

Diseño de un prototipo de monitoreo vehicular en zona urbana utilizando sistema operativo Android

Designing an urban vehicle monitoring prototype using Android operating system

Wbeymar Carvajal P.
ZTE Colombia

wcarvajalp@correo.udistrital.edu.co

Johann S. Orozco L.
Claro S.A.

jsorozcol@correo.udistrital.edu.co

El prototipo planteado en este artículo busca ayudar a minimizar el congestionamiento de las vías, permitiendo a los usuarios de una empresa ya sea pública o privada, conocer la ubicación de los vehículos pares. Se desarrolló una aplicación móvil en la cual un usuario puede conocer las últimas 20 posiciones de otro, y sabrá si está retrasado o adelantado con respecto a este. Además podrá deducir si hay congestión o no por la distancia que haya entre estos 20 puntos mostrados, ya que la diferencia de tiempo entre estos es de solo un minuto. También tiene la posibilidad de un aplicativo Web donde se puede ver todos los usuarios que han estado conectados y la ubicación que han tenido en todo momento, esto permite tener un registro de la posición de los vehículos y con esto poder generar estrategias que permitan prestar un mejor servicio.

Palabras clave: Android, monitoreo, vehículos

The prototype proposed in this paper aims to help minimize the congestion of the roads, allowing users of a company whether public or private, the location of the pairs vehicles. We developed a mobile application in which a user can to know the latest 20 positions of the others and to know if is delayed or advanced with respect to this. It also may be deducted if there is congestion or not according the distance between the 20 points shown as the time difference between these it is only 1 minute. It also has the possibility of a Web application where you can see all users have been connected and location that have had at all times, this allows to keep track of the position of vehicles and thereby to generate strategies to render a better service.

Keywords: Android, monitoring, vehicles

Introducción

En este artículo se reflejan los procesos desarrollados para el diseño del prototipo de monitoreo vehicular en

zona urbana, el cual fue planteado como una solución al problema de movilidad presente en muchas ciudades del país, donde los conductores de diferentes servicios públicos y privados necesitan conocer la ubicación de otros usuarios (Uppoor, Trullols-Cruces, Fiore, y Barcelo-Ordinas, 2014).

Este prototipo fue planteado para ser desarrollado en sistema operativo Android (Martínez, Barreto, y Wanomen, 2014). Este toma muestras desde el usuario final (cliente del aplicativo Android) para compilarlas en una base de datos que alberga toda la información. Esta información es utilizada para el ingreso a la plataforma y para administrar la ubicación de todos los usuarios conectados. Cuando el usuario hace un ingreso correcto a la plataforma se conecta por medio de tecnología *restfull* al servidor y envía su ubicación al servidor cada determinado tiempo permitiendo a los demás usuarios conocer la ubicación de los otros usuarios conectados (Farfán, Quevedo, y Hernández, 2012).

Fecha recepción del manuscrito: Agosto 29, 2014

Fecha aceptación del manuscrito: Septiembre 22, 2014

Wbeymar Carvajal P., ZTE Colombia; Johann S. Orozco L., Claro S.A.

Esta investigación fue financiada por: Universidad Distrital Francisco José de Caldas.

Correspondencia en relación con el artículo debe ser enviada a: Johann Smit Orozco Larrota. Email: johannorozcol@gmail.com

Este sistema es controlado desde un servidor Web apache con servicios Web restfull, el cual registra los usuario, habilita permisos, y almacena la información en bases de datos, además permite conocer la ubicación de todos los usuarios conectados, ya sea uno a uno o todos al mismo tiempo (Safaripour, Khendek, Glitho, y Belqasmi, 2014).

El sistema fue desarrollado de tal manera que permite a todos los usuarios conocer la ubicación exacta en latitud y longitud a través de la utilización de la aplicación Maps de google, de cualquier usuario conectado a la aplicación cliente en el momento de hacer la consulta al servidor (Meng, Zhao, y Xu, 2012).

Metodología

Dentro de la metodología utilizada para el desarrollo del proyecto se optó por la utilización de herramientas de desarrollo libre, como lo son Netbeans (Java) y el aplicativo XAMPP (que permite la descarga de servidor Apache, de MySQL, FileZilla, Mercury), y Eclipse que es otra plataforma de desarrollo Java para el diseño de aplicaciones en el sistema operativo móvil Android.

Debido a la gran problemática vehicular que se vive en las grandes y pequeñas ciudades principales del mundo se requiere diseñar un prototipo de monitoreo vehicular que permita mantener un informe casi en tiempo real que mantenga informada una central de cuales son los recorridos más convenientes para rutas particulares ya sean escolares o de empresas (Li et al., 2014).

Este tipo de información se compila en una base de datos controlada desde un aplicativo Web, el cual permitirá no solo controlar la información generada por el aplicativo cliente (aplicativo de Android instalado en cada equipo y asignado a un vehículo), si no poder generar usuario y perfiles para estos usuario a fin de llevar un consolidado con los datos recibidos como lo son: posición, velocidad de recorrido, entre otros.

La problemática de nuestra ciudad, así como la de muchas ciudades no sólo de nuestro país sino de muchas partes del mundo, es poder generar un control en tránsito ciudadano, al menos en el tránsito no particular, (servicio público o rutas vehiculares). Estos vehículos son los mayores culpables del mal rendimiento de tráfico vehicular. En nuestra ciudad Bogotá, se han creado puestos de trabajo informales encargados de medir los tiempos de retraso entre un vehículo y otro en el caso de los buses de servicio publico, lo cual a ayudado a tener un mejor control de la ubicación de los vehículos en rutas y la diferencia de tiempos entre ellas. No solo buses de servicio publico necesitan conocer la ubicación de sus demás compañeros, esta necesidad también se ve en empresas dedicadas al reparto de mercancía o al transporte de personal de empresas.

Esto ha ayudado, pero al no ser un trabajo formalizado se ve opacado por no tener una correcta organización ni funcionamiento. Además, al ser informal cualquier persona

se ubica en un punto y lo hace, no genera un reporte preciso de la ubicación y se da una sobre demanda del servicio. El objetivo primordial es optimizar el trabajo de las personas que están ubicadas en ciertos puntos de la ciudad utilizando aplicativos Android y WEB encargados de dar un reporte más preciso del movimiento de todos los vehículos en tránsito, y que se pueda generar una consulta, primero más precisa, y segundo más general del trayecto de todos estos vehículos en tránsito, consiguiendo con esto optimizar el trabajo y mejorando el proceso de reportes de ubicación entre vehículos. Esto no solo mejora la movilidad en la ciudad sino que permite el mejor manejo de la ubicación, mejorando los ingresos para conductores de servicio público y privado y permitiendo prestar un servicio de mejor calidad a los usuarios.

El desarrollo de la herramienta sigue los principios básicos definidos en ingeniería de software, en particular el modelo en cascada o ciclo de vida del software (Fig. 1) (Ramone, 2012).

1. Planificación: Realiza un estudio de factibilidad del software así como contemplar los posibles costos que pueden surgir mediante su implementación.

2. Análisis y diseño de requerimientos: Involucra la identificación de las características que guían para determinar las funcionalidades del software de acuerdo al medio donde se pretende implementar. Es muy importante notar que trata de responder a las preguntas ¿Quiénes intervienen en el uso del Software?, ¿Qué restricciones tendrá el software?

3. Diseño: Se identifica y describe las abstracciones del software y cumplir con los requerimientos plasmando todas esas características en un diseño que permite visualizar y contemplar adicionalmente situaciones no previstas.

4. Implementación: Realizar las pruebas pertinentes y verificar que se cumplen con las características identificadas.

5. Operación y mantenimiento: Se instala dentro del ambiente, dependerá que pasará a partir de ahí, ya que esta etapa aún puede considerar nuevamente la existencia de características que no han sido contempladas y/o características innecesarias, implicando la modificación del software para la adaptación de estas anomalías.

6. Crecimiento y cambio: Se evalúa el software de modo que se determina si se puede emplear dentro de la nueva tecnología no afectando la integridad del mismo, de modo que si no es posible que exista una adaptación a lo nuevo, el proceso de diseño del software nuevamente se repite desde el principio

Resultados

La Fig. 2 detalla el diagrama estructural del software.

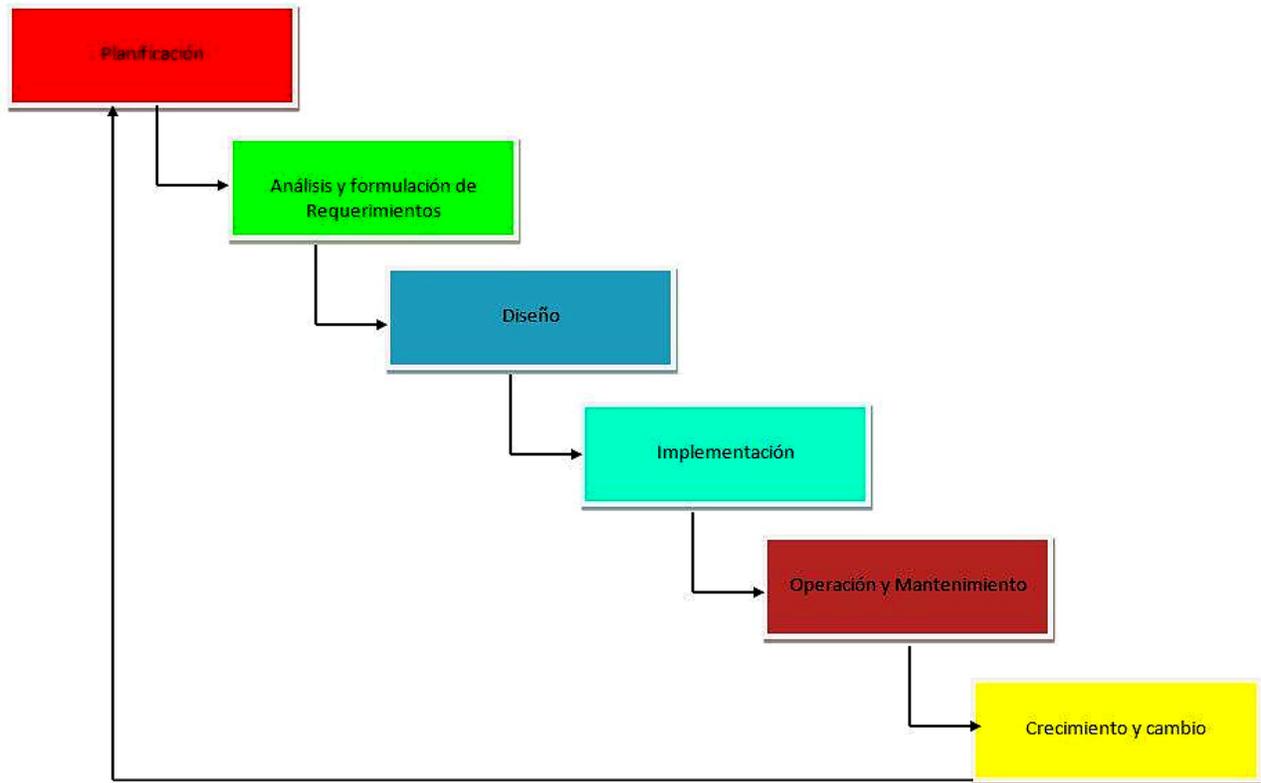


Figura 1. Modelo de cascada (Ramone, 2012).

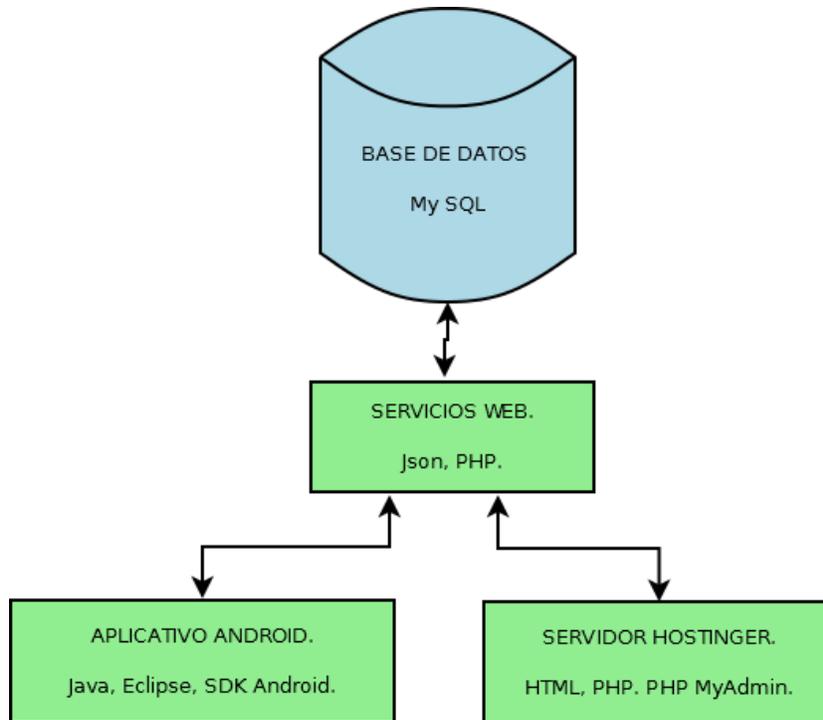


Figura 2. Diagrama de software.

Servicios Web

Los servicios Web utilizados están montados sobre tecnología JSON (Javascript Object Notation). Este formato utilizado por plataformas de desarrollo Java. Es el más simple y ligero para intercambio de datos y no requiere el uso de XML.

Se optó por la utilización de este tipo de comunicación entre cliente y servidor por:

- Siendo un estándar que se encuentra integrado dentro de las plataformas de desarrollo Java, no es necesario importar librerías para su utilización.
- El flujo de datos entre cliente y servidor se limita solo a la información útil. Es decir, no utiliza ningún tipo de protocolo para el envío. Aunque esto puede verse mal para la seguridad, dado el tipo de aplicación donde la información que se envía no tiene ningún nivel de confidencialidad es más que suficiente.
- La sencillez de su código e implementación, tanto en la plataforma Netbeans para el aplicativo WEB, como en Eclipse para el aplicativo Android.

Acceso a servicios Web desde Tracking Location

Dentro del aplicativo Android desarrollado en la plataforma Eclipse se utilizó JSON para establecer la comunicación con el servidor en diferentes momentos de uso del App.

Servicios en plataforma Web

En el aplicativo WEB se generó un único archivo PHP que contiene todos los métodos asociados a los servicios WEB sobre JSON.

Diagrama de metodología de acceso

En el prototipo existen dos tipos de usuario: un usuario cliente y un usuario Admin. Cada uno con permisos de acceso diferentes. En las Fig. 3 y Fig. 4 se muestran los diagramas de la metodología de acceso de cada usuario tanto al aplicativo Web, como al aplicativo Android.

Aplicativo Android (Tracking Location)

El aplicativo Android creado para el proyecto, llamado TRACKING LOCATION, permite la localización de otros usuarios a través de GPS y el uso de un servidor.

Cuando se selecciona un usuario, y se hace una conexión correcta con el servidor, este mostrara las últimas 20 posiciones almacenadas en la base de datos (Fig. 5).

Estando dentro de Maps, se tiene la opción de ver cual posición es (enumeradas del 1 al 20), que día y a que horas fue registrada, simplemente parándose sobre cada uno de los indicadores (Fig. 6). Al retroceder se volverá al listado de usuarios conectados, y se podrá seleccionar otro para conocer

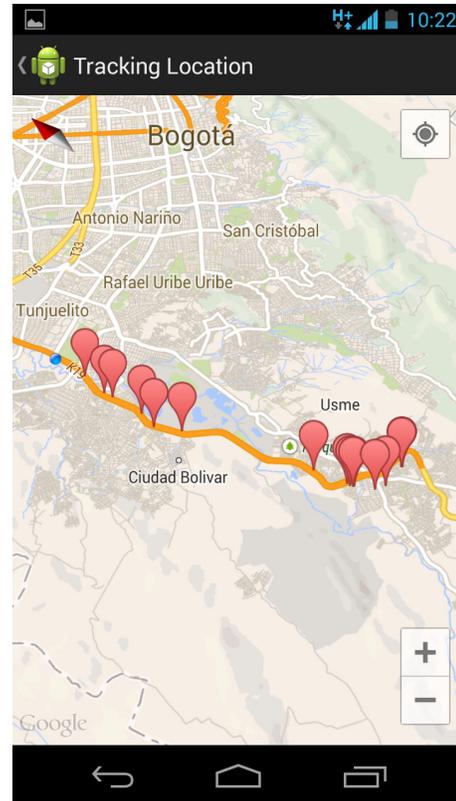


Figura 5. Dibujo de las últimas 20 posiciones.

su ubicación. La Fig. 7 muestra un diagrama de todas las actividades programadas dentro de la aplicación, y el enlace entre cada una de ellas.

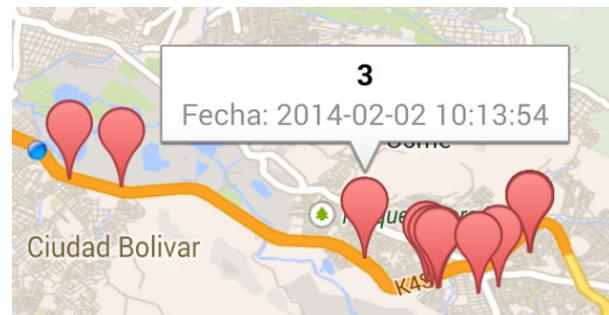


Figura 6. Información de cada indicador.

Aplicación Web

Para el diseño de la aplicación Web se utilizó la plataforma de desarrollo Netbeans 7.3, la cual permite la programación en HTML5 y CSS3. Adicional a esto se utilizó el aplicativo XAMPP que contiene el servidor Apache y MySQL, que son los de interés. Además, para los formularios y el registro de usuarios se manejo el lenguaje PHP que va ligado al HTML, pero que a comparación de este no se

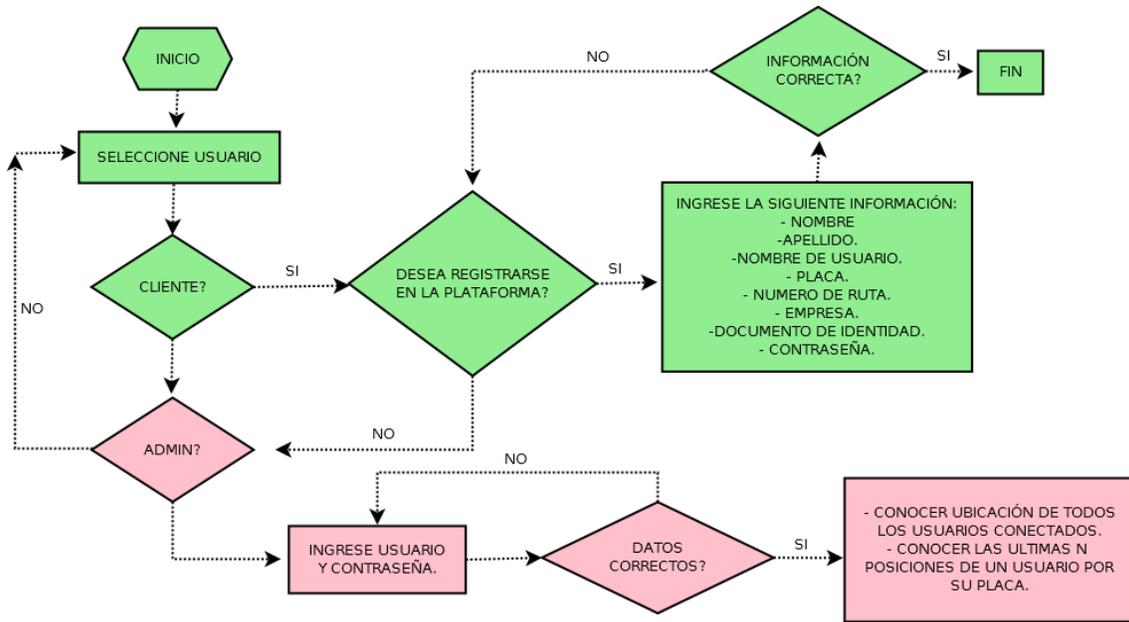


Figura 3. Diagrama de acceso aplicativo WEB.

resuelve por el navegador sino por el servidor Apache que proporciona el XAMPP. En la Fig. 8 se muestra el diagrama de la arquitectura del prototipo de monitoreo vehicular desarrollado.

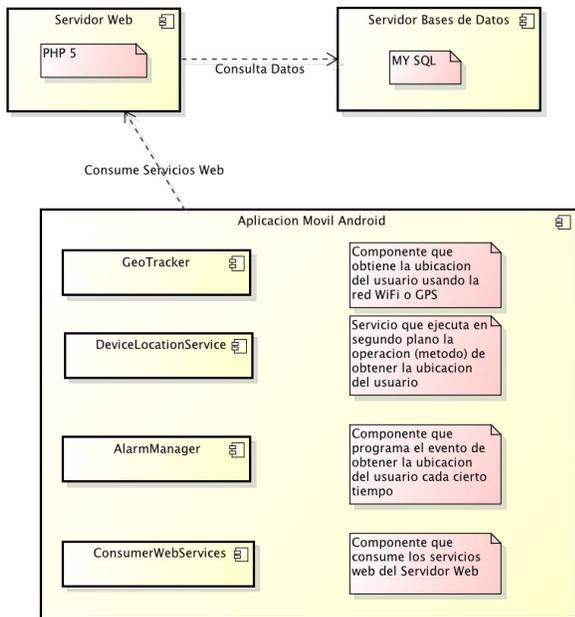


Figura 8. Arquitectura.

Pruebas de estrés

Para corroborar que la página soporte cierta carga de usuarios y baja latencia de usuarios, se realizó una prueba

de estrés utilizando la página <https://loadimpact.com>, la cual permite simular varios usuarios simultáneos utilizando la página en un tiempo determinado. Nuestro servidor se encuentra en USA, y la prueba fue realizada desde un servidor encontrado en Singapur (Fig. 9).

La prueba consistió en agregar 50 usuarios gradualmente durante un periodo de cinco minutos. Al llegar al límite de 50 usuarios se detiene la prueba. La Fig. 10 muestra los tiempos de carga de los usuarios y la cantidad de usuarios en el momento.

Se puede observar que el tiempo de carga de los usuarios varía entre uno y tres segundos, con un comportamiento constante sin importar el número de usuarios conectados. El test también muestra una tabla donde se detallan los tiempos de cargo y el número total de peticiones de la página inicial y de cada uno de los elementos que componen la página (Fig. 11).

Por ultimo, el sistema también entrega unas gráficas (Fig. 12) con los tipos de contenidos y sus tiempos de carga.

Se realizó el mismo test con la URL <http://monitoreocar.esy.es/api/funtions.php>, la cual contiene los servicios, es decir, es el lugar donde se realiza toda la interacción en segundo plano (Fig. 13 y Fig. 14).

Se puede observar un tiempo promedio de respuesta de 268,45 ms. Además, en la gráfica se puede apreciar como el servidor empieza a tardar más tiempo en responder después del usuario 30.

Conclusiones

El prototipo de monitoreo vehicular desarrollado puede ser utilizado por cualquier empresa publica o privada

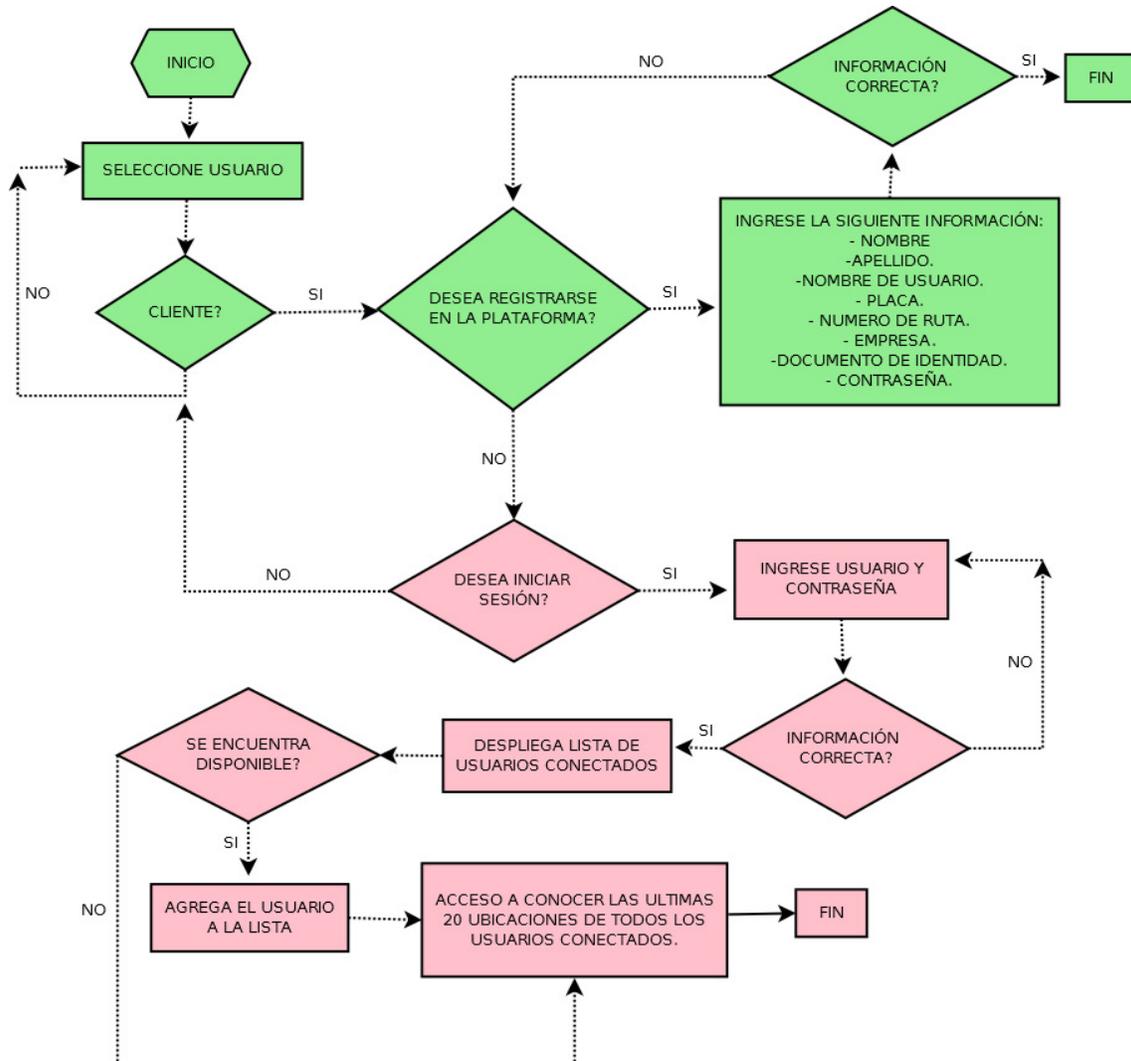


Figura 4. Diagrama de acceso aplicativo Android.

que necesite mantener un monitoreo constante sobre sus vehículos, conociendo su ubicación actual, y la que ha tenido en todo momento. Se puede inferir con el uso de la aplicación en ciertas situaciones, los eventos que están ocurriendo o que ocurrieron viendo las distancias entre los diferentes indicadores de posición de los usuarios.

El prototipo mostró un comportamiento dentro de lo esperado teniendo en cuenta las condiciones de red presentes en diferentes zonas de la ciudad. Este tipo de eventos fueron tenidos en cuenta en el momento del desarrollo y por tanto el aplicativo sigue funcionando sin mostrar errores, hasta que logra conectarse de nuevo a un red.

La utilización de este tipo de aplicaciones móviles se está expandiendo, primero por la gran cantidad de usuarios de teléfonos inteligentes y segundo por la gran utilidad que dan al cliente. Por estos motivos, el mercado se está enfocando

solo en desarrollar este tipo de dispositivos y aplicaciones para los mismos.

Finalmente se puede concluir, que el uso de este tipo de aplicaciones de movilidad ayudan a mejorar la prestación de servicio hacia los usuarios, ya que permiten conocer eventos viales y con esto realizar acciones que permitan minimizarlos.

Referencias

- Farfán, D., Quevedo, M., y Hernández, C. (2012). Prototipo de una plataforma web de comunicación virtual. *Tekhnê*, 9(1), 13-20. (ISSN 1692-8407)
- Li, Y., Ren, W., Jin, D., Hui, P., Zeng, L., y Wu, D. (2014). Potential predictability of vehicular staying time for large-scale urban environment. *IEEE Transactions on Vehicular Technology*, 63(1), 322-333. (ISSN 0018-9545)

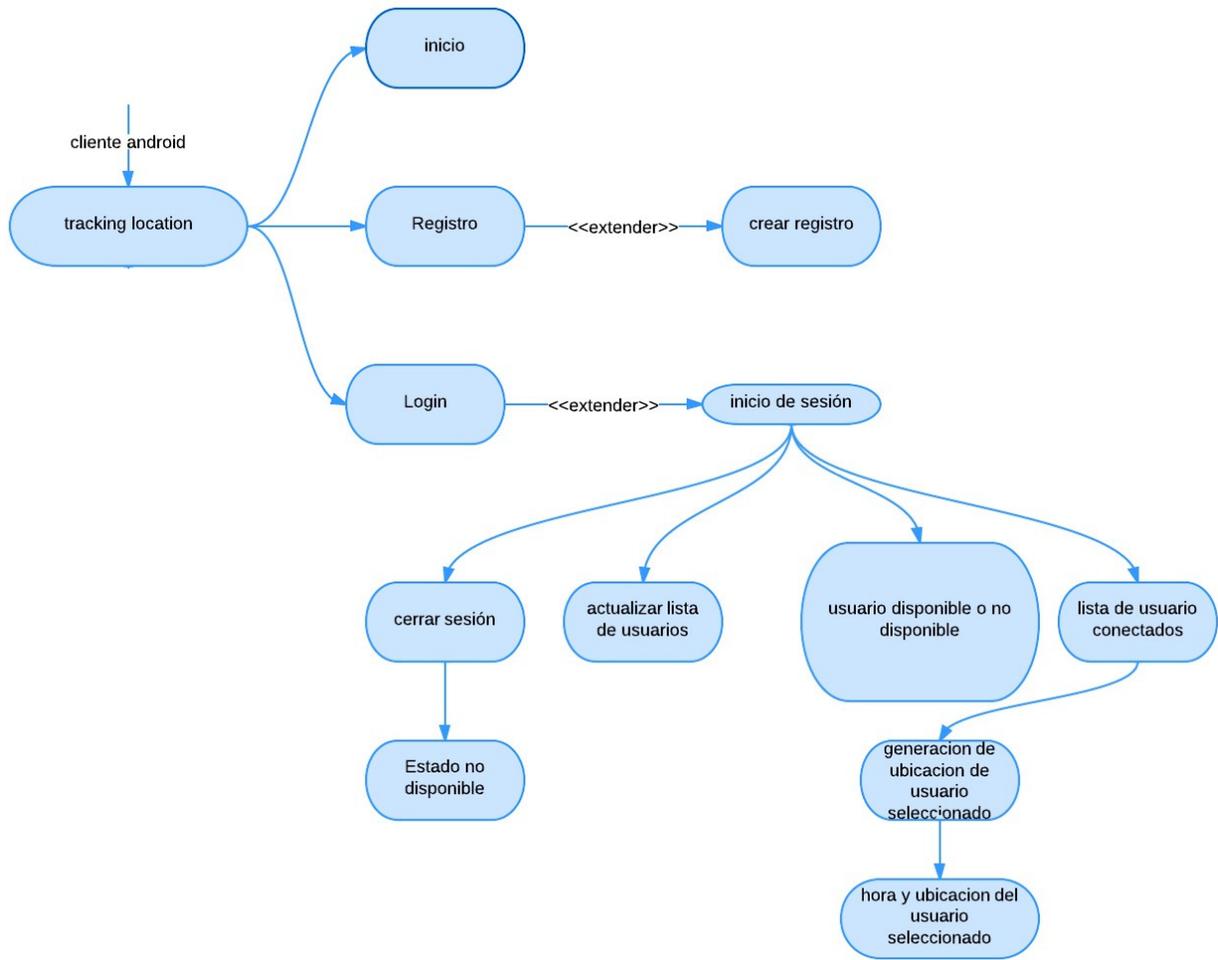


Figura 7. Diagrama actividades aplicación.

Martínez, L., Barreto, M., y Wanomen, L. (2014). Aplicativo para ubicación de sitios turísticos en bogotá a través de dispositivos tablet (samsung galaxy tab 10.1, touchpad y xyboard 10.1) implementando realidad aumentada y geolocalización por proximidad. *Tekhnê*, 11(1), 5-8. (ISSN 1692-8407)

Meng, J., Zhao, H., y Xu, Y. (2012). Power failure monitoring and locating system based on google maps. En *2012 fourth international conference on computational and information sciences (iccis)* (p. 337-340).

Ramone, M. (2012, Febrero). *Ingeniería del software. modelo en cascada*. Retrieved from [http://](http://ingenexescom.blogspot.com/2012/02/modelo-en-cascada.html)

ingenexescom.blogspot.com/2012/02/modelo-en-cascada.html

Safaripour, R., Khendek, F., Gliho, R., y Belqasmi, F. (2014). A restfull architecture for enabling rapid development and deployment of companion robot applications. En *2014 international conference on computing, networking and communications (icnc)*.

Uppoor, S., Trullols-Cruces, O., Fiore, M., y Barcelo-Ordinas, J. (2014). Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5), 1061-1075.

Load test: Auto-generated (lunes, 10 de febrero de 2014 11:08:25 p. m.)

Target URL: <http://monitoreocar.esy.es/> Test result public URL: <http://loadimpact.com/load-test/monitoreocar.esy.es-8584da>

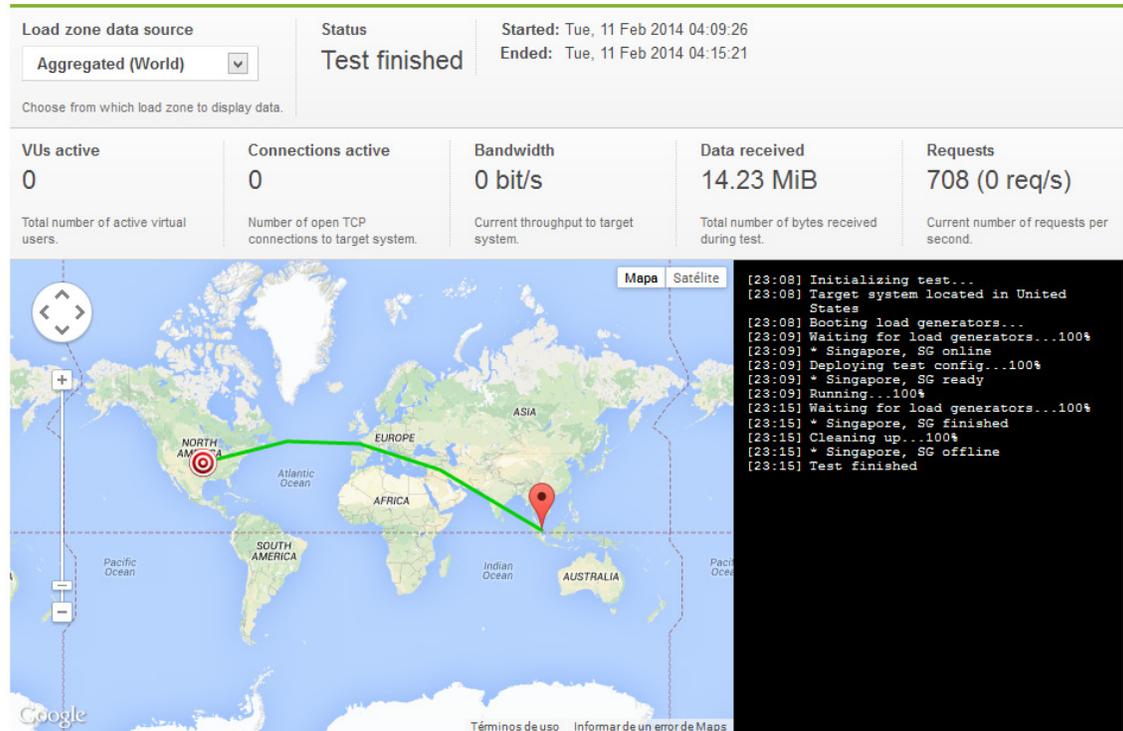


Figura 9. Test Loadimpact.

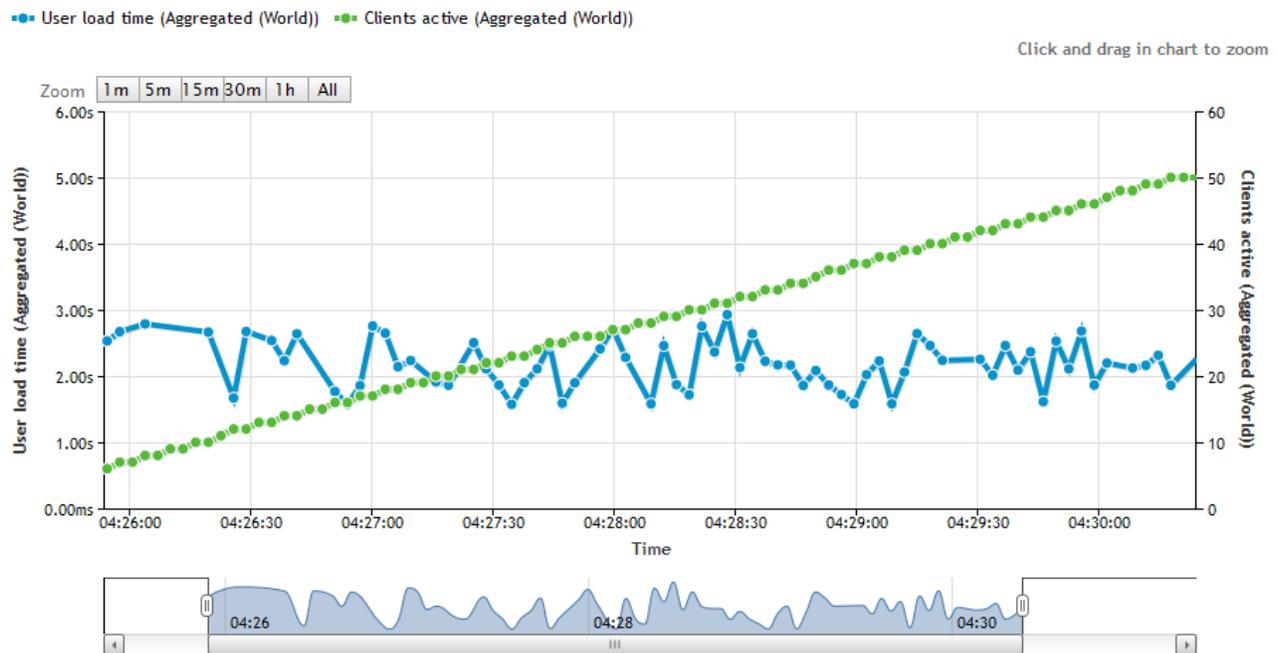


Figura 10. Gráfica tiempo de carga de usuario vs usuarios conectados.

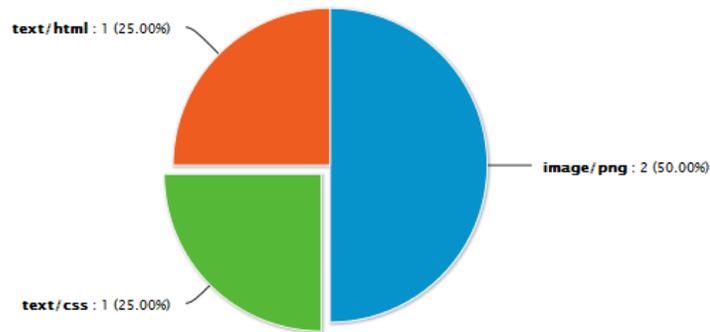
Show URLs

URL	Load zone	User scenario	Successful	Failed	Last avg
http://monitoreocar.esy.es/	Aggregated (World)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	173	16	0.00ms
http://monitoreocar.esy.es/	Tokyo, JP (Amazon)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	173	16	0.00ms
http://monitoreocar.esy.es/bg.png	Aggregated (World)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	163	0	1.00s
http://monitoreocar.esy.es/bg.png	Tokyo, JP (Amazon)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	163	0	1.00s
http://monitoreocar.esy.es/div.png	Aggregated (World)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	163	0	193.15ms
http://monitoreocar.esy.es/div.png	Tokyo, JP (Amazon)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	163	0	193.15ms
http://monitoreocar.esy.es/monitoreo.css	Aggregated (World)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	170	0	1.6m
http://monitoreocar.esy.es/monitoreo.css	Tokyo, JP (Amazon)	Auto-generated (martes, 11 de febrero de 2014 11:24:43 p. m.)	170	0	1.6m

Figura 11. Tabla de tiempos de carga promedio de los usuarios.

URL content type distribution

Click on slice to highlight rows in URL table



URL content type load time distribution

Click on slice to highlight rows in URL table

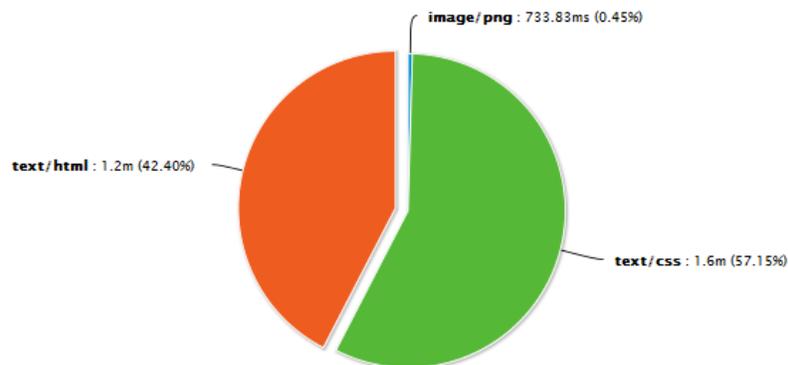


Figura 12. Gráficas de contenidos.

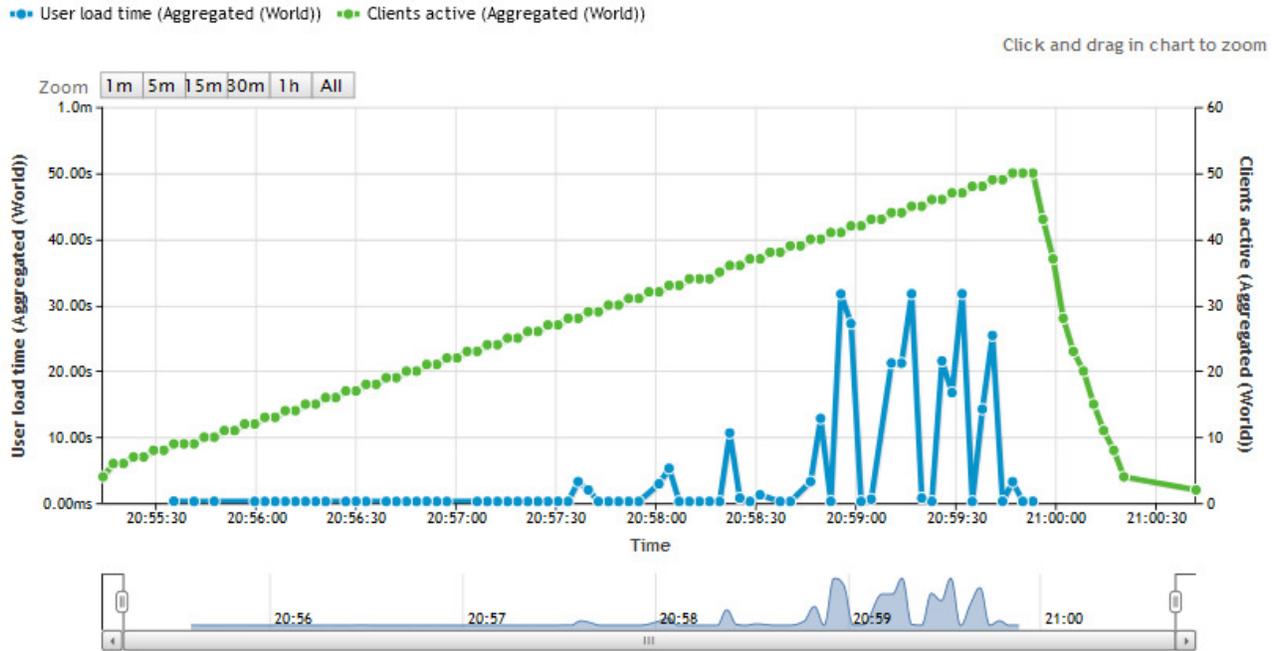


Figura 13. Gráfica tiempo de carga de usuario vs usuarios conectados test 2.

Show 10 URLs

URL	Load zone	User scenario	Successful	Failed	Last avg
http://monitoreocar...es/api/functions.php	Aggregated (World)	Auto-generated (miércoles, 12 de febrero de 2014 8:53:01 p. m.)	173	20	268.45ms
http://monitoreocar...es/api/functions.php	Ashburn, US (Amazon)	Auto-generated (miércoles, 12 de febrero de 2014 8:53:01 p. m.)	173	20	268.45ms

Figura 14. Tabla de tiempos de carga promedio de los usuarios test 2.