

ARQUITECTURAS DE SOFTWARE PARA ENTORNOS MÓVILES

SOFTWARE ARCHITECTURE FOR MOBILE ENVIRONMENTS

Einer Zapata Granada¹, Luis Eduardo Sepúlveda Rodríguez²,
Carlos Eduardo Gómez Montoya³, Christian Andrés Candela Uribe⁴.

¹. Universidad del Quindío Armenia, Colombia. ezapata@grid.edu.co

². Universidad del Quindío Armenia, Colombia. lesepulveda@grid.edu.co

³. Universidad del Quindío Armenia, Colombia. carloseg@grid.edu.co

⁴. Universidad del Quindío Armenia, Colombia. christiancandela@grid.edu.co, Grupo de Investigación en Redes, Información y Distribución – GRID, Universidad del Quindío

Recibido: 07 de febrero de 2014

Aceptado: 08 de marzo de 2014

*Correspondencia del autor. Universidad del Quindío- Armenia: Carrera 15 Calle 12 Norte. Bloque de Ingeniería

Tel: +57 (6) 7359355 Armenia-Colombia.

E-mail: {ezapata}@grid.edu.co

RESUMEN

El gran impacto que están causando los smartphones en la actualidad se debe en gran medida a la evolución de sus sistemas operativos, los cuales cada vez son más estables y robustos, lo que permite a los desarrolladores de software crear aplicaciones móviles de mayor tamaño y complejidad. Las aplicaciones móviles han ido más allá de los algoritmos y estructuras de datos, la especificación y diseño de la estructura general de un sistema de software móvil surge como un nuevo tipo de problema, y la arquitectura de software móvil como la solución a este. Esta investigación tuvo como objetivo definir una solución arquitectónica móvil que compartiera algunos de los principios más reconocidos de la arquitectura de software en general, y de esta forma ayudar a estandarizar y adaptar metodologías, procesos, enfoques y demás aportes significativos de la arquitectura de software moderna a entornos de desarrollo de software pequeños. La metodología de investigación adoptada para este proyecto es una adaptación del método científico de Francis Bacon, el cual consta de seis etapas: observación, inducción, hipótesis, prueba de hipótesis, demostración de hipótesis y teoría científica. El artículo se compone de 6 secciones: Introducción, Marco de referencia, Metodología, Proceso para la construcción de una arquitectura de software móvil, Resultados y Conclusiones.

Palabras claves: smartphones, estilo arquitectónico, arquitectura de software, arquitectura de software móvil, atributos de calidad, escenarios de calidad, screening question.

ABSTRACT

The great impact that smartphones are causing currently is due largely to the evolution of their operation systems, which every time are more stable and robust, this allow to the software developers create mobile applications of more size and complexity. The mobile applications have gone beyond of the algorithms and data structure, the specification and design of the general structure of a mobile software system arise as a new kind of problem, and the mobile software architecture as the solution. It research had as goal define a mobile software architecture that will share some of the more important principles of the general software architecture, and so help to standardize and adapt methodologies, process, approaches and others significant contributions to the modern architecture software to smaller software develop environments. The methodology of investigation used for this project is an adaptation of the Francis Bacon's scientific method, which is compound of six phases: ask a question, do background research, construct a hypothesis, test your hypothesis by doing an experiment, analyze your data and draw a conclusion and communicate your results. The article is compound of six sections: Introduction, Framework, Methodology, Process for the construction of a mobile software architecture, Results and Conclusions.

Keywords: smartphones, architectural styles, software architecture, mobile software architecture, quality attribute, quality attribute scenarios, screening question.

1. INTRODUCCIÓN

Los Smartphones están revolucionando el sector tecnológico mundial (Columbus, 2013), su auge actual les ha permitido desempeñar un papel de gran relevancia en diferentes disciplinas, y se prevé que en el futuro dichos dispositivos tendrán un impacto mayor (IDC, 2013).

El presente de los teléfonos inteligentes se debe en gran parte a sus sistemas operativos (SO), estos cada vez son más estables y robustos (Android Developers), lo que ha permitido a los desarrolladores de software crear aplicaciones de mayor tamaño y complejidad. La construcción de los aplicativos móviles ha ido más allá de los algoritmos y estructuras de datos, la especificación y el diseño de la estructura general de un sistema de software móvil, surgen como un nuevo tipo de problema (Garlan & Shaw, Abstract, 1994), y la Arquitectura de Software Móvil (ASM) como la solución a este.

La arquitectura de software (AS) tuvo sus comienzos en los 80's y han evolucionado hasta la época actual (Mary & Rodrigues, 2012), pero aún en gran parte de Centro y Sur América es inmadura, esto debido a que grandes aportes de la arquitectura moderna aún no se han adaptado a este tipo entornos. A pesar que universidades como la Autónoma Metropolitana de México ya han buscado adaptar metodologías de gran importancia a entornos pequeños como el latino-americano (Cervantes Maceda), aún no se han estandarizado

adaptaciones de dichas metodologías, por lo que en muchos países de Centro y Sur América la AS aún es inmadura, y mucho más en el campo del software móvil que apenas viene creciendo a nivel mundial.

Esta investigación tuvo como objetivo definir una solución arquitectónica móvil para el proyecto llamado "Desarrollo de Framework Android basado en arquitectura de movilidad para la empresa EtherealGF S.A.S" que compartiera algunos de los principios más reconocidos de la AS en general, y de esta forma ayudar a aumentar la calidad del software. Aplicar los principios de la AS a la ASM, ayudara a estandarizar y adaptar metodologías, procesos, enfoques y demás aportes significativos de la AS moderna a entornos de desarrollo de software más pequeños.

La presente investigación es un trabajo realizado dentro del Programa de Jóvenes Investigadores de Colciencias en la modalidad interinstitucional entre la Universidad del Quindío a través del Grupo de Investigación en Redes, Información y Distribución – GRID y EtherealGF S.A.S (EtherealGF S.A.S). La metodología de investigación adoptada para este proyecto es una adaptación del método científico de Francis Bacon, el cual consta de seis etapas: observación, inducción, hipótesis, prueba de hipótesis, demostración de hipótesis y teoría científica (Gaspar, 2009).

Este artículo se encuentra estructurado de la siguiente forma: la sección 2 muestra la importancia de la ASM

y la necesidad de adatar procesos arquitectónicos reconocidos a entornos más pequeños. En la sección 3 se explica la metodología en la cual se basó la investigación. La sección 4 explica el proceso que se llevó a cabo para construir la ASM. A continuación en la sección 5, se muestran los resultados obtenidos en la investigación. Para finalizar en la sección 6 se muestran las conclusiones.

2. MARCO DE REFERENCIA

A. El auge de los Smartphones

Los smartphones están revolucionando el sector tecnológico mundial (Columbus, 2013), su auge actual les ha permitido desempeñar un papel de gran relevancia en diferentes disciplinas. Muchos dispositivos electrónicos, entre estos las computadoras, han sufrido un decrecimiento en el sector de su mercado, debido a las múltiples funcionalidades que tienen los teléfonos inteligentes, las cuales están remplazándolos (Meerker & Wu, INTERNET TRENDS D11 CONFERENCE, 2013).

La reconocida analista tecnológica Mary Meerker en el informe (Meerker, Internet Trends, 2012), logro predecir exitosamente en el 2012 como para el año 2013, la base instalada de teléfonos inteligentes y tabletas superó las de las computadoras portátiles y de escritorio. Por otra parte, un estudio realizado por International Data Corporation (IDC) coincidió con la proyección de Meerker (IDC, 2013), y prevé que para los próximos años aumentará significativamente la venta de dispositivos móviles inteligentes, lo que causará que los smartphones tengan un impacto mayor que el actual.

En gran parte, el presente de los teléfonos inteligentes se debe a sus sistemas operativos, los cuales cada vez son más estables y robustos. Un ejemplo es Android, que a partir de sus constantes versiones de API (Interfaz de programación de aplicaciones) (Android Developers), ha aumentado el número de funcionalidades y mejorado su desempeño.

Como lo dicen David Garlan and Mary Shaw en (Garlan & Shaw, Abstract, 1994), con el paso del tiempo los sistemas de software aumentan su tamaño y complejidad. La evolución de los SO móviles está causando este efecto en su propio entorno, el diseño de sus aplicativos ha ido más allá de los algoritmos y estructuras de datos, la especificación y el diseño de

la estructura general de un sistema de software móvil, surgen como un nuevo tipo de problema, y la Arquitectura de Software Móvil como la solución a este.

B. Actualidad de la arquitectura de software

La arquitectura de un sistema de software se debe de enfocar en satisfacer los atributos de calidad (Microsoft, 2009), para esto se usan una serie de patrones, estilos y tácticas arquitectónicas que dan la posibilidad de centrarse en aspectos como: desempeño, seguridad, mantenibilidad, etc. Típicamente la AS se compone de la combinación de estilos arquitectónicos, casi nunca se limita a uno solo (Microsoft, 2009).

Los principios básicos de la AS vienen siendo aplicados desde los años 80 (Mary & Rodrigues, 2012), y han evolucionado hasta la época actual, pero aún muchos países son inexpertos. En Colombia la AS viene tomando fuerza de la mano de universidades como UniAndes y Javeriana (UniAndes) (Pontificia Universidad Javeriana), o empresas como Asesoftware (Asesoftware). Aunque las actividades realizadas por dichas entidades ayudan a incrementar el número de expertos en AS, aún se está abarcando una mínima parte de los involucrados en este sector.

El SEI (Software Engineering Institute) es uno de los precursores en la investigación de la ingeniería de software y AS a nivel mundial (Software Engineering Institute (SEI)). Reconocidos investigadores pertenecientes a este instituto han realizado aportes en los cuales se base la AS moderna, uno de estos es Len Bass, quien propuso el método llamado Diseño Dirigido por Atributos (ADD) (Limón), este hace parte de un proceso de diseño y evaluación de AS que se descompone en 4 etapas (SEONTI, 2012):

- QAW (Quality Attribute Workshop)
- ADD (Attribute Driven Design)
- VaB (Views and Beyond)
- ATAM (Architecture Tradeoff Analysis Method)

Aunque este proceso ha aportado gran calidad a la AS, en entornos como el latino-americano, no es adecuado implementarlo debido a su alta robustez. Este problema ya ha sido notado por la Universidad Autónoma Metropolitana de México, la cual ha realizado especializaciones como (Cervantes Maceda), para adaptar el proceso del SEI a entornos pequeños.

Es claro que la AS aún es muy inmadura en la mayoría

de países de Centro y Sur América, y mucho más en el campo del software móvil que apenas viene creciendo a nivel mundial. Si se tiene en cuenta todo lo anterior, se puede afirmar que es de gran necesidad la adaptación de los estilos arquitectónicos de software típicos a entornos de desarrollo móviles, ya que en base a estos y las tácticas arquitectónicas se satisfacen los atributos de calidad (Microsoft Cooperation, 2009), los cuales permiten construir una ASM (Arquitectura de Software Móvil) exitosa.

Aunque el objetivo de este artículo se puede resumir en el párrafo anterior, cabe resaltar la necesidad de una investigación más a fondo, que se centre en la adaptación del proceso SEI nombrado anteriormente a entornos de desarrollo de software móvil.

3. METODOLOGÍA

La metodología que se empleó en el desarrollo del proyecto se basa en una adaptación del método científico según Francis Bacon (Gaspar, 2009), el cual consta de las siguientes fases:

- Observación: se realizó una investigación detallada sobre las necesidades que se presenta en EthereumGF al momento de desarrollar aplicaciones Android. En esta fase las screening questions fueron claves.
- Inducción: Con base en los resultados obtenidos en la fase 1, se investigaron y seleccionaron posibles frameworks, APIs y librerías que podían dar soporte a las necesidades planteadas. Además se identificaron las calidades sistémicas que debía tener el framework Android.
- Hipótesis: Con base en la información resultante de aplicar las primeras dos fases, se planteó el documento arquitectónico que definió los atributos de calidad, restricciones y requisitos funcionales que debe tener el framework para satisfacer las necesidades de EthereumGF.
- Probar la hipótesis: se desarrolló el framework Android para EthereumGF en base a la ASM planteada en la fase de Hipótesis.
- Demostración de hipótesis: se desarrolló un proyecto Android que usa algunos módulos del framework Android.
- Teoría científica: en esta fase se realizó toda la documentación que quedó como resultado de realizar la investigación.

4. PROCESO PARA LA CONSTRUCCIÓN DE UNA ARQUITECTURA DE SOFTWARE MÓVIL GRID

presenta gran interés en realizar investigaciones sobre temas emergentes que puedan aportar a la ingeniería de software (Grupo de Investigación en Redes, Información y Distribución - GRID), siendo este el caso de la ASM y AS en general. La presente investigación es un trabajo realizado dentro del Programa de Jóvenes Investigadores de Colciencias en la modalidad interinstitucional entre la Universidad del Quindío a través del Grupo de Investigación en Redes, Información y Distribución – GRID y EthereumGF S.A.S (EthereumGF S.A.S). Esta investigación surgió debido a la necesidad de buscar una solución arquitectónica al proyecto denominado “Desarrollo de Framework Android basado en arquitectura de movilidad para la empresa EthereumGF S.A.S”. Para la construcción de la ASM de dicho proyecto se realizó un proceso típico para crear AS en cualquier entorno, pero teniendo en cuenta las consideraciones nombradas en (J.D., y otros, 2008). La creación de la ASM comenzó con la realización de screening questions, las cuales son preguntas abiertas estructuradas para hacer reflexionar a los cuestionados y no darle espacio de dar respuesta cortas (Oracle). Luego, se establecieron las calidades sistémicas del framework y sus respectivos escenarios de calidad. A continuación, se satisficieron los atributos de calidad por medio de la selección y combinación de los estilos arquitectónicos. Para finalizar, se realizó la documentación de la arquitectura en base a el modelo de vistas 4+1 (Kruchten, Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software, 1995).

A. Definición de los atributos de calidad

Como se ha dicho anteriormente los atributos de calidad son la base para la selección de los estilos arquitectónicos, y por consiguiente también la de una AS exitosa, por eso cuando se inicia con la creación de una AS se debe realizar un estudio y análisis profundo que permita definir las calidades sistémicas que necesita el proyecto. Al ser el objetivo de esta investigación la construcción de un framework, además de la realización de las screening question, se consideró el informe (Riehle D. , 2000), el cual nombra atributos de calidad que debe tener este tipo de software.

El proceso de recolección de la información necesaria para iniciar el proyecto, comenzó con la realización

de dos fases de screening question al grupo de trabajo involucrado en la construcción de aplicaciones Android en EthereumGF; la primera fase se basó en un cuestionario de 57 preguntas en su mayoría abiertas, y la segunda fue una serie de entrevistas construidas con base en la información que se tabuló de la primera fase. Los resultados obtenidos sirvieron para definir los requisitos funcionales, restricciones y calidades sistémicas. Estos últimos se dividieron en Integridad conceptual, Mantenibilidad, Rendimiento, Confiabilidad y Reusabilidad.

Los atributos de calidad son difíciles de descubrir, escribir y verificar, por eso para su definición se construyeron escenarios de calidad que permitieron establecer el contexto y las circunstancias específicas de cada uno de ellos (Bass, Clements, & Kazman, 2012). El siguiente es uno de los escenarios de calidad realizados para delimitar el rendimiento del framework:

“Todas las operaciones que realice el framework deben ser ejecutadas en un intervalo de tiempo menor a un segundo, de rebasar esta barrera las operaciones se debe de ejecutar en un hilo secundario”.

| Elemento | Descripción |
|------------------------|--|
| Origen del estímulo | Stakeholder |
| Estímulo | Algunas funcionalidades no dependen solamente del framework. |
| Ambiente | Condiciones normales. |
| Componentes | Todo el framework |
| Respuesta | Realizar la mayoría de operaciones en menos de 1 segundo y las que sobrepasen este tiempo, ejecutarlas en un hilo secundario. |
| Medida de la respuesta | La mayoría de operaciones que realiza el framework se ejecutaran en máximo un 1 segundo, las demás se operaran en hilos secundarios. |

Figura 1. . Escenario de calidad de rendimiento.

B. Construcción de la arquitectura de software móvil

La definición de los atributos de calidad, restricciones y requisitos funcionales, establecen la base para una ASM, pero esto no garantiza que sea exitosa, y aunque nada más que el software final puede definir si hubo éxito o no, sí existe una manera de aproximarse a este resultado, y es usando como referencia una ASM que ya haya tenido éxito (Reed, 2002). Para esta investigación se usó como arquitectura de referencia la Mobile Solutions on Google Cloud Platform (Google Cloud Platform).

Los estilos arquitectónicos fueron seleccionados con base en las definiciones dadas en (Microsoft Corporation, 2009), pero teniendo en cuenta las consideraciones al desarrollar software móvil descritas en (J.D., y otros, 2008). El resultado de todo este proceso fue la selección y combinación del estilo arquitectónico por capas, basado en componentes y orientado a objetos.

Como se dijo anteriormente el modelo de vistas usado en esta investigación es el 4+1, este contiene 5 vistas que se deben documentar con enfoques diferentes. En el caso en que la información de una vista no presente valor para los interesados en la arquitectura, se recomienda no documentar dicha vista, pues una ASM solo debería de contener información relevante.

Cabe aclarar que la vista de escenarios siempre debe ser implementada, pues esta relaciona a las demás vistas por medio de escenarios reales y permite identificar irregularidades en el diseño arquitectónico (Krucchten, Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software, 1995). En este proyecto solo se documentaron la vista lógica, la vista de despliegue y la vista de escenarios.

El paso final del proceso arquitectónico fue compartir la arquitectura resultante con sus interesados, brindándoles de esta forma una vista estructural de alto nivel que proporcionó un lenguaje común y objetivos claros.

5. RESULTADOS

El documento arquitectónico definió el software a construir, mostró la estructura general del framework Android para EthereumGF y brindó a los stakeholders

un objetivo y vocabulario común para facilitar la comunicación. El resultado anterior fue alcanzado gracias al documento arquitectónico, el cual contiene la información que será descrita a continuación.

Aplicar las screening questions dio como resultado la información de mayor relevancia para la construcción de la ASM, esta etapa fue realizada lo más detalladamente posible debido a que se definió solo una solución arquitectónica candidata, esto con el fin de acelerar el proceso de selección del documento arquitectónico, y evitar la construcciones de multiplex soluciones candidatas, o de realizar procesos como el del SEI nombrado anteriormente.

El documento arquitectónico comenzó estableciendo las metas, restricciones y alcance de la ASM, allí se definieron los objetivos a cumplir con la construcción del framework, versiones Android a soportar, tiempo que debe tardar el desarrollo del proyecto, librerías, SDKs y APIs externas a usar, además de a quién va dirigido el documento arquitectónico y los temas a tratar en el framework.

En total los temas o módulos a tratar fueron 15: Facebook, XML, Twitter, REST, Drawable, Google Maps, Intenciones implícitas, BroadcastReceiver, AsyncTask, Canvas, Gestures, YouTube, Sockets, Views y SharedPreferences. Con base en dichos temas se documentaron la vista lógica, la vista de despliegue y la vista de escenarios.

La primera vista en documentarse fue la de escenarios, allí se definieron las calidades sistémicas del framework y sus escenarios de calidad. La segunda, la vista lógica, explicó la representación de la estructura general del framework Android, definió las piezas fundamentales de la arquitectura, el papel que desempeñan dentro del diseño y su comportamiento respecto a los otros elementos.

La figura 2 es la representación arquitectural del framework Android para EtherealGF, la capa de componentes del framework Android (en la cual se centra la información de mayor importancia para la arquitectura), consta de 3 capas, la primera, capa dependiente agrupa las funcionalidades de los temas que necesitan de librerías externas para su funcionamiento, este es el caso de Facebook, Twitter, XML, Google Maps y YouTube. La segunda, capa independiente, son las funcionalidades de los temas que solo necesitan

del framework de aplicación Android para realizar sus operaciones, ese es el caso de REST, Drawable, BroadcastReceiver, AsyncTask, Canvas, Gestures e Intenciones implícitas. La tercer es la capa conceptual, su objetivo principal es documentar por medio de ejemplos de codificación y manuales, las propiedades de temas como Sockets, Views, SharedPreferences e incluso temas de las capas dependientes e independiente.

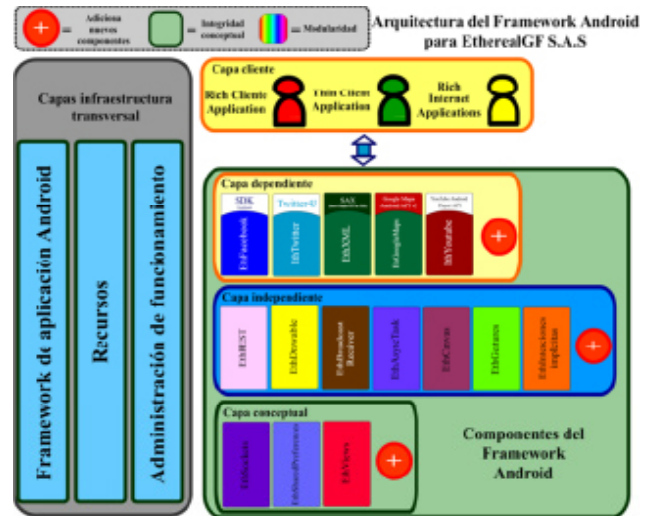


Figura 2. Diagrama arquitectural del framework Android para EtherealGF.

La última vista, la de despliegue, muestra la forma en que los módulos del framework descritos en la vista lógica deben ser desplegados en los proyectos. La figura 3 es una representación de dicho despliegue.

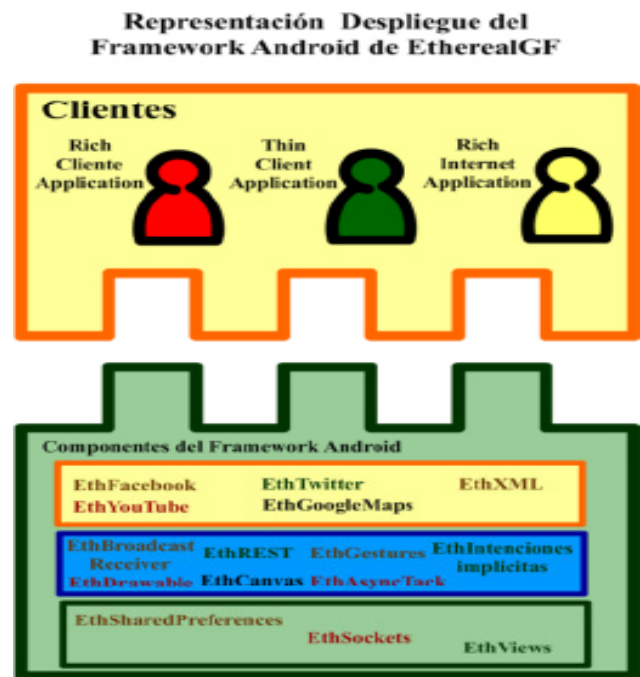


Figura 3. Representación del despliegue del framework Android

La ASM resultante es basada en estilos arquitectónicos ampliamente conocidos, por eso puede ser fácilmente comprendida por arquitectos de otros entornos.

6. CONCLUSIONES

El objetivo de la arquitectura de un sistema de software es construir aplicaciones de calidad, por eso la AS no se debe ver como una paso para retrasar el desarrollo, si no como la posibilidad de establecer la estructura general del sistema, y brindarle a los involucrados en la construcción del proyecto un lenguaje y objetivo común, antes de poner la primera línea de código.

Los estilos y tácticas arquitectónicas son la base para la construcción de cualquier AS, por eso la adaptación de los más importantes servirá como plataforma para aplicar metodologías, procesos, enfoques y demás aportes significativos de la arquitectura de software moderna, a entornos de desarrollo de software pequeños.

Para finalizar, se debe aclarar que a pesar de lograr aportar una gota de agua al mar de la AS, el conocimiento adquirido durante la construcción de este artículo servirá para realizar investigaciones de mayor robustez, como lo es adaptar el proceso arquitectónico del SEI, o cualquier otro aporte significativo de la AS moderna que sea de interés.

BIBLIOGRAFÍA

1. Android Developers. (n.d.). <uses-sdk>. (Android Developers) Retrieved Octubre 20, 2013, from <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>
2. Asesoftware. (n.d.). DISEÑO DE ARQUITECTURA DE SOFTWARE. Retrieved Enero 22, 2014, from Asesoftware: <http://www.asesoftware.com/web/es/capacitacion/124-talento-en-ingenieria-de-software/289-diseno-de-arquitecturas-de-software.html>
3. Bass, L., Clements, P., & Kazman, R. (2012). Understanding Quality Attributes. In *Software Architecture in Practice Third Edition* (pp. 63-78).
4. Cervantes Maceda, H. (n.d.). Evaluación de la metodología de diseño arquitectural propuesta por el SEI. Distrito Federal.
5. Columbus, L. (2013, Enero 17). 2013 Roundup of Smartphone and Tablet Forecasts & Market Estimates. (Forbes) Retrieved Octubre 19, 2013, from <http://www.forbes.com/sites/louiscolombus/2013/01/17/2013-roundup-of-mobility-forecasts-and-market-estimates/>
6. EtherealGF S.A.S. (n.d.). Inicio. (The Ethereal Game Factory) Retrieved Diciembre 17, 2013, from <https://www.etherealgf.com/?lang=es>
7. Facebook developers. (n.d.). Facebook SDK for Android. Retrieved Diciembre 03, 2013, from Facebook developers: <https://developers.facebook.com/docs/android/>
8. Garlan, D., & Shaw, M. (1994). Abstract. In *An Introduction to Software Architecture*. Pittsburgh.
9. Garlan, D., & Shaw, M. (1994). Introduction. In *An Introduction to Software Architecture* (p. 2).
10. Gaspar, M. M. (2009, Julio). La toma de decisiones metodológicas en la investigación social: Un devenir entre la subjetividad y la objetividad. Retrieved Enero 4, 2013, from http://www.uv.mx/cpue/num9/opinion/completos/margarito_decisiones.html
11. Google Cloud Platform. (n.d.). Mobile Solutions on Google Cloud Platform. (Google Cloud Platform) Retrieved Diciembre 22, 2013, from <https://cloud.google.com/developers/articles/mobile-application-solutions>
12. Google Developers. (n.d.). Mobile Solutions on Google Cloud Platform. Retrieved Diciembre 13, 2013, from Google Cloud Platform: <https://cloud.google.com/developers/articles/mobile-application-solutions>

13. Grupo de Investigación en Redes, Información y Distribución - GRID. (n.d.). Información general del GRID. Retrieved Noviembre 20, 2013, from http://ingenieriauiq.edu.co/index.php?option=com_k2&view=item&layout=item&id=123&Itemid=156
14. IASA. (n.d.). Colombia Chapter. (IASA) Retrieved Octubre 27, 2013, from http://www.iasaglobal.org/iasa/colombia_chapter.asp
15. IDC. (2013, Enero 14). Android and iOS Combine for 91.1% of the Worldwide Smartphone OS Market in 4Q12 and 87.6% for the Year, According to IDC . Retrieved Diciembre 12, 2013, from IDC Anlyze the Future: <http://www.idc.com/getdoc.jsp?containerId=prUS23946013>
16. IDC. (2013, Mayo 16). Android and iOS Combine for 92.3% of All Smartphone Operating System Shipments in the First Quarter While Windows Phone Leapfrogs BlackBerry, According to IDC . Retrieved Octubre 19, 2013, from <http://www.idc.com/getdoc.jsp?containerId=prUS24108913>
17. J.D., M., Alex, H., David, H., Jason, T., Prashant, B., Lonnie, W., et al. (2008). Mobile Application Architecture Guide .
18. Kruchten, P. (1995, Noviembre). Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software. IEEE Software 12.
19. Kruchten, P. (n.d.). Planos Arquitect´onicos: El Modelo de “4+1” Vistas de la Arquitectura del Software*.
20. Limón, C. R. (n.d.). La arquitectura de software en el desarrollo de sistemas de software. In Las vistas arquitectónicas de software y sus correspondencias mediante la gestión de modelos (p. 14). Valencia.
21. Mary, S., & Rodrigues, P. (2012). Software Architecture- Evolution and Evaluation. (IJACSA) International Journal of Advanced Computer Science and Applications, 3(8), 82-88.
22. Meerker, M. (2012). Internet Trends. Menlo Park.
23. Meerker, M., & Wu, L. (2013). INTERNET TRENDS D11 CONFERENCE. Menlo Park.
24. Microsoft. (2009). Combining Architectural Styles. In Microsoft Application Architecture Guide (p. 21). Microsoft press.
25. Microsoft. (2009). Determine the Quality Attributes . In Microsoft Application Architecture Guide (p. 16). microsoft press.
26. Microsoft. (2009). Quality Attributes. In Microsoft, Application Architecture Guiden (pp. 191-204).
27. Microsoft Cooperation. (2009). Quality Attributes. In Microsoft Application Architecture Guide 2nd Edition (pp. 191-204).
28. Microsoft Corporation. (2009). Architectural Patterns and Styles . In Microsoft Application Architecture Guide 2nd Edition (pp. 19-35).
29. Oracle. (n.d.). Setting Up Screening Questions. Retrieved Diciembre 18, 2013, from http://docs.oracle.com/cd/E39904_01/hcm92pbr0/eng/hcm/herm/task_SettingUpScreeningQuestions-e36a19.html
30. Pontificia Universidad Javeriana. (n.d.). Arquitectura Empresarial de Software. Retrieved Enero 6, 2014, from Pontificia Universidad Javeriana: http://puj-portal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/pos_arq_empresarial
31. Reed, P. R. (2002, Septiembre 15). Reference Architecture: The best of best practices. (developerWorks) Retrieved Diciembre 22, 2013, from <http://www.ibm.com/developerworks/rational/library/2774.html>
32. Riehle, D. (2000). Framework Design A Role Modeling Approach. Hamburg .
33. Riehle, D. (2000). Framework Design A Role Modeling Approach. Hamburg.
34. SEONTI. (2012). Diseño de Arquitecturas de Software .
35. Software Engineering Institute (SEI). (n.d.). About Us - Overview. (Software Engineering Institute (SEI)) Retrieved 10 27, 2013, from <http://www.sei.cmu.edu/about/?location=secondary-nav&source=1358>
36. Twitter Developers. (n.d.). Twitter Libraries. Retrieved Diembre 3, 2013, from Twitter Developers: <https://dev.twitter.com/docs/twitter-libraries>
37. UniAndes. (n.d.). CURSO IASA. (Universidad de los Andes) Retrieved Octubre 27, 2013, from <http://sistemas.uniandes.edu.co/main/noticias/390-curso-iasa>