

Análisis Comparativo de Transferencia Asíncrona de Imágenes codificadas en RFC3548 y MIME en Android

Comparative Analysis of Asynchronous Transfer of Images encoded in RFC3548 and MIME in Android

Ms. Johnny Alexander Salazar Cardona¹
Juan David Montoya Montealegre²

Recibido: 02/18/2016 - Aceptado: 04/30/2016

Cómo citar este artículo: J. Salazar y J. Montealegre, “Análisis Comparativo de Transferencia Asíncrona de Imágenes codificadas en RFC3548 y MIME en Android”, *IngEam*, vol. 3, n.º 3, pp. 14 - 21, 2016

Resumen

La transferencia de elementos multimedia como imágenes en los dispositivos móviles es un componente que es utilizado masivamente por los usuarios de este tipo de tecnologías, cargando y descargando imágenes para diferentes propósitos como redes sociales, entretenimiento, educación y ocio. Para el envío de este tipo de información se aplican diferentes métodos, entre los que destaca el envío utilizando codificación (codificar el archivo a una cadena de caracteres, enviar dicha cadena y decodificarla al lado del servidor). El presente artículo busca comparar dos enfoques de transferencia aplicando codificación Base64 como RFC3548 y MIME, utilizándolos en el sistema operativo Android para dispositivos móviles, y así determinar cuál de estos es más eficiente en este tipo de tecnologías.

Palabras clave: envío imágenes, base64, Android, dispositivos móviles, multimedia

Abstract

The transfer of multimedia elements as images on mobile devices is a component that is used massively by users of such technologies, loading and unloading images for different purposes such as social networking, entertainment, education and leisure. For sending this type of information different methods are applied, among them the sending using coding (encode the file to a string of characters, send that string and decode it next to the server). This article attempts to compare two transfer approaches applying Base64 coding as RFC3548 and MIME, using them in the Android operating system for mobile devices, and thus determine which of these is more efficient in this type of technology.

Keywords: sending images, base64, Android, mobile devices, multimedia

¹ Correo electrónico: alexander9052@gmail.com

² Correo electrónico: jdm@eam.edu.co

Introducción

Desde la aparición del internet, todos los servicios que involucran elementos multimedia se enfocaron en los computadores, brindando diferentes servicios de comunicación y entretenimiento, como las redes sociales y streaming de video. Cuando se alcanzó cierto auge en los computadores y laptops del momento, las tecnologías móviles como los celulares que para la época era aún una tecnología incipiente, vieron una oportunidad de masificar el acceso a estas áreas utilizando las redes de comunicaciones existentes, sin la necesidad de utilizar cableado o estar limitado a un área específica de corto alcance y cobertura [1].

Las tecnologías móviles actuales como smartphones, tablets y smartwatches se han convertido en una herramienta cotidiana de los seres humanos, facilitando muchas de las actividades diarias como el acceso a la información, comunicación, aprendizaje, distribución multimedia y ocio. Toda esta información multimedia que se transmite en la actualidad a través de redes sociales, foros, educación virtual y cualquier enfoque de comunicación masiva hace que se generen múltiples formatos de archivos, siempre buscando una mayor calidad de contenido y menor tamaño para agilizar el acceso a este. Además, se generan diferentes métodos de transmisión para estos archivos multimedia, buscando seguridad y agilidad en su transferencia.

Es por esto que en el presente artículo se busca evaluar 2 métodos de transferencia en Android, el RFC3548 y MIME, en el cual se evidencian los tiempos de los diferentes envíos a un servidor de las imágenes codificadas a Base64. Para lograrlo, se desarrolló una aplicación nativa enfocada en la evaluación de dicho experimento, utilizando librerías oficiales para los

diferentes tipos de codificación. Esta permite tomar una fotografía con el celular donde se encuentre instalada la aplicación, y es enviada 50 veces por cada uno de los métodos evaluados, registrando el tiempo en la transferencia de cada uno de los archivos. Con el fin de dar claridad al proceso realizado, el artículo se encuentra estructurado de la siguiente manera: Primero se realizara un contexto de los enfoques de envío utilizado, descrito en el Marco teórico. Luego se explicara en detalle la metodología utilizada para la ejecución del experimento, profundizando en el proceso experimental, el software y hardware utilizado. Después se mostraran todos los resultados obtenidos y finalmente se expondrán las conclusiones.

Marco teórico

Como explica Zhenxing, et al. [2] existen 2 funciones principales de codificación de datos, uno es para el almacenamiento y/o transferencia de datos limitados a ASCII, y la otra es aplicada para manipular dichos datos con editores de texto [3]. En la categoría de transferencia y almacenamiento se encuentra la codificación a Base64, el cual es un sistema de codificación por posiciones utilizando la mayor potencia de 2 que se puede representar por los caracteres de ASCII, el cual es 64. En resumen se utiliza para representar datos binarios con un formato de cadena ASCII. En este existen diferentes esquemas de codificación, entre los que se encuentran el MIME RFC 2048, RFC 3548, PEM, UTF-7, OpenPGP, IRCu, RFC 4648. Este método como cualquier otro, tiene sus desventajas, entre la que se destaca que un archivo codificado a Base64, aumentara su tamaño según el enfoque utilizado. Otro aspecto relevante, es que si el archivo codificado pesa más de 8kb no podrá ser enviado por GET a través de un navegador, por lo que se debe evaluar otro método de envío como POST [4]. Zhenxing,

et al. [2] también indican que existen diferentes bases para codificar archivos aparte del Base64, como base 2, 8 y 16, pero la Base64 es ampliamente utilizada tecnológicamente hablando, debido a que emplea 52 alfabetos, 10 digitales y 2 símbolos para contar cualquier binario de 6 bits.

A. Codificación RFC 3548

Las codificaciones a Base X son usadas en muchas situaciones para almacenar o enviar datos que por alguna razón se encuentren restringidos a caracteres ASCII [5]. En la codificación a Base X se encuentran múltiples enfoques, entre ellas la RFC 3548, la cual por razones ya estipuladas también se restringe solo a caracteres de ASCII. Este utiliza 65 caracteres, utilizando 6 bits para representar cualquier carácter. De los 65 caracteres, 64 son para representar letras entre mayúsculas y minúsculas, como también números del 0 al 9. El carácter 65 es un '=', utilizado para indicar una función de procesamiento especial [6]. Este tipo de codificación no es totalmente transparente, o por lo menos no para la implementación de la prueba realizada en el marco de este proceso comparativo, debido a que se debe codificar manualmente para el envío de la información y su posterior decodificación en el servidor. Además los archivos codificados por este medio pesan un 33% más respecto al tamaño original [4, 7].

B. Codificación MIME

MIME (Multipurpose Internet Mail Extensions) es un conjunto de especificaciones para la transferencia de datos de forma transparente. Este permite soportar caracteres distintos al ASCII. La librería utilizada para implementar esta codificación fue `httpmime-4.2.5.jar` facilitado por la organización apache, permitiendo el transporte de imágenes. En la implementación de dicha librería permite por

medio de tecnología Java realizar una solicitud HTTP al servidor para que esté mismo se prepare para recibir una imagen en el servidor.

En la documentación de la librería se especifica que utiliza RFC 2048, la cual es de los protocolos SMTP más utilizados por los proveedores de correo para el envío de información con o sin archivos adjuntos, la librería esta licenciada y es muy utilizada en protocolos SMTP, por lo tanto es sugerido para trabajar en internet y en envío de correo con archivos adjuntos a los servidores. El uso de esta librería es de uso libre bajo las licencias Apache, permitiendo envíos de datos de una manera transparente [8].

C. Estado del arte

Actualmente la base de datos indexada por excelencia sobre temas relacionados con ingenierías IEEE (Institute of Electrical and Electronics Engineers), cuenta con un total de 10 artículos relacionados con dispositivos móviles android y transferencia de imágenes con MIME y RFC3548, con publicaciones entre el año 2001 y el 2016. Además estas se encuentran en ubicaciones como Orlando – Florida, Zúrich, Takamatsu, Kuala Lumpur y Palermo.

Entre las investigaciones existentes destacan 2 en particular. Una investigación realizada en Servicios de imagen en red para aplicaciones en telemedicina y la otra en reducción de acceso a memoria para el algoritmo MIME. La primera investigación muestra los resultados obtenidos en una red de servicios médicos en el campo patológico basado en el envío síncrono y asíncrono de imágenes en formato JPG con tamaños entre 4,9kb y 31.3kb para una evaluación diagnóstica a través del protocolo MIME, donde demostró ser una codificación efectiva para tal fin [9]. La otra investigación logro la optimización del

algoritmo MIME para deducir el costo computacional de memoria, siendo este un impacto notorio cuando el procesamiento de las imágenes llega a cantidades considerables [10].

Metodología

Durante el desarrollo del análisis del sistema se identificaron una serie de parámetros experimentales para comprobar la hipótesis planteada. Primero se estableció la tecnología de envío de los archivos y segundo, se determinó que antes de la transferencia del archivo, se podría tomar la imagen con la cámara del dispositivo. También se definieron otros parámetros experimentales durante el diseño, como la arquitectura física y lógica del funcionamiento del sistema para establecer las responsabilidades tanto del cliente como del servidor.

En la etapa de desarrollo experimental se utilizó una clásica arquitectura MVC (Modelo - Vista - Controlador), además se utilizaron librerías externas con el fin de soportar el envío de las imágenes. Para la transferencia de imágenes por el enfoque MIME se utilizó de la librería HTTPMIME versión 4.2.5 de la compañía apache, la cual se encarga de encapsular el archivo de manera transparente y de enviarlo por el método POST al servidor (ver figura 1). Para las imágenes con el enfoque RFC3548 se utilizó la clase Base64 para codificar en texto el archivo a transferir y la librería GSON para enviar la cadena codificada como un objeto JSON. Finalmente con el fin de ejecutar el envío de las imágenes en un proceso paralelo todo fue controlado por un hilo.

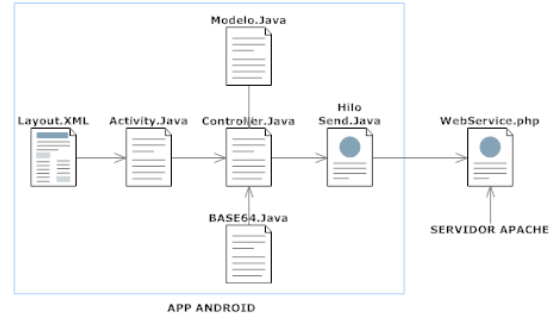


Figura 1 Diagrama de componentes del sistema.

A. Proceso experimental

El diseño experimental establecido para comprobar y comparar la eficiencia de los 2 enfoques establecidos se basa en el envío de una imagen tomada con la cámara del dispositivo, enviándola 50 veces por cada uno de los métodos a comparar.

La imagen fue tomada con una cámara de 5 megapíxeles, ocupa un tamaño digital de 1,26Mb, tiene una resolución de 1496 x 2592 y se encuentra en formato JPG (ver figura 2).



Figura 2 Fotografía enviada.

Luego de tomada la imagen y según la metodología a validar, se realizó un proceso determinado, como la codificación manual o automática, empaquetado HTTP, captura del tiempos desde el dispositivo móvil, recepción de los datos en el servidor para su

almacenamiento o decodificación y almacenamiento (ver figura 3 y ver figura 4).

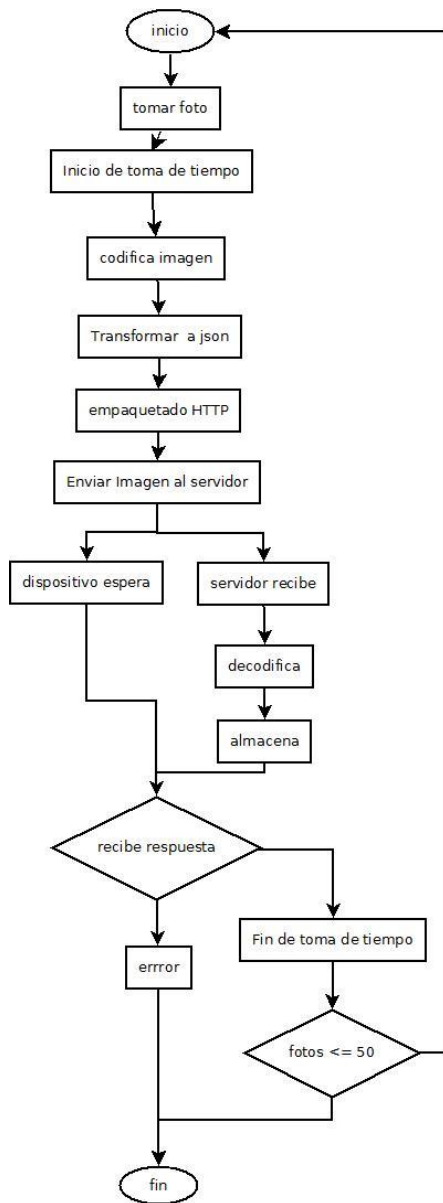


Figura 3 Proceso envío imágenes por RFC3548.

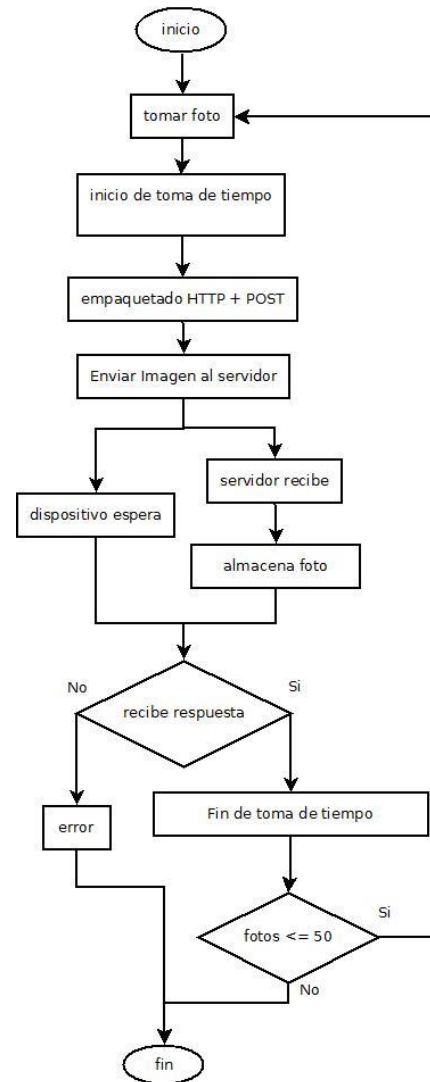


Figura 4 Proceso envío imágenes por MIME.

B. Software utilizado

Las tecnologías usadas para el funcionamiento del proyecto fueron: 1) IDE de desarrollo utilizado fue el eclipse versión Luna 2) Plugin ADT de android versión 23.0.5 3) Wamp versión 2.4 (Apache 2.4.4 y php 5.4.16) 5) Android 4.4.4 API 20.

C. Hardware utilizado

Dispositivo móvil con procesador quad-core 1.2 GHz, GPU andreno 305 y 1GB RAM. El servidor utilizado contaba con un procesador quad core 1.7 GHz, 4 GB en RAM a 1666

MHz con un Sistema Operativo Windows 7 ultimate a 64 Bits.

Resultados

Luego de transmitir 50 veces la imagen establecida en el software desarrollado, se obtuvieron una serie de tiempos para cada uno de los métodos utilizados. Adicionalmente, para la transmisión de las imágenes codificadas con el enfoque RFC3548 se tomó el tiempo que tarda la codificación no transparente de las imágenes en el dispositivo móvil, determinando este proceso de conversión que tanto afecta en la diferencia de resultados. En este se registraron tres tiempos: El tiempo de envío con RFC3548, que es la duración de una determinada transferencia desde el dispositivo móvil codificando la imagen, hasta que se recibe la imagen en el servidor, se decodifica y se almacena. El segundo valor registrado es el tiempo del envío con el enfoque MIME, que corresponde a la transferencia de la imagen hasta que se recibe y se almacena en el servidor. Finalmente el tercer valor capturado es el tiempo que tardo el dispositivo móvil en codificar la imagen a texto desde el dispositivo móvil con RFC3548, con el fin de determinar que tanto tarda esta codificación no transparente en el proceso de transferencia (ver figura 5, 6 y 7). En este apartado destaca el hecho de que la diferencia entre el promedio del tiempo de envío con RFC3548 y el promedio del tiempo del envío MIME es de 1,945 segundos, y el promedio que tarda en codificar una imagen con RFC3548 es de 0,819 segundos.

Nro	Tiempos en Sg. Envio por RFC3548	Tiempos en Sg. Envio por MIME	Sg. Envio por RFC3548 sin tiempo codificacion
Envio 1	20,264	17,344	18,997
Envio 2	19,468	17,301	18,666
Envio 3	19,453	17,322	18,63
Envio 4	19,456	17,302	18,622
Envio 5	19,523	17,352	18,747
Envio 6	19,286	17,421	18,455
Envio 7	19,213	17,683	18,445
Envio 8	19,434	17,482	18,588
Envio 9	19,382	17,829	18,611
Envio 10	19,314	17,51	18,505
Envio 11	19,529	18,091	18,65
Envio 12	19,508	17,399	18,716
Envio 13	19,346	17,638	18,548
Envio 14	19,343	17,49	18,557
Envio 15	19,549	17,64	18,733
Envio 16	19,885	17,504	19,111
Envio 17	19,701	17,394	18,816
Envio 18	19,587	17,571	18,843
Envio 19	19,695	17,94	18,924
Envio 20	19,462	17,414	18,671
Envio 21	19,363	17,497	18,572
Envio 22	19,224	17,362	18,444
Envio 23	19,544	17,466	18,751
Envio 24	19,495	17,389	18,682
Envio 25	19,4	17,38	18,64

Promedio RFC3548: 19,453

Promedio MIME: 17,508

Promedio RFC3548 sin T codificacion: 18,634

Figura 5 Tiempos de envío por método, fotos 1-25.

Nro	Tiempos en Sg. Envio por RFC3548	Tiempos en Sg. Envio por MIME	Sg. Envio por RFC3548 sin tiempo codificacion
Envio 26	19,447	17,576	18,627
Envio 27	19,427	17,471	18,656
Envio 28	19,997	17,721	18,998
Envio 29	19,37	17,739	18,583
Envio 30	19,455	17,64	18,686
Envio 31	19,302	17,413	18,498
Envio 32	18,585	17,648	17,81
Envio 33	19,29	17,442	18,509
Envio 34	19,174	17,503	18,413
Envio 35	19,577	17,475	18,78
Envio 36	19,233	17,458	18,455
Envio 37	19,729	17,473	18,901
Envio 38	19,232	17,846	18,457
Envio 39	19,293	17,449	18,459
Envio 40	19,45	17,484	18,623
Envio 41	19,403	17,509	18,522
Envio 42	19,262	17,368	18,484
Envio 43	19,787	17,408	18,753
Envio 44	19,37	17,401	18,6
Envio 45	19,413	17,382	18,643
Envio 46	19,665	17,299	18,893
Envio 47	19,637	17,35	18,693
Envio 48	19,444	17,465	18,641
Envio 49	19,279	17,321	18,51
Envio 50	19,399	17,844	18,561

Promedio RFC3548: 19,453

Promedio MIME: 17,508

Promedio RFC3548 sin T codificacion: 18,634

Figura 6 Tiempos de envío por método, fotos 26-50.

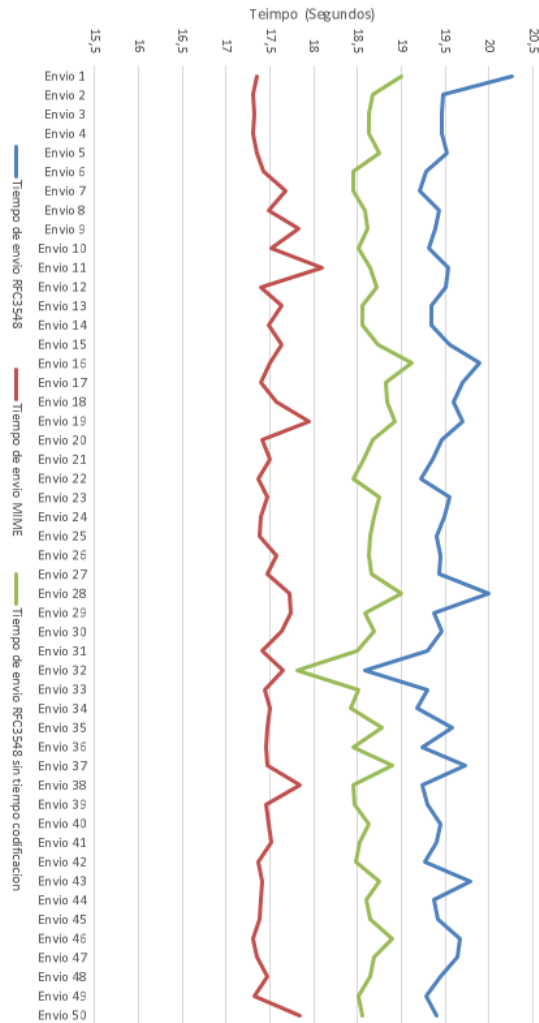


Figura 7 Comparación de rendimiento métodos de envío.

Conclusiones

La transferencia de imágenes codificadas utilizando RFC3548 es más lenta que la transferencia de imágenes con codificación MIME RFC 2048 en dispositivos móviles, tardando cerca de 2 segundos con la imagen establecida para las pruebas realizadas. Estos resultados se deben a que codificar y decodificar una imagen tanto en el dispositivo móvil como en el servidor, es un tiempo adicional en el proceso, y como el envío con RFC3548 no es totalmente transparente hace una fuerte diferencia en el

proceso de codificación en el dispositivo móvil y de decodificación en el servidor.

Cabe resaltar que el mejor tiempo en la transferencia de una imagen codificada con RFC3548, no es tan bueno como el peor tiempo obtenido en la transferencia de la imagen codificada con MIME, teniendo una diferencia cercana a medio segundo. Esto tampoco se da quitando el tiempo completo de la codificación con RFC3548 en el dispositivo móvil, teniendo en cuenta que esta codificación hace que el archivo pese un 33% más que el original y en la codificación sea mayor tiempo en el proceso, aunque este tiempo no fue posible determinarlo con precisión en el servidor [4, 7].

Referencias bibliográficas

- [1]C. Cantillo, M. Roura, and A. Sanchez, "Tendencias actuales en el uso de dispositivos móviles en educación," *Department of Human Development, Education and Culture*, vol. 147, 2012.
- [2]L. Zhenxing, L. Lu, H. Richard, and Z. Yongzhao, "Base62x: An Alternative Approach to Base64 for non-Alphanumeric Characters " *Distributed and Intelligent Systems Research Group*, vol. 11, 2011.
- [3]S. Josefsson. (2006, Diciembre). *The Base16, Base32, and Base64 Data Encodings*. Available: <http://www.ietf.org/rfc/rfc4648.txt>
- [4]PHP Group. (2015, base64_encode. Available: <http://php.net/manual/es/function.base64-encode.php>
- [5]V. Cerf, "ASCII format for Network Interchange," *RFC 20*, October 1969.
- [6]S. Josefsson. (2003, Diciembre). *The Base16, Base32, and Base64 Data Encodings*. Available: <https://tools.ietf.org/html/rfc3548>
- [7] Microsoft. (2015, Diciembre). *A description of Base64 MIME content transfer encoding (3 ed.)*. Available:

<https://support.microsoft.com/en-us/kb/323489>

- [8] L. Wood, P. Holliday, and D. Floreani, "Moving data in DTNs with HTTP and MIME," *IEEE*, 2009.
- [9] R. Vito and D. M. Vincenzo, "IMAGENCIES: Network Image Services for Telemedicine Applications," *IEEE*, pp. 501-506, 2001.
- [10] S. Goel, M. Shaaban, T. Darwish, H. Mahmoud, and M. Bayoumi, "Memory Accesses Reduction for MIME Algorithm," *IEEE*, pp. 805-808, 2003.