



ARTÍCULO DE INVESTIGACIÓN / RESEARCH ARTICLE

Estudio de QoS (Quality of Service) y QoE (Quality of Experience) para servicios web consumidos desde un mashup móvil

*QoS (Quality of Service) and QoE (Quality
of Experience) Study for Web Services
Consumed from a Mobile Mashup*

JUAN GABRIEL ENRIQUEZ *
SANDRA ISABEL CASAS **

*Docente Asistente. Magister en Informática y Sistemas, Universidad Nacional de la Patagonia Austral – Unidad Académica Río Gallegos – Instituto de Tecnología Aplicada - Grupo de Ingeniería de Software Pragmática (ITA-GISP) .
jenriquez@uarg.unpa.edu.ar. Orcid: 0000-0002-0699-6924

**Docente Investigador Asociado. Doctora en Sistemas e Ingeniería de Software, Universidad Nacional de la Patagonia Austral – Unidad Académica Río Gallegos -- Instituto de Tecnología Aplicada - Grupo de Ingeniería de Software Pragmática (ITA-GISP).
sicasas@uarg.unpa.edu.ar. Orcid: 0000-0002-8289-6132

Correspondencia: Enriquez Juan Gabriel, teléfono: 549 2966 442376, jenriquez@uarg.unpa.edu.ar. Lisandro de la Torre 860, Río Gallegos, Santa Cruz, Argentina.

Origen de subvenciones o apoyos recibidos: Los resultados forman parte del proyecto de investigación: “Estrategias para el desarrollo de web mashup”, código: 29 A 449.



Resumen

El objetivo de este trabajo consistió en el estudio de la QoS (por sus siglas en inglés) y la QoE (por sus siglas en inglés) de API (por sus siglas en inglés) de servicios web consumidos por aplicaciones *mash-up* móvil. El estudio se enfoca en recolectar y analizar indicadores de QoS y de QoE. La metodología usada para el desarrollo de este trabajo es experimental, puesto que se ha realizado una experiencia que permitió recolectar los datos del estudio. Para tal fin, se desarrolló un *mash-up* móvil que consume tres API REST (por sus siglas en inglés) y una librería que permite recolectar indicadores de QoS y de QoE. El estudio se realizó durante 60 días y se recolectaron 1.012 valores de indicadores. Los resultados obtenidos prueban que realizar un análisis parcial de QoS y de QoE que consideren indicadores de manera aislada puede llevar a conclusiones inexactas, por lo que se debe realizar un análisis y evaluación más integral, en atención a la mayor cantidad de factores que afectan la QoS y la QoE.

Palabras clave: mash-up móvil, QoE, QoS, servicio web.

Abstract

The objective of this work is to study the QoS and QoE of web services APIs consumed by mobile mashup applications. The study focuses on collecting and analyzing QoS and QoE indicators. The methodology used for the development of this work is experimental: an experience has been carried out that allowed for collecting the study data. For this purpose, a mobile mashup was developed that consumes three REST APIs and a library that allows collecting QoS and QoE indicators. The study was carried out for 60 days, and 1.012 indicator values were collected. The obtained results prove that performing a partial analysis of QoS and QoE, considering indicators in isolation, can lead to inaccurate conclusions, a more comprehensive analysis and evaluation should be carried out, taking into account the greater number of factors that affect QoS and QoE.

Keywords: Mobile Mashup, QoE, QoS, Web service.

1. INTRODUCCIÓN

Los dispositivos móviles se vuelven cada vez más populares y potentes debido a múltiples factores, entre ellos la capacidad de ejecutar aplicaciones y contenidos móviles de todo tipo (mensajería/chat, entretenimiento, redes sociales, comercio, banca móvil, servicios sociales y públicos, ocio, multimedia, etc.) y a los avances de las tecnologías de redes inalámbricas de banda ancha. Sin embargo, las aplicaciones móviles o servicios móviles están expuestos a diversos y múltiples factores que afectan no solo su desempeño (calidad de servicio, *quality of service* [QoS]), sino también la satisfacción del usuario final (calidad de experiencia, *quality of experience* [QoE]) [1], [2], [3].

La QoS se ha convertido en un tema importante tanto para el proveedor de servicios como para los usuarios. Los proveedores de servicios están preocupados por los parámetros de QoS en red, como el rendimiento, la tasa de pérdida, la fluctuación o el retraso para medir el rendimiento del servicio. QoS es un término técnico que depende de los factores del nivel de la red. Por otro lado, el rendimiento del servicio por parte del usuario es un término más subjetivo; el servicio debe prestarse en un plazo razonable de tiempo. La percepción subjetiva del usuario generalmente se denomina QoE [4]. En particular, en América Latina y el Caribe, los informes de economía móvil como [5], [6] reconocen el problema de la calidad (QoS) y su percepción en los usuarios (QoE) como una barrera para el acceso digital móvil y como una propiedad que a los usuarios les importa más que el precio.

Los *mash-up* móviles resultan un interesante caso de estudio para analizar. A pesar de ser aplicaciones simples que reúsan, integran y combinan diversos componentes externos como servicios web, por ejemplo, a partir del consumo de API (por sus siglas en inglés) web, pueden ser afectados por los rendimientos de QoS y de QoE deficientes. Un componente de *mash-up* es cualquier segmento de datos, lógica de aplicación o interfaz de usuario que puede ser reutilizado y que es accesible local o remotamente [7]. La selección de componentes *mash-ups* es el proceso de evaluación y elección de componentes, se realiza en etapas tempranas del desarrollo de la aplicación e influye en el éxito del *mash-up* (los componentes deben cumplir con los objetivos del *mash-up* y atraer a los usuarios finales). El proceso de selección implica asignar requisitos del *mash-up* a componentes, seleccionar componentes candidatos y evaluar la calidad de estos [8].

Los componentes *mash-up* pueden ser de diferentes tipos, y en este trabajo se consideran los componentes del tipo API de servicio web. Los proveedores de API, que van desde desarrolladores individuales hasta las compañías más importantes de la industria (Facebook, Google, etc.), integran contenido y funcionalidad en un componente que expone una interfaz de programación de aplicaciones (API) y es accesible con protocolos web como REST (por sus siglas en inglés) y SOAP (por sus siglas en inglés) [8].

La selección de API es difícil para los desarrolladores, dado que su oferta con funcionalidades y prestaciones similares es realmente amplia, como se puede constatar al recorrer los catálogos web API Harmony, PublicAPI o ProgrammableWeb [9], [10], [11]; solo ProgrammableWeb lista más de 23.721 API públicas, debido a que las API web se convirtieron en la columna vertebral de la web, la nube y las aplicaciones móviles. A medida que aumenta la oferta de nuevas API web, el proceso de selección, evaluación y monitorización de los componentes *mash-up* necesario para asegurar la calidad deseada se vuelve más complejo y artesanal. En este sentido, la elección de los servicios también debe considerar la QoS y la QoE, en atención a propiedades como tiempo de ejecución o tiempo de respuesta del servicio.

Los estudios de investigación se han dedicado a comprender, medir y modelar la QoS y la QoE para los tipos de servicios más populares (VoIP, *streaming*, servicios web) [12], [13], [14], [15], [16], [17]. En el caso de los servicios web, aún son escasos los estudios que analicen la QoS y la QoE, que se han centrado en los navegadores y no tanto en aplicaciones nativas. Además, por lo general, hacen énfasis en el funcionamiento de la API y no en los factores de QoS que pueden afectar dicho funcionamiento desde el punto de vista del usuario final. Entre esos factores que influyen en la calidad de la API, se pueden mencionar disponibilidad, latencia, retardo y pérdida de paquetes de la red.

Realizar el análisis de la QoS y la QoE desde una aplicación móvil o *mash-up* móvil ofrece ciertas ventajas relacionadas con el entorno real de uso. Las aplicaciones que se ejecutan en dispositivos cliente tienen potencialmente acceso a las capacidades del dispositivo, información de contexto, comportamiento de uso del usuario y métricas en la aplicación [18]. Disponer de estos datos permitiría realizar análisis e interpretaciones más integrales y útiles.

El objetivo de este trabajo consiste en el estudio de la QoS y la QoE de API de servicios web consumidos por aplicaciones *mash-up* implementadas en Android. Para tal fin, se ha desarrollado un *mash-up* móvil que consume tres API REST (por sus siglas en inglés) y una librería que permite recolectar indicadores de QoS y de QoE.

El estudio se realizó durante 60 días, se recolectaron 1.012 valores de indicadores y se enfocó en recolectar y analizar, por un lado, datos sobre la QoS en capa de red y, por otro, se desea obtener datos que afecten la calidad del servicio percibida por el usuario final (QoE). Estos indicadores recolectados ayudan a los desarrolladores de *mash-up* móviles en el proceso de selección o evaluación de componentes *mash-up*. Los resultados obtenidos nos indican que realizar un análisis parcial de QoS y de QoE considerando indicadores de manera aislada puede llevar a conclusiones inexactas, por lo que se debe realizar un análisis y evaluación más integral, en atención a la mayor cantidad de factores que afectan la QoS y la QoE.

Este artículo está organizado de la siguiente manera. La sección dos describe los trabajos relacionados con el tema de investigación, la sección tres presenta los materiales y métodos utilizados para desarrollar el estudio, la sección cuatro muestra los resultados obtenidos, la sección cinco incluye una breve discusión de los resultados y la sección seis detalla las conclusiones derivadas del estudio propuesto.

2. TRABAJOS RELACIONADOS

Se revisan enfoques relacionados con diferentes aspectos de nuestra propuesta. Primero, se proporciona un detalle de nuestro trabajo previo sobre herramientas y enfoques para recolectar datos a fin de analizar o evaluar diversos aspectos de la calidad de aplicaciones nativas móviles. Luego, revisamos otras propuestas y enfoques en relación con la recolección y evaluación de QoS o de QoE móvil.

Nuestro Grupo de I+D en Ingeniería de Software Pragmática (GISP) presentó dos herramientas de soporte relacionadas con la gestión de calidad en aplicaciones móviles. La primera se llama Q2M [19] que es una librería para computar métricas de QoS y de QoE en aplicaciones Android y la segunda es Nexo [20] que es una herramienta para la visualización de indicadores de QoS y de QoE cuya finalidad es facilitar el análisis de las relaciones que puedan existir entre dichos indicadores. Las herramientas desarrolladas se encuentran disponibles en el repositorio <https://github.com/gispunpauarg>.

Por otro lado, el objetivo en [21] es establecer una relación entre QoS y QoE en un servicio web mediante pruebas de laboratorio desde el dispositivo móvil utilizando la conexión inalámbrica LAN (por sus siglas en inglés). En este entorno de laboratorio, existen bajas posibilidades de tener, por ejemplo, demoras, fluctuaciones o ruido en la red. Para las pruebas, el usuario móvil solicita datos al servicio Google Maps a través de un servidor Apache local, el cual es el encargado de medir la QoS (retraso, fluctuación y fiabilidad). Por otro lado, el usuario utiliza el servicio y luego lo califica manualmente respondiendo un cuestionario MOS (por sus siglas en inglés). La conclusión es que la QoS tiene relación directa con la QoE y que el diseño móvil puede afectar al usuario que puede percibir una mejor respuesta.

En [22] se describe un enfoque para usar la QoE como criterio para la selección de un servicio web, que incluye un análisis de los diferentes factores que influyen la calidad percibida y una metodología para medir la QoE y establecer una correlación entre la QoS y la QoE. Se definen los atributos QoS que describen el rendimiento de un servicio web y cómo se pueden medir (duración de ejecución, disponibilidad, fiabilidad, costo de ejecución y reputación), luego se especifica un conjunto de pruebas basadas en combinaciones de las variables de control (atributos QoS) y se realiza la evaluación utilizando un cuestionario MOS. Los resultados del trabajo muestran cómo el tiempo de ejecución

del servicio web, la disponibilidad y la confiabilidad afectan la calidad percibida por el usuario final e indica que tanto la disponibilidad como la confiabilidad tienen mayor influencia en la QoE que el tiempo de ejecución del servicio.

El trabajo realizado en [23] propone un *framework* que se puede utilizar para analizar y evaluar la QoE sobre requisitos para servicios y aplicaciones multimedia en un entorno móvil para la plataforma Android. El *framework* es centrado en el usuario y sensible al contexto y permite medir la QoE en teléfonos inteligentes con una alta precisión y la consideración de múltiples parámetros sobre el usuario, la red y el sistema. Los datos recolectados pertenecen a las siguientes categorías: información relacionada con el usuario (cuestionario), información relacionada con la aplicación (parámetros del video, errores), información relacionada con el dispositivo móvil (uso del CPU —por sus siglas en inglés—, memoria, ubicación) e información relacionada con la red (retraso, fluctuación, tipo de red).

En [13] se propone un modelo de evaluación de QoS/QoE dirigida por un algoritmo genético basado en SA (*annealing base*) para la selección de servicios web. Primero, se presenta un modelo de cálculo de QoS con los siguientes indicadores: costo del servicio, tiempo de ejecución, disponibilidad y fiabilidad. Para unificar la dimensión y dirección de los indicadores, el trabajo define un procesamiento estandarizado para la QoS antes de calcular la calidad integral mediante una función F de QoS global. Segundo, para expresar la percepción subjetiva de los clientes, el indicador QoE puede ser calificado en cinco grados (excelente, bueno, promedio, malo, terrible). Por último, basándose en el modelo, un algoritmo genético para selección de servicios web es dado para mejorar la eficiencia en la selección de servicio web.

En [12] se examina el servicio Periscope. El trabajo estudia los protocolos utilizados y la calidad típica de los indicadores de QoE, tales como latencia de reproducción, calidad de video y consumo de energía de la aplicación de Android. Periscope es un servicio de Twitter que permite mediante su API transmitir video en vivo a una gran cantidad de espectadores usando un dispositivo móvil. Entre el dispositivo móvil y el servicio Periscope, se instala un *proxy* transparente. El *proxy* intercepta las solicitudes HTTPS (por sus siglas en inglés) y permite examinar y registrar el intercambio de solicitudes y respuestas entre el cliente y el servicio Periscope.

El objetivo del trabajo en [24] es presentar un enfoque y un conjunto de herramientas para comparar la calidad según el rendimiento y la disponibilidad de las API web en consideración a la movilidad geográfica de los clientes. El kit de herramientas de medición propuesto está parametrizado con una lista de puntos finales de web API. Basados en esa lista, envía periódicamente solicitudes de *ping*, HTTP (por sus siglas en inglés) y HTTPS. Luego, se registran los resultados detallados que se analizan cuando se completa la ejecución de las solicitudes enviadas.

3. MATERIALES Y MÉTODOS

El estudio consistió en utilizar una aplicación durante mayo y junio de 2020, la cual se desarrolló específicamente para este estudio. En esta, se integró una librería desarrollada para recolectar indicadores de QoS y de QoE. Estos se recogieron y calcularon en el dispositivo móvil del usuario cada vez que se ejecutó la aplicación. Esta, junto con la librería, se encuentra disponible en <https://github.com/gispunpauarg/QoSQoE-MashupMovil>. A continuación, se presentan la aplicación, los detalles de la implementación de la librería y el diagrama de funcionamiento en su conjunto.

Aplicación

Para describir el enfoque del trabajo, se desarrolló una aplicación *mash-up* móvil simple para la plataforma Android, cuya interfaz se muestra en la figura 1. La aplicación permite realizar búsquedas de puntos de interés (POI, por sus siglas en inglés) según un término de búsqueda en el área donde se encuentra el usuario según las coordenadas GPS (por sus siglas en inglés) del dispositivo móvil. Una vez realizada la búsqueda, los POI obtenidos se visualizan sobre un mapa; se pueden buscar, por ejemplo, restaurantes, museos, pizzerías, clubes, etc. (figura 1).

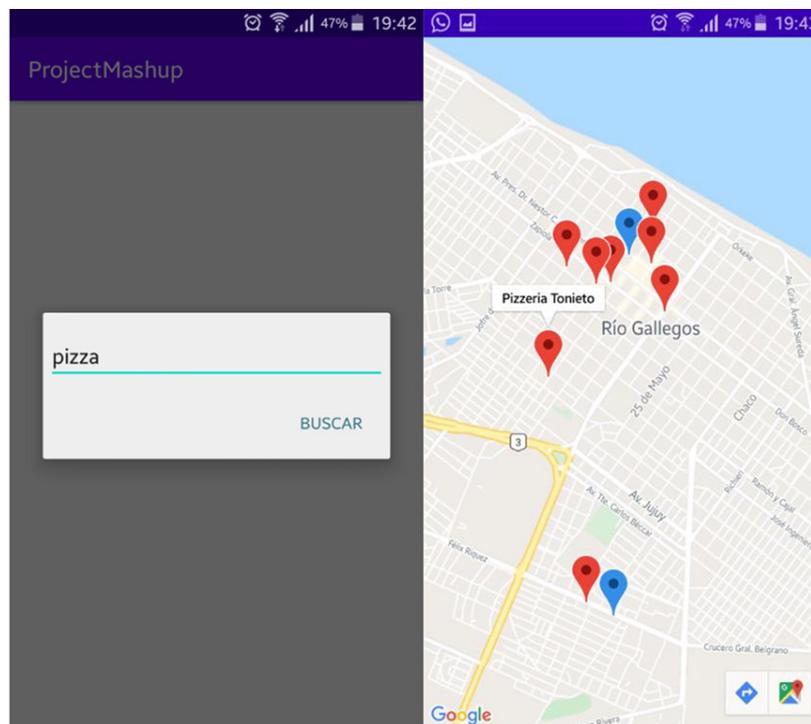


FIGURA 1. APLICACIÓN MASH-UP MÓVIL

La aplicación está compuesta por tres servicios web: Google Maps, TomTom y Foursquare que se detallan a continuación. La API de Google Maps [25] ofrece imágenes de mapas desplazables donde se pueden observar los POI obtenidos. Para obtenerlos, se utilizan dos servicios similares mediante llamadas a sus API: la API de TomTom (Search API) [26] y la API de Foursquare (Venue Search) [27]. Los POI de TomTom se representan en el mapa de color rojo y los POI de Foursquare de color azul.

Search API (TomTom) es una API RESTful diseñada para desarrolladores que permite la búsqueda de direcciones y POI. La búsqueda asigna una latitud/longitud a una dirección específica, cruce de calles, característica geográfica o punto de interés (POI).

Request: GET <https://api.tomtom.com/search/2/search>

Response: Object JSON

Venue Search (Foursquare): API que devuelve una lista de lugares cercanos o próximos de la ubicación actual, opcionalmente que coincida con un término de búsqueda

Request: GET <https://api.foursquare.com/v2/venues/search>

Response: Object JSON

Librería de indicadores

En la aplicación descrita, se incorporó la librería, que mediante llamadas a sus métodos permite recolectar los indicadores seleccionados. Durante el uso de la aplicación, se fueron registrando los valores de los indicadores en un archivo XML (por sus siglas en inglés). Posteriormente, estos datos fueron analizados y evaluados para ser considerados como un criterio de selección o evaluación del servicio web.

El registro de los indicadores se inicia cada vez que la aplicación *mash-up* invoca a los servicios web para solicitarle los POI, proceso que se ejecuta en segundo plano sin afectar la interacción del usuario. Los servicios que se evalúan son los que proveen los POI buscados (TomTom y Foursquare), que se usan mediante llamadas a sus puntos finales (*endpoints*) de acuerdo con su respectiva API REST.

En este trabajo, y mediante su implementación en la librería, se consideran los indicadores que se listan en la tabla 1.

TABLA 1. INDICADORES DE QoS Y QoE CONSIDERADOS

Indicadores QoS	
Latencia	El <i>delay</i> o latencia mide la cantidad de tiempo en milisegundos que tarda un paquete en viajar en una red desde el origen hasta el destino y volver.
Pérdida de paquetes	Es el porcentaje de paquetes transmitidos que se descartan en la red; se pierden.
Tipo de conexión	Tipo de conexión (wifi o 4G) a la que está conectado el dispositivo del usuario mientras utiliza la aplicación.
Indicadores QoE	
Latencia percibida por el usuario	El tiempo en milisegundos que el usuario espera la respuesta del servicio llamado desde la aplicación.
Número de resultados de la respuesta del servicio	El número de POI que tiene la respuesta del servicio llamado desde la aplicación.
Tamaño de la respuesta del servicio	El tamaño en bytes de la respuesta del servicio llamado desde la aplicación.

4. DIAGRAMA DE FUNCIONAMIENTO

Para realizar el estudio del funcionamiento y de la recolección de indicadores, se instaló la aplicación desarrollada en un dispositivo de marca Samsung modelo Galaxy A5 con Android versión 5.0.2. El área de búsqueda de POI se realizó sobre Río Gallegos (Santa Cruz, Argentina) que se encuentra en la latitud $-51,62261$ y longitud $-69,21813$.

En el periodo que se realizó el estudio, se asignó el dispositivo a un usuario con perfil informático para que utilice la aplicación diariamente, recomendando su utilización varias veces al día. Durante el estudio los indicadores se almacenaron en un archivo XML que tiene el siguiente formato: denominación del indicador recolectado, fecha y hora de registro, en *tag* se asienta la denominación del servicio asociado al indicador (TomTom o Foursquare), y por último el valor del indicador.

```
<indicador nombre="latencia" fecha="2020-06-24 12:32" tag="serviceName"> valor
</indicador>
```

Ejemplos:

```
<indicador nombre="latencia" fecha="2020-07-22 12:43:21" tag="tomtom">62.4</
indicador>
```

```
<indicador nombre="latencia" fecha="2020-07-22 12:43:27" tag="foursquare">60.4</indicador>
```

En la figura 2, se pueden observar las interacciones entre los distintos componentes de *software*. La aplicación *mash-up* móvil (*app mash-up* móvil) consume tres diferentes servicios web (Google Maps, TomTom y Foursquare) mediante el uso de REST API. Por medio de llamadas a la librería propuesta, antes de que la aplicación invoque a los servicios solicitando POI, se realiza la captura de indicadores definidos por el desarrollador. En el siguiente ejemplo, se muestra la llamada a la librería para recolectar el indicador “latencia” antes de invocar el servicio de Foursquare; el valor del indicador recolectado es registrado en el archivo XML en el formato antes mencionado:

```
myURL = new URL("https://api.foursquare.com/v2/venues/explore?client_id=xxxxxxxxx  
&client_secret=xxxxxxxxx&v=20180323&ll=-51.6333,-69.2167&query="+txtaBuscar);  
indicadores.getLatency("api.foursquare.com", "foursquare");  
HttpURLConnection conn = (HttpURLConnection) myURL.openConnection();  
conn.setRequestMethod("GET");  
conn.connect();
```

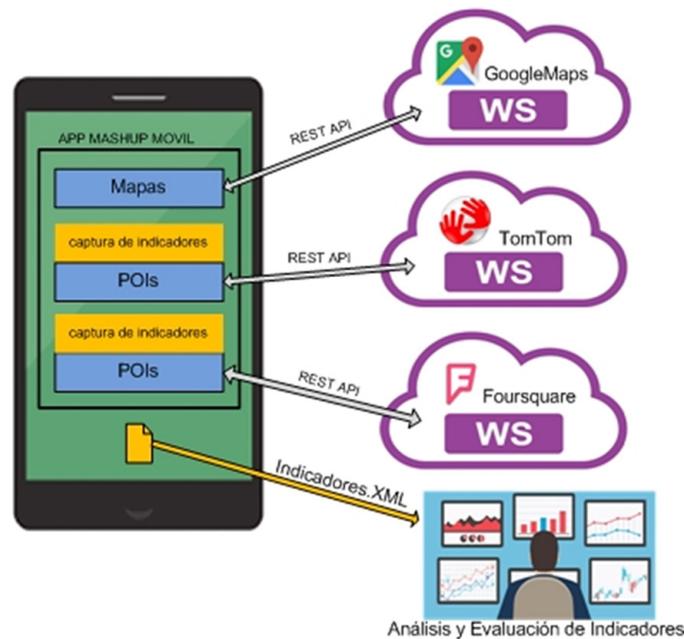


FIGURA 2. DIAGRAMA DE FUNCIONAMIENTO

5. RESULTADOS

Durante el periodo de prueba, se registraron 1.012 valores de indicadores. Para el análisis y la evaluación de los indicadores obtenidos, se procedió a calcular la media truncada y se descartó un 5 % de elementos en el extremo inferior y superior.

Las siguientes figuras corresponden a los indicadores que aportan información más relevante para el análisis. El indicador “pérdida de paquetes” se descartó debido a que solo el 2 % registró 100 % de pérdidas.

En la figura 3, se puede observar, por un lado, la media de la latencia de la red contra los servicios, y por otro, la media de la latencia percibida por el usuario desde que la aplicación hace la llamada al servicio hasta que recibe la respuesta. Se observa que, a pesar de que la media de la latencia de la red del servicio TomTom es inferior a la de Foursquare, luego la media de la latencia percibida por el usuario de Foursquare es de 514 ms, mientras que la de TomTom es de 900 ms.

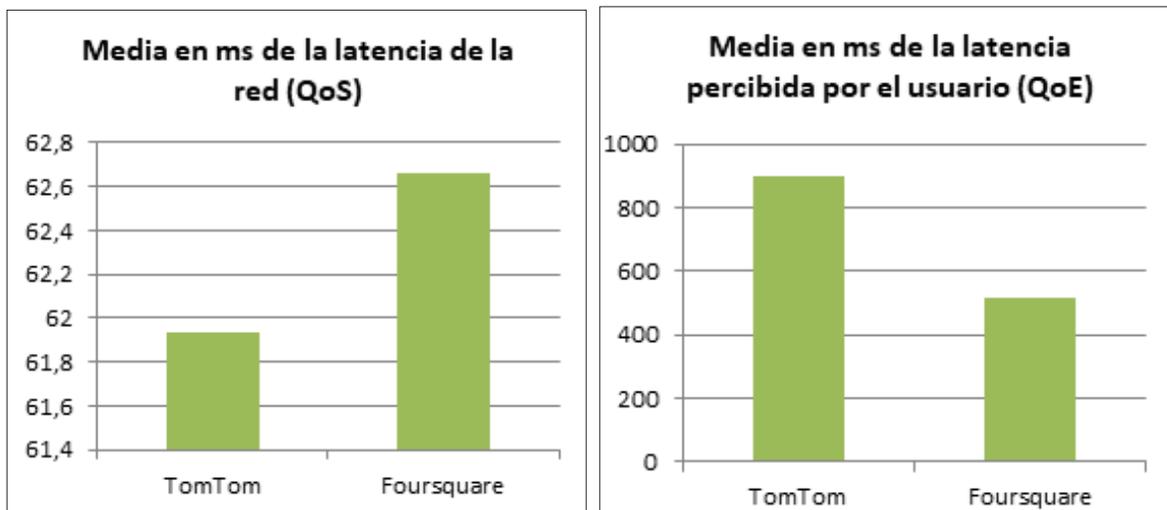


FIGURA 3. RESULTADOS DE LOS INDICADORES LATENCIA Y LATENCIA PERCIBIDA

Del mismo modo, analizando la figura 4, se observa que la media del número de respuestas del servicio TomTom (4,2) a búsquedas realizadas por el usuario es muy superior a la de Foursquare (1,6), en el servicio TomTom el 26 % de las búsquedas retornaron o respuestas, mientras que Foursquare lo hizo en el 46 %. Esto tiene relación con el tamaño de las respuestas: a mayor número de respuestas, mayor va a ser su tamaño.

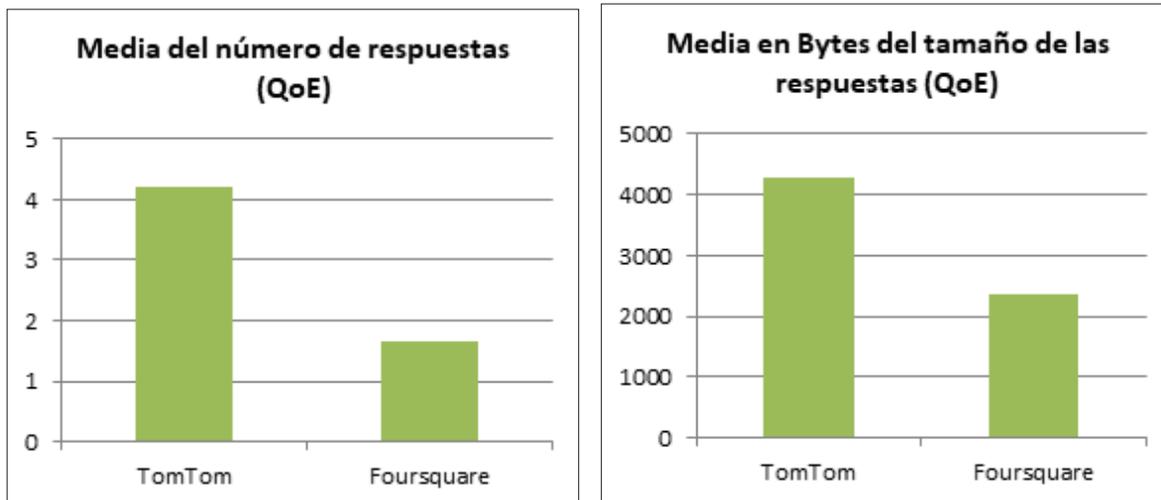


FIGURA 4. RESULTADOS DE LOS INDICADORES NÚMERO Y TAMAÑO DE LAS RESPUESTAS

6. DISCUSIÓN

En este trabajo, observamos que la media de la latencia de la red del servicio TomTom (61,93 ms) es 0,73 ms menor que la de Foursquare (62,66 ms). Si solo consideramos este indicador, hace suponer que sería mejor para el usuario seleccionar un servicio que tiene un tiempo de acceso menor. Sin embargo, necesariamente no se corresponde con una mayor satisfacción del usuario con ese servicio, debido a que luego el tiempo de procesamiento de las búsquedas es mayor para el servicio TomTom.

El 2 % de los indicadores de pérdida de paquetes que registró el 100 %, no fue considerado para el análisis, debido al bajo porcentaje y a que la pérdida se pudo haber dado por falta de conexión del dispositivo y no por un problema propio del servicio.

La principal diferencia de nuestro trabajo con otros estudios relacionados es que la recolección de los indicadores de QoS y de QoE fueron realizados desde un dispositivo móvil con la funcionalidad de recolección integrada en la aplicación *mash-up* móvil, lo que permite a su desarrollador disponer de información desde el punto de vista de la aplicación y del contexto real de uso del usuario final. De esta manera, el desarrollador, ante la falta de calidad de algún indicador, puede mejorar su aplicación o cambiar el proveedor del servicio web por otro de mejor calidad.

Consideramos que otro factor que puede influir en el proceso de selección es la calidad de las respuestas. Si realmente estas son relevantes para el usuario y también la fecha de actualización de los POI devueltos en las búsquedas, pueden hallarse POI que en la actualidad no existen. Este tipo de indicador que estima la calidad de la respuesta no fue considerado en este trabajo.

Las limitaciones de nuestro estudio están relacionadas con la selección parcial de indicadores de QoS/QoE que se registró, dado que existen más factores que pueden tener impacto en la satisfacción del usuario al utilizar una aplicación o servicio móvil. De igual forma, al recolectar indicadores de QoE sobre servicios web que ofrecen información sobre puntos de interés y no considerar la amplia variedad de contenidos, datos y tipos de servicios como servicios de clima, de negocios, de control, etc.

7. CONCLUSIONES

Mediante este estudio se exploraron algunos de los factores de QoS y de QoE que influyen en la percepción del usuario al utilizar una aplicación que consume servicios web. Para realizarlo, se desarrolló una aplicación que utiliza dos servicios web similares y una librería que permite recolectar desde la aplicación indicadores de QoS (latencia, pérdida de paquetes y tipo de conexión) y de QoE (latencia percibida por el usuario, número de resultados de la respuesta del servicio y tamaño de la respuesta del servicio).

Como conclusión, recomendamos que para realizar análisis o evaluaciones de QoS y de QoE debe considerarse la mayor cantidad de indicadores y la relación entre ellos, de esta manera se minimizan las interpretaciones parciales o imprecisas respecto de los servicios. En este trabajo, se probó que analizar los indicadores de forma aislada puede derivar en conclusiones equivocadas.

Como trabajo futuro, se plantea mejorar el estudio realizado analizando la calidad de las respuestas retornadas por los servicios, extender los tipos de indicadores recolectados tanto de QoS como de QoE y ampliarlo considerando *mash-ups* móviles más complejos.

REFERENCIAS

- [1] G. Devashish, "Mobile computing", *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 9, pp. 846-855, sept. 2013.
- [2] M. Hefeeda y C. Hsu, "Mobile video streaming in modern wireless networks", en *MM'10: Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1779-1780. <https://doi.org/10.1145/1873951.1874368>
- [3] H. Luo y M. Shyu, "Quality of service provision in mobile multimedia: a survey", *Hum. Cent. Comput. Inf. Sci.*, vol. 1, no. 1, pp. 1-15, nov. 2011. <https://doi.org/10.1186/2192-1962-1-5>
- [4] J. Shaikh, M. Fiedler y D. Collange, "Quality of experience from user and network perspectives", *Ann. Telecommun.*, vol. 65, pp. 47-57, dic. 2010. <https://doi.org/10.1007/s12243-009-0142-x>

- [5] GSMA (20 sept. 2016), *The mobile Economy Latin America and the Caribbean 2016* [En línea]. Disponible en: <https://www.gsma.com/latinamerica/resources/mobile-economy-latin-america-caribbean-2016>
- [6] GSMA (31 oct. 2017), *The Mobile Economy Latin America and the Caribbean 2017* [En línea]. Disponible en: <https://www.gsma.com/latinamerica/resources/mobile-economy-latin-america-caribbean-2017>
- [7] F. Daniel y M. Maristella, *Mashups: concepts, models and architectures*. Milán: Springer, 2014.
- [8] S. Aghaee, “A quality-based framework for leveraging the process of mash-up component selection”, Tesis de maestría, University of Gothenburg, Gotemburgo, Suecia, 2010 [En línea]. Disponible en: <http://hdl.handle.net/2077/21953>
- [9] API Harmony - Catalog of web API [En línea]. Disponible en: <https://apiharmony-open.mybluemix.net/public>
- [10] Public API - Collection of public and open APIs [En línea]. Disponible en: <https://public-apis.xyz>
- [11] ProgrammableWeb - APIs, Mashups and the Web as Platform [En línea]. Disponible en: <https://www.programmableweb.com>
- [12] M. Siekkinen, E. Masala y T. Kämäräinen, T. “A first look at quality of mobile live streaming experience: the case of periscope”, en *IMC'16: Proceedings of the 2016 Internet Measurement Conference*, 2016, pp. 477-483. <https://doi.org/10.1145/2987443.2987472>
- [13] G. Zhi-Peng, C. Jian, Q. Xue-Song y M. Luo-Ming, “QoE/QoS driven simulated annealing-based genetic algorithm for Web services selection”, *J. China Univ. Posts Telecommun.*, vol. 16, pp. 102-107, sept. 2009. [https://doi.org/10.1016/S1005-8885\(08\)60347-7](https://doi.org/10.1016/S1005-8885(08)60347-7)
- [14] S. Joshi y O. Ramanaiah, “An integrated QoE and QoS based approach for web service selection”, en *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, 2016, pp. 1-7. Doi: 10.1109/ICTBIG.2016.7892671
- [15] K. Bouraquia, E. Sabir, M. Sadik y L. Ladid, “Quality of experience for streaming services: measurements, challenges and insights”, en *In IEEE Access*, 2020, pp. 13341-13361. Doi: 10.1109/ACCESS.2020.2965099
- [16] S. Cardeal, F. Neves, S. Soares, F. Tavares y P. Assunção, “Arqos®: system to monitor QoS/QoE in VoIP”, en *2011 IEEE EUROCON - International Conference on Computer as a Tool*, 2011, pp. 1-2. Doi: 10.1109/EUROCON.2011.5929310
- [17] S. Jelassi, G. Rubino, H. Melvin, H. Youssef y G. Pujolle, “Quality of experience of VoIP service: a survey of assessment approaches and open issues”, en *IEEE Communications Surveys & Tutorials*, 2012, pp. 491-513. Doi: 10.1109/SURV.2011.120811.00063

- [18] L. Skorin-Kapov, M. Varela, T. Hoßfeld y K. Chen, “A survey of emerging concepts and challenges for QoE management of multimedia services”, en *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2018, pp. 1-29. <https://doi.org/10.1145/3176648>
- [19] A. Machini, J. Enriquez y S. Casas, “Q2M, una librería para computar métricas de calidad en aplicaciones móviles”, *ICT-UNPA*, vol. 11, no. 2, pp. 1-17, ag. 2019. <https://doi.org/10.22305/ict-unpa.v11i2.783>
- [20] A. Machini, J. Enriquez y S. Casas, “Nexo: una herramienta para la visualización y análisis de indicadores QoS y QoE móviles”, *ICT-UNPA*, vol. 12, no. 2, pp. 47-62, nov. 2020. <https://doi.org/10.22305/ict-unpa.v12.n2.731>
- [21] M. Chowdhury, “Measuring impact of QoS on QoE in mobile web services”, Tesis de maestría, Royal Institute of Technology, Estocolmo, Suecia 2012 [En línea]. Disponible en: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-90334>
- [22] F. Lalanne, A. Cavalli y S. Maag, “Quality of experience as a selection criterion for web services”, en *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, 2012, pp. 519-526. Doi: 10.1109/SITIS.2012.81
- [23] B. Chihani, K. Rehman Laghari, E. Bertin, D. Collange, N. Crespi y T. Falk, “User-centric quality of experience measurement”, en *Mobile Computing, applications, and services: 5th International Conference, MobiCASE 2013, Paris, France, November 7-8, 2013, revised selected papers*, G. Memmi y U. Blanke, Eds. Cham: Springer, 2014, pp. 33-43. https://doi.org/10.1007/978-3-319-05452-0_3
- [24] D. Bermbach y E. Wittern, “Benchmarking web API quality”, en *Web engineering: 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings*, A. Bozzon, P. Cudre-Maroux y C. Pautasso, Eds. Cham: Springer, 2016, pp. 188-206. https://doi.org/10.1007/978-3-319-38791-8_11
- [25] Google Cloud, *Maps* [En línea]. Disponible en: <https://cloud.google.com/maps-platform/maps>
- [26] TomTom (30 jun. 2020), *Search API and Extended Search API* [En línea]. Disponible en: <https://developer.tomtom.com/search-api>.
- [27] Foursquare, *Venue Search* [En línea]. Disponible en: <https://developer.foursquare.com/docs/api-reference/venues/search>