

Gestión de Calidad Móvil: Contratos, QoS y QoE

Juan Enriquez¹, Antonio Collazo García², Ariel Machini¹, Sandra Casas¹

¹Instituto de Tecnología Aplicada, Universidad Nacional de la Patagonia Austral,

²Centro de Investigaciones y Transferencias Santa Cruz, CONICET

Piloto Rivero S/N, Río Gallegos, Argentina

{jenriquez,scasas}@unpa.edu.ar

Abstract. La gestión de la calidad, la medición y evaluación, el monitorio, la optimización de la QoS / QoE de los servicios móviles en sus diversos aspectos, es un problema actual. Este trabajo explora y analiza la integración de QoS y QoE móvil en contratos, que permitan especificar las necesidades y/o preferencias de los usuarios de las aplicaciones y a la vez puedan ser ejecutados en los dispositivos. Para ello presentamos mecanismos para la especificación e implementación de los contratos y la composición de los mismos con las aplicaciones móviles. En particular se cubre la plataforma Android.

Keywords: Aplicaciones Móviles, Calidad de Servicio, Calidad de Experiencia, Contratos.

1 Introducción

Los dispositivos móviles se vuelven cada vez más populares y potentes, esto se debe a múltiples factores, entre ellos la capacidad de ejecutar aplicaciones de contenidos multimediales y por otro lado, a partir del avance de las tecnologías de redes inalámbricas de banda ancha, lo que también ha sido denominado como multimedia móvil.

Sin embargo, las aplicaciones y/o servicios móviles están expuestos a diversos y múltiples factores que afectan no solo su desempeño (calidad de servicio – QoS), sino también la satisfacción del usuario final (calidad de experiencia - QoE). Ante un inaceptable desempeño en la ejecución de las aplicaciones móviles y una experiencia no satisfactoria del usuario, resulta común que no se utilicen o se abandonen las aplicaciones. Algunos factores ampliamente reconocidos como causantes de los problemas de desempeño se encuentran, canales físicos poco fiables y con ancho de banda limitado, movilidad de nodos, enrutamiento, limitaciones de recursos (la CPU, la batería y memoria de los dispositivos, tamaño de pantalla y métodos de entrada), métricas de evaluación, condiciones climáticas [1][2][3].

En este escenario, sumamente heterogéneo y cambiante, una de las áreas de mayor interés entre la comunidad científica, las prestadoras de servicios, los desarrolladores de aplicaciones móviles y los usuarios, es la calidad. La gestión de la calidad, la medición y evaluación, el monitorio, la optimización de la QoS / QoE de los servicios móviles en sus diversos aspectos, es un problema actual.

La literatura y la industria han hecho distinciones entre QoS y QoE, no obstante, se proponen enfoques de tratamiento más integrales [4][5][6][7] que desde la hipótesis de que una QoS degradada generará un QoE inaceptable, admiten sus estrechas relaciones causales, ya que la QoS ofrece la infraestructura subyacente de base para la medición de QoE, es así que la QoS define las características no funcionales de un sistema, afectando la calidad percibida de los resultados.

Una visión más amplia de QoS, la determina como el conjunto de aspectos no funcionales del sistema que determinan el nivel de satisfacción de los usuarios de la funcionalidad ellos proveen [8], con el fin de cumplir con el objetivo general de obtener niveles de QoS que coincidan con las necesidades de los usuarios y así impacten en la QoE. Esta forma de entender la QoS implica una traducción de nivel superior de los requisitos de QoS a nivel de usuario a los requisitos de QoS de nivel inferior.

En esta perspectiva entran a jugar mayor cantidad de factores, propiedades, métricas, parámetros e indicadores, que atraviesan varias de capas del ecosistema móvil. Esta necesidad se convierte en una tendencia por definir mecanismos para permitir que las aplicaciones declaren sus necesidades en términos de QoE (objetivo), resultando coherente con los requisitos 5G expresados por NGMN Alliance en [9], que afirma que el próximo sistema 5G debe ser consciente de los usuarios finales y las necesidades de QoE de las aplicaciones, midiéndolas donde se percibe (es decir, en los dispositivos) [10].

Este trabajo explora y analiza la integración de QoS y QoE móvil en contratos, que permitan especificar las necesidades y/o preferencias de los usuarios de las aplicaciones y a la vez puedan ser ejecutados en los dispositivos. Para ello presentamos mecanismos para la especificación e implementación de los contratos y la composición de los mismos con las aplicaciones móviles. En particular se cubre la plataforma Android.

2 QoS y QoE Móvil

Los antecedentes de gestión de la calidad, remiten casi unívocamente a la QoS, ámbito en el cual se definen con mayor precisión funciones, fases, elementos y componentes. La gestión de QoS se puede considerar como un área especializada de la administración de sistemas distribuidos, y refiere a la supervisión y el control necesarios para garantizar que se alcancen y se mantengan las propiedades de calidad de servicio deseadas, que se aplica tanto a las interacciones continuas de los medios como a las interacciones discretas [11]. En el contexto de la computación móvil, la Gestión de la QoS se refiere al conjunto de actividades dedicadas a la vigilancia y control de los recursos involucrados en la provisión de un nivel de QoS adecuado. La Gestión de QoS es una tarea global en el sentido de que tiene en cuenta todo tipo de recursos que soportan el servicio (s) proporcionado (s) por un sistema: recursos de red, recursos del sistema, recursos de middleware, aplicaciones, etc., con el fin de cumplir con el objetivo general de obtener niveles de QoS que coincidan con las necesidades de los usuarios. Esta forma de entender la QoS se conoce como “end-to-end QoS” e implica una traducción de nivel superior de los requisitos de QoS a nivel de usuario a los requisitos de QoS de nivel inferior [8].

Los diversos aspectos de la interacción y los tipos de garantías requeridos, deben sintetizarse en una especificación de requisitos y en las relaciones de compensación para permitir la gestión de la QoS entregada [12], estas especificaciones a menudo constituyen “contratos”. La gestión de QoS se suele dividir en funciones o fases, entre las cuales básicamente se distinguen dos fases o tipos de funciones: estáticas (definición del contrato que consiste en la especificación de requerimientos, capacidades, indicadores, parámetros, de carácter principalmente cuantitativos, especificación de las reglas de negociación/re-negociación: acciones que se ejecutarán ante diversas circunstancias, y reserva de recursos en los casos que se requiera) y dinámicas (los aspectos dinámicos de la gestión de la calidad de servicio responden al cambio dentro del entorno, permitiendo que un contrato se cumpla de forma continua). Las funciones estáticas, aplicadas al inicio de una interacción, y funciones dinámicas, aplicadas según sea necesario durante una interacción (sesión u ejecución del sistema).

En cuanto a la QoE, no existe un andamiaje de “gestión” como conjunto de funciones, sino a tareas centradas en la evaluación, monitorio y/o predicción, basándose en métricas objetivas y/o subjetivas. Existen varios métodos para cuantificar las métricas subjetivas [13][14][15][16]. Las métricas subjetivas se encuentran relacionadas a la opinión del usuario, evaluando la calidad de la aplicación o servicio a partir de su experiencia con la aplicación [15], dependiendo de factores subjetivos como el contexto y expectativas del usuario sobre la aplicación. En general, las métricas subjetivas se basan en la realización de encuestas y cuestionarios a los usuarios para la obtención de información, presentando la dificultad de ser costosas en tiempo y dinero [4][17].

Con respecto a las métricas objetivas se destaca su carácter sistemático, exacto y repetible, y refieren a distintas propiedades como, el tiempo de presentación de los datos, que mide el tiempo transcurrido entre la acción del usuario y la respuesta de la aplicación, el consumo de los datos móviles, consumo de energía, interfaz de usuario y presentación del contenido [15]. Varios estudios analizan la QoE en estrecha relación con la QoS, en tanto que una degradación en la QoS provoca una inaceptable QoE. Parámetros de QoS como retardo, inestabilidad, tasa de pérdida, tasa de error, ancho de banda y tasa de éxito de la señal se usan para determinar el valor de la QoE, dependiendo únicamente del cálculo de la QoS [7]. La QoS se compone de un conjunto de métricas relacionadas a los parámetros de rendimiento de la red que representan un impacto en la QoE. La cantidad de datos que se recibe del servicio como respuesta, el tiempo transcurrido entre el envío y la recepción de los datos y la pérdida de información, son algunas de las métricas asociadas a la QoS [4].

3 Integrando QoS y QoE en contratos

En esta Sección se presenta el enfoque para la gestión de calidad, a partir de la integración de QoS y QoE en contratos. La misma cubre las fases estática y dinámica de la gestión.

3.1 Definición de Contratos de Calidad

La definición del contrato, estará supedita al concepto subyacente de enfoque de calidad que se pretende administrar. La Figura 1, representa el enfoque adoptado. La calidad como un todo (parte izquierda), que a la vez reconoce como partes componentes, la QoS y la QoE, cada una de estas propone sus propias propiedades de análisis (parte derecha).

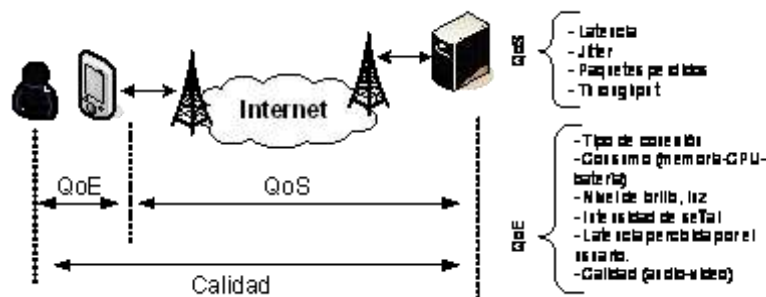


Fig.1: Enfoque de Gestión de Calidad

Así en términos generales y genéricos (Fig.2), un contrato es una lista o conjunto de tuplas, cada tupla establece una relación entre una propiedad de calidad (QoS o QoE) requerida y una acción a realizar si la propiedad no se cumple. Una propiedad requerida, y un valor (un valor específico o en un rango) que se define estáticamente pero se controla dinámicamente.

```

MPropiedades = {latencia, jitter, paquetes_perdidos, conexion,
consumo_memoria, consumo_bateria, consumo_CPU,
nivel_brillo, nivel_luz, intensidad_senal,
latencia_percibida, calidad_audio, calidad_video,...}

MAcciones = {no_accion, monitorear, controlar_flujo, almacenar_datos,
restringir_calidad, reservar_recurso, reasignar_recurso,
enviar_advertencia, suspender_act, abortar_otras_act}

MContrato = {(p, a): p ∈ MPropiedades y a ∈ MAcciones}

```

Una lista posible de acciones a realizar se han tomado de [18], ninguna acción, monitorear la QoS/QoE lograda, controlar el flujo de información, actuar para restringir la QoS/QoE lograda, reservar o reasignar recursos, advertencia de que se están cruzando límites o umbrales, suspender o abortar la actividad, suspender o abortar otras actividades que compiten por los recursos.

3.2 Implementación de Contratos

La especificación del contrato se realiza desde una aplicación móvil instalada en el dispositivo, la cual es independiente del resto de las aplicaciones de usuario. La Figura 2, presenta la secuencia del proceso. Primero, debe elegirse la aplicación sobre

la cual se definirá el contrato (2.a). El contrato es particular para la aplicación que se pretende controlar, en razón que distintas aplicaciones requieren distintas propiedades (parámetros y valores) y a la vez diferentes acciones. A continuación, se selecciona las propiedades que formará parte del contrato, para lo cual, se presenta una lista de las mismas (2.b). A medida que el usuario selecciona las propiedades, indica la cualidad o desempeño que debe cumplir (regular, medio, bueno) y la acción a realizar si el valor no se logra (monitoreo, alerta informativa y alerta confirmatoria) (2.c).



Fig. 3. App ContractGenerator.

La especificación de los valores de las propiedades, se realiza a partir de categorías cualitativas, que tienen su correspondiente valor numérico asignado o un indicador, en la Tabla 1 se presenta un subconjunto de las 11 que han sido aplicadas. Mientras que solo un valor de propiedad puede ser asignado a una propiedad, más de una acción puede ser relacionada a la propiedad. En (2.d) se presenta un contrato definido para YouTube que especifica las propiedades Latencia, Jitter y Paquetes perdidos.

Tabla 1. Propiedades, Valores cualitativos, indicadores y umbrales.

Propiedad	Bueno	Medio	Regular
Latencia	<100 ms	<= 200 ms	>200 ms
Jitter	< 50ms	<= 60 ms	>60
Paquetes perdidos	< 0.001	<= 0,002	>0,002
Tipo de Conexión	4G - WiFi	3G	GSM
Intensidad de Señal	40 a 60 dBm	60 a 80 dBm	>80dBm
Latencia percibida por el usuario	<100 ms	<=200 ms	>200 ms
Consumo de memoria	<100 um	<=200 um	>200

La especificación se almacena en formato XML, la Figura 4, presenta la estructura del contrato y una instanciación en la que se ha especificado para una aplicación diversas propiedades, valores y acciones.

<pre><?xml version="1.0" encoding="utf-8"?> <contract app="com.google.android.youtube" date="2018-07-02 17:19:37"></pre>	<pre><?xml version="1.0" encoding="utf-8"?> <contract app="com.google.android.youtube" date="2018-07-02 17:19:37"></pre>
--	--

<pre> <property name="ScreenBrightness ConnectionType SignalStrength Jitter Latency UserPerceivedLatency LightEnvironment PacketLoss Proximity MemoryConsumption CPUConsumption "> <range> regular middle bad </range> <actions> <monitoring/> <informative/> <confirmatory/> </actions> </property> </contract> </pre>	<pre> <property name="ScreenBrightness"> <range>regular</range> <actions> <confirmatory/> </actions> </property> <property name="Latency"> <range>middle</range> <actions> <monitoring/> </actions> </property> <property name="Proximity"> <range>bad</range> <actions> <monitoring/> <informative/> </actions> </property> </contract> </pre>
---	---

Fig. 4. Contrato e instanciación.

La composición del contrato, es el mecanismo que permite que un contrato se aplique o ejecute durante la interacción del usuario con la aplicación, es decir, permite llevar adelante la fase dinámica de la gestión de la calidad. Para ello, se ha adoptado un enfoque de separación de concerns y la modularización de las tareas correspondientes a la fase dinámica se han diseñado e implementado en AspectJ [19]. AspectJ es una extensión de Java que permite implementar requerimientos no funcionales y/o de calidad, como en este caso, a partir de una serie de mecanismos tales como pointcuts y advices y un proceso de tejido. De esta forma, un aspecto puede interceptar eventos (por ejemplo métodos) de clases correspondientes a la funcionalidad central de un sistema para incorporar nueva funcionalidad. En este caso, el aspecto ContractCheck, encapsula toda la lógica de la verificación de las propiedades y valores definidos en el contrato y la ejecución de las acciones definidas si corresponde (Fig. 5).

```

@Aspect
public class ContractCheck
@Pointcut("execution(* *. onCreate(..)")
public void metodo() {}
@Before("metodo()")
// Leer el contrato
// Para cada propiedad definida calcular el valor actual
// Verificar valores del contrato con los valores calculados
// Ejecutar la acción si no se cumple la restricción

```

Fig. 5. Aspecto ContractCheck.

Para ello, un pointcut intercepta cada vez que se crea una actividad y previamente a partir de un aviso “before” (fragmento de código que se ejecutará antes que la creación de la actividad se ejecute), realiza las operaciones necesarias: leer el contrato, analizar que propiedades y valores establecidos, y comparar con el cálculo

dinámico de dichas propiedades. Para los casos que no se cumpla la especificación, se dispara la acción indicada. Para el cálculo del valor de cada propiedad en ejecución, que es contrastada con los valores de las propiedades definidas en el contrato, hemos desarrollado una biblioteca de propiedades, que implementa los diversos algoritmos.

El listado de la Figura 6 es un recorte del fichero XML generado en una de las pruebas que hemos realizado, cuyo contrato especificaba la acción monitorear para las propiedades latencia, latencia percibida por el usuario, tipo de conexión, etc. Se registra la propiedad controlada, la fecha y hora del control y el valor calculado, el cual difiere del especificado.

```
<property id="latencia" date="2018-07-05 20:22:15">150.0</property>
<property id="latenciaPercibidaSegundos" date="2018-07-05 20:22:15">0</property>
<property id="brilloPantalla" date="2018-07-05 20:22:42">12</property>
<property id="tipoConexion" date="2018-07-05 20:22:42">3G</property>
<property id="brilloPantalla" date="2018-07-05 20:26:47">12</property>
<property id="tipoConexion" date="2018-07-05 20:26:47">3G</property>
<property id="latenciaPercibidaSegundos" date="2018-07-05 20:26:47">0</property>
<property id="brilloPantalla" date="2018-07-05 20:30:59">12</property>
<property id="tipoConexion" date="2018-07-05 20:30:59">GSM</property>
<property id="latenciaPercibidaSegundos" date="2018-07-05 20:31:00">1242</property>
<property id="brilloPantalla" date="2018-07-05 20:34:21">12</property>
<property id="tipoConexion" date="2018-07-05 20:34:21">GSM</property>
<property id="latenciaPercibidaSegundos" date="2018-07-05 20:34:21">0</property>
<property id="luzEntornoLx" date="2018-07-05 20:34:21">0.0</property>
<property id="brilloPantalla" date="2018-07-05 20:35:01">12</property>
```

Fig. 6. Monitoreo de un Contrato.

5. Trabajos Relacionados

A continuación se detallan varias herramientas (librerías y/o framework) que dan soporte a la medición de QoS o QoE móvil. Estas se centran en una de las dimensiones, proponen propiedades limitadas y en menor medida la posibilidad de definir acciones. Aunque algunas de estas propuestas consideran la participación del usuario final, el concepto de contrato no es explícito, principalmente el objetivo de la mayoría es realizar mediciones para el análisis posterior de las dimensiones.

El trabajo de [20] propone un framework para Android para el desarrollo de aplicaciones QoS distribuidas con capacidades de auto-reconfiguración. Las funcionalidades se incorporan mediante la extensión de sus clases (Activities, Services e Intents). Este framework se utiliza para desarrollar estrategias para múltiples parámetros de QoS de aplicaciones. El esquema QoS adoptado define, para cada dominio de aplicación, los atributos y valores de calidad, así como también permite al usuario expresar las preferencias personales de QoS en una solicitud de servicio. Esto posibilita que los usuarios y los proveedores de servicios sean capaces de definir los requisitos y las propuestas del servicio a fin de llegar a un acuerdo sobre la prestación del mismo.

Netradar [<https://www.netradar.org/>] es un sistema de software para mediciones de calidad de internet móvil que soporta tanto datos móviles como redes WiFi y los resultados obtenidos se pueden observar en un mapa. Proporciona clientes de

medición para múltiples sistemas operativos y los resultados de la medición se almacenan de manera centralizada. Las propiedades que se miden son, velocidad de bajada y subida de datos, latencia e intensidad de la señal.

Fastah Network Kit [<https://getfastah.com/>] es una biblioteca para Android para llevar a cabo la estimación de calidad de una red. Es útil para aplicaciones que requieren adaptar la experiencia y comportamiento de usuario en respuesta a diferentes condiciones de la red (4G, 3G, 2G y WiFi). La latencia y la congestión se miden utilizando servidores de baja latencia. La información se almacena en los servidores. Para realizar la estimación de calidad mediante la librería se requiere agregar como dependencia la misma a la aplicación que se va a controlar y configuraciones. Si se necesita un control manual sobre las mediciones, la integración se debe realizar además sobre el código fuente de la aplicación móvil.

Existen varias herramientas para la plataforma Android, entre ellas Speedtest.net, nPerf, Speed Test Pro, Speedcheck, que apuntan a medir la calidad de la red, sin embargo son muy limitadas y están enfocadas básicamente en considerar solamente el ancho de banda de la red, proporcionan valores absolutos sin realizar ningún tipo de análisis desde el punto de vista de los usuarios finales. Estas herramientas son aplicaciones que se instalan en los dispositivos móviles para monitorizar la calidad de servicio de la conexión, experimentada en el dispositivo.

Para la medición objetiva de la QoE en forma automática existen diferentes herramientas. Las herramientas Monsoon Power Monitoring y PowerTutor [15] se basan en la medición de métricas referentes al consumo de energía de los dispositivos móviles. Prometheus [21] se enfoca en el uso de métricas objetivas con el fin de estimar la QoE en aplicaciones móviles, prestando atención en métricas asociadas al tráfico de la red como rendimiento, volúmenes de datos, tasa de pérdida y otras como la intensidad de la señal. AppInsight [22] ayuda a diagnosticar fallos de rendimiento experimentados por sus aplicaciones, indicando las optimizaciones necesarias para mejorar la experiencia de usuario. Analiza las app móviles interponiéndose en los manejadores de eventos del dispositivo, teniendo en cuenta la latencia percibida por el usuario. AppInsight se enfoca en el monitoreo analizando métricas como el porcentaje de transferencia de datos extras y la cantidad de bytes por transmisión. Timecard [23] es una herramienta para estimar la QoE a partir de los tiempos de respuesta de las aplicaciones a los usuarios, basándose en el análisis de métricas como el tiempo transcurrido desde que el usuario inicia la solicitud y la estimación del tiempo que tarda en transmitir la respuesta del servidor al cliente y procesar la respuesta en el cliente. Panappticon [24] es una herramienta de monitoreo de eventos para Android que permite identificar problemas de rendimiento y diagnosticar las razones del bajo rendimiento percibido por el usuario en la realización de transacciones que inician por un toque de pantalla o pulsación de tecla y finalizan por una actualización de pantalla, midiendo el intervalo de tiempo entre la acción del usuario y la respuesta mostrada por la aplicación. Proteus [25] es una herramienta de recopilación de información sobre el rendimiento de la red, enfocada en aplicaciones de comunicación en tiempo real para predecir el rendimiento futuro de la red y la ocurrencia de pérdida de paquetes dentro de un intervalo de tiempo, analizando métricas como la tasa de pérdida, retardo unidireccional y rendimiento. QoE Doctor [17], la cual se basa en la medición de métricas objetivas como la latencia percibida por el usuario, el consumo de datos móviles y consumo de energía. Además cuenta con un conjunto de métricas para aplicaciones específicas como Facebook, Youtube y navegadores. En [26] se

describe una herramienta de evaluación capaz de estimar la QoE en términos de MOS para el servicio de YouTube. La misma permite mapear la QoS en QoE respecto de las sesiones de YouTube. Además, analiza posibles causas que conducen a una baja QoE y proporciona algunas sugerencias para mejorarla. En este trabajo, proponen un procedimiento genérico para estimar la calidad percibida por el usuario final siguiendo tres pasos: estimar o medir la QoS de la red (rendimiento, tiempo de ida y vuelta, tasa de pérdida, etc.); convertir métricas QoS de red en métricas QoS de la aplicación; y mapear la QoS a la QoE del usuario final (en términos de MOS).

6. Conclusiones

En este trabajo se presenta una propuesta para integrar QoS y QoE en contratos para aplicaciones móviles, con el objetivo de gestionar la calidad en las capas finales del ecosistema móvil. Algunas características de nuestra propuesta son, la especificación de los contratos es flexible en razón que pueden incluir distintos tipos de propiedades a controlar y acciones, se pueden realizar diversas combinaciones, atendiendo a los diferentes contextos (aplicaciones, usuarios, dispositivos); la composición de los contratos con las aplicaciones con aspectos además de garantizar una mejor modularización y evitar propiedades no deseables como el código invasivo, permite un más fácil desacople de los contratos.

Este trabajo exploratorio abre varias líneas de trabajo que continuaremos, identificar las limitaciones de los contratos, en cuanto a las propiedades y acciones que no puedan ser controladas y/o ejecutadas desde los dispositivos; analizar e incorporar las declaraciones de requerimientos de las aplicaciones, en la especificación de los contratos, resulta otra vía de trabajo complementaria; estimar el costo de la gestión de contratos, en términos de rendimiento, consumo de memoria y energía; y por último, analizar métodos de composición alternativos a AspectJ, para llevar la propuesta a otras plataformas móviles.

Referencias

1. Luo H. Shyu M.: Quality of service provision in mobile multimedia - a survey - Human-centric Computing and Information Sciences 1:5 Springer. (2011)
2. Devashish G.: Mobile Computing, International Journal of Advanced Research in Computer Science and Software Engineering, vol 3, num 9, pp 846-855 (2013)
3. Hefeeda M, Hsu C-H.: Mobile Video Streaming in Modern Wireless Networks. Proceedings of the ACM International Conference on Multimedia 1779-1780 (2013)
4. Kim H., Lee D., Lee J., Lee K., Lyu W., y Choi S: The QoE Evaluation Method through the QoS-QoE Correlation Model, en 4th International Conference on Networked Computing and Advanced Information Management, vol. 2, pp. 719-725. (2008)
5. Kosinski J., Nawrocki P., Radziszowski D., Zielinski K., Zielinski S., Przybylski G., Wnek P.: SLA Monitoring and Management Framework for Telecommunication Services, Networking and Services, LCNS 2008. 4th International Conference on, pp. 170-175, IEEE Computer Society (2008).

6. Moor K., Ketyko I., Joseph W., Verdejo J., Pessemier T., Marez L.: Evaluating Quality of Experience of mobile applications and services in a Living Lab setting. *Mobile Networks and Applications*, vol. 15 (2010).
7. Nawrocki P. y Śliwa A.: Quality of Experience in the context of mobile applications, *Comput. Sci.*, vol. 17, n.o 3, p. 371 (2016).
8. Asensio J., Villagra V., Lopez de Vergara J., Berrocal J.: UML Profiles for the Specification and Instrumentation of QoS, Management Information in Distributed Object-Based Applications, Technical University of Madrid, Spain. (2001).
9. GMN Alliance: 5g white paper, Tech. Rep. (2015).
10. Viti R., Pecorella T., Brilli L., Fantacci R., Carlini C.: Enhance the QoE for Smartphone Applications: an API approach for QoS control in LTE-A, AEIT International Annual Conference. (2015)
11. Blair G., Stefani J.: *Open Distributed Processing and Multimedia* – Addison-Wesley, 1997.
12. Chalmers D., Sloman M.: A survey of quality of service in mobile computing environments, *Communications Surveys & Tutorials*, IEEE, 2(2):2-10 (1999).
13. Hektner J., Schmidt J., y Csikszentmihalyi M.: *Experience Sampling Method: Measuring the Quality of Everyday Life*. SAGE (2007).
14. Kahneman D., Krueger A., Schkade D., Schwarz N. y Stone A.: A Survey Method for Characterizing Daily Life Experience: The Day Reconstruction Method, *Science*, vol. 306, n.o 5702, pp. 1776-1780 (2004).
15. Ickin S.: *Quality of Experience on Smartphones: Network, Application, and Energy Perspectives*, Tesis Doctoral - Diva Portal (2015).
16. Strejtl R., Winkler S, y Hands D.: Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives, *Multimed. Syst.*, vol. 22, n.o 2, pp. 213-227(2016).
17. Haokun Luo Q. y Sanae Rosen Z.: QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis, *IMC Proceedings de Conference on Internet Measurement*, Canada. Pp 151-164 (2014)
18. ITU-T Recommendation X.641- *Information Technology – Quality of Service: Framework - International Telecommunication Union* (1997)
19. Kiczales G., Hilsdale E., Hugunin J., Kersten M., Palm J., Griswold W.: An Overview of AspectJ, *Proceeding de 15th European Conference on Object-Oriented Programming* pp 327-353 UK(2001)
20. Gonçalves J., Ferreira L.: *A Framework for QoS-Aware Service based, Mobile Systems* (2010).
21. Aggarwal V., Halepovic E., Pang J., Venkataraman S., y Yan H.: Prometheus: Toward Quality-of-Experience Estimation for Mobile Apps from Passive Network Measurements, *HotMobile Proceedings de 15th Workshop on Mobile Computing Systems and Applications*. Article No. 18 (2014).
22. Ravindranath L., Padhye J., Agarwal S., Mahajan R., Obermiller I. y Shayandeh S.: AppInsight: Mobile App Performance Monitoring in the Wild, en *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, USA, pp. 107–120. (2012)
23. Ravindranath L., Padhye J., Mahajan R., y Balakrishnan H.: Timecard: Controlling User-perceived Delays in Server-based Mobile Applications, en *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, USA, pp. 85–100.(2013)
24. Zhang L., Bild D., Dick R., Mao Z., Dinda P.: Panappticon: Event-based tracing to measure mobile application and platform performance, *International Conference on Hardware/Software Codesign and System Synthesis* (2013).
25. Xu Q., Mehrotra S., Mao Z., y Li J.: PROTEUS: Network Performance Forecast for Real-Time, Interactive Mobile Applications, *Microsoft Res.* (2013).
26. Gómez G., Hortigüela L., Pérez Q., Lorca J., García R., Aguayo-Torres M.: YouTube QoE evaluation tool for android wireless terminals. *EURASIP J. Wireless Communications and Networking*, vol. 164, pp. 1-14 (2014).