latlng,
            content: '<p>Tu posición actual</p>'+latlng});

        //Se define un marcador que se muestra en la posición obtenida
        var marker = new google.maps.Marker({
            position: latlng,
            map: map,
            title: "Mi posición!",
            icon: "css/icons/location-pin.png",
            animation: google.maps.Animation.DROP});
    }

```

```

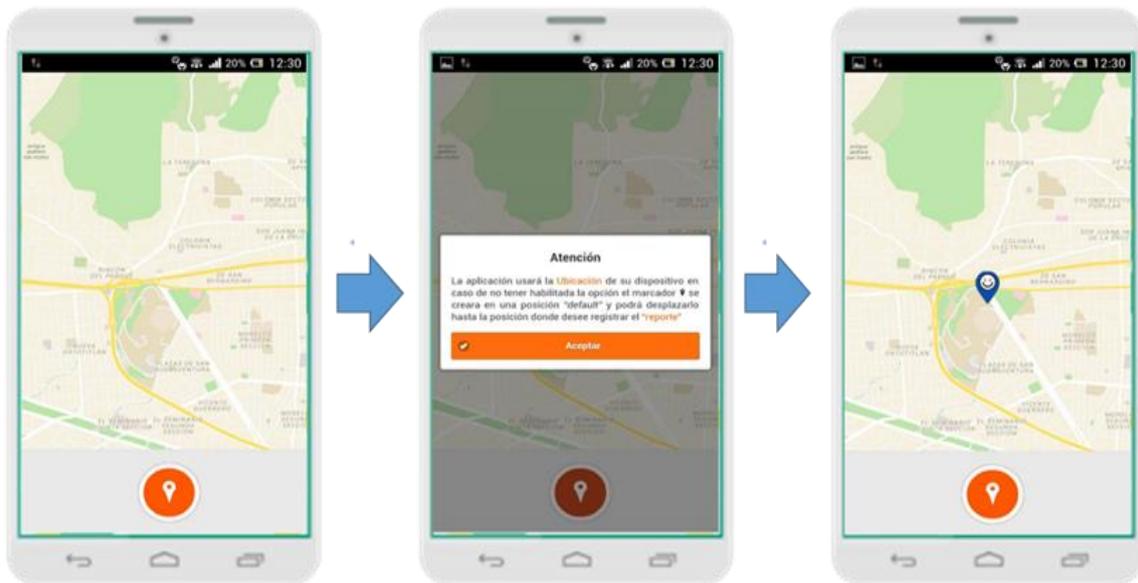
        google.maps.event.addListener(marker, 'click', function()
{infowindow.open(map,marker);});

    }//fin DibujaMapa
});

```

Código 2. Ejemplo de geolocalización con PhoneGap y la API de Google Maps

En el código anterior se integraron los servicios de la API del Google Maps para mapear los datos obtenidos de la posición de los dispositivos móviles, para realizar la integración se utilizó el método "getCurrentPosition" ya que se consideró que era el método más adecuado para obtener la latitud y longitud y mostrarlos en un mapa. Posteriormente se determinó un punto en el mapa ubicado en Municipio de Toluca, Estado de México con coordenadas decimales (Latitud y Longitud) por defecto ya que si acaso el método "getCurrentPosition" fallase, el centro del mapa se muestre un marcador en la posición por defecto. Si el método utilizado se ejecutara se determinó que el centro del mapa estaría determinado por la latitud y longitud obtenidas con la API de geolocalización y esta localización fuera un marcador en el centro del mapa. En la aplicación el proceso se ve de la siguiente manera:



Se abre la aplicación

Si se presiona el botón naranja se avisa que se utilizara la posición que el dispositivo capture

Se representa con un marcador las coordenadas obtenidas

Ilustración 11. Funcionamiento de proceso de geolocalización en aplicación para reportar fallas en servicios públicos

### 2.2.1.3. Integración de geocodificación inversa a aplicación.

Una de las ventajas de la API de Google Maps es que permite utilizar el potencial de su base de datos cartográfica para desarrollar aplicaciones personalizadas que a su vez pueden acceder a más funciones del servicio de Google Maps. Uno de estos servicios es la Geocodificación a la inversa que se utilizó en la aplicación que se desarrolló para ejemplificar esta investigación.

La geocodificación es el proceso de buscar una ubicación geográfica a partir de una dirección. La geocodificación inversa es el proceso contrario: encontrar la dirección más cercana a una ubicación geográfica es decir calle, número de portal de coordenadas conocidas en este caso las que se obtienen por medio de la aplicación híbrida.

En este caso la geocodificación inversa servirá como medio de complementación de información del reporte de una falla publica que se desee reportar con la aplicación sin embargo el trasfondo está en la complementación de información haciendo uso de distintas herramientas, en el caso particular de la aplicación desarrollada es el servicio de geocodificación de Google Maps, sin embargo si se hablara de otras aplicaciones se podrían usar otros servicios es decir de clima, trafico, servicios comerciales y otros. Todo lo anterior con el objetivo de potenciar los registros que se deseen generar, complementando la información que puede ir de solo puntos en el espacio a un concepto más completo en el que los registros ya sean puntos, líneas o polígonos que contengan información que pueda asociarse a otras bases de datos para generar análisis del espacio geográfico.

De tal modo que en la aplicación que se desarrolló se le añadió la función de geocodificación inversa para obtener datos más allá que las coordenadas en bruto a través del objeto `google.maps.Geocoder()`; de la API de Google Maps. La codificación JavaScript base que se utilizó en la aplicación para reportar fallas en servicios públicos para obtener la dirección del lugar de la falla utilizando la geocodificación inversa fue esta:

```
geocoder.geocode({'latLng': latLng}, function(resultados, status) {  
  
    if (status == google.maps.GeocoderStatus.OK) {  
        if (resultados[0]) {  
  
            //Nivel de resultados de la direccion  
            //Suele corresponder con la "dirección postal" ej.111  
            8th Avenue, New York, NY  
            var direccion=resultados[0].formatted_address;  
            /*
```

```

        ...código del marcador aquí
        */
    }
} //end IF
}); //end geocoder

```

Código 3. Codificación base para generar la geocodificación inversa

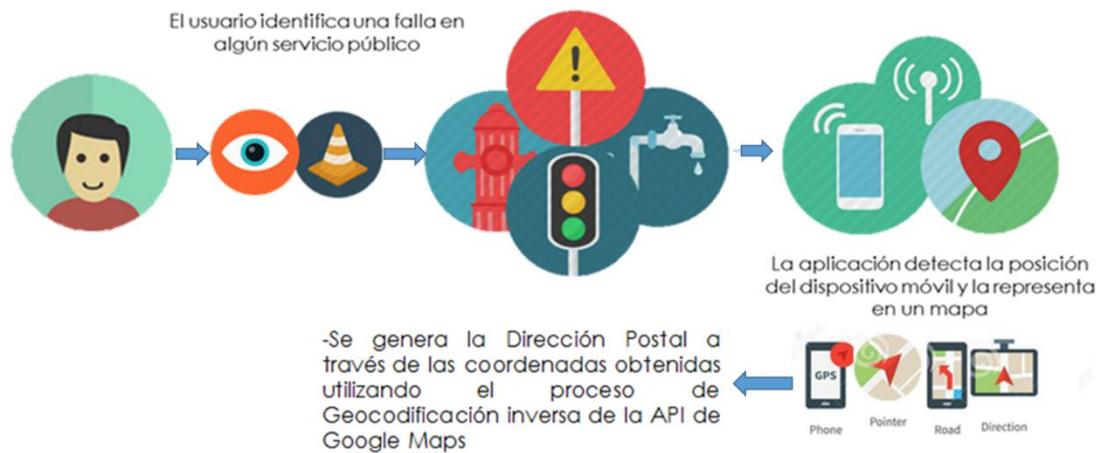


Ilustración 12. Proceso de geocodificación inversa en la aplicación desarrollada. Elaboración propia

## 2.3. Conexión a bases de datos externa en aplicaciones en PhoneGap

Las bases de datos en aplicaciones móviles funcionan de dos diferentes maneras de almacenamiento que puede ser local o remoto en un servidor.

Cuando se habla de conexiones remotas a base de datos con dispositivos móviles es necesario mencionar que en su totalidad el enlace se hace a través de una conexión a una red ya sea celular, WIFI o alguna tecnología para transferir datos, es por eso que la arquitectura de una aplicación móvil nativa o híbrida que se conecta con una base de datos remota siempre tiene contemplada una aplicación de servidor que responda a las peticiones del celular o tableta.

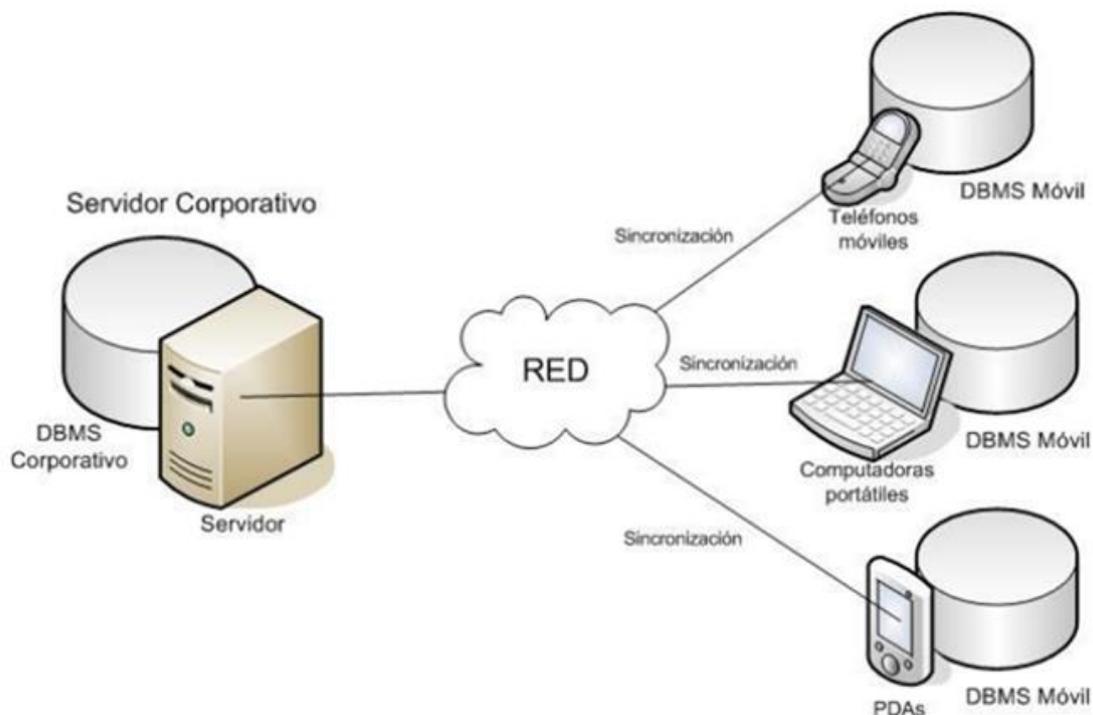


Ilustración 13. Sistema de Bases de Datos Móvil. Fuente: Evolución de las Bases de Datos: de Fijas a Móviles (Delgado Fernández & Gama Moreno, 2009)

Las aplicaciones móviles actuales requieren compartir y transferir datos entre plataformas por medio de una conexión a una red para llegar así a lograr una mayor interacción entre las aplicaciones y los usuarios, tal afirmación se ve reflejada en las redes sociales, servicios de mapas como Google Maps incluso en aplicaciones de almacenamiento en la nube. A su vez los datos que se transfieren deben estar disponibles con rapidez además de utilizar los menores recursos para mejorar la experiencia de usabilidad y funcionalidad de una aplicación móvil, ya que las

actividades que se llevan a cabo con los dispositivos móviles inteligentes son tareas que requieren respuestas rápidas, por ejemplo al momento de localizar un sitio al cual se requiere llegar y se desconoce el lugar mientras se conduce una auto por una ciudad.

Entrando a un plano más concreto en cuestión de los usuarios y las aplicaciones para dispositivos móviles lo importante es el tiempo, si se habla de una aplicación como la que se desarrolló en esta investigación, una tarea tal como reportar fallas en servicios públicos no deberá demorar mucho tiempo debe ser una acción que lleve la mayor fluidez posible, pero mucho dependerá del diseño de la aplicación, sin embargo lo importante es cuando se trabaja con transferencias de datos en un entorno multiusuario y además distribuido el tiempo de carga de cualquier proceso es importante y más si se trata de un servicio prestado en un dispositivo móvil inteligente, *“las aplicaciones y webs móviles deben ser diseñados para un número limitado de tareas, fáciles de realizar en poco tiempo y a la vez que con otras tareas. Asumir interrupciones, reducir los pasos necesarios y elementos en pantalla”* (E-interactive, 2013).

Lo mencionado anteriormente cobra relevancia cuando se trata de aplicaciones con connotación espacial como en este trabajo de investigación ya que cada vez se puede percibir que algunas de las aplicaciones móviles tienden a estar diseñadas para una mayor relación entre el espacio y las actividades del usuario, y es una tarea de la geoinformática aprovechar el hecho de que los usuario también puedan interactuar espacialmente mediante sus dispositivos móviles inteligentes no solo para identificar el entorno que los rodea porque también pueden ser partícipes de la generación de información y una herramienta que puede servir para llevar a cabo lo sugerido son las aplicaciones híbridas aprovechando o maximizando los recursos que los lenguajes de programación web como HTML5 y JavaScript brindan.

Acciones	Aplicación Móvil	Web Móvil	Indiferente	Ninguno
Consultar mapas	50%	13%	30%	6%
Redes sociales	46%	12%	26%	16%
email	41%	19%	33%	7%
Bancos	30%	15%	22%	33%
Noticias	24%	27%	40%	10%
Restaurantes, entretenimiento	18%	27%	46%	10%
Información viajes	17%	25%	39%	20%
Compras	14%	23%	40%	22%
Información sobre Productos	11%	35%	42%	13%

Tabla 4. Preferencias por acción entre aplicaciones y webs móviles. Fuente: Keynote Systems, Inc

Una forma de comprobar lo que se ha apuntado anteriormente acerca de la transferencia de datos en dispositivos móviles quizá sea el incremento del tráfico de datos móviles a nivel global lo que puede sugerir que los usuarios estaría enviando y requiriendo mayor información por medio de aplicaciones móviles en los próximos años y ciertamente es una tendencia que se toma a favor para la generación de nuevos datos.

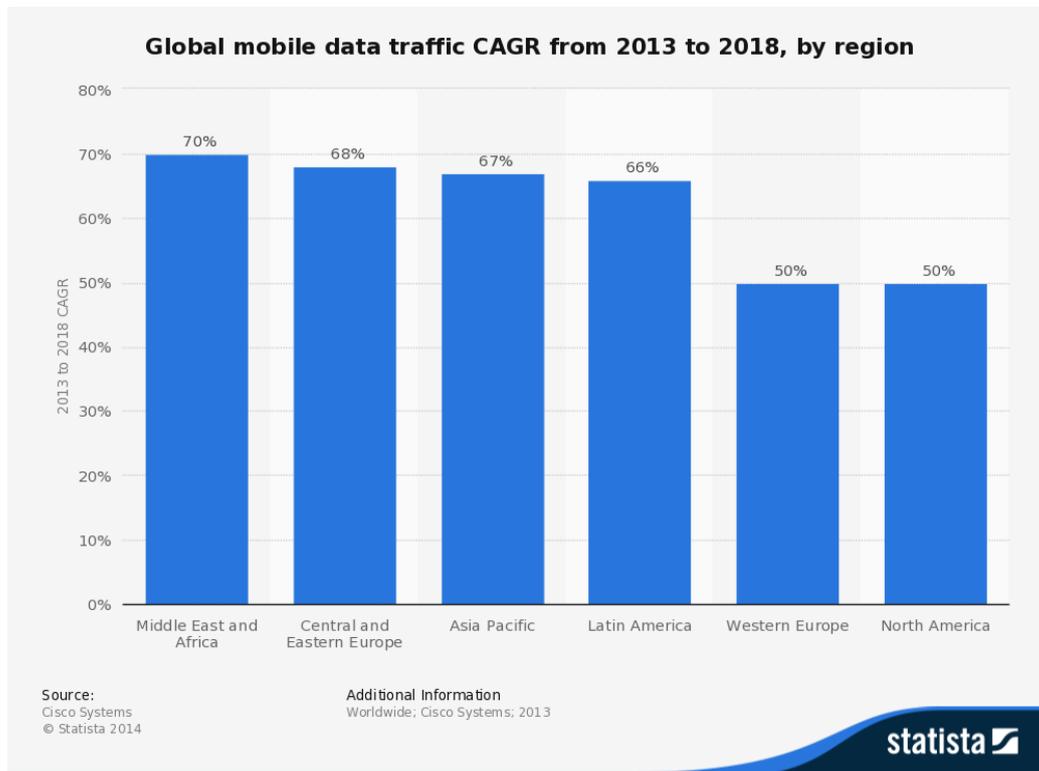


Ilustración 14. Tasa de crecimiento de tráfico de datos móviles 2013- 2018. Fuente: IDC & Statista

Por otra parte analizando la arquitectura del proyecto que se desarrolló en esta investigación este involucra una aplicación *Front-end* (aplicación móvil) que requiere una conexión a un servidor remoto a una aplicación *Back-end* (aplicación de servidor) entonces se buscó una solución que no convirtiera la conexión y la transferencia de datos desde el servidor en un problema que involucrara el consumo de recursos tanto de software, hardware y conexión a internet de las dos partes involucradas; tal solución se encontró en los Web Service REST.

Los servicios REST por su parte son una alternativa intercambiar datos gracias a que hacen uso de un medio de transferencia ligero es el lenguaje JSON por medio de peticiones HTTP (POST, DELETE, UPDATE, PUT y PASTE), aunque los servicios REST pueden utilizar XML este lenguaje en comparación con JSON hace más pesada la transferencia y por ende el consumo de datos.



web o móviles sean consumidos por otras aplicaciones con el fin de que los datos obtenidos por medio de multiusuarios puedan ser también aprovechados por los mismos o nuevos usuarios.



Ilustración 16. Usuarios de dispositivos móviles y servicios REST

Por otra parte la obtención de datos con coordenadas por los servicios web es una forma de mantenerse cerca de los usuarios y entender sus actividades en las aplicaciones porque de esta manera pueden potenciar sus estrategias de mercadotecnia y al mismo tiempo brindar un servicio de datos, un ejemplo puede ser el análisis de la interacción de social con respecto a un evento en una zona en específico con la API REST de Twitter a través de hashtags.

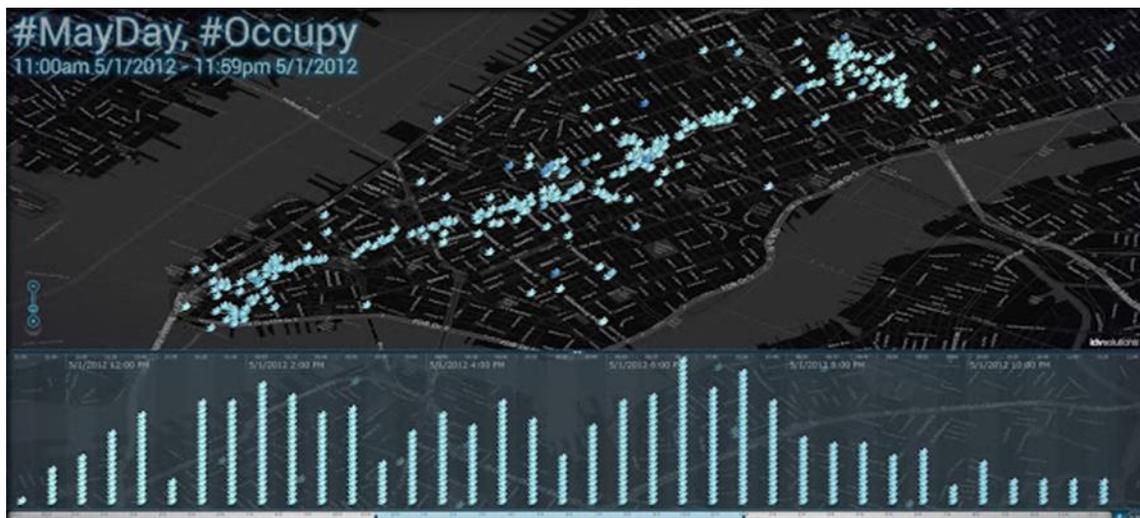


Ilustración 17. Visualización de Manhattan en primero de mayo, 2012, representando el momento y lugar de los tweets que contengan las palabras clave, MayDay y Occupy. Fuente: <http://www.socialancer.com/>

Pedraza (2010) sostuvo que “un servicio basado en localización no necesariamente es usado para localizar personas”.

Otro de los puntos a destacar también es que gracias a la localización en las aplicaciones móviles y pasando por los servicios web se pueden recopilar y cartografiar eventos o fenómenos que ocurren en el espacio geográfico transfiriéndolos a un plano virtual.

Los usuarios claramente tienen influencia directa en la conformación de las bases de datos en un servicio con esta estructura, así que tanto las aplicaciones como los servicios deben de brindar funcionalidad al mismo tiempo que interactividad y rapidez de carga.

Tomando en cuenta lo anterior en la aplicación que se desarrolló como ejemplo para esta investigación se eligió que los servicios REST eran una alternativa ligera para la transacción de datos.

### 2.3.2. Servicios REST: Transacción de datos georreferenciados con aplicaciones híbridas

Uno de los elementos más importantes en esta investigación es mostrar la forma en cómo se pueden almacenar los datos georreferenciados que se obtienen de los usuarios por medio de una aplicación móvil, además también cómo los datos que se han guardado en el servidor pueden ser consumidos por los mismos usuarios de dispositivos móviles inteligentes utilizando servicios web que a su vez pueden ser usados en otras plataformas como navegadores web o aquellas que interpreten los protocolos de transferencia en este caso HTTP ya que el tipo de arquitectura para la transferencia de datos utilizada en esta ocasión es REST.

Los servicios REST son una alternativa para llevar a cabo el intercambio de datos entre aplicaciones cliente-servidor en un entorno móvil, una razón importante para utilizar esta arquitectura en este trabajo de investigación es que las transacciones de datos son ligeras ya que se utiliza JSON un lenguaje con el que se puede trabajar sin requerir metadatos esto en comparación con XML, de esta forma se gasta menos ancho de banda si se considera que el internet en los Smartphone se paga por Megabytes consumidos. Hacer que las conexiones sean ligeras tanto para capturar o consumir datos de los reportes en fallas servicios públicos para la aplicación desarrollada como ejemplo, además se refleja el concepto que se trató de seguir, haciendo que el tiempo de respuesta en las transacciones no afectara el rendimiento de la aplicación ni el tiempo de respuesta.

Una vez que con aplicación para reportar fallas en servicios públicos "FaSeP" se detecte y desee hacer un reporte estarán en contacto dos

aplicaciones una el dispositivo móvil que hará las llamadas HTTP a la aplicación *Back-end* y otra que responderá esas solicitudes en desde el servidor.



Ilustración 18. Conexión de aplicación móvil a servicio REST.

Los patrones de solicitud de respuesta en una arquitectura REST son los mismos utilizados por los navegadores web es decir aplican el protocolo HTTP. Sin embargo los servicios REST no se limitan solo a las peticiones GET y POST, también se puede implementar acciones concretas con métodos como PUT o DELETE. Y a diferencia de la estructura de un navegador las respuestas no serán HTML u otros recursos binarios, más bien en un formato ligero que puede ser JSON o XML. También se debe mencionar antes de continuar que no todos los servidores, servicios web, lenguajes o clientes soportan en todos los métodos HTTP.

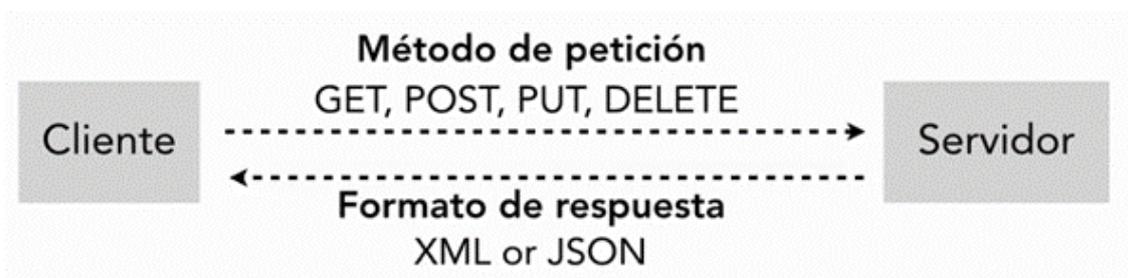


Ilustración 19. Peticiones y respuestas REST. Fuente: Video2brain.com

Método	Significado
GET	Leer datos
POST	Insertar datos
PUT o PATCH	Actualizar datos, o insertar con un nuevo ID
DELETE	Eliminar datos

Tabla 5. Métodos HTTP y su significado. Fuente: Video2brain.com

Las aplicaciones híbridas por su parte se pueden considerar en un contexto afín a los servicios web derivado de su núcleo web y

principalmente por el lenguaje JavaScript. El lenguaje mencionado tiene tal importancia en las aplicaciones web actuales que en los últimos tiempos se han generado librerías para facilitar el desarrollo procesos entre los que están la transacción de datos utilizando AJAX y precisamente gran parte de la importancia que tiene JavaScript se la debe al surgimiento de AJAX. La influencia es tal que algunos autores consideran que “cuando todos pensaban que JavaScript era un lenguaje casi muerto, apareció AJAX y el lenguaje tomo un nuevo vuelo impensado hace unos años” (Firtman, 2007).

La importancia en cuanto a los servicios REST haciendo referencia a las aplicaciones que utilizan mapas con los que se puede entender el espacio está en que los datos sean capas de información que sean posibles de representar incluso de analizar. Con lo que respecta a las aplicaciones móviles híbridas cuando los datos son obtenidos y representados en un mapa se están especializando eventos o fenómenos que ocurren en un lugar o varios, las capas de información generadas son una herramienta para comprender o conocer el espacio geográfico utilizando capas de información que la mayoría de las veces no tiene complejidad mayor a la geolocalización de sitios de interés y referencia espacial.



Ilustración 20. Capas de información en un mapa. Fuente: Dreamstime.com

Al utilizar un servicio REST para recuperar capas de información es importante decir que soporta diferentes formatos de datos, la posibilidad de utilizar XML, JSON o variantes es una ventaja, principalmente cuando se usan adaptaciones de JSON para datos espaciales como GeoJSON o TopoJSON ya que en cuanto a dispositivos móviles se refiere normalmente es más rápido, ligero y menos costoso para un cliente consumir datos de esta manera; incluso Google Maps y Open Layers soportan formatos como GeoJSON para añadir capas de información.

### 2.3.2.1. Construcción de servicio REST

Algunos de los objetivos primarios que se establecen para un servicio REST tienen que ver con los usuarios y es porque siempre se espera que un sistema así reciba y envíe solicitudes de transferencias de datos, se puede comprender así que los servicios REST son populares en aplicaciones en dispositivos móviles inteligentes ya que las bases de datos son llenadas por gran cantidad de usuarios y que a su vez pueden requerir los datos almacenados. Es el caso de la aplicación para reportar fallas en servicios públicos que se desarrolló como ejemplificación de la investigación en este trabajo, ya que si bien la aplicación al ser probada no involucra gran cantidad de conexiones al mismo tiempo, pero la idea es que al generar este tipo de aplicaciones los usuarios colaboren con mayor frecuencia, un ejemplo como ya se ha mencionado puede observarse en aplicaciones como Foursquare © o Waze ©.

Para generar las respuestas a solicitudes de transferencia de datos en la aplicación híbrida para reportar fallas en servicios públicos se desarrolló en un servidor con PHP y MySQL un servicio REST capaz de enviar y recibir solicitudes específicamente GET y POST. Para reducir la complejidad de la aplicación *back-end* se utilizó “Slim” un framework para PHP que permite generar aplicaciones web y APIs con muy poco código.

Se hizo una primera prueba donde se recuperaría un “Hola Mundo” al ingresar a una URI<sup>11</sup> desde un navegador, el objetivo fue hacer probar la solicitud HTTP GET, en el código se puede observar que se define el método y se recupera el contenido de la función “Hola Mundo” cuando se accede a una dirección “/hola” en este caso “http://direccion/hola”

```
<?php
require 'Slim/Slim.php'; //Instalación de Slim PHP
\Slim\Slim::registerAutoloader();
$app = new \Slim\Slim();
//Primera URL
$app->get('/hola', 'HolaMundo');//URL 1 GET "Hola Mundo"
$app->run();

function HolaMundo() {echo "Hola Mundo";}
?>
```

Código 4. Instalación Slim Framework y ejemplo Hola Mundo. Fuente: Elaboración propia

En ejecución el ejemplo del código anterior se puede observar de la siguiente manera (Ver p. 57):

---

<sup>11</sup> **URI**: Uniform Resource Identifier— es una cadena de caracteres que identifica los recursos de una red de forma unívoca

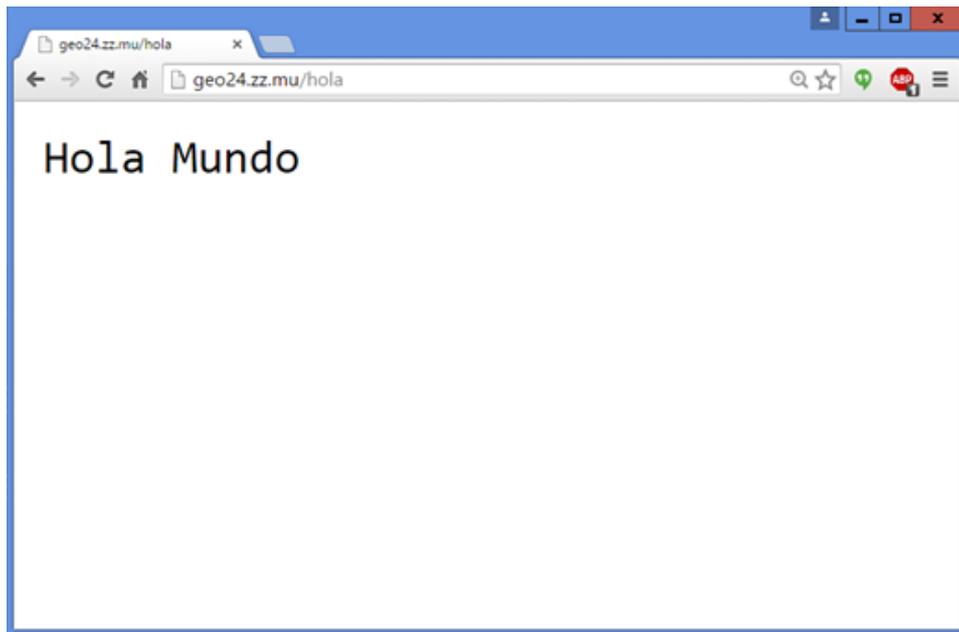


Ilustración 21. Ejemplo Hola Mundo servicio REST con Slim PHP.

Posterior a la primera prueba se desarrolló el código para conectar el servicio con la base de datos, el cual no comprendió una gran dificultad, este proceso se puede observar en el siguiente código donde se estableció la conexión a la base de datos "reporte" y la tabla "fasep" en MySQL.

```
function getConnection() {  
    //Datos de conexión  
    $dbhost="localhost";  
    $dbuser="usuario";  
    $dbpass="contraseña";  
    //Base de Datos  
    $dbname="reporte";  
    $dbh = new PDO("mysql:host=$dbhost;dbname=$dbname;charset=utf8",  
$dbuser, $dbpass);  
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    $dbh->exec("set names utf8");  
  
    return $dbh;  
}
```

Código 5. Conexión a base de datos de aplicación para reportar fallas en servicios públicos.  
Fuente: Elaboración propia

Con estos dos partes de código ilustrados anteriormente se tiene la base de un servicio que responde a una solicitud GET, aunque el resultado es solo un "Hola Mundo" se ejemplifico de manera básica cómo funcionan las solicitudes, por otra parte se puede apreciar que la cantidad de código es muy poca, en algunos caso se pueden generar servicios REST hasta con treinta líneas de código si se utilizan frameworks que aligeren la carga de codificación.

### 2.3.2.2. Método POST

Este método HTTP corresponde a uno de los verbos disponibles para usar en una arquitectura REST y es utilizado para crear un recurso nuevo, en la aplicación para reportar fallas en servicios públicos el método POST es con el cual se creara un reporte por medio de la aplicación móvil que se conectara con la aplicación *back-end* para enviar los datos de los datos resultantes del proceso de geolocalización de un desperfecto en un servicio público.

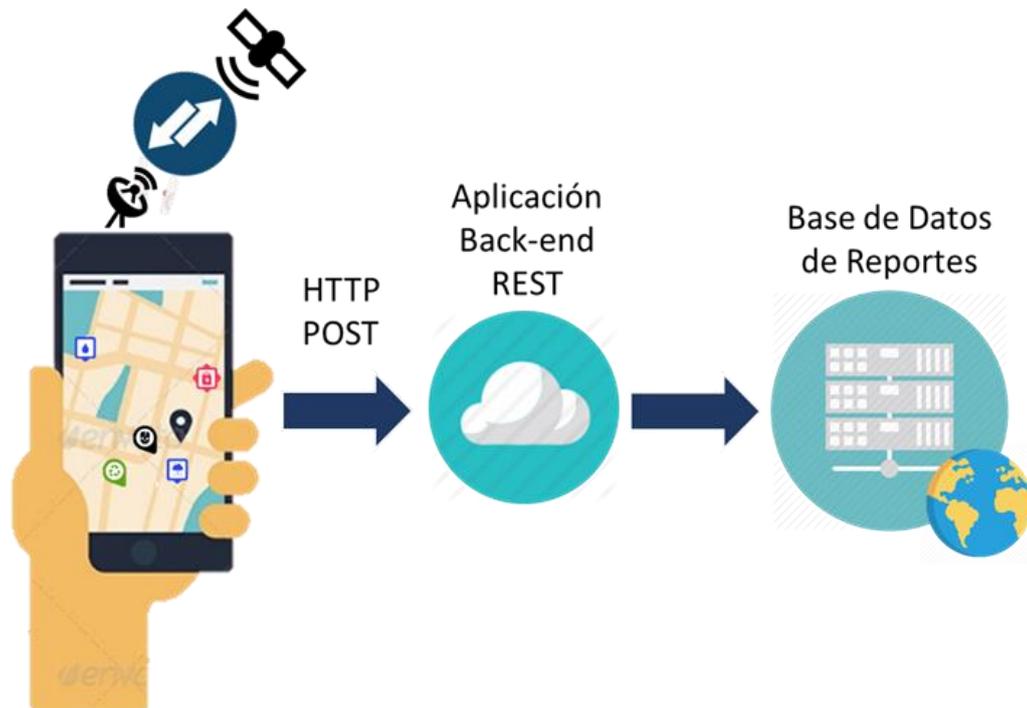


Ilustración 22. Método POST. Fuente: Elaboración propia

Para obtener los datos de una falla en un servicio público se desarrolló un formulario para así no solo obtener los datos de geolocalización del punto, también se solicitó el tipo de falla conforme a un catálogo de opciones donde se pueden elegir una, también se genera en un campo la dirección postal de la falla que se obtiene a través del proceso de geocodificación inversa gracias a las coordenadas obtenidas por el GPS del dispositivo, se ingresaron dos opciones una para tomar una foto o elegir una de la galería del y optativamente se colocaron dos campos el de “nombre” e “e-mail”, para tener información adicional. Una vez que el formulario es llenado se presiona el botón “Enviar Reporte” el cual por medio de una conexión XMLHttpRequest AJAX con jQuery envía los datos con método POST al servicio REST en formato JSON para almacenar el registro en una base de datos.

La vista del formulario de la aplicación de ejemplo “FaSeP” se muestra a continuación (Ilustración 23).



Ilustración 23. Formulario de aplicación FaSeP. Fuente: Elaboración propia

El botón enviar está configurado con una función que al dar clic ejecuta la interfaz de comunicación que es XMLHttpRequest<sup>12</sup> simplificada con la librería de JavaScript jQuery, al ejecutar esta interfaz se logra la comunicación con el servidor para enviar o recibir una petición en este caso se manda el método POST con un objeto que contiene las propiedades de la falla en un servicio público en formato JSON

```
$.ajax({
  type: "POST", //Método HTTP
  //URI en Servicio REST que recibirá solicitud
  url: "http://Host/fasepFull",
  //Generación de objeto JSON
  data: ({
    Falla: fallas,
    Lugar: lugares,
    Punto: puntos,
    LatY: laty,
    LongX: longX,
    Nombre: nombres,
    Email: emails,
    Foto: foto
  }),
  cache: false,
  dataType: "text",
  success: onSuccess
});
```

Código 6. Envío de datos a través de método POST con JavaScript y AJAX.

<sup>12</sup> **XMLHttpRequest (XHR):** es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML y codificaciones particulares específicas.

De lado del servidor, la aplicación *back-end* recibe el objeto JSON que fue enviado desde la aplicación móvil a la URI <http://Host/fasepFull> donde se interpreta el contenido de cada atributo para insertarlo por medio de variables en una consulta SQL con la cual se crean los registros en la tabla “*fallas*” de la base de datos “*reporte*” en MySQL cabe resaltar que las coordenadas se guardan en campos separados de Latitud y Longitud pero también se generan geometría en formato punto.

```
<?php

//Habilitar el acceso de las aplicaciones externas ej. App móviles
header("Access-Control-Allow-Origin: *");
require 'Slim/Slim.php';
\Slim\Slim::registerAutoloader();
$app = new \Slim\Slim();

$app->post('/fasepFull', 'addFS');//URL 4 POST

$app->run();

/*Se respontde a la solicitud POST para añadir un registro nuevo*/
function addFS() {

    //incluimos el archivo php para generar el folio del reporte
    include '/php/GenerarFolio.php';
    //Conexion a mysql
    $db = getConnection();

    $Folio = date('Y'. 'm'. 'd'. $clave7);
    $Falla = $_POST['Falla'];
    $Lugar = $_POST['Lugar'];
    $Punto = $_POST['Punto'];
    $LatY = $_POST['LatY'];
    $LongX = $_POST['LongX'];
    $Nombre = $_POST['Nombre'];
    $Email = $_POST['Email'];
    $Foto = $_POST['Foto'];
    $Reporte = "1" ;

    //generación de consulta SQL
    $consulta="
    INSERT INTO `fasep`
    (`Folio`,`Falla`,`Lugar`,`Punto`,`LatY`,`LongX`,`Nombre`,`Email`,`Foto`
    ,`Reportado`) VALUES
    ('$Folio','$Falla','$Lugar',GeomFromText('POINT(".$Punto.")',0),'$LatY'
    , '$LongX','$Nombre','$Email','$Foto','$Reporte)";

    //Ejecución de consulta
    $db->query($consulta);

    //Cerrar conexión
    $db = null;
}

function getConnection() {Código... }
?>
```

Código 7. Servicio REST, método POST para recibir objeto JSON con PHP

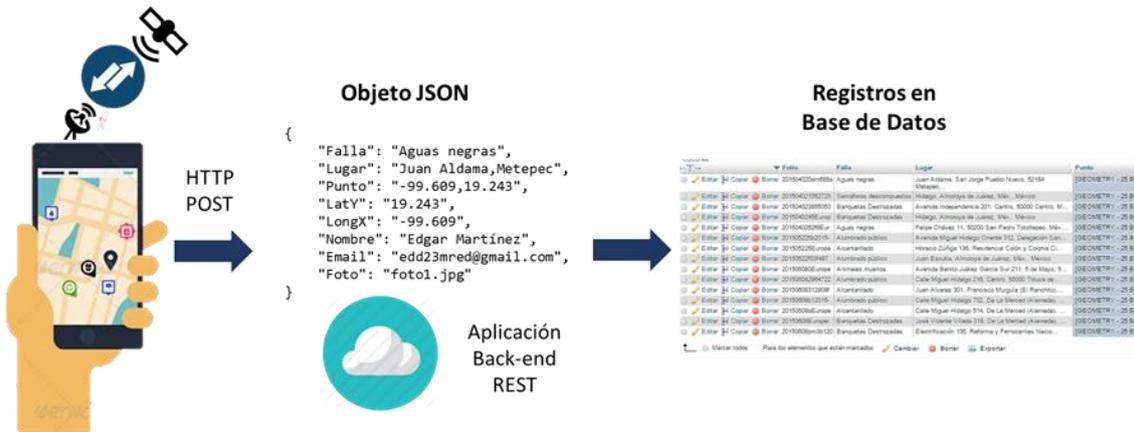


Ilustración 24. Método POST envió de objeto JSON a base de datos. FaSeP

### 2.3.2.3. Método GET

Es el método con el cual se manda la solicitud para consumir los datos, es decir para que los distintos dispositivos puedan leer los datos en formato XML o JSON que están almacenados en el servidor, en las aplicaciones híbridas en dispositivos móviles tanto las nativas al consumo de datos de servicios es una de las características que más se usan tan solo basta con mencionar que apps como Facebook, Twitter, Foursquare, Waze, etc., que hacen uso de servicios que basados en REST en algunas de sus funciones.

El consumo de datos en una aplicación móviles es una característica resultado de un almacén de datos, en apps donde los usuarios son los encargados de la alimentación de base de datos la visualización de los registros realizados es una herramienta importante para conocer el alguna forma espacio geográfico, un ejemplo es cuando un usuario de Foursquare añade un nuevo lugar, otros usuarios han visitados ese mismo lugar, y ellos mismos puntúan el sitio, escriben opiniones que puede observar otras personas en distintas plataformas ya sea un teléfono, una Tablet o un navegador en una computado personal.

Todo lo mencionado anteriormente involucra un trabajo colaborativo donde están implícitos muchas veces los términos ya mencionados como los métodos HTTP POST y GET, los lenguajes de transferencia, los servicios REST o SOAP, aplicaciones *front-end*, *back-end*, las bases de datos y principalmente los usuarios. Se puede mencionar también que poder usar la geolocalización en un dispositivo móvil ha potenciado a las aplicaciones móviles que utilizan la georreferenciación, sin embargo el éxito se trata de una sinergia entre distintos sistemas en diferentes

plataformas, avances tecnológicos e incluso se podría decir que todo lo anterior llevado a un mejor plano gracias a los usuarios.



Ilustración 25. El usuario manda. Fuente: ESRI España

Ya se mencionó un ejemplo con una aplicación conocida como Foursquare, sin embargo la aplicación desarrollada para ejemplificar esta investigación contiene las bases principales para darle importancia al usuario, es decir los reportes los hace un usuario de un dispositivo móvil Android donde este instalada la aplicación, y al mismo tiempo lo puede consumir el mismo Smartphone u otro dispositivo. En esta investigación el tema se ejemplificó con fallas en servicios públicos, sin embargo el panorama es amplio ya que los eventos o fenómenos en el espacio geográfico que pueden georreferenciarse en el momento con este tipo de aplicaciones son diversos ya sea que se trate de una avería de un automóvil, pedir comida a domicilio, identificación de postes, dibujo de áreas de parques, entre otros. En la actualidad se trata que a partir de la geolocalización se promuevan actividades en su mayoría tienen que ver con mercadotecnia, pero existen y pueden haber aplicaciones que utilicen el espacio geográfico para facilitar actividades diarias o incluso laborales a los usuarios.

Por las razones mencionadas anteriormente es importante que los datos lleguen rápido a los usuarios. En el momento de consumir datos provenientes de un servicio REST con método GET se debe entender que la ligereza de los datos se debe al estado representacional de los objetos, es decir la forma en la que viajan los datos.

Con la aplicación FaSeP se recuperaron los datos de los reportes realizados a través de una conexión a con AJAX al servidor utilizando el método GET como medio de solicitud y respuesta, de esta forma las fallas en servicios públicos pueden ser observados después de ser almacenados con el método POST del servicio REST.

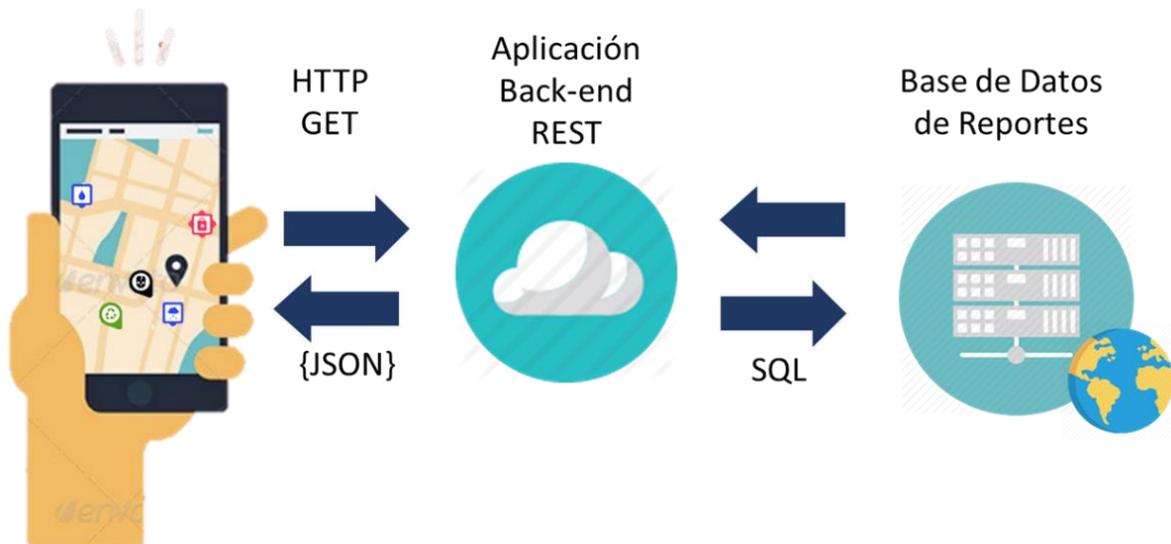


Ilustración 26. Método GET. Fuente: Elaboración propia

El envío de datos desde la aplicación back-end REST puede ser diverso, ya sea por el formato de transferencia o por los datos que se deseen enviar, es decir pueden enviarse datos específicos según la aplicación cliente lo requiera, o pueden configurarse resultados que se han establecido desde la planeación del servicio. Ya sea una u otra la alternativa, todo tiene que ver con los requerimientos de las aplicaciones.

La aplicación híbrida FaSeP fue desarrollada para visualizar las fallas en el mapa y el tipo de fallas que son, esto porque solo se estableció ejemplificar el método GET, sin embargo más adelante se mostrara como los datos se usaron en otras plataformas en distintas vistas considerando que los servicios REST deben poder adaptarse a otros modelos de negocios.

La aplicación de que envía los datos solicitados desde el dispositivo móvil con la app FaSeP es ligera no requiere más que conformar los objetos JSON con la base de datos para poder utilizarlos como marcadores múltiples en el mapa de la aplicación para reportar fallas en servicios públicos.

```
<?php
//Habilitar el acceso de las aplicaciones externas ej. App móviles
header("Access-Control-Allow-Origin: *");
require 'Slim/Slim.php';
```

```

\Slim\Slim::registerAutoloader();
$app = new \Slim\Slim();

$app->get('/fasepMob', 'getFSPm');//URL 4 POST
$app->post('/fasepFull', 'addFS');//URL 4 POST

$app->run();

function getFSPm() {

    $sql = "SELECT Falla,LatY,LongX FROM fasep";
    try {
        $db = getConnection();
        $reportes = $db->query($sql)->fetchAll(PDO::FETCH_ASSOC);

        $return = array();
        foreach ($reportes as $row){
            $return[]=array(
                'Falla'=>$row['Falla'], 'LatY'=>$row['LatY'],
                'LongX'=>$row['LongX']);
        }
        print unicode_decode('{"markers": '. json_encode($return) .
    '}');
        $db = null;

    } catch(PDOException $e) {
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
}

function addFS() { Código...}
function getConnection() {Código... }

?>

```

Código 8. Servicio REST, método GET para enviar una respuesta en formato JSON con PHP

En esta solicitud GET a la URI <http://Host/fasep/fasepMob> el dispositivo móvil recibirá solo los datos del tipo de falla y las coordenadas para representar cada punto en el mapa ya que el estado representacional de la petición contempla pedir de la base de datos los campos Falla, LatY, LongX, por lo que la aplicación REST hace una consulta SQL que solo contempla solicitar los datos que han sido requeridos por la aplicación híbrida.

```
$sql = "SELECT Falla,LatY,LongX FROM fasep";
```

Posterior a hacer la consulta el servidor envía los datos a la aplicación *back-end* donde se codifica la respuesta en formato JSON para que puedan ser consumidos por la aplicación para reportar fallas en servicios públicos utilizando JavaScript para generar los marcadores en el mapa con las coordenadas regresadas de los registros almacenados.

```

{"markers":
  [

```

```

    {
      "Falla": "TipodeFalla 1",
      "LatY": "Latitud 1",
      "LongX": "Longitud 1"
    },
    {
      "Falla": "TipodeFalla 2",
      "LatY": "Latitud 2",
      "LongX": "Longitud 2"
    }
  ]
}

```

Código 9. Estructura de objeto JSON generado para aplicación FaSeP

La estructura del resultado JSON arrojado en la petición GET es un conjunto de objetos en una categoría cadena “*markers*” que encapsula a los distintos reportes que se visualizan en un mapa en la aplicación FaSeP, los objetos contienen atributos de que representan el tipo y las coordenadas de las fallas en servicios públicos que han sido reportadas virtualmente con la aplicación móvil.

El resultado al acceder <http://Host/fasepMob> de manera estándar arroja una respuesta GET, en caso de que se accediera con un navegador web al cargar la página mostraría al objeto JSON que a su vez puede utilizarse para otras aplicaciones podría ser visualización en una tabla, generación de una lista de fallas, etc.

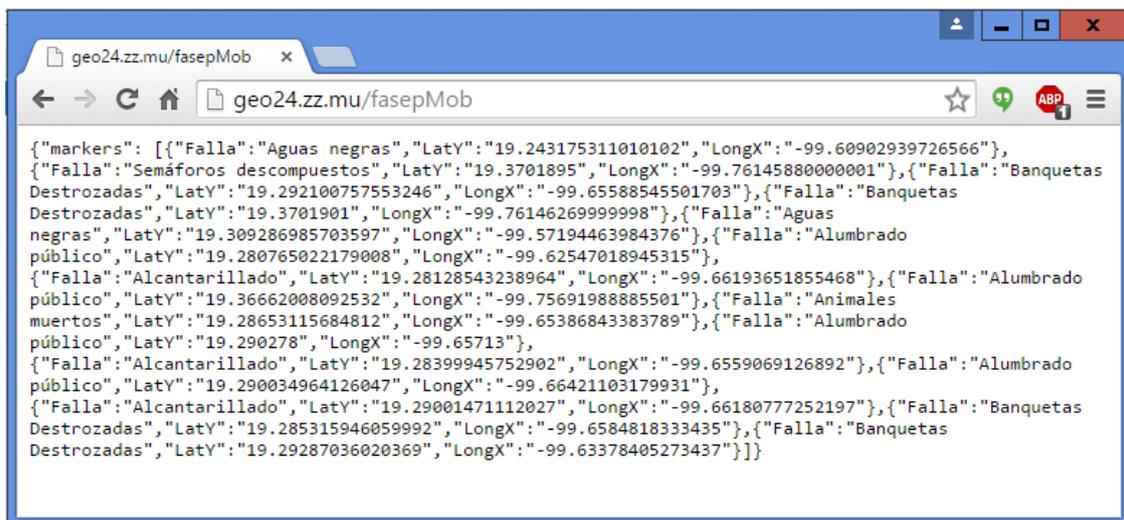


Ilustración 27 Resultados JSON del servicio REST para aplicación FaSeP

Una vez que el servicio que devuelve los datos estuvo listo, solo basto con consumir los datos del servidor con una solicitud GET XMLHttpRequest, en las aplicaciones híbridas es bastante sencillo hacer una solicitud de este tipo si se utilizan librerías como jQuery el código es bastante sencillo ya que en una aplicación con núcleo web como la que se desarrolló en esta investigación para reportar fallas en servicios públicos, la solicitud para

establecer contacto con el servidor para hacer una petición GET bastó con añadir la línea que se muestra enseguida.

```
$.getJSON('http://host/fasepMob', function(data) { código... })
```

De esta forma solo se ha logrado establecer la conexión con una línea de código gracias jQuery para hacer una solicitud GET únicamente, para transformar los datos recibidos en marcadores en el mapa es necesario decodificar el resultados JSON.

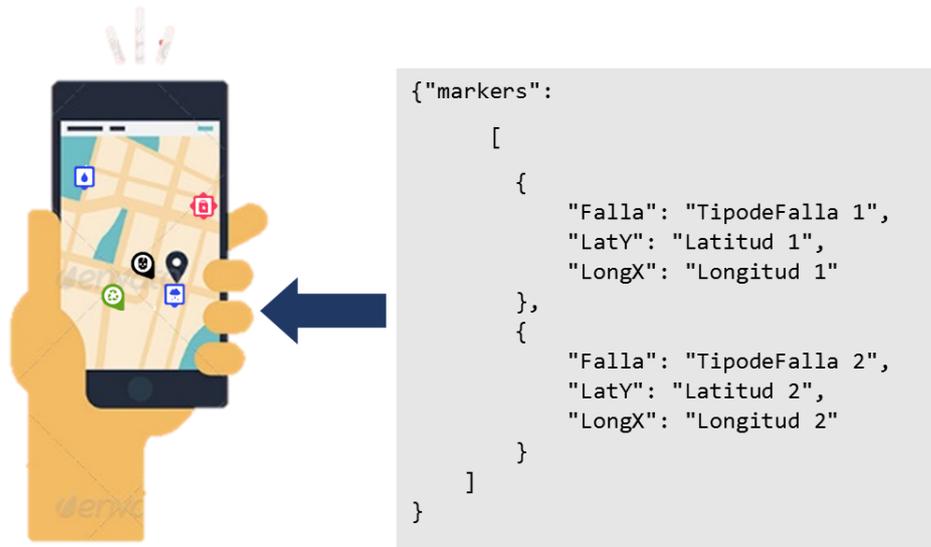


Ilustración 28. Decodificación de resultado JSON a marcadores en el mapa

Para poder convertir los datos en marcadores basta con identificar el objeto "markers" para poder utilizar los multi-objetos que contiene con datos de coordenadas, entonces en el momento en que se creó un nuevo objeto Marker con la API de Google más en JavaScript, se generaron puntos en el mapa para cada posición recuperada de los datos en el objeto LatLng, gracias a que con la función "each" se identificó cada uno de los elementos y datos del objeto "markers" ya que la función mencionada hace un proceso de recorrido parecido al de un "while", de esta manera cuando se solicitó un nuevo marcador la aplicación identificó cada registro de falla en la respuesta JSON y la convirtió a un marcador en el mapa.

```
$.getJSON('http://Host/fasepMob', function(data) {  
  $.each(obj.markers, function(key, data) {  
    var posit = new google.maps.LatLng(data.LatY, data.LongX);  
    var markerR = new google.maps.Marker({  
      position: posit,  
      map: map,  
      icon: 'css/img/accident.png',  
      title: data.Falla  
    });  
    var details = data.Falla;  
    bindInfoWindow(markerR, map, r, details);  
  });  
});
```

```
    });  
  }  
}
```

Código 10. Transformación de objeto JSON a marcadores en Google Maps con JavaScript

Por lo tanto el resultado en la aplicación híbrida se refleja en la siguiente captura de pantalla donde se pueden apreciar los distintos registros realizados transformados en marcadores en el mapa de Google.



Ilustración 29. Transformación de objeto JSON a marcadores en aplicación FaSeP

### 2.3.3. Visualización de resultado en otras interfaces

Recapitulando el tema de la transferencia de datos habrá que mencionar que en algunos procesos la conexión a una red con internet deberá ser más veloz que en otros casos. Un ejemplo que puede mencionarse se observa en el envío de reportes en la aplicación FaSeP, ya que en el formulario se requiere seleccionar o tomar una fotografía, esta característica suele aumentar el peso de la transferencia y por lo tanto el gasto de ancho de banda, el punto anterior es importante ya que al ser usuario de un teléfono inteligente es trascendental tener en cuenta el costo del uso de las aplicaciones que se conectan a internet. Si se habla de la aplicación desarrollada para ejemplificar esta investigación se puede decir que para aligerar la carga de las transferencias se optó por la conversión de la calidad de la fotografía seleccionada o tomada a una menor, sin embargo aunque se redujo el ancho de banda utilizado para generar los reportes aún se gastó para almacenar uno entre 550 kilobytes y 1 megabyte, por otro lado el costo

del gasto de internet tendrá que ver mucho con las tarifas del servicio de telefonía contratado por el usuario de un Smartphone.

Dependiendo entonces del tipo de dispositivo, el consumo de datos del servicio REST de fallas en servicios públicos creado para la aplicación FaSeP será diferente, es decir si busca aminorar el peso de la transacción para un dispositivo móvil el servicio web que provee los datos genera resultados JSON adecuados para ser consumidos en por un teléfono inteligente o Tablet y así de igual manera genera resultados más completos para otras plataformas.

Al intentar complementar el proceso de generar datos a través de la geolocalización con dispositivos móviles se quiso cerrar la brecha en el ciclo que va desde la captura de la posición hasta el consumo de los datos almacenados. Por esta razón cuando se generó la aplicación para reportar fallas en servicios públicos hubo que pensar en que hace una herramienta así, es decir que la tarea de recopilación de información espacial con dispositivos móviles para ser almacenado no basta solo el proceso de la obtención de coordenadas sí que va de que los datos se transferirían también al servidor y que posteriormente esos datos puedan ser consumidos en distintas plataformas. Entonces se desarrolló un servicio REST que captura los datos provenientes de la aplicación y que además arroja resultados JSON para ser consumidos.

El consumo de datos para la aplicación FaSeP consta de interpretar los objetos JSON de reportes provenientes del servicio web, la lectura de los datos se realizó por medio de peticiones AJAX debido a que la aplicación como ya se sabe fue desarrollada con PhoneGap y por ello el código que conforma el proyecto radica en HTML5, CSS3 y JavaScript.

FaSeP se trata de una aplicación híbrida para Android con núcleo web, pero los resultados generados por los reportes en fallas públicos pueden visualizarse también en otras interfaces web, ya sea objetos HTML o Mapas, aun así la importancia del servicio REST no radica en las interfaces que pueden consumir los resultados si no en el formato de datos que se genera en las respuestas.

Los datos que el servicio REST de reportes de fallas en servicios públicos genera datos en formato JSON y una variante GeoJSON<sup>13</sup>, la diferencia radica en que en primero las coordenadas capturadas se dividen en dos objetos (Latitud y longitud) y en el segundo existe un objeto que define el tipo de geometría y otro que plasma las coordenadas.

---

<sup>13</sup> **GeoJSON:** Es un formato de codificación de diversas estructuras de datos geográficos.

Un objeto GeoJSON puede representar una geometría, o una colección de geometrías, GeoJSON soporta los siguientes tipos de geometrías: puntos, líneas, polígonos, multi-puntos, multi-líneas, multi-polígonos y colecciones de geometrías.

“Una estructura GeoJSON contiene un objeto geometría y otras propiedades adicionales y un colección representa una lista de estructura” (Butler, et al., 2008)

Lo mencionado anteriormente puede ejemplificarse gráficamente en el código mostrado en la siguiente tabla donde se representa un objeto geometría de un Punto y otro de una Línea:

Punto	Línea
<pre>{   "type": "Point",   "coordinates": [     -80.66252,     35.04267   ] }</pre>	<pre>{   "type": "LineString",   "coordinates": [     [       -80.66198322805866,       35.04296808121376     ],     [       -80.66207649424241,       35.04274941454224     ],     [       -80.66219679439743,       35.04262648135723     ]   ] }</pre>

Tabla 6. Ejemplo de objetos geometry GeoJSON

Ahora que se dio a conocer los dos tipos de formatos que pueden ser consumidos se hablara del primero, que son los resultados en formato JSON del servicio que al momento de realizar investigación se alojaron en la siguiente URI: <http://geo24.zz.mu/fasepFull> que genera objetos JSON con más características de la base de datos que el que se genera para dispositivos móviles.

```
{
  "markers": [
    {
      "Folio": "201506182f7137a",
      "Falla": "Recolección Basura",
      "LatY": "19.295280408251518",
      "LongX": "-99.65891098678588",
      "Lugar": "Xochitl 119, De El Coporo, 50050 Toluca de Lerdo, Méx., México",
      "Foto": "http://geo24.zz.mu/Upimages/72080.jpg",
      "Reportado": "1"
    }
  ]
}
```

```

        "Folio": "20150618am71891",
        "Falla": "Pavimentación",
        "LatY": "19.290278",
        "LongX": "-99.65713",
        "Lugar": "Calle Miguel Hidalgo 216, Centro, 50000 Toluca de Lerdo,
Méx., México",
        "Foto": "http://geo24.zz.mu/Upimages/72082.jpg",
        "Reportado": "1"
    }
]
}

```

Código 11. Ejemplo de reportes en formato JSON de la URI fasepFull

En el resultado JSON mostrado se observa un ejemplo de la conformación de los objetos generados desde la base de datos para la URI fasepFull donde se puede apreciar que se añaden objetos como "Folio, Lugar, Foto y Reportado" a comparación del servicio para la aplicación FaSeP híbrida donde solo se generan objetos "Falla, LatY y LongX". Es importante mencionar que el campo "Reportado" muestra resultados con el carácter "1", ya que se simula que si es 1 la falla aun no es solucionada y si es 0 la falla en el servicio público ha sido resuelta, y que se contempla que si se tratara de una aplicación para uso distribuido a público en general este aspecto de señalar cuales fallas han sido resueltas y cuáles no, suelen ser importantes.

Se desarrolló una página web ligera en un servidor que consumiera los datos del servicio REST en la primera sección se introdujeron los resultados en forma de una tabla HTML, para simplificar el procesos de peticiones AJAX se hizo uso de la librería de JavaScript jQuery de esta manera se pudo reducir la cantidad de líneas de código y la complejidad del mismo. Lo mencionado anteriormente se puede observar en la parte fundamental de la codificación en JavaScript que realiza el proceso general del llenado de una tabla.

```

$.getJSON("http://geo24.zz.mu/fasepFull", function(data) {
    $.each(obj.markers, function(key,data) {
        $('#tablajson tbody').append(Código...);
    });
});

```

Código 12. Solicitud AJAX GET con jQuery

De manera global en código anterior como ya se había plasmado anteriormente, se hace una solicitud GET AJAX con jQuery donde se accede a un URI del servicio REST para obtener los datos de fallas públicos y realizar una iteración con la función `each()`.

La función `each()` de jQuery nos sirve para iterar uno por uno los elementos de una colección, usualmente estos son el resultado de un selector jQuery, pero también pueden ser otras colecciones de Javascript

como un array o un objeto que en este caso son resultados JSON del servicio REST.

Una vez que se ha generado la iteración en código se procedió a utilizar la función `.append` de jQuery con la que se puede añadir contenido nuevo a un elemento sin eliminar el que ya está hecho en este caso se generó una tabla en HTML con un identificador "tablajson" y se llenó utilizando la función mencionada en cada iteración generando una nueva fila en la estructura `<tbody>`.

```
<table id="tablajson">

  <thead>
    <th>Folio</th>
    <th>Falla</th>
    <th>Lugar</th>
    <th>Foto</th>
    <th>Estado</th>
  </thead>

  <tbody> </tbody>

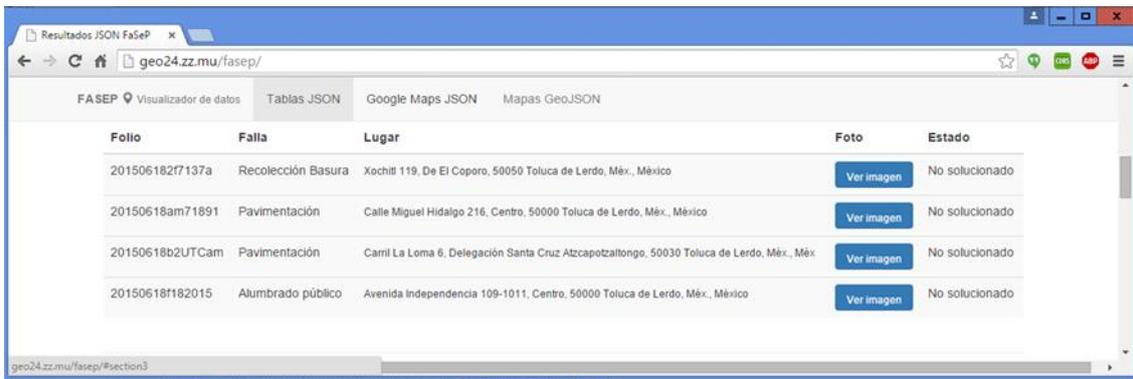
</table>
```

Código 13. Estructura de tabla HTML

```
$.getJSON("http://geo24.zz.mu/fasepFull", function(data) {
  $.each(obj.markers, function(key, data) {
    //introducir datos en tabla
    $('#tablajson tbody').append(
      function() {
        if (data.Reportado == "1") {
          data.Reportado = "No solucionado"
        } else if (data.Reportado == "0") {
          data.Reportado = "Solucionado";
        };
        return "<tr>" +
          "<td>" + data.Folio + "</td>" +
          "<td>" + data.Falla + "</td>" +
          "<td><small>" +
          data.Lugar + "</small></td>" +
          "<td>" +
          '<a class="btn btn-primary input-sm" href="' +
          data.Foto +
          '" target="_blank">' +
          "<small>Ver imagen</small></a>" +
          "</td>" +
          "<td>" +
          "<w id='st'" +
          data.Reportado + "</w></td>" +
          "<tr>";
      }
    );
  });
});
```

Código 14. Solicitud y llenado de tabla HTML con datos JSON de servicio REST de fallas en servicios públicos

El resultado puede obtenido puede apreciarse en la siguiente ilustración (pág. Siguiente, ver Ilustración 31).



Folio	Falla	Lugar	Foto	Estado
201506182f7137a	Recolección Basura	Xochitl 119, De El Coporo, 50050 Toluca de Lerdo, Méx., México	<a href="#">Ver imagen</a>	No solucionado
20150618am71891	Pavimentación	Calle Miguel Hidalgo 216, Centro, 50000 Toluca de Lerdo, Méx., México	<a href="#">Ver imagen</a>	No solucionado
20150618b2UTCam	Pavimentación	Carril La Loma 6, Delegación Santa Cruz Atzacapotzallongo, 50030 Toluca de Lerdo, Méx., Méx	<a href="#">Ver imagen</a>	No solucionado
20150618f182015	Alumbrado público	Avenida Independencia 109-1011, Centro, 50000 Toluca de Lerdo, Méx., México	<a href="#">Ver imagen</a>	No solucionado

Ilustración 30. Vista de tablas con datos JSON

Otras vistas que se pueden generar en HTML son los mapas en un caso puede ser una representación con la API de Google Maps, procedimiento que ya se ha realizado en la aplicación FaSeP y son los mismo pasos y código, la razón como ya se ha mencionado es que las aplicaciones híbridas tienen un núcleo web de HTML, CSS y JavaScript.

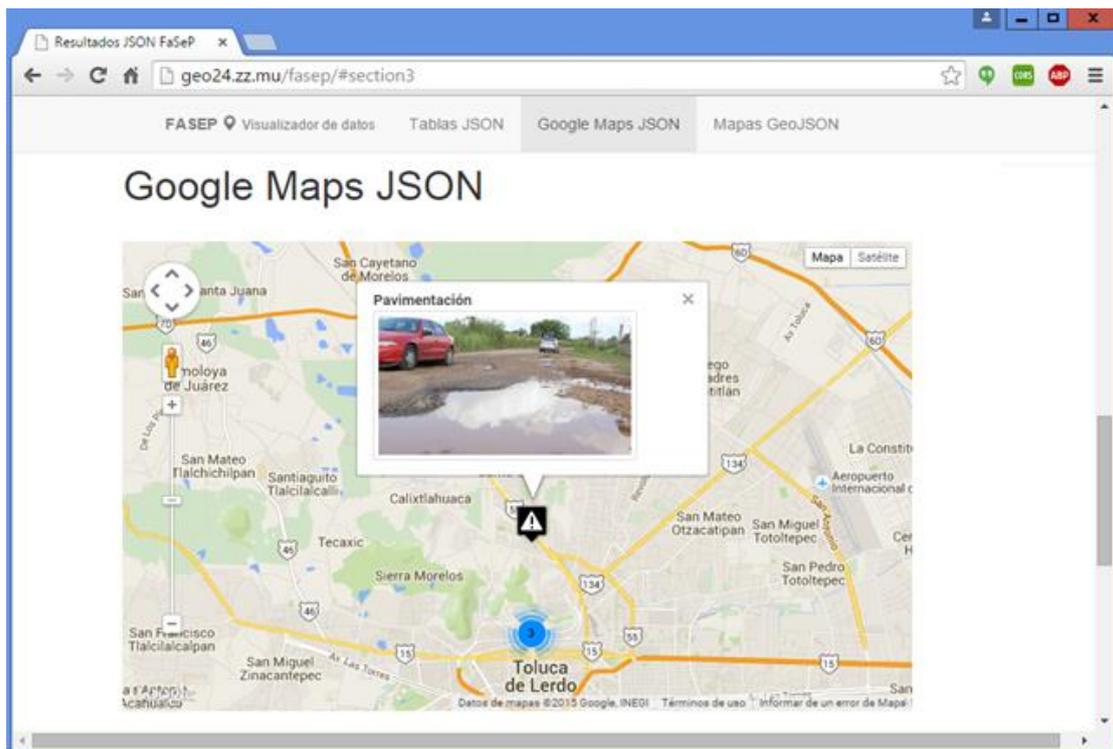


Ilustración 31. Visualizador de datos JSON de fallas en servicios públicos en Google Maps

Otro formato que el servicio REST de fallas en servicios públicos genera es GeoJSON, desde el servidor se integró una librería libre llamada GeoPHP creada por el programador Patrick Hayes, para generar las coordenadas por una conversión de datos GEOMETRY de MySQL a objetos GeoJSON

a través de la función interna WKT, cabe destacar que otros formatos o archivos que puede generar la librería son KML, GPX, GeoRSS.

Al acceder a la URI "fasepGeo" (<http://geo24.zz.mu/fasepGeo>) se genera una estructura de objetos geometría en este caso solo puntos con datos acerca de fallas en servicios públicos en el municipio de Toluca de Lerdo generados por los reportes de la aplicación FaSeP.

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -99.658417460327,
          19.321019126969
        ]
      },
      "properties": {
        "Falla": "Pavimentación",
        "Foto": "http://geo24.zz.mu/Upimages/72081.jpg"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -99.653997179871,
          19.292100757553
        ]
      },
      "properties": {
        "Falla": "Alumbrado público",
        "Foto": "http://geo24.zz.mu/Upimages/72083.jpg"
      }
    }
  ]
}
```

Código 15. Ejemplo de reportes en formato GeoJSON de la URI fasepGeo

El formato de objetos GeoJSON contiene características como tipo de falla y fotografía además de la geometría aunque las propiedades pueden aumentarse, es decir agregar el Folio, la Dirección, etc.

El objeto que manda la URI "fasepGeo" puede ser consumido por distintas plataformas, por una parte integrarse a vistas HTML como ya se ha realizado con objetos JSON sencillos, pero no es el fin, ya que vale más poder aplicar los datos a plataformas SIG que en muchos casos con

reconocidas por OGC, por esta razón se generó un mapa con Leaflet que es una librería JavaScript para generar mapas interactivos, la base cartográfica de esta librería es Open Street Maps, si bien se pudo trabajar con OpenLayers donde también se pueden consumir datos en formato GeoJson se eligió Leaflet porque la codificación es amigable y ligera, además al igual que OpenLayers puede integrar servicios Tiles.

El procedimiento de obtener los datos en una aplicación híbrida o web a través de una petición GET es el mismo que se utilizó para introducir los registros en el mapa Leaflet, es decir también se agregó la función `each()` para recorrer los objetos y generar de esta forma una capa de marcadores GeoJSON que se agregó al mapa Leaflet.

```
$.getJSON("http://geo24.zz.mu/fasepGeo", function (data) {

    //Generación de objeto para crear Clusters
    var markers = L.markerClusterGroup();

    //Generación de icono
    var myIcon = L.icon({
        iconUrl: 'css/img/accident.png'
        , iconSize: [32, 37]
        , iconAnchor: [16, 37]
        , popupAnchor: [0, -28]
    });

    //Generación de marcadores y características
    //por medio de función each()
    function onEachFeature(feature, layer) {
        var popupContent = "<div class='an'><strong class=' ' id='i'>" +
            feature.properties.Falla + "</strong>" +
            "<img class='img-thumbnail ma' src='" +
            feature.properties.Foto + "'/></div>";
        layer.bindPopup(popupContent);
    }

    var geojson = L.geoJson(data
        , {
            onEachFeature: onEachFeature
            , pointToLayer: function (feature, layer) {
                return L.marker(layer
                    , {
                        icon: myIcon
                    }
                );
            }
        }
    );
    //integración de capa de marcadores geojson a capa Leaflet
    markers.addLayer(geojson);
    map1.addLayer(markers);
    map1.fitBounds(markers.getBounds());
});
```

Código 16. Solicitud de datos y proceso para generar capa de marcadores GeoJSON en un mapa Leaflet

El resultado generado por el código mostrado es un visualizador de datos GeoJSON en un mapa Leaflet que puede observarse a continuación (pág. Siguiete, Ilustración 33.).

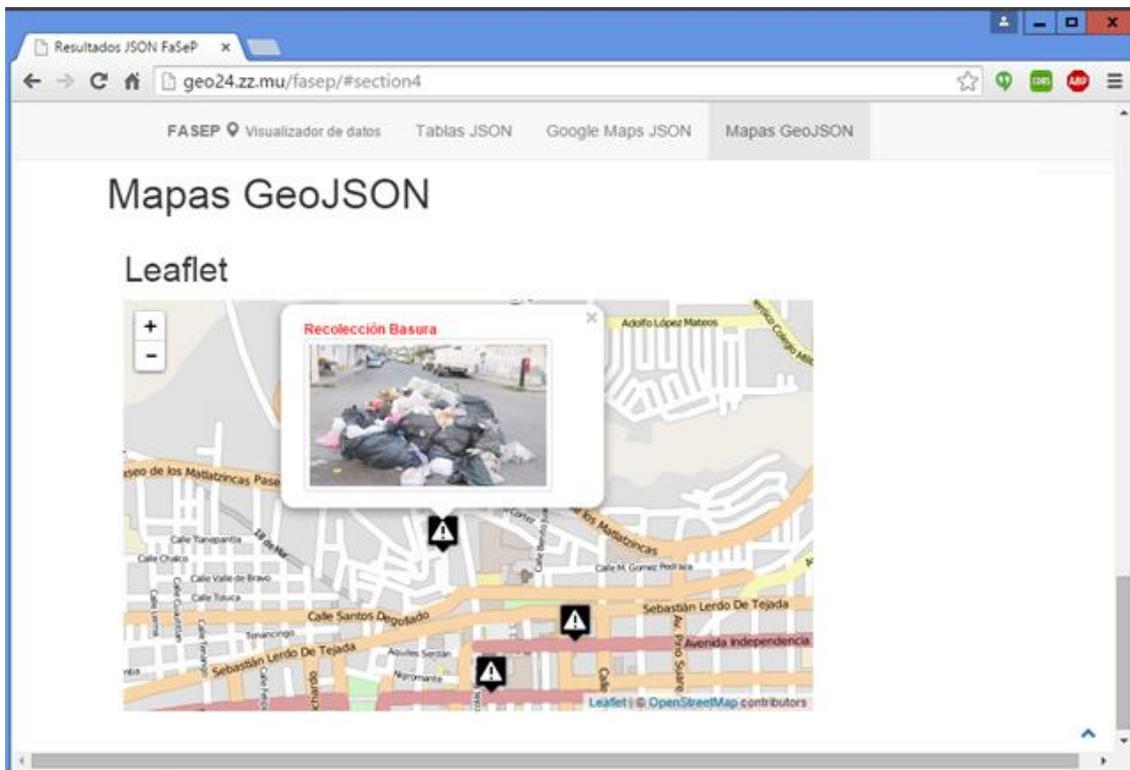


Ilustración 32. Visualizador de datos GeoJSON de fallas en servicios públicos en un mapa Leaflet

Acerca de los datos GeoJSON una de las ventajas es que no solo pueden consumirse en interfaces de aplicaciones web, también son compatibles en plataformas SIG de escritorio, para comprobarlo se utilizó QGIS Desktop 2.8.1 para añadir una capa GeoJSON del servicio REST.

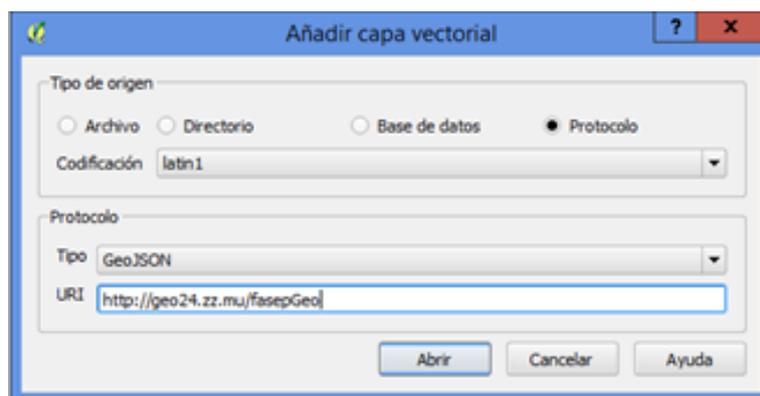


Ilustración 33. Agregación de capa vectorial GeoJSON desde servicio REST en QGIS

Con la carga del servicio web REST FaSep en QGIS se pueden visualizar los datos online consumidos en representaciones vectoriales (pág. Siguiete, Ilustración 35), pero también se puede pensar en análisis

espacial considerando que los datos incluyen propiedades de las fallas en servicios públicos.

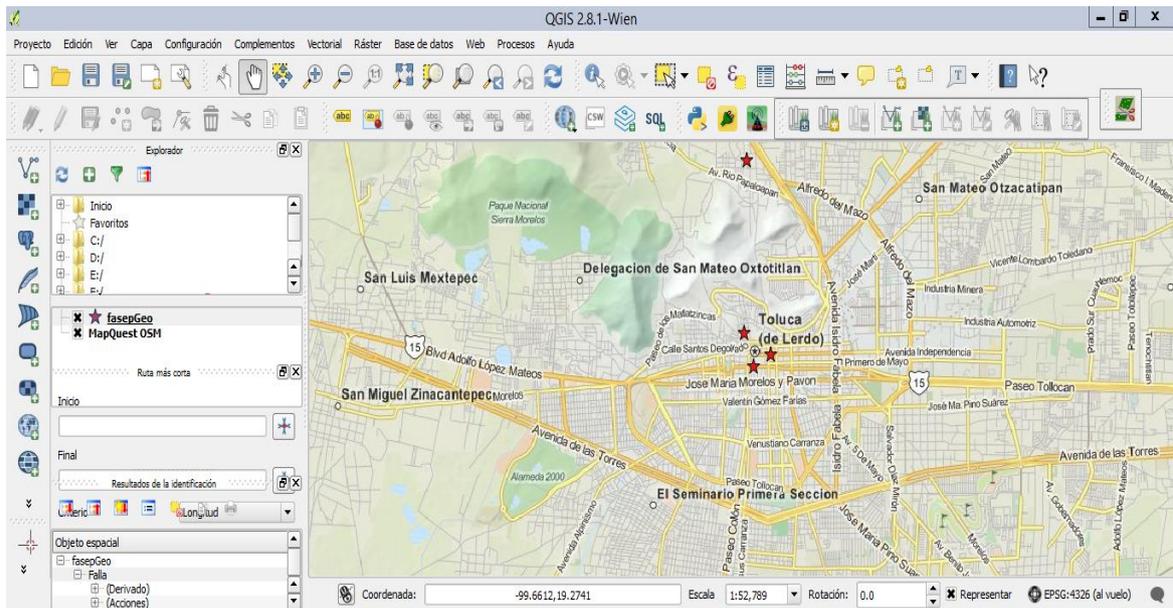


Ilustración 34. Visualización de capa de datos GeoJSON en QGIS

### 3. RESULTADOS

Las aplicaciones híbridas son alternativa para generar aplicaciones para dispositivos móviles multiplataforma, al utilizar lenguajes web para desarrollar una herramienta para generar datos a través de la captura de coordenadas para analizar que tanto las aplicaciones híbridas tienen potencial para ser utilizadas con el fin de capturar eventos y fenómenos en el espacio geográfico, se encontró en primer lugar que un sistema así conlleva no solo tener un buen rendimiento en el dispositivo móvil ya que también se necesita agilizar los procesos de servidor donde se hacen las solicitudes de almacenamiento y consumo de datos, lo anteriormente mencionado también está relacionado con la conexión a una red con internet es decir las redes móviles (CDMA, GSM, 3G y 4G), el ancho de banda utilizado para realizar las transacciones de datos es el aspecto que más se tuvo y debe tomarse en cuenta. “La integración de los dispositivos móviles, Internet y la conectividad inalámbrica ofrece una oportunidad extraordinaria para extender información y servicios” (Microsoft, 2004)

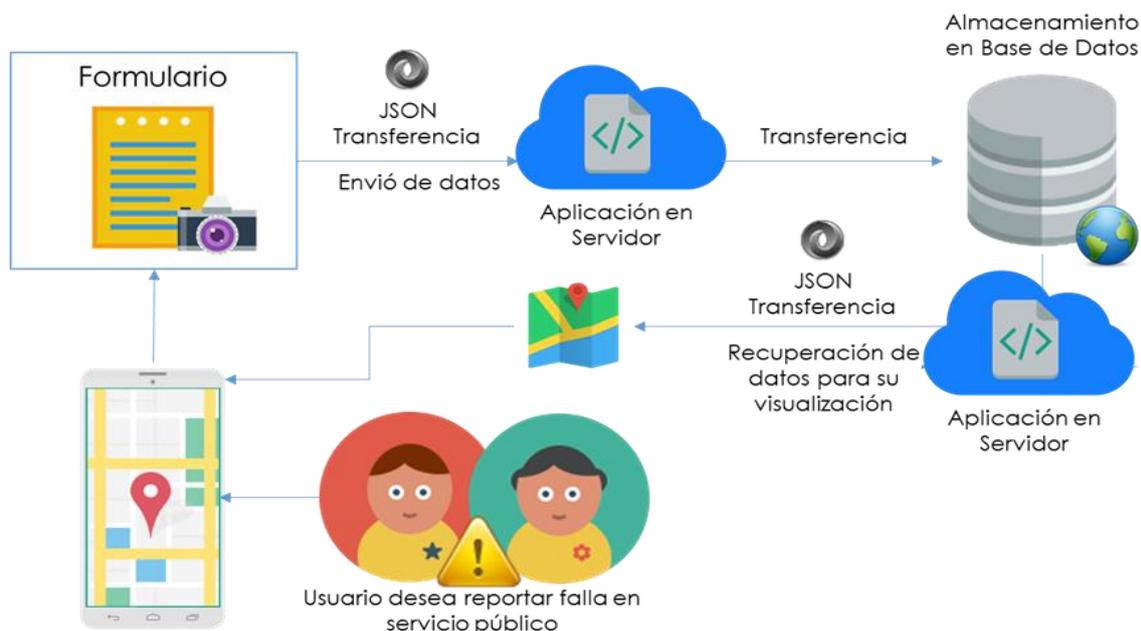


Ilustración 35. Diagrama de funcionamiento de aplicación para reportar fallas en servicios públicos en el municipio de Toluca. Elaboración propia

Como se muestra en la *ilustración 6* se generó un diagrama general del sistema donde se puede apreciar interactúan diversos elementos, uno que es la aplicación móvil y el uso de los sensores hardware que tienen instalados y la aplicación de servidor que responde a las peticiones que se hacen en la interacción entre el entorno *Front-end* y *Back-end*. Los elementos de la aplicación para reportar fallas en servicios públicos al

igual que en el diagrama mostrados interactúan entre sí, no se puede separar una función de la otra.

El análisis de las funciones de un sistema para generar datos a través del GPS de un dispositivo móvil no puede aislar a los elementos del mismo sistema sin tomar en cuenta que interactúa uno con otro casi la mayoría del tiempo. En el sistema FaSeP los reportes de fallas públicas se hacen a través de transferencias con el servidor y la visualización por medio peticiones de datos, por ello uno de los elementos importantes a analizar es el ancho de banda que utiliza la aplicación al generar los reportes.

### 3.1 Costos aproximados

Tomando en cuenta lo anteriormente señalado, para lograr concebir una visión más clara del gasto de datos móviles de la aplicación FaSeP se generaron 20 reportes de fallas en servicios públicos en la calle de la ciudad de Toluca. Los primeros aspectos a analizar fue el consumo de datos en la primera ejecución de la aplicación y el primer reporte mientras el dispositivo móvil se encontraba conectado en una red 3G realizado el resultado se muestra en la siguiente ilustración.

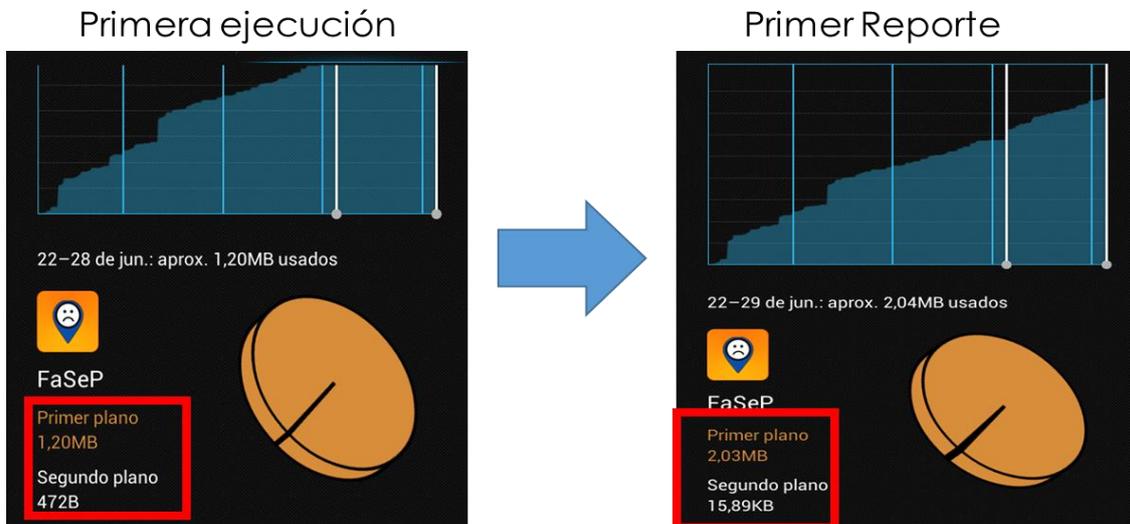


Ilustración 36. Comparación de datos gastados entre primera ejecución y primer reporte en la aplicación FaSeP

En la ilustración 36 claramente se puede observar el uso de datos en los procesos ya mencionados, formando un total aproximado a los 2.05 megabytes contando el consumo en segundo plano, el resultado que representa es significativo si se quiere conocer que tanto saldo o dinero se gastó en consumir estos datos mientras se estaba conectado a la red móvil de un servicio de telefonía de prepago.

Empresa	Tarifa de Internet 	Calculo de gasto FaSeP
 movistar	\$0.85 por Mb	\$1.7425 pesos
 nextel	\$2.00 por Mb	\$4.1 pesos
 telcel	\$0.85 por Mb	\$1.7425 pesos
 USACEL	\$1.19 por Mb*	\$2.4395 pesos
 Unefon	\$1.1264 por Mb*	\$2.30912 pesos

\* sujeto a condiciones

Tabla 7. Calculo de gasto de internet al momento de realizar la investigación en primera ejecución y primer reporte de falla con la aplicación FaSeP según servicio de telefonía. Tabla elaborada con datos de la IFT.

En cálculos netos la primera ejecución y reporte de falla en un servicio público con la aplicación FaSeP resulta menos costosa en servicios de prepago en empresas telefonía como como Movistar y Telcel, sin embargo este resultado depende también del estado de la conexión, en esencia quizá resulte más económico realizar los reportes desde un dispositivo móvil que tenga contratado cierto servicio de telefonía, sin embargo la señal y la rapidez serán factor, incluso los precios pueden cambiar si se tiene contratado un servicio con plan de datos.

Entonces el peso aproximado de una transacción de datos en un reporte es 0.85 megabytes, que comprende el proceso de carga de un reporte al servidor que contiene los datos de la falla y una fotografía, y también la recuperación de un nuevo registro para representarlo en el mapa en un marcador. El precio en cálculos brutos para servicios como Telcel y Movistar es de \$0.7225 pesos, esto independientemente de que se haga un paneo al mapa, ya que si esto pasa el precio de uso de datos incrementa. Es importante no confundir entre el precio de la transacción y la carga del mapa ya que los otros 1.24 megabytes (\$1.054 pesos) son el gasto de carga de la vista del mapa de Google, esto sin considerar los movimientos o acciones de zoom.

Aparentemente el precio de los veinte reportes conociendo el costo aproximado debería acercarse al que se muestra en la siguiente tabla, aunque ya se ha mencionado que los cálculos están basados en datos brutos que no consideran otras variables más que la carga del mapa y el reporte enviado.

Empresa	Costo virtual aproximado de 20 reportes en fallas en servicios públicos según primer reporte
 movistar	\$34.85 pesos
 nextel	\$82.0 pesos
 telcel	\$34.85 pesos
 USACEL	\$48.79 pesos*

Unefon 	\$46.1824 pesos*
* sujeto a condiciones	

Tabla 8. Cálculo bruto aproximado de costo de veinte reportes considerando el costo del primer reporte

El proceso de cálculo se realizó de esta manera para identificar de manera aproximada el peso en megabytes de una transacción en comparación con los reportes realizados considerando las variables de la realidad de los servicios de telefonía. Por otro lado los cálculos apoyaron a tener un punto de vista aproximado de los gastos por reporte, con lo anteriormente mencionado se puede decir que los el mayor consumo de ancho de banda lo hace la carga del mapa y no en si los reportes, así entonces también se considera que un servicio REST es una alternativa ligera para almacenar datos de una aplicación como FaSeP que genera datos de fenómenos que ocurren en el espacio.

Comparación entre tiempo de respuesta y peso de archivo respuesta JSON del servicio REST de fallas en servicios públicos		
URI	Tiempo de Respuesta	Peso de archivo JSON
<a href="http://geo24.zz.mu/fasepMob">http://geo24.zz.mu/fasepMob</a>	388 ms	1 KB
<a href="http://geo24.zz.mu/fasepFull">http://geo24.zz.mu/fasepFull</a>	647 ms	2 KB
<a href="http://geo24.zz.mu/fasepGeo">http://geo24.zz.mu/fasepGeo</a>	1244 ms	2 KB

Tabla 9. Comparación entre tiempo de respuesta y peso de archivo respuesta JSON del servicio REST

Como se puede observar en la tabla 9 el servicio REST generado crea resultados ligeros tomando en cuenta que son veinte fallas reportadas que en comparación con los precios mencionados anteriormente el costo del consumo al menos de los datos de la URI <http://geo24.zz.mu/fasepMob> pueden ser relativamente bajos, y con esto se puede asegurar que los servicios RESTful son una alternativa ligera para aquellas aplicaciones que se desarrollen para generar datos a través de coordenadas obtenidas por un dispositivo móvil que quieran almacenarse en un servidor.

### 3.2. FaSeP aplicación híbrida para Android

Como ya se ha mencionado en esta investigación, las aplicaciones híbridas mantienen un carácter multiplataforma en comparación con las nativas, en esta investigación se desarrolló una aplicación llama FaSeP resultado de ejemplificar el proceso de generación de datos georreferenciados utilizando el GPS de los dispositivos móviles inteligentes. El ejemplo consto en generar una aplicación para reportar fallas en

servicios públicos en el municipio de Toluca, de esta forma se buscó representar fenómenos o eventos que ocurrían en el espacio geográfico, específicamente en este caso averías en servicios públicos en Toluca.

FaSep de lado de la aplicación móvil tiene una base de programación HTML5, CSS3 y JavaScript; si se quisiera definir el proceso de desarrollo se señalaría que la complejidad no constituye un obstáculo, principalmente porque en las aplicaciones nativas se tiene que aprender la codificación para los procesos en distintos lenguajes.

En las aplicaciones híbridas gracias a que se pueden utilizar lenguajes web se pueden añadir a los proyectos librerías JavaScript que aligeran las cargas de código, además librerías de interfaz de usuario que asemejan el entorno *front-end* a los gráficos de una aplicación nativa.

	Device Access	Speed	Development Cost	App Store	Approval Process
Native	Full	Very Fast	Expensive	Available	Mandatory
Hybrid	Full	Native Speed as Necessary	Reasonable	Available	Low Overhead
Web	Partial	Fast	Reasonable	Not Available	None

Tabla 10. Comparación entre aplicación móviles. Fuente: Aplicacionesmovil.com

La comparación entre aplicaciones móviles lleva a un punto en que se analizó FaSeP de una manera más integral donde si bien una aplicación híbrida desarrollada en PhoneGap, una característica que es cierta es que gracias a los *plugins* se puede acceder al control de los sensores del dispositivo de una manera local aunque no con la misma velocidad que en una aplicación nativa, en comparación con aplicaciones populares en el mercado como Waze que tiene una programación más robusta en plataformas nativas y que maneja más datos FaSeP suele tardar más en la carga de la aplicación en sí, en cuanto al servicio de datos REST el tiempo de carga es mínimo aunque no puede apreciarse debido a la carga del resto de la aplicación.

En cuanto costo se puede mencionar el tiempo de desarrollo que es la medida en que se puede cuantificar en esta ocasión la aplicación híbrida FaSeP, debido a la cantidad tan basta de documentación la creación de tiempo no demora un cantidades extensas de tiempo, incluso aunque se añadieron características como captura de fotos o

archivos de la galería de imágenes la complejidad no incremento de manera que tuviera impacto en el tiempo de desarrollo de la aplicación.

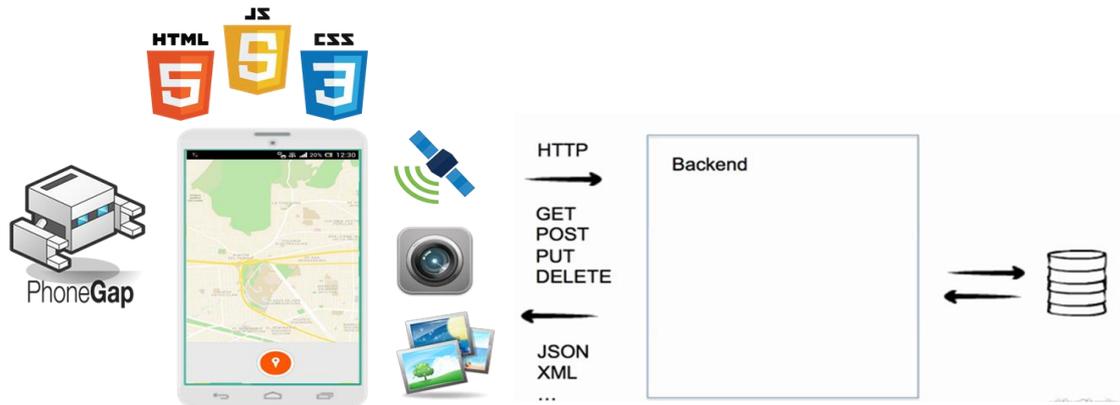


Ilustración 37. Aplicación FaSeP.

Los resultados de la aplicación FaSep para generar información utilizando la localización en dispositivos móviles pueden ser un tanto subjetivos, ya que se puede responder de una manera positiva al cuestionamiento ¿Se puede generar información georreferenciada con aplicaciones híbridas para dispositivos móviles?, sin embargo no es el cuestionamiento correcto, ya que fácilmente se puede comprobar que utilizando tecnologías web estándar dentro de la aplicación de servidor y la móvil se puede lograr una interacción de transacción de datos en formatos ligeros. Los datos provenientes del servicio REST influirán hasta cierto punto en el rendimiento de la aplicación, pero la aplicación debe lograr una estabilidad que no dependa de los servicios externos, PhoneGap es una herramienta potente para generar aplicaciones híbridas, existen otros frameworks que son capaces de emular la misma acción, sin embargo los puntos que deben tratarse en el desarrollo de este tipo de aplicaciones es el desarrollo de la lógica de la app donde interviene el lenguaje JavaScript, ya que desde esta capa se puede optimizar el resto de la infraestructura de la aplicación móvil, aunque además se debe tomar en consideración que la plataformas para generar aplicaciones híbridas también avanzan en cada versión liberada.

Regresando al tema de los cuestionamientos que deben tomarse en cuenta para desarrollar una aplicación para generar datos georreferenciados van más allá del concepto de la aplicación instalada en un teléfono o Tablet si no en los usuarios, el objetivo que se busca cubrir al generar datos, el tiempo que se invertirá en desarrollo, los costos, etc.

No es tan ajeno el concepto SIG en una aplicación del tipo ya mencionado ya que se busca capturar, almacenar, desplegar información de carácter geográfico que posteriormente puede ser analizada en la aplicación de ejemplo desarrollada en esta investigación se generó información puntual, a través de un proceso de captura que

llego hasta el punto de crear resultados en formatos compatibles con procesadores de información geográfica como QGIS.

Teniendo en cuenta la influencia de los SIG en la aplicación FaSep se puede proceder a un cuestionamiento como: ¿Quién va a usar la aplicación?

Al igual que en los SIG los usuarios son importantes en esta investigación se centró la atención en usuarios comunes de dispositivos móviles como puede ser cualquier persona caminando en la calle que identifica una falla en un servicio público y puede reportarlo haciendo uso de su teléfono en el momento que observa el fenómeno.

Este tipo de usuarios necesita que las aplicaciones funcionen con rapidez, funcionales con interfaces amigables, en la aplicación FaSeP se trató de minimizar la cantidad de botones y hacer los que tenía más intuitivos.

Sin embargo aunque un colectivo de usuario siempre genere mayor cantidad de datos, siempre habrá que desconfiar en la calidad y veracidad de los mismos, aunque como ya se sabe en aplicaciones como Waze los usuarios generan datos por un bien común que es agilizar el tránsito por las ciudades. El resultado que se espera los usuarios puedan actuar por bien común y bien colectivo, y que de estas acciones se generen datos para ser analizados posteriormente.

*“Sin duda los dispositivos móviles han transformado nuestra forma de interactuar con el mundo, además de ser una plataforma que permite la comunicación directa de las marcas con los usuarios”* (Richaud, 2013)

Las consideraciones pertinentes tomar por otro lado es que no todos los sistemas y los usuarios de estos son iguales, por ello habrá ocasiones en que las aplicaciones se utilizaran para solventar problemas más afines a las organizaciones, por ejemplo recientemente empresas de televisión por cable buscan una manera de identificar los postes e instalaciones donde se toma la señal de cable, una solución podría usar aplicaciones del tipo que se desarrolló en esta investigación.

Otro cuestionamiento razonable es: ¿Es necesario el acceso a los sensores del dispositivo?, anteriormente se decía que las aplicaciones híbridas no eran capaces de acceder a funciones nativas de los dispositivos, por otro lado por suerte los frameworks han evolucionado también, por ejemplo si bien se era capaz de acceder a funciones como la cámara o el GPS en aplicaciones híbridas, aún quedaban funciones que no podían realizarse como lectura de códigos QR, todo cambió cuando las necesidades de desarrollo multiplataforma no tan robustos apareció, ya que desarrolladores y comunidades de desarrolladores empezaron a generar *plugins* propios, tal es el caso de PhoneGap que siempre ha trabajado en

un concepto OpenSource, de esta forma los desarrolladores pueden adecuar los *plugins* a los objetivos de la aplicación que desarrollan y puede verse que gran variedad de los *plugins* creados se publican en internet razón por lo cual no hay excusa en poder utilizar muchas de las funciones nativas de los dispositivos, y es verdad que no tienen el mismo rendimiento que una aplicación nativa sin embargo llevan consigo la ventaja de ser multiplataforma.

Existe un árbol de decisiones generado por la empresa OutSystems para tomar la decisión de qué tipo de aplicación debe generar dependiendo los objetivos del desarrollador, se muestra a continuación para complementar la parte de los cuestionamientos necesarios aplicables a cuando se desea desarrollar una aplicación híbrida para generar información de eventos y fenómenos en el espacio geográfico a través del uso de los sensores de localización de un dispositivo móvil.

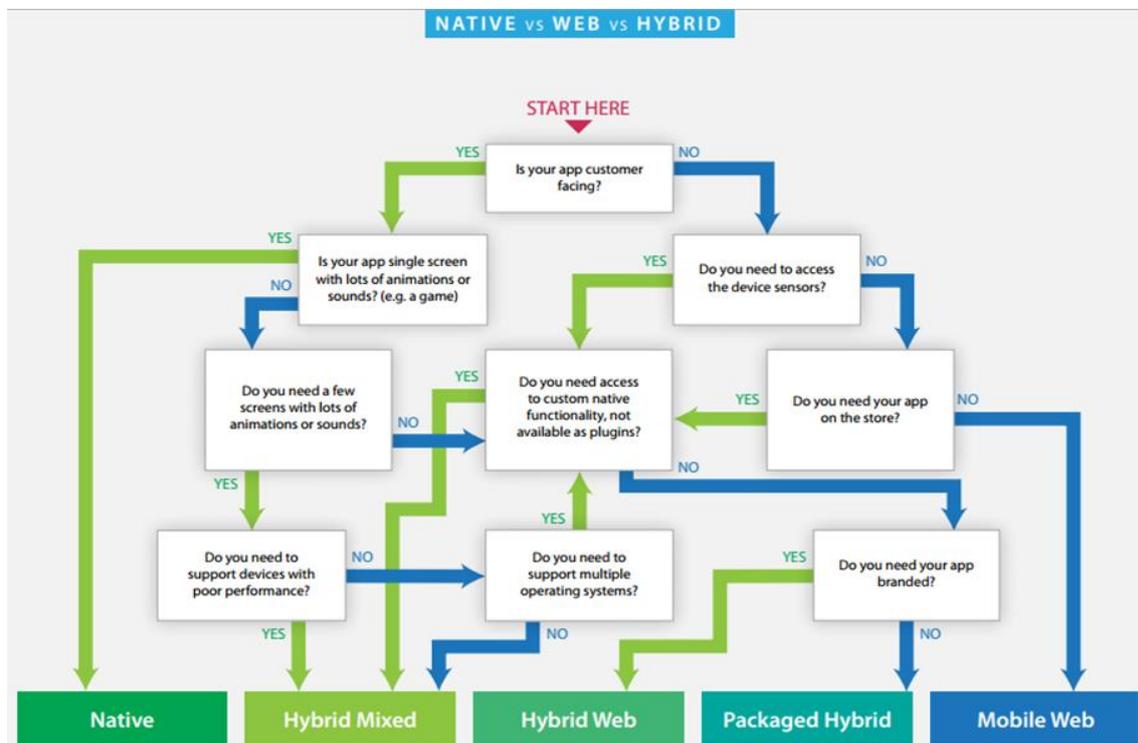


Ilustración 38. Árbol de decisiones para el desarrollo de aplicaciones nativas, híbridas y web. Fuente: OutSystems

Enfocándose en resultados más técnicos e independientemente de las consideraciones que se mencionaron anteriormente, al desarrollar la aplicación híbrida FaSeP para Android que generar datos a través de reportes en fallas en servicios públicos se cumplió el objetivo que se persiguió a lo largo del trabajo de investigación, que fue generar datos aprovechando el GPS de los dispositivos móviles. A pesar de las limitaciones que ya se ha mencionado que puede tener una aplicación híbrida para dispositivos móviles se logró llegar a obtener un diseño

conciso y eficiente en cuanto a la captura de los datos para las fallas en servicios públicos desde el punto de vista de la interfaz del usuario. Además se logró integrar sensores o fuentes de datos del dispositivo móvil para añadir valor agregado a los reportes, de lado del servidor se desarrolló un servicio RESTful ligero que arroja resultados en formato JSON y GeoJSON.

Recapitulando las actividades y procesos que la aplicación FaSeP que se fueron mostrando progresivamente en el desarrollo del documento se pudo apreciar la esencia de la investigación que como ya se ha mencionado es generar información georreferenciada a través de un dispositivo móvil y que consta de un ciclo que involucra la interacción entre la aplicación móvil y la aplicación de servidor además de la interacción del usuario y la importancia que este tiene en la generación de los datos.

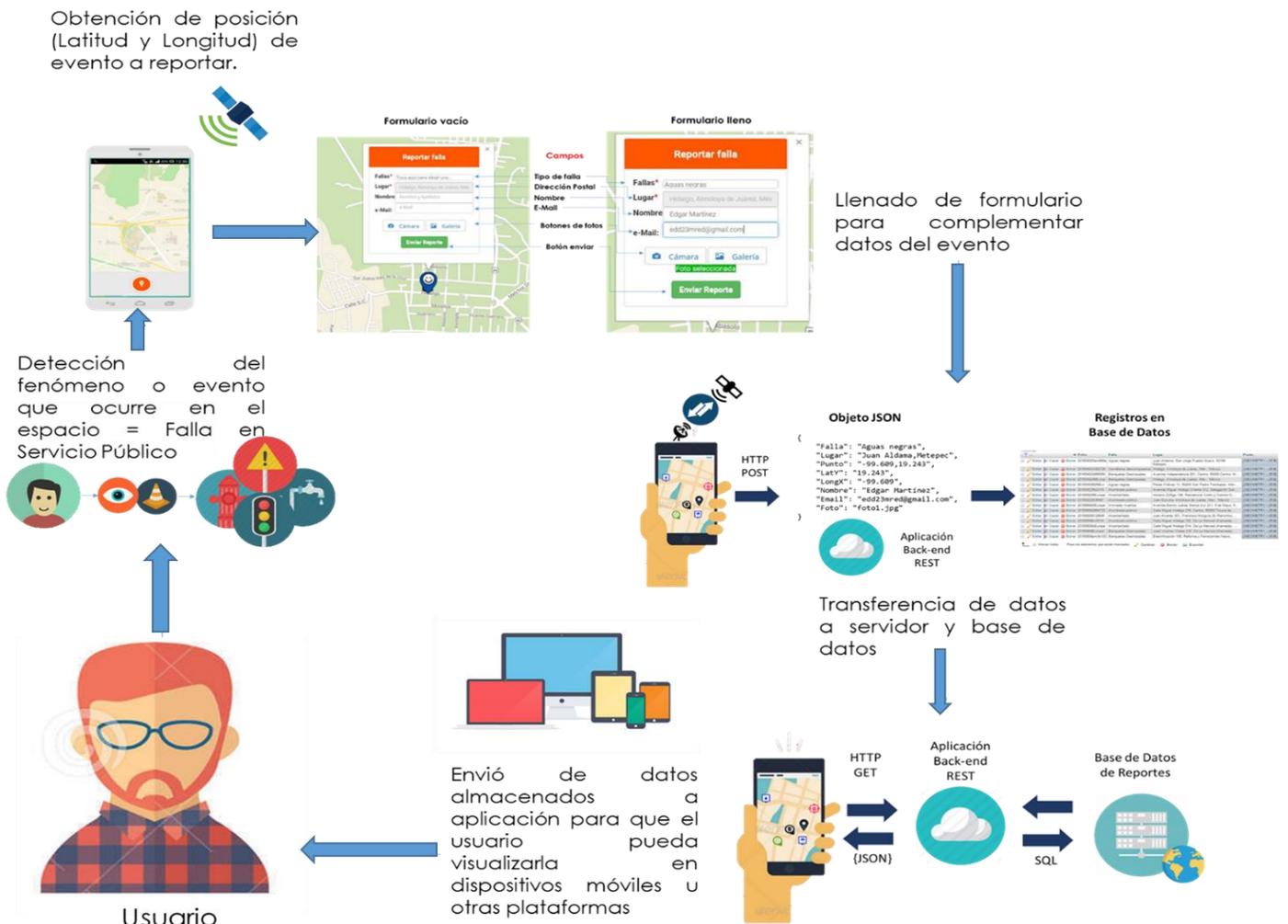


Ilustración 39. Actividades y procesos realizados en la aplicación FaSeP



## 4. CONCLUSIONES Y RECOMENDACIONES

Para concluir este trabajo de tesis, en este capítulo se expondrán las conclusiones y recomendaciones derivadas de este proyecto. Lo anterior con el fin de mostrar análisis de lo argumentado y lo probado a lo largo de las páginas de este documento y así como la posible continuidad del proyecto.

### 4.1. Conclusiones

A lo largo de esta tesis se buscó demostrar como las aplicaciones híbridas son una herramienta funcional para la generación de datos geoespaciales a partir del uso de la geolocalización en dispositivos móviles inteligentes.

Así pues, se observó en parte que debido a la creciente comunidad usuarios de internet que acceden al mismo desde sus dispositivos móviles, las aplicaciones móviles pueden desarrollarse para tener la característica de generar datos que compartan los usuarios, por ello la aplicación para reportar fallas en servicios públicos en el municipio de Toluca (FaSeP), que se desarrolló como ejemplo para esta tesis fue un ejercicio enriquecedor que ayudo a demostrar el potencial de las aplicaciones móviles híbridas para generar datos de geolocalización con dispositivos como *smartphones*.

Con el desarrollo de la aplicación FaSeP se demuestra que usando distintas tecnologías web se pueden generar aplicaciones móviles para Android con PhoneGap. El desarrollo involucró la generación de datos geoespaciales y también una constante interacción entre dispositivo móvil y la aplicación en el diseño y la ejecución de los elementos *front-end* y *back-end*. En este sentido:

- Las características de los dispositivos móviles requeridos en la aplicación FaSep como la cámara, el acceso a la galería, notificaciones y GPS, fueron fácilmente añadidas con PhoneGap, en cuanto la aplicación requiere de su uso los sensores solicitados funcionan de una manera rápida.
- Además, dentro de la codificación, la manipulación del hardware solo se hace utilizando JavaScript, lo anterior hizo que la tarea de integrar los sensores con otras funciones de la aplicación fuera ágil, como en el caso de la geolocalización.

- El sensor GPS se integró fácilmente con librerías de mapas como la API de Google Maps y OpenLayers, con la API Geolocation W3C de esta forma se logró obtener la ubicación representarla en un mapa en la aplicación, para la detección de la ubicación geográfica de la falla en un servicio público con FaSeP, sin embargo la geolocalización del reporte depende mucho de la red de internet a la que el dispositivo esté conectado, ya que de la conexión depende la mayor exactitud de las coordenadas detectadas.
- Por otro lado, la transferencia de los datos obtenidos al servidor es un elemento importante en la aplicación. Desde la parte front-end, el uso de JavaScript y AJAX fueron fundamentales para lograr la conexión para enviar los reportes al servidor en formato JSON y desde la parte back-end, la aplicación REST de servidor desarrollada con PHP mostró un buen desempeño en línea para recibir y almacenar los objetos JSON recibidos en la base de datos MySQL.
- Así mismo, el consumo de los datos del servicio REST desarrollado comprende una ventaja en cuanto a la ligereza del formato de transferencia ya sea JSON o GeoJSON, con el último se observó que cuando se requiere un resultado con formato vectorial georreferenciado este mismo puede ser compatible con múltiples plataformas incluyendo software SIG de escritorio.
- Un punto aparte que es benéfico en el desarrollo de aplicaciones híbridas como FaSep es el tiempo, se debe decir que aunque se capturan datos en un servidor a partir de la geolocalización, FaSeP es una solución ligera que integra diversas técnicas de programación web sin añadir complejidad extra, de esta forma se obtuvieron resultados aceptables.

No obstante a los puntos anteriormente mostrados, un aspecto que involucra una mención especial es la generación de datos, pero no visto solo desde la perspectiva técnica, más bien ampliando el panorama a un concepto más integral. Lo mencionado es debido a que en la aplicación FaSeP mantiene una relación indisoluble entre el usuario y el proceso de creación de un reporte en una falla en un servicio público.

Haciendo referencia a lo mencionado se concluye que los datos almacenados tendrán una calidad proporcional al proceso de captura, disminuida o mejorada en relación al tiempo de carga y transacción de los datos. Todo lo anterior tiene que ver en mayor parte con el desarrollo y el diseño de la aplicación híbrida pensada en el usuario, ya que son la

pieza principal en un sistema basado en la ubicación y generación de datos para dispositivos móviles, como el que se trató de ejemplificar con FaSeP.

## 4.2. Recomendaciones

Si bien la aplicación FaSeP fue un ejercicio de ejemplificación donde se desarrolló un prototipo de una aplicación híbrida con PhoneGap para generar datos geoespaciales a partir de dispositivos móviles, esta herramienta puede convertirse en un proyecto ambicioso si en un futuro algún estudiante quisiera retomarlo. Si ese fuera el caso los principales puntos que se pueden recomendar con base en los resultados que dejó la investigación son los siguientes:

- Ya que se trata de un proyecto que se implementa en un entorno urbano se recomienda cambiar los mapas a la librería de OpenLayers, esto permitirá poder agregar una capa Tiles offline, para disminuir el consumo de datos móviles añadiendo mapas precargados en la aplicación a cierto nivel de acercamiento.
- Modificar los *scripts* del formulario que utilizan JQuery a la librería Angular JS, esto disminuirá el tiempo de carga. También se recomienda mejorar la validación de los campos.
- Generar un portal donde los usuarios de la aplicación puedan consultar si la falla que reportaron se ha reparado o no, así mismo, añadir esta característica a la aplicación, lo anterior conlleva por su parte añadir un sistema de cuentas de usuario.
- Sería recomendable poder aplicar la gama de plataformas compilando las aplicaciones para distintos sistemas operativos móviles, y así ampliar un mayor número de usuarios.

Para la culminación de este documento se agrega como comentario final que existen áreas de aprendizaje dentro de la línea de acentuación “*desarrollo de sistemas geotecnológicos*”, que fortalecieron el criterio de desarrollo de este proyecto. Unidades de aprendizaje tales como: *almacén de datos geoespaciales, desarrollo de proyectos geotecnológicos y seminario de innovaciones geotecnológicas* ayudaron a sentar las bases teóricas de este proyecto de tesis, sin embargo, también existen áreas donde se pueden aplicar herramientas como la que se desarrolló (FaSeP), las cuales tienen que ver con análisis de datos geoespaciales como: *infraestructura de datos espaciales y geomarketing*.

## REFERENCIAS

- “Consideraciones acerca del impacto de Google Earth en la valoración y difusión de los productos de georrepresentación. (2006). GeoFocus, [online] 6(1578-5157), pp.3-4. Available at: [http://geofocus.rediris.es/2006/Editorial\\_2006.pdf](http://geofocus.rediris.es/2006/Editorial_2006.pdf) [Accessed 10 Nov. 2014].
- ABEYSINGHE, S. (2008). RESTful PHP web services. Birmingham [U.K.]: Packt Pub.
- ANBIAH, R., BHATTARAI, R. and SEDLIAK, M. (2011). PHP Ajax cookbook. Birmingham, U.K.: Packt Pub.
- BERGANZA, (2014). JSON. [online] video2brain. Available at: <https://www.video2brain.com/mx/cursos/json> [Accessed 14 Sep. 2014].
- COBO ROMANÍ, C. and PARDO KUKLINSKI, H. (2007). Planeta Web 2.0. Inteligencia colectiva o medios fast food. Barcelona: Grup de Recerca d'Interaccions Digitals, Universitat de Vic.
- COMPARADOR.IFT.ORG.MX, (2015). Comparador de Tarifas. [online] Available at: <http://comparador.ift.org.mx> [Accessed 11 May 2015].
- DAVIS, S. (2007). GIS for web developers. Raleigh, NC: Pragmatic Bookshelf.
- DELAGDO, M. (2013). Aplicaciones híbridas: presente y futuro del desarrollo móvil. 1st ed. [ebook] Madrid, España: atsistemas, pp.1-3. Available at: [http://atsistemas.com/wp-content/uploads/2013/04/20130429\\_aplicaciones\\_hibridas.pdf](http://atsistemas.com/wp-content/uploads/2013/04/20130429_aplicaciones_hibridas.pdf) [Accessed 26 May 2014].
- DELGADO FERNÁNDEZ, P. and GAMA MORENO, L. (2009). Evolución de las Bases de Datos: de Fijas a Móviles. Zacatepec, Morelos, México. 1st ed. [ebook] Ensenada, Baja California: 1 Universidad Autónoma de Baja California. Available at: <http://campusv.uaem.mx/cicos/imagenes/memorias/7mocicos2009/Articulos/Evolucion%20de%20las%20Bases%20de%20Datos.pdf> [Accessed 18 May 2014].
- DELISLE, M. (2004). Mastering phpMyAdmin for effective MySQL management. Birmingham, UK: Packt Publishing.
- DEVELOPER.ANDROID.COM, (2014). Download Android Studio and SDK Tools | Android Developers. [online] Available at: <http://developer.android.com/sdk/index.html> [Accessed 30 Aug. 2014].
- DOCS.PHONEGAP.COM, (2014). PhoneGap API Documentation. [online] Available at:

- [http://docs.phonegap.com/en/edge/guide\\_platforms\\_index.md.html](http://docs.phonegap.com/en/edge/guide_platforms_index.md.html) [Accessed 11 Aug. 2014].
- E-interactive, (2013). Comportamiento de usuarios en dispositivos móviles. [online] Available at: <http://www.e-interactive.es/wp-content/uploads/Informe-de-Marketing-Movil.pdf> [Accessed 29 Jul. 2014].
  - EROSKI CONSUMER, (2009). Geolocalización en el móvil: Geolocalización: asistente personalizado en el móvil | Revista | EROSKI CONSUMER. [online] Available at: <http://revista.consumer.es/web/es/20091001/internet/75113.php> [Accessed 14 Feb. 2015].
  - Esri.com, (2008). ArcNews Summer 2008 Issue -- GIS and the GeoWeb. [online] Available at: [http://www.esri.com/news/arcnews/summer08articles/gis-and-geoweb\\_sp.html](http://www.esri.com/news/arcnews/summer08articles/gis-and-geoweb_sp.html) [Accessed 17 Jan. 2014].
  - Esri.com, (n.d.). What is GIS?. [online] Available at: <http://www.esri.com/what-is-gis/howgisworks> [Accessed 4 Jun. 2014].
  - europapress.es, (2015). Las aplicaciones híbridas representarán el 60% de las apps en 2015. [online] Available at: <http://www.europapress.es/portaltic/software/noticia-aplicaciones-hibridas-representaran-60-apps-2015-20140109151118.html> [Accessed 6 Apr. 2014].
  - Firtman, M. (2008). Ajax. México, D. F.: Alfaomega.
  - FRAMINGHAM, M. (2013). Android and iOS Combine for 91.1% of the Worldwide Smartphone OS Market in 4Q12 and 87.6% for the Year, According to IDC | Business Wire. [online] Businesswire.com. Available at: <http://www.businesswire.com/news/home/20130214005415/en/Android-iOS-Combine-91.1-Worldwide-Smartphone-OS#.VcogNvmqqko> [Accessed 20 Nov. 2014].
  - Geojson.org, (2014). GeoJSON Specification. [online] Available at: <http://geojson.org/geojson-spec.html> [Accessed 10 Sep. 2014].
  - GitHub, (2014). phayes/geoPHP. [online] Available at: <https://github.com/phayes/geophp> [Accessed 1 Apr. 2014].
  - Google Developers, (2013). Google Maps API licensing. [online] Available at: <https://developers.google.com/maps/licensing?hl=es> [Accessed 31 Oct. 2014].
  - h1lab - simplicity is reliability - desarrollo web & mobile, (2014). H1Lab :: Desarrollo de aplicaciones multiplataforma. [online] Available at: [http://www.h1lab.com/servicio/28/5/desarrollo\\_mobile\\_hibridos\\_web\\_app\\_multiplataforma.html](http://www.h1lab.com/servicio/28/5/desarrollo_mobile_hibridos_web_app_multiplataforma.html) [Accessed 16 Apr. 2014].

- Headwaydigital.com, (2014). Qué es el Geofencing o Geotargeting: geolocalización en móviles -. [online] Available at: <http://www.headwaydigital.com/es/que-es-el-geofencing-o-geotargeting-geolocalizacion-en-moviles/> [Accessed 13 Jan. 2015].
- Hébuterne, S., Pérochon, S. and Piqueres Juan, F. (2014). Android. Cornellà de Llobregat (Barcelona): ENI.
- Hwang, S. and Yu, D. (2012). GPS Localization Improvement of Smartphones Using Built-in Sensors. [online] Available at: <http://connection.ebscohost.com/c/articles/80176321/gps-localization-improvement-smartphones-using-built-sensors> [Accessed 19 Apr. 2014].
- labmexico.com, (2013). Los dispositivos móviles están transformando la manera en la que los mexicanos se relacionan con el mundo | IAB MEXICO. [online] Available at: <http://iabmexico.com/usos-habitos-dispositivos-moviles-2013> [Accessed 18 Aug. 2014].
- IBM, (n.d.). IBM Desarrollo de aplicaciones móviles. [online] Available at: <http://www-03.ibm.com/software/products/es/category/mobile-application-development> [Accessed 18 Jun. 2014].
- jquery.org, j. (2015). jQuery API Documentation. [online] Api.jquery.com. Available at: <http://api.jquery.com/> [Accessed 9 Sep. 2014].
- jquery.org, j. (n.d.). jQuery Mobile API Documentation. [online] Api.jquerymobile.com. Available at: <http://api.jquerymobile.com/> [Accessed 6 Sep. 2014].
- Leafletjs.com, (2014). Documentation - Leaflet - a JavaScript library for interactive maps. [online] Available at: <http://leafletjs.com/reference.html> [Accessed 7 Dec. 2014].
- Levinsohn, A. (2001). LA INTEROPERABILIDAD GEOESPACIAL. [online] Redgeomática.rediris.es. Available at: <http://redgeomática.rediris.es/metadatos/publica/articulo04.htm> [Accessed 11 Nov. 2014].
- López Michelone, M. (2013). La historia de Android. [online] unocero. Available at: <https://www.unocero.com/2013/09/23/la-historia-de-android/> [Accessed 14 Apr. 2014].
- Maclean, M. (2014). Leaflet Tips and Tricks. 1st ed. [ebook] LeanPub, pp.73-80. Available at: <https://leanpub.com/leaflet-tips-and-tricks> [Accessed 18 Aug. 2014].
- Mark Otto, a. (2014). Getting started · Bootstrap. [online] Getbootstrap.com. Available at: <http://getbootstrap.com/getting-started/> [Accessed 19 Nov. 2014].
- Marketing Directo, (2013). Los datos de localización en tiempo real cada vez cobran más importancia en la segmentación móvil - Marketing Directo. [online] Available at: <http://www.marketingdirecto.com/especiales/mobile-marketing->

- blog/los-datos-de-localizacion-en-tiempo-real-cada-vez-cobran-mas-importancia-en-la-segmentacion-movil/ [Accessed 14 Sep. 2014].
- Marqués, A. (2013). Conceptos sobre APIs REST. [online] Asier Marqués. Available at: <http://asiermarques.com/2013/conceptos-sobre-apis-rest/> [Accessed 15 May 2015].
  - Microsoft.com, (n.d.). Microsoft TechNet - Artículos Técnicos. [online] Available at: <https://www.microsoft.com/spain/technet/recursos/articulos/3010023.aspx> [Accessed 11 May 2015].
  - Mitchell, T. (2005). Web mapping illustrated. Beijing: O'Reilly.
  - Nielsen, (2013). THE MOBILE CONSUMER. A GLOBAL SNAPSHOT. [online] Nielsen, pp.7-14. Available at: <http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Mobile-Consumer-Report-2013.pdf> [Accessed 10 Oct. 2014].
  - Piqué, D. (n.d.). Curso PhoneGap. [online] Pixelpro.es. Available at: <http://pixelpro.es/cursos-mobile/aplicaciones-phonegap.html> [Accessed 16 Sep. 2014].
  - Retos en la publicación de contenidos atractivos y de calidad en la web 3.0. (2010). 1st ed. [ebook] ESRI, pp.10-12. Available at: [http://ligit0.uab.es/web/not/act06/Seminari\\_Esri\\_doc.pdf](http://ligit0.uab.es/web/not/act06/Seminari_Esri_doc.pdf) [Accessed 14 Jan. 2015].
  - Rodríguez Benito, E. (2010). La Geolocalización, Coordenadas hacia el Éxito. In: II Congreso Internacional Comunicación 3.0. [online] Salamanca, España: Universidad de Salamanca, pp.1-7. Available at: <http://campus.usal.es/~comunicacion3punto0/comunicaciones/042.pdf> [Accessed 9 May 2014].
  - Shotts, K. (2014). PhoneGap 3.x mobile application development. Birmingham, UK: Packt Pub.
  - Simoes, C. (2015). INUSUAL | REST vs SOAP al servicio de la web. [online] INUSUAL. Available at: <http://inusual.com/articulos/rest-vs-soap-al-servicio-de-la-web/> [Accessed 11 Jul. 2015].
  - Sriparasa, S. (2013). JavaScript and JSON essentials. Birmingham, UK: Packt Publishing.
  - The Denitive Guide to Choosing Your Enterprise Mobile Application Architecture. (2015). 1st ed. [ebook] Atlanta, GA: OutSystems, pp.9-11. Available at: <http://www.outsystems.com/offer/html5-native-hybrid-mobile-architecture/> [Accessed 11 Jun. 2015].
  - Video2Brain, (2015). Qué es un web service. [video] Available at: <https://www.video2brain.com/mx/cursos/web-services-con-php>.