**TABLA DE CONTENIDO**

1. **Introducción ……………………………………………………………………………………………………………………………………. 2**
2. **Justificación …………………………………………………………………………………………………………………………………….. 2**
3. **Alcance de la evaluación …………………..…………………………………………………………………………………………….. 2**
4. **Marco Teórico ……………………………………………………………………………………………………………………………....... 2**
   1. **Computación Móvil………………………………………………………………………………………………………………….. 2**
   2. **Comunicación entre dispositivos móviles en tiempo real …………………………………………………………. 3**
   3. **Patrón *Publish-Suscribe*……………………………………………………………………………………………………………. 4**
5. **Implementaciones …………………………………………………………………………………………………………………………… 5**
   1. ***Google Cloud Messaging* (GCM) ……………………………………………………………………………………………….. 5**
   2. ***Message Queue Telemetry Transport* (MQTT) ………………………………………………………………………….. 6**
   3. ***Extensible Messaging and Presence Protocol* (XMPP) ……………………………………………………………….. 6**
6. **Conclusiones ……………………………………….................................................................................................. 7**
7. **Introducción**

La idea del estudio es evaluar diferentes técnicas de mensajería “*machine to machine*” (M2M) entre dispositivos móviles, las cuales tienen como funcionalidad permitir que desde un servidor central exista una comunicación constante con una máquina, con el fin de solicitar y/o evaluar información que será procesada y a partir de ella poder comunicar al dispositivo mediante mensajes cortos los cuales generan un consumo mínimo de datos. Se desea que quien acceda a este documento tenga un punto de partida para la elección de la tecnología más acorde para el envío de mensajes M2M a partir de los pequeños ejemplos funcionales con las principales tecnologías y el comparativo entre ellas presentado por este proyecto.

1. **Justificación**

El objetivo de la realización de un estudio que permita evaluar las diferentes tecnologías usadas para envío de mensajes cortos entre dispositivos móviles es brindar un punto de partida frente a la elección de la tecnología más acorde según los requerimientos específicos de los proyectos a partir de ejemplos prácticos que brinden una visión más detallada de la forma en que se implementan las diferentes tecnologías y que tipo de soluciones pueden aportar a las necesidades de proyectos de desarrollo específicos.

1. **Alcance de la evaluación**

Según las investigaciones realizadas sobre las tecnologías se quieren evaluar las bondades ofrecidas por cada una de ellas según enfoques particulares para proyectos que en su desarrollo impliquen la comunicación con dispositivos sin necesidad que este se encuentre constantemente realizando verificaciones frente a un servidor para obtener algún tipo de respuesta o notificación; sino que esta tarea se encuentre delegada al servidor y que se encargue de enviar algún tipo de información según el contexto en que se encuentre al dispositivo.

1. **Marco teórico**
   1. **Computación móvil:**

Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.

De esta forma se define la computación móvil como la posibilidad de movilizar estos dispositivos de un espacio físico a otro sin ningún tipo de restricción y sin la necesidad de conexión mediante cables, sino a partir de redes inalámbricas o satelitales.

La computación móvil ha permitido un amplio avance en el procesamiento y manejo de la información por parte de las personas, debido a que mediante ella se tiene acceso a la información en el momento en que se desea por medio de dispositivos móviles (teléfonos inteligentes, tabletas, computadores portátiles, etc.). Los cuales sin alojar necesariamente toda la información solicitada por el usuario, permite mediante una simple conexión a un servidor central donde dicha información se encuentra almacenada y que permite la actualización, consulta, procesamiento y mantenimiento continuo de la misma. Para que desde los dispositivos mediante una interfaz gráfica los usuarios puedan realizar peticiones a dicho servidor para la consulta y/o modificación de la misma en tiempo real.

* 1. **Comunicación entre dispositivos móviles en tiempo real**
* ***Machine-to-Machine* (M2M):**

La comunicación M2M consiste en el intercambio automático de datos entre máquinas, equipos, módulos individuales y sistemas completos, sin ningún tipo de intervención humana, con el fin de a partir de conocimiento generado desde la máquina servidor se pueda transmitir a la máquina cliente (dispositivo) información específica de su interés. En ella se incluyen la transmisión de mediciones por **SMS** (*Short Message Service*), **GPRS** (*General Packet Radio Service*) y **UMTS** (*Universal Mobile Telecommunications System*), la radiotransmisión de datos de consumo y el acceso y control remoto de dispositivos.

Ha sido sobre todo la rápida y rentable transmisión datos a través de las redes de telefonía móvil lo que ha dado un importante impulso a la comunicación M2M.

* ***Internet of Things* (IoT)**

Se basa en la interconexión mediante internet de dispositivos electrónicos integrados a la vida cotidiana donde cada uno de ellos puede ser identificado y gestionado por otros sin necesidad de interacción humana.

La expansión de los dispositivos inteligentes y objetos conectados, produjo un cambio en la forma en que los consumidores y las compañías interactúan con la tecnología. La Internet de las cosas, que conjuga dispositivos móviles, objetos inteligentes y tecnologías embebidas, con Internet y la Computación en la nube, conectados mediante NFC (*Near Field Communication*), WiFi, Bluetooth, o radiofrecuencia, fue el paso de la evolución hacia una sociedad más conectada. Internet de las cosas nos permite recibir información en cualquier momento y lugar, entre diversos emisores y receptores (*peers*).

Una asociación internacional de grandes organizaciones de normalización, llamada oneM2M[[1]](#footnote-1), una solución M2M para realizar una "combinación de dispositivos, software y servicios que funcionan con una intervención humana escasa o nula"[[2]](#footnote-2).

* **Mensajes Push**

Es un mensaje que una aplicación servidor envía a una aplicación (móvil) cliente indicándole que tiene algún tipo de información nueva disponible. Lo que distingue a las notificaciones pushes que el servidor es quien inicia la comunicación, no espera a que el cliente pregunte si hay algo nuevo. La ventaja es la inmediatez: la información llega al cliente (y por tanto al usuario) en cuanto está disponible en el servidor.

* **Protocolo MQTT**

Es un protocolo de conectividad enfocado a M2M y al IoT ya que se ha diseñado para ser un protocolo de mensajería extremadamente ligero basado en TCP. Es útil para conexiones con sitios remotos donde es fundamental un pequeño *footprint[[3]](#footnote-3)* (como Arduino[[4]](#footnote-4)) y el ancho de banda es muy importante, debido a su bajo consumo.

Una característica muy importante es que al ser un protocolo ligero existen clientes y servidores MQTT en diversos lenguajes[[5]](#footnote-5).

* 1. **Patrón *Publish-Suscribe***

Es un patrón de mensajería donde quienes envían los mensajes (Editores) no programan directamente los mensajes para que sean enviados directamente a receptores específicos (Suscriptores). Sino que los mensajes publicados se separan en clases para que los Suscriptores expresen interés en una o más clases y de esta forma sólo recibirán mensajes de su interés.

1. **Implementaciones**
   1. ***Google Cloud Messaging* (GCM):**

Es un servicio que permite enviar datos desde un servidor a dispositivos cliente[[6]](#footnote-6), al igual que recibir mensajes de dichos dispositivos mediante la misma conexión.

Este servicio provee un mecanismo que el servidor puede usar para decirle a las aplicaciones móviles que contacten directamente al servidor.

Es un servicio totalmente gratuito, Google define como restricción que los mensajes no pueden tener un tamaño superior a 4Kb.

Para usar el servicio que provee GCM se debe generar un token permanente[[7]](#footnote-7), con el fin de poseer un identificador único que será usado posteriormente como el ID del remitente.

**Ventajas:**

* Es un servicio gratuito
* Soportado directamente por Google.
* Comunicación bidireccional.
* 100% compatible con dispositivos Android.

**Desventajas:**

* Limita los mensajes a un máximo de 4Kb.
* Límite de mensajes en cola 100[[8]](#footnote-8).

**Casos de éxito:**

* **Symantec Mobile Security:**

GCM permite a los servidores de Symantec comunicarse con los dispositivos mediante comandos push sin tener que esperar a que el dispositivo se comunique previamente con ellos mediante la aplicación[[9]](#footnote-9).

* **Galegram:** Aplicación de mensajería instantánea en “Gallego”, la cual permite notificaciones de mensajería instantánea en los dispositivos móviles mediante GCM[[10]](#footnote-10).
* Aplicaciones que usan GCM mediante las notificaciones hechas al usuario sin previa solicitud realizada en el dispositivo: Google Plus, Twitter, Pinterest, Facebook, LinkedIn, Instagram, Viber, Line, Skype.
  1. ***Message Queue Telemetry Transport* (MQTT):**

Es un protocolo de conectividad abierto *Machine-to-Machine* (M2M) que permite enviar datos estilo telemetría como mensajes a través de redes de alta latencia o con restricciones. Está diseñado para ser una mensajería *Publish-Suscribe* muy ligera y útil para conexiones donde es importante no consumir mucho ancho de banda (Comunicaciones entre sensores, GPRS, etc.), al igual que para aplicaciones móviles debido a su envío eficiente.

**Casos de éxito:**

* **Facebook Messenger App:** El uso del protocolo MQTT permite enrutar los mensajes del chat en milisegundos[[11]](#footnote-11).
* **IBM WebSphere MQ Telemetry:** Permite mediante el protocolo MQTT el transporte de mensajes robustos incluso para dispositivos muy pequeños de manera casi inmediata[[12]](#footnote-12).

* **IBM Lotus Expeditor Micro Broker[[13]](#footnote-13)**
  1. ***Extensible Messaging and Presence Protocol* (XMPP):**

Es un protocolo abierto que se creó para ser usado en sistemas de mensajería instantánea, basado en XML. Originalmente se conocía como **Jabber**, y el proyecto fue iniciado en 1998 por Jeremie Miller. Actualmente XMPP y sus múltiples extensiones soportan mensajería instantánea, videoconferencia, vista del estado en línea de los usuarios, y transferencia de archivos a través de clientes.

**Casos de éxito:**

Adoptado como estándar para servicios de chat entre los cuales se encuentran:

* WhatsApp: Provee soporte para chats multiusuario, también brinda seguridad para los mensajes durante la capa de transporte.
* Facebook Messenger: Sirve para habilitar el chat de Messenger en aplicaciones de terceros.
* GTalk: Antes de Hangouts se usaba el protocolo XMPP para la autenticación de usuarios y envío de mensajes, así como mensajes de voz y video peer-to-peer.

1. **Conclusiones**

* GCM al ser una tecnología reciente brinda a los usuarios mejoras en el rendimiento de la batería y en el uso de recursos.
* GCM al ser soportado por Google da por sentado que existe personal dedicado a su mejora continua.
* Los sistemas de mensajería ofrecen mensajería orientada a eventos. La aplicación cliente ya no tiene que enviar una solicitud ni estar después comprobando continuamente si hay respuesta. En vez de eso, el servidor de mensajería notifica a la aplicación cliente cuando se produce algún evento por medio de una alerta tan pronto como se produce dicho evento.
* El uso de GCM garantiza el uso del protocolo XMPP permitiendo obtener lo mejor de los dos mundos: Las ventajas de conexiones mediante GCM así como la confiabilidad y seguridad brindada por XMPP.
* El uso del protocolo MQTT es altamente recomendado debido a que es de los más rápidos y fáciles de extender.[[14]](#footnote-14)

El siguiente cuadro contiene cada una de las tecnologías y sus respectivas ventajas y desventajas, el cual puede servir como base para la toma de la decisión de cuál de ellas usar:

|  |  |  |
| --- | --- | --- |
| TECNOLOGÍA | VENTAJAS | DESVENTAJAS |
| GCM | * Servicio soportado por Google. * No se debe preocupar por el desarrollo y la implementación por parte del servidor. | * Limitado para su uso a través de una cuenta de Google. * Soportado para versiones Android superiores a la 2.2. * Su implementación es demasiado compleja para dispositivos con sistema operativo diferente a Android. |
| MQTT | * Es un protocolo simple, compacto, escalable. | * No es muy maduro. * Algunos de sus componentes no son Open Source.[[15]](#footnote-15) |
| XMPP | * Es un protocolo potente que puede ser aplicado a sistemas de chat. | * Protocolo complejo, redundante; lo cual aumenta el consumo de recursos de los dispositivos. |

1. <http://www.onem2m.org/> [↑](#footnote-ref-1)
2. Alain Louchez, Georgia Tech Research Institute, Atlanta, Georgia [↑](#footnote-ref-2)
3. Cantidad de memoria de la máquina que usa un programa cuando se está ejecutando. http://en.wikipedia.org/wiki/Memory\_footprint [↑](#footnote-ref-3)
4. http://www.arduino.cc/ [↑](#footnote-ref-4)
5. http://mqtt.org/software [↑](#footnote-ref-5)
6. Originalmente pensada para Android pero también puede ser implementada con otros sistemas operativos como iOS de Apple aunque es un proceso mucho más complejo: <https://cloud.google.com/developers/articles/ios-push-notifications/> [↑](#footnote-ref-6)
7. Clave proporcionada por la API de google. [↑](#footnote-ref-7)
8. Cuando se supera el límite de mensajes en espera se empieza a descartar el envío de los mensajes según su antigüedad. [↑](#footnote-ref-8)
9. <http://www.symantec.com/business/support/index?page=content&id=TECH191945> [↑](#footnote-ref-9)
10. <http://www.eldiario.es/turing/redes_sociales/galegram_0_240076240.html> <https://play.google.com/store/apps/details?id=org.telegram.galegram.release&hl=es> [↑](#footnote-ref-10)
11. <https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920> [↑](#footnote-ref-11)
12. <http://www-03.ibm.com/software/products/en/wmq-telemetry> [↑](#footnote-ref-12)
13. <http://www-03.ibm.com/software/products/en/expeditor> [↑](#footnote-ref-13)
14. <http://www.programering.com/a/MDO4cDNwATg.html> [↑](#footnote-ref-14)
15. <https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=d5bedadd-e46f-4c97-af89-22d65ffee070> [↑](#footnote-ref-15)