Revista Científica yTecnológica UPSE, Vol. IV, N°2 Pág. 62-67 Junio 2017

8

Mobile Cloud Computing- Google Cloud Messaging

Patricio Navas M., Henry Llamuca, Diego Jácome, Jéssica Castillo.

Recibido: noviembre 2016
Aprobado: abril 2017

Mobile Cloud Computing- Google Cloud Messaging

Patricio Navas Moya[1], Henry Llamuca[1], Diego Jácome[2], Jéssica Castillo[2]
[1]Universidad de las Fuerzas Armadas ESPE; [2]Universidad Técnica de Cotopaxi
mpnavas@espe.edu.ec

Resumen

En este artículo se presenta un análisis de GCM (Google Cloud Messaging); rendimiento de la biblioteca, la integración con Google App Engine y también las herramientas de desarrollo, incluyendo la API. Los resultados revelan que GCM en relación a su predecesor C2DM (Cloud to Device Messaging) presenta mejoras significativas que se deben tomar en cuenta al momento del desarrollo de Apps en tiempo real que facilita y potencia el desarrollo de aplicaciones cada vez más potentes, fiables, rápidas y sobre todo optimizando recursos. Por lo tanto, mediante la utilización de la API de GCM se logró analizar correctamente los tiempos reales de este tipo de aplicaciones siendo tiempos menores en relación a la API anterior de C2DM, ya que en la obtención de un Token que se realiza solo una vez al día en un promedio de 5,951 segundos y en el envío de mensajes un promedio de 0,181314763 segundos dependiendo de la red. Por lo tanto, Mobile Cloud Computing es una nueva tendencia que mediante la utilización de servicios en la nube está incursionando en los últimos años y que servicios como Google Clould Message permiten dar un paso más a la hora de hablar de aplicaciones móviles.

Keywords— Google App Engine, C2DM, GCM, Token

Abstract

This article presents an analysis of GCM (Google Cloud Messaging); Library performance, integration with Google App Engine and also development tools, including API. The results reveal that GCM in relation to its predecessor C2DM (Cloud to Device Messaging) presents significant improvements that must be considered when developing Apps in real time that facilitates and enhances the development of increasingly powerful, reliable, Fast and above all optimizing resources. Therefore, using the GCM API could correctly analyze the actual times of this type of applications being shorter times in relation to the previous API of C2DM, since in obtaining a Token that is performed only once Per day in an average of 5.951 seconds and in the sending of messages an average of 0.181314763 seconds depending on the network. Therefore, Mobile Cloud Computing is a new trend that using services in the cloud is penetrating in recent years and services such as Google Clould Message allow to take a step further when talking about mobile applications.

Keywords— Google App Engine, C2DM, GCM, Token

I. INTRODUCCIÓN

Internet usualmente se visualiza y conceptualiza como una gran nube donde todo está conectado y donde al conectarse se suministran todos los servicios requeridos. A este esquema de trabajo se lo denomina Cloud Computing, un modelo de aprovisionamiento de recursos IT que potencia la prestación de servicios IT y servicios de negocio, facilitando la operativa del usuario final y del prestador del servicio[1].

Por otro lado, en los últimos años, las aplicaciones dirigidas a los dispositivos móviles han comenzado a ser abundante, con aplicaciones en diversas categorías como entretenimiento, salud, juegos, negocios, redes sociales, viajes y noticias. [5]Debido a las exigencias y a las características limitadas de los dispositivos móviles, se hace necesario depender de la cloud para la distribución de los servicios al dispositivo móvil.

La convergencia de Cloud Computing y la Internet Móvil ha permitido el desarrollo del Mobile Cloud Computing, el cual más que una tecnología es una filosofía de trabajo. MCC se puede definir como:

"Un servicio que permite a los usuarios móviles con recursos limitados para ajustar de forma adaptativa capacidades de procesamiento y almacenamiento mediante la separación y la descarga de los puestos de trabajo computacionalmente intensivas y almacenamiento exigentes en recursos de la nube tradicionales al proporcionar acceso inalámbrico omnipresente transparente".[2]

Según la investigación “Mobile Cloud Computing” de ABI Research [3], cloud será la tendencia dominante en este espacio. Utilizando tecnologías cloud, se crearán aplicaciones más sofisticadas que serán usadas por mayor cantidad de suscriptores. Las aplicaciones comenzarán a almacenar los datos en la cloud en lugar de un dispositivo móvil ya que serán más poderosas debido a que el poder de procesamiento también es descargado de la cloud.

El estudio de Juniper Research, establece que se espera que el mercado de consumo y empresarial para las aplicaciones móviles basadas en la nube a la altura de $ 9.5 mil millones en 2014[4]

Los puntos fuertes de la computación en nube pueden ser descritos en términos de los servicios ofrecidos por los proveedores de servicios en la nube: software como servicio (SaaS), plataforma como servicio (PaaS) e infraestructura como servicio (IaaS)[6]

En los últimos años, MCC ha sido un área de investigación activa. Como MCC está en las etapas preliminares, por tanto, este articulo científico examinará una tecnología llamada GCM (Google Cloud Messaging) y lo bien que se integra con la computación en nube. En esta investigación nos fijaremos en el rendimiento de la biblioteca, la integración con Google App Engine y también las herramientas de desarrollo, incluyendo la API. Crearemos una aplicación mediante GCM y haremos pruebas iniciales de rendimiento. En si nuestro objetivo es proporcionar una mejor comprensión de este servicio ampliamente utilizado. También es esencial para entender el tiempo real propiedades de GCM para los investigadores que trabajan en sistemas móviles como multitud de abastecimiento, la respuesta a preguntas y otros sistemas de tiempo real.

II. MENSAJERÍA DE NUBE A DISPOSITIVO (C2DM) Y GOOGLE CLOUD MESSAGING (GCM)

C2DM se puso en marcha en 2010 para que las aplicaciones Android envíen datos desde los servidores a sus aplicaciones. El servicio C2DM maneja todos los aspectos de la puesta en cola de mensajes y la entrega a la aplicación de destino que se ejecuta en el dispositivo de destino.

La nube de Android para dispositivos de mensajería (C2DM) está en desuso. El servicio C2DM se cerró por completo el 10/20/2015. Actualmente, C2DM no acepta los nuevos usuarios, y no otorga nuevas cuotas. Se recomienda a los desarrolladores C2DM para mover a Google Cloud Messaging (GCM).GCM es la próxima generación de C2DM.

GCM reemplaza C2DM. El enfoque de GCM es el siguiente:

- Facilidad de uso. No hay formularios de registro.
- No hay cuotas.
- GCM y C2DM estadísticas están disponibles a través de la consola de desarrollo.
- eficiencia de la batería.
- Rico conjunto de nuevas API. [13]

III. GOOGLE CLOUD MESSAGING

GCM es un servicio que facilita la mensajería entre aplicaciones para dispositivos móviles y aplicaciones de servidor. Permite a los desarrolladores para enviar empujar mensajes a dispositivo Android desde el servidor. Google Cloud Messaging es un servicio popular como cliente / servidor solución de comunicación para la plataforma Android [7], [8].

IV. TRABAJOS RELACIONADOS

Hoy en día, todos los sistemas operativos móviles más populares apoyan empuje servicios de notificación [9], [10], [11], [12]. No está aún en curso trabajar para optimizar la publicación / suscripción modelo de constante desarrollo tecnología móvil. Por ejemplo, Mobius [12] de Yahoo! La investigación y el MIT utiliza el almacenamiento en caché y hace un uso más eficiente de hardware ad hoc para ese propósito. MobilePush [8] propone un diseño para un sistema a escala de Internet móvil que soporta los dispositivos conectados a la LAN inalámbrica y para la red GPRS, y en [10], los autores presentan una evaluación de su móvil publicación / suscripción de ajuste. A lo mejor de nuestro conocimiento, sólo un estudio retrospectivo que evalúa el rendimiento del empuje servicios de notificación para los sistemas operativos móviles más comunes [11].

En este trabajo, el rendimiento de la nube al dispositivo Android Mensajería (C2DM), es decir, el predecesor en desuso de GCM, se compara con algunos otros servicios de notificación de inserción, que se pueden configurar para trabajar con clientes de Android. Sus hallazgos revelan que, C2DM recibe alrededor de 4 a 6 segundos el tiempo de respuesta dado que el dispositivo

que recibe la notificación de inserción inicia el mensaje, es decir, la notificación de inserción rebota para el mismo dispositivo. Por lo tanto, sus resultados se limitan a de comunicaciones periódicas a un solo dispositivo, y carece comparativa análisis de rendimiento de las notificaciones push en masa.

V. DESCRIPCIÓN DE LA ARQUITECTURA DE GCM

Google Cloud Messaging para Android (MCG) es un servicio que le permite enviar datos desde su servidor a los dispositivos Android de los usuarios cubriendo aspectos como de almacenamiento, hacer cola y la entrega de los mensajes a la aplicación Android de destino.

A. Descripción de la arquitectura

Una aplicación GCM incluye un servidor de Google conexión, un servidor de aplicaciones en el entorno que interactúa con el servidor de conexión a través de HTTP o el protocolo XMPP, y una aplicación cliente.

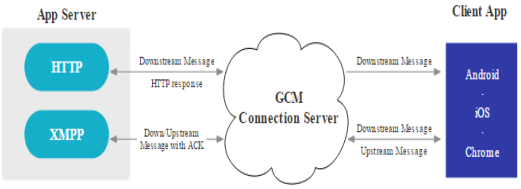


Figura 1: Arquitectura

B. Cómo interactúan estos componentes:

Google servidores de conexión: GCM aceptan mensajes desde el servidor de aplicación y los envían a una aplicación de cliente. El XMPP servidor de conexión también puede aceptar mensajes enviados desde la aplicación cliente y desviarlas a su servidor de aplicaciones.

En el servidor de aplicaciones: se implementa el HTTP y / o XMPP protocolo para comunicarse con el servidor (s) de conexión GCM. Servidores de aplicaciones para enviar mensajes a un servidor de conexión GCM; los encola servidor de conexión y almacena el mensaje y lo envía a la aplicación cliente. Si implementa XMPP, el servidor de aplicación puede recibir los mensajes enviados desde la aplicación cliente.

El cliente de aplicación es una aplicación cliente de GCM-habilitado. Para recibir y enviar mensajes de GCM, esta aplicación debe registrarse con GCM y obtener un identificador único llamado una ficha de registro. [14]

C. Interacción con GCM

1. El Dispositivo Android envía ID del remitente, el id de aplicación al servidor de GCM para su registro.
2. El servidor de GCM da un ID de registro al dispositivo Android.
3. El Dispositivo Android envía este ID de registro al servidor local.
4. El servidor local almacena el ID de registro en la base de datos para su uso posterior.

a) Cuando se proporciona una notificación a través de la página web, el servidor envía el mensaje al servidor de GCM junto con el ID registrado.

b) GCM servidor envía el mensaje al dispositivo usando particular, que la ID de registro.

Google Cloud Messaging como se muestra en la Figura 2 actuará como la principal plataforma. El usuario primero se registra en la plataforma GCM y recibirá una ficha de identificación el cual se almacenará en el servidor que identifica al usuario del teléfono sobre la base de que el ID. El servidor es donde la organización va a tener un cliente web basado en PHP que se va a enviar el notificaciones a los individuos con respecto a su ID de registro.[15].

Figura 2. Muestra el proceso general de registro y envío de una notificación se produce. El usuario los registros de la GCM en el paso 1 del GCM proporciona el ID de registro en el paso 2. En el paso 3 y 4 de las tiendas de móviles en la identificación en el servidor y el paso a. Y b. son las fases en las que el servidor está enviando la notificación al teléfono a través de la arquitectura GCM.[16]

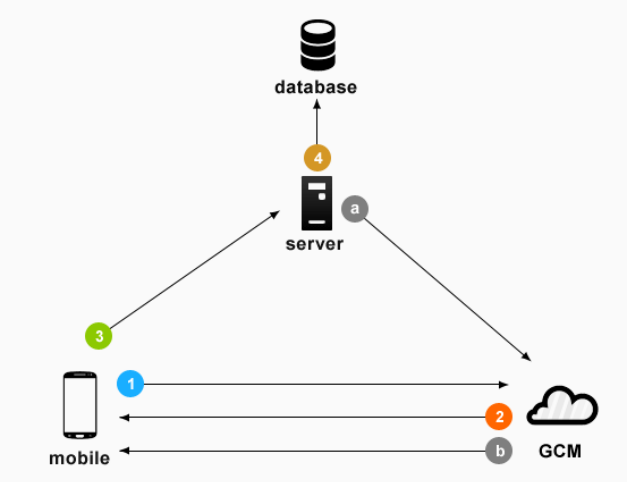


Figura 2: Trabajando con GCM

D. Dispositivo Android

La aplicación Android es la interfaz principal mediante la cual el usuario va a utilizar para recibir notificaciones Push. A continuación, se enumeran los requisitos importantes para el desarrollo de la aplicación:

- Aplicación de Cliente [17]: Esta aplicación está desarrollada utilizando estudio de desarrollo de Android en junto con herramientas de SDK.
- Herramientas de SDK [17]: Las herramientas del SDK de Android compilar el código junto con todos los datos y recursos archivos. El SDK de Android proporciona las herramientas y APIs necesarios para comenzar a desarrollar aplicaciones en la plataforma Android usando el lenguaje de programación Java.
- .apk [17]: todo el código en un solo archivo .apk se considera que es una aplicación y es el archivo que los dispositivos con Android utilizan para instalar la aplicación.
- Sistema operativo Android [17]: Android es un sistema operativo basado en Linux diseñado principalmente para el tacto dispositivos móviles de pantalla, como los teléfonos inteligentes y Tablet PC.

Se permite reemplazar y La reutilización de componentes.

E. Servidor de aplicaciones:

La aplicación de servidor es una parte esencial de las notificaciones push siendo la principal la actividad del servidor proporcionar el administrador de la facilidad de enviar el mensaje al cliente y recibir un acuse de recibo por parte del cliente que se ha recibido el mensaje. El componente necesario para configurar un servidor se enumera a continuación:

- Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.[18]

F. Componentes y Credenciales de GCM:

Componentes:

Componente	Descripción
Conexión GCM	Es el servidor de envío de mensajes de Google entre el servidor y el cliente
Aplicación Cliente	Aplicación cliente con habilitado para GCM
Aplicación Servidor	Esta aplicación servidor envía los datos a una aplicación cliente utilizando el servidor de conexión GCM.

Credenciales:

Credencial	Descripción
Sender ID	Este ID se utiliza en el proceso de registro para autenticar el servidor de aplicaciones para enviar mensajes a la aplicación cliente.
API Key	Una clave de API guardada en la aplicación servidor que proporciona el servicio de Google.
Application ID	Id que recibe una aplicación cliente que se está registrando para recibir mensajes.
Registration Token	Una identificación dada por la conexión GCM para la aplicación cliente que permite que le permite recibir mensajes.

VI. PRUEBAS

Dentro de esta investigación fue de interés el rendimiento de la biblioteca, la integración con Google App Engine y también las herramientas de desarrollo, incluyendo la API. Se creó una aplicación implementando GCM misma que proporciona una mejor comprensión de este servicio y esto permitió la realización de pruebas de rendimiento.

GCM identifica los dispositivos Android mediante fichas de registro ('tokens'). Los 'tokens' de registro se actualizan cada día. Por eso, cada app Android que use GCM debe tener una InstanceIDListenerService que gestione esas actualizaciones. Es por esto que dentro de nuestra aplicación existe una clase Java llamada TokenRefreshListenerService.java, una subclase de InstanceIDListenerService.

```
public class TokenRefreshListenerService
extends InstanceIDListenerService {
    @Override
    public void onTokenRefresh() {

        Intent i = new Intent(this,
RegistrationService.class);
        startService(i);

    }
}
```

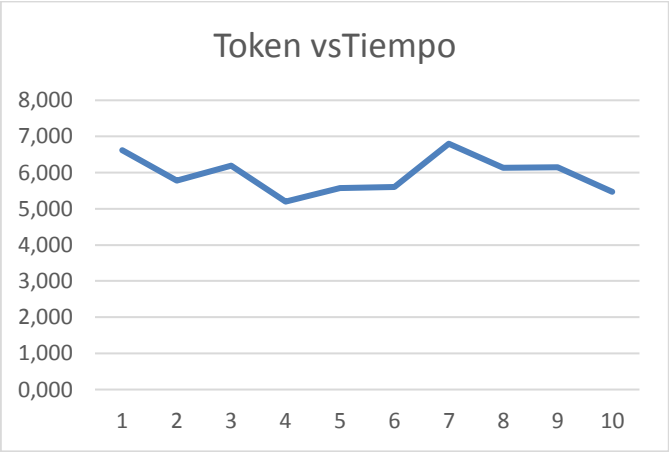
Mediante esta clase nuestra aplicación se conecta al servicio de google y obtiene un identificador para la aplicación mismo que mediante el proceso de bluid en Android Studio puede ser medio el tiempo de respuesta que demora en obtener un Nuevo Token. La tabla 1 muestra el tiempo de respuesta de este proceso en segundos.

Tabla 1: datos de obtención de nuevo token.

N	Token	Tiempo
1	fTSb3QvC85w:APA91bFJ3mv2-...	6,617
2	fwi9RVOeor8:APA91bFVR6-...	5,783
3	dcwK357hkq4:APA91bEBBQJAcmc_...	6,196
4	cBy-qyF6cYc:APA91bFw4	5,198
5	cJJKnnrLT3o:APA91bHFe_	5,568
6	dfZETD7n48w:APA91bHtn..	5,600
7	flykm4mv3wU:APA91bGS6S..	6,799
8	fsVL-x3xfCM:APA91bGC-...	6,127
9	fYMEsbWtSGk:APA91bF-...	6,151
10	dy0pSVbeecc:APA91bF7_..[1]	5,474

Grafica 1: Token vs Tiempo

[1] Los tokens obtenidos son cadena de caracteres extensa por tal motivo se toma los primeros caracteres de la cadena.



Como se puede observar en la tabla 1 los tiempos de respuesta por parte del servidor de Google para la asignación de un nuevo token para nuestra aplicación son relativamente bajos siendo la media de estos tiempos medidos en segundos igual a 5,951 procesos que realiza GCM diariamente en segundo plano, al ser tiempos relativamente bajos ya que su máximo es 6.799 segundos y su mínimo de 5,198 segundos mismo que dependen del tráfico de la red esto permite que las aplicaciones con GCM sean útiles en tiempo real.
Para la siguiente prueba una vez finalizada la aplicación cliente e implementada la aplicación de servidor en lenguaje Python para poder enviar mensajes desde el servidor al cliente se ejecutará la misma.

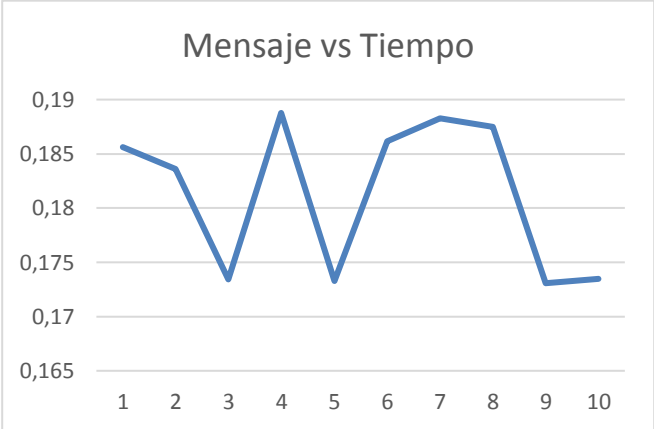
```
from timeit import timeit
from urllib2 import *
import urllib
import json
import sys
MY_API_KEY="AIzaSyDFd2eNoun3STu20cqbACFbx6B3B0-sKM0"
messageTitle = sys.argv[1]
messageBody = sys.argv[2]
data={
   "to" : "/topics/my_little_topic",
   "notification" : {
      "body" : messageBody,
      "title" : messageTitle,
      "icon" : "ic_cloud_white_48dp"
   }
}
dataAsJSON = json.dumps(data)

request = Request(
   "https://gcm-http.googleapis.com/gcm/send",
   dataAsJSON,
   { "Authorization" : "key="+MY_API_KEY,
     "Content-type" : "application/json"
   }
)

print urlopen(request).read()
print(timeit(urlopen(request).read()))
```

En este punto, estamos listos para enviar notificaciones push a todos los dispositivos en los que esté instalada nuestra app. Obteniendo como resultado de haber enviado exitosamente el mensaje el id de nuestro mensaje y el tiempo que le lleva realizar el proceso hasta llegar a la aplicación cliente. En la tabla 2 se presenta los resultados del envío de mensajes medido en segundos.

Tabla 2: Datos de tiempo de envío de notificaciones Push.

N	Mensaje ID	Tiempo
1	{"message_id":5878589687675917119}	0,185626646
2	{"message_id":6030502813993436770}	0,183578135
3	{"message_id":7698771505396956396}	0,173434642
4	{"message_id":5435391298791731076}	0,18876691
5	{"message_id":7633299173569349958}	0,173273471
6	{"message_id":6309391902108523131}	0,186146088
7	{"message_id":8652302404126376398}	0,188280831
8	{"message_id":4637924330646485496}	0,187480111
9	{"message_id":7297079743157580924}	0,173074317
10	{"message_id":8270096040601687783}	0,173486483

Grafica 2: Mensajes vs Tiempo



En relación a la tabla 1 donde la obtención de un token lleva un poco más de 5 segundos en la tabla dos se puede observar que el envío de mensajes desde el servidor al cliente es más corto aun lo que permite una eficacia significativa a la hora de la implementación de este servicio(GCM) siendo así que la media de los tiempos de envío de notificaciones push hacia los dispositivos es de 0,181314763 medido en segundos, siendo el punto máximo de 0,188766910 y un punto mínimo de 0,173074317 segundos lo que conlleva a mayor comunicación en aplicaciones de tiempo real.

VII. CONCLUSIÓN

Google Cloud Messaging ofrece una gran comodidad y una manera flexible para manejar mensajes entre el cliente y el servidor en aplicaciones móviles (Mobile Cloud Computing) siendo la siguiente generación de C2DM actualmente en desuso por tanto es importante proporcionar información para lograr una mejor comprensión de esta nueva generación. En nuestra investigación se analizó el rendimiento de la biblioteca, la integración con Google App Engine y también las herramientas de desarrollo, incluyendo la API.

En cuento a la biblioteca de google se puede decir que proporciona eficazmente información en relación a la migración de C2DM hacia CGM, llevando a que las aplicaciones con este servicio sean desarrolladas eficazmente mediante la documentación oficial de Google para desarrolladores. Siendo esta la biblioteca más eficaz al momento de implementar este servicio que ofreces beneficios con relación a su versión anterior como a la hora de hacer cola y entrega de mensajes.

Mediante el desarrollo de esta aplicación implementando este servicio se logró entender y comprender de manera ágil la arquitectura que ofrece Google Cloud message, muestra de ello es que se logró mediante Google APP Engine (plataforma que ofrece los servicios de Google) implementar correctamente la aplicación-servido. En sí, se puede decir que Google APP Engine es una plataforma que ofrece grandes servicios en este caso GCM que ayudan a generar aplicaciones potentes optimizando recursos y tiempo importantes para las aplicaciones móviles, además que a través de la implementación de los servicios de esta plataforma se puede generar software de calidad para aplicaciones de tiempo real en conjunto con un IDE como Android Studio que provee las herramientas necesarias para estos servicios y Python un lenguaje con el que se creó la aplicación de servidor con lleva a decir que actualmente existen herramientas óptimas para el desarrollo.

Por ultimo, mediante la utilización de la API de GCM en nuestra aplicación se logró analizar correctamente los tiempos reales de este tipo de aplicaciones siendo tiempos menores en relación a la API anterior de C2DM, ya que en la obtención de un token que se realiza solo una vez al día en un promedio de 5,951 segundos y en el envío de mensajes un promedio de 0,181314763 segundos dependiendo de la red podemos darnos cuenta que es una API útil para el desarrollo de aplicaciones móviles de tiempo real que a su vez facilita y potencia el desarrollo de aplicaciones cada vez más potentes, fiables, rápidos y sobre todo optimizando los recursos de un Smartphone. Lo que conlleva a concluir que Mobile Cloud Computing es una nueva tendencia que mediante la utilización de servicios en la nube está incursionando en los últimos años y que servicios como Google Clould Message permiten dar un paso más a la hora de hablar de aplicaciones móviles.

VIII. REFERENCIAS

[1] Murazzo, M. A., & Rodríguez, N. R. (2010). Mobile cloud computing. In XII Workshop de Investigadores en Ciencias de la Computación.

[2] Khan, A. N., Kiah, M. M., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. Future Generation Computer Systems,29(5), 1278-1299.

[3] ABI Research, Mobile Cloud Computing, http://www.abiresearch.com/research/1003385

[4] S. Pérez, nube de computación móvil: $ 9.5 mil millones para el año 2014, http://exoplanet.eu/catalog.php de 2010.

[5] Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. Future Generation Computer Systems, 29(1), 84-106.

[6] J. Carolan, S. Gaede, J. Baty, G. Brunette, A. Licht, J. Remmell, L. Tucker, J. Weise, Introduction to cloud computing architecture—white paper, 2009.

[7] "Push Notifications for Windows Phone." [Online]. Available: http://msdn.microsoft.com/enus/library/ff402537(v=VS.92).aspx

[8] "Push Service_ BlackBerry Developer." [Online]. Available: https: //developer.blackberry.com/services/push/?CPID=PUSHAPI00

[9] "Google Cloud Messaging for Android — Android Developers." [Online]. Available: http://developer.android.com/google/gcm/index.html

[10] "Local and Push Notification Programming Guide: Apple Push Notification Service." [Online]. Available: http://developer.apple. com/library/mac/\#documentation/NetworkingInternet/Conceptual/ RemoteNotificationsPG/ApplePushService/ApplePushService.html

[11] "Push Notifications for Windows Phone." [Online]. Available: http://msdn.microsoft.com/en-us/library/ff402537(v=VS.92).aspx

[12] "Push Service - BlackBerry Developer." [Online]. Available: https: //developer.blackberry.com/services/push/?CPID=PUSHAPI00

[13] "Google cloud Messaging" [online] Available: https://developers.google.com/cloud-messaging/c2dm#history

[14] "Google Cloud Messaging: Overview" [online] Available: https://developers.google.com/cloud-messaging/gcm#send-msg

[15] AndroidHive, "Android Push Notifications using Google Cloud Messaging" [Online]. Available: http://www.androidhive.info/

[16] Ganatra, N., & Patel, R. GOOGLE CLOUD MESSAGING (GCM): Alight WEIGHT COMMUNICATION MECHANISM.

[17] Tutorialspoint: "Android-Architecture" [Online]. Available: http://tutorialspoint.com/android/android_architecture

[18]Van Rossum, G., & Drake Jr, F. L. (2004). Tutorial Python. Disponível gratuitamente em http://python. org.