

PROPUESTA DE UN MODELO DE GESTIÓN DE SERVICIOS EN LA NUBE PARA LA MANIPULACIÓN DE SISTEMAS ROBÓTICOS CON EL USO DE DISPOSITIVOS MÓVILES

PROPOSAL OF A MANAGEMENT SERVICE MODEL TO HANDLE CLOUD ROBOTIC SYSTEMS BY MEANS OF MOBILE DEVICES

Jaime Alberto Guzmán-Luna, PhD.

*Grupo de Investigación Sintelweb
Universidad Nacional de Colombia
Medellín, Colombia
jaguzman@unal.edu.co*

Ingrid-Durley Torres-Pardo, PhD (C).

*Grupo de Investigación GEA Institución
Universitaria Salazar y Herrera
Medellín, Colombia
l.torres@jush.edu.co*

Paola Andrea Galeano-Hincapié, Ing.

*Grupo de Investigación Sintelweb
Universidad Nacional de Colombia
Medellín, Colombia
pagaleanoh@unal.edu.co*

(Recibido el 10-05-2014. Aprobado el 20-06-2014)

Resumen. Este documento propone un modelo para la tele-operación de robots con el uso de dispositivos móviles (celulares o tablets) como medios de comunicación entre el usuario y la máquina; para este propósito se sugiere utilizar un servidor que permita almacenar información sobre las especificaciones de cada uno de los dispositivos móviles. Paralelamente, se propone acudir a un conjunto de servicios web que operan desde la nube para construir una interfaz personalizada para el usuario la cual facilitará la tele-operación del robot por medio de la captura de datos y de una heurística.

Palabras clave: Dispositivos móviles; Servicios en la nube; Sistemas robóticos; Tele-operación.

Abstract. This paper proposes a model for tele-operation of robots using mobile devices (cellular phones and/or tablets) as a means of communication between the user and the machine; for this purpose it is suggested to use a server, which will store information on different specifications of each mobile devices; In parallel, intends to go to a set of Web services, that operate from the cloud to build them a custom user interface, which will facilitate tele-operation of the robot, through the collection of data and heuristics.

Keywords: Cloud services; Mobile devices; Robotic systems; Tele-operation.

1. INTRODUCCIÓN

La computación en la nube brinda una alternativa que busca solucionar diversos problemas que atacan a la robótica; dos de los más comunes lo constituyen la capacidad de almacenamiento y cómputo junto con la independencia de la plataforma [3]. Se trata de una alternativa que, a través de un dispositivo con acceso a la web, interactúa con distintas aplicaciones para teleoperar un robot.

El modelo se apoya en los servicios web con el fin de aprovechar la abstracción de los datos implementada sobre la web moderna y permitir el consumo de diferentes servicios sin necesidad de que tengan conocimiento sobre la infraestructura que está detrás convirtiéndose en una buena opción en el momento de la manipulación remota en dispositivos de media o baja capacidad de cómputo [2]

El objetivo de este documento es presentar un modelo basado en la gestión de servicios en la nube para facilitar la teleoperación de un robot a través de interfaces personalizadas para el usuario. Con este propósito se utiliza un robot Lego Mindstorm [1], con el que se hace la integración del sistema operativo Android y el lenguaje Java Lejos bajo los IDE Eclipse o NetBeans para una mejor unificación entre ambos dispositivos.

Con el propósito de brindar un mayor detalle de la propuesta este artículo se organiza de la siguiente forma: la sección 2 realiza un breve recorrido por los principales conceptos y trabajos de la literatura que hacen uso de una arquitectura de cómputo en la nube y operan de manera remota un robot. Esta exposición termina con la presentación de la propuesta del modelo de gestión de servicios en la nube orientado hacia la manipulación del robot. En la sección 3 se exponen las características físicas y funcionales del robot usado dentro del modelo. La 4 expone un caso de estudio de tele-operación basado en realidad aumentada, seguido de los agradecimientos y conclusiones asociados con el proyecto.

2. MARCO REFERENCIAL

Esta sección recopila, de manera sintética, los conceptos asociados con las principales temáticas de la propuesta. Además, se citan los trabajos más relevantes de la literatura sobre esta temática específica.

2.1. Dispositivos móviles

Un dispositivo móvil se puede definir como un aparato electrónico de tamaño mediano o pequeño, con alguna capacidad de procesamiento y posibilidad de conexión a una red, diseñado para una función específica, pero que puede llevar a cabo otras más generales. De esta manera, la definición abarca desde reproductores de audio portátiles hasta tablets o smartphone [12]. Con cada nueva versión del dispositivo se actualiza también el sistema operativo y las características del hardware. Esto incluye modificaciones y nuevas adiciones; tal es el caso de los últimos dispositivos móviles de la familia Galaxy (por parte de Samsung y Google) o el último iPhone (de Apple) que traen consigo una variedad de sensores y accesorios novedosos, como por ejemplo el “Sensor identificador de huellas digitales” integrado en el iPhone 5s o el pulsómetro incorporado en el Samsung Galaxy S5 y se destacan, además, los sensores gestuales que vienen implementando [13], [14].

A manera de resumen, en la Tabla 1 se muestran algunos de los dispositivos o sensores que se utilizan en la actualidad, en celulares de alta gama, algunos de ellos comunes en la mayoría de dispositivos móviles, mientras que otros (sensor de temperatura y humedad o el barómetro) son menos frecuentes. Junto a los sensores, existen otros métodos de entradas actualmente implementados en dispositivos móviles como el reconocimiento facial o el control por voz [15]. La Tabla 1 resume algunas de los citados en dispositivos [16].

2.2. Arquitectura de cómputo en la nube

La computación en la nube se compone de tres modelos principales [2]: i) *Software as a Service (SaaS)* en el que se presentan las aplicaciones para el consumo a través de la red, sin conocimiento alguno de la infraestructura. ii) *Platform as a Service (PaaS)*, que presenta un entorno de desarrollo para el control y creación de aplicación, según las tecnologías que soporta el proveedor. Y, finalmente, iii) *Infrastructure as a Service (IaaS)* para el acceso a los recursos fundamentales o físicos, necesarios para el funcionamiento y comunicación. De manera paralela, para el manejo remoto de robots (teleoperación), existen varias propuestas referenciales que usan la arquitectura de cómputo en la nube, a partir de un servidor y diferentes conexiones con módulos de servicios o plataformas de implementación.

Tabla 1. Métodos de entrada comunes disponibles en dispositivos móviles

SENSOR - DISPOSITIVO	DESCRIPCIÓN
Acelerómetro	Mide la aceleración en el celular y la precisión es de un radio de 2G con un error aproximado de 0.18G.
GPS	Sistema de posicionamiento por satélite (error aproximado de 10m).
Giroscopio	Calcula el cambio en la orientación en radianes.
Pantalla	Dispositivo de entrada y salida principal y permite la interacción entre el usuario y el dispositivo móvil.
Luminosidad	Mide la intensidad de luz del exterior.
Proximidad	Calcula la proximidad de un objeto al celular y su rango no exceda los 5cm.
Wifi	Establece conexión y brinda acceso a la red.
BlueTooth	Transmisión de datos entre distintos dispositivos y artefactos electrónicos que posean el mismo dispositivo.
Temperatura y humedad	Medición de temperatura y humedad del aire.
Barómetro	Medición de la presión atmosférica.
Geomagnético	Detección de cambios en el campo electromagnético.
Gestual (guiado por infrarrojo)	Detección de movimientos por encima de la pantalla.
Joystick (externo)	Medio de entrada alternativo a la pantalla.

Tabla 2. Métodos de entrada seleccionados

DISPOSITIVO	SENSOR	DESCRIPCIÓN
Samsung Galaxy note	S pen	Permite escribir con la precisión de un bolígrafo e incluso incluye funciones adicionales en los botones laterales de algunos modelos.
Samsung Galaxy S4+	Air Gestures + Cámara + Acelerómetro	Detección de gestos de las manos sin tocar la pantalla del dispositivo, sigue el movimiento de las manos para realizar acciones y permite su manejo en tiempo real más preciso.
General (se compra por separado)	Stylus	El input del stylus es traducido a toque y usa método nativo del dispositivo para permitir la manipulación.
Android e iOS	Cámara + Acelerómetro	Estos sistemas operativos se pueden trabajar con licencias generales de software libre (código abierto sin fines de lucro).

El objetivo es conectar el robot a un servidor web. Este servidor envía el estado actual de la máquina a un usuario (computador o dispositivo), que, finalmente, puede controlar indirectamente el robot mediante un dispositivo de entrada. Se describen a continuación algunas de las propuestas de la literatura que se orientan hacia esta corriente.

AD - *Automatic Deliberative* [4], presenta los principales problemas cuando se utiliza la web como un medio de comunicación; además, expone soluciones de varios autores, en los que se evidencia la necesidad de profundizar en la estabilidad mediante una estrategia de *Supervisory Control* a partir de una arquitectura de “skills” (AD - *Automatic Deliberative*).

Por otra parte, Corba [5], expone la implementación de una arquitectura basada en Corba para el control de una mano robótica como ayuda para las personas de edad avanzada, mediante un “nivel de tareas”, un procesamiento de imagen y un “feedback server”.

En paralelismo [6], se propone obtener ventaja del paralelismo de la computación en la nube para reducir el procesamiento cuando se ha de adquirir información y, por tanto, contribuir de manera eficiente en la generación de rutas, mediante la comunicación en paralelo de diversos sistemas robóticos.

En SW-SOAP [7], se exploran varias alternativas para el control de un dispositivo robótico a partir de los servicios web. Se destaca la implementación de un protocolo SOAP. En el proceso se incorpora un servidor intermedio que proporciona la interfaz de servicios web. Tales servicios procesan los mensajes SOAP recibidos desde el usuario y los traduce en comandos adecuados para el robot y los envía al robot a través de la conexión Bluetooth inalámbrica; muestra, además, una vista en tiempo real de la operación remota del robot.

En SW Restfull [8], los autores exponen las ventajas de esta arquitectura en comparación con las arquitecturas antes mencionadas. Cabe resaltar (según el autor) la simplicidad, la capacidad de transmitir datos

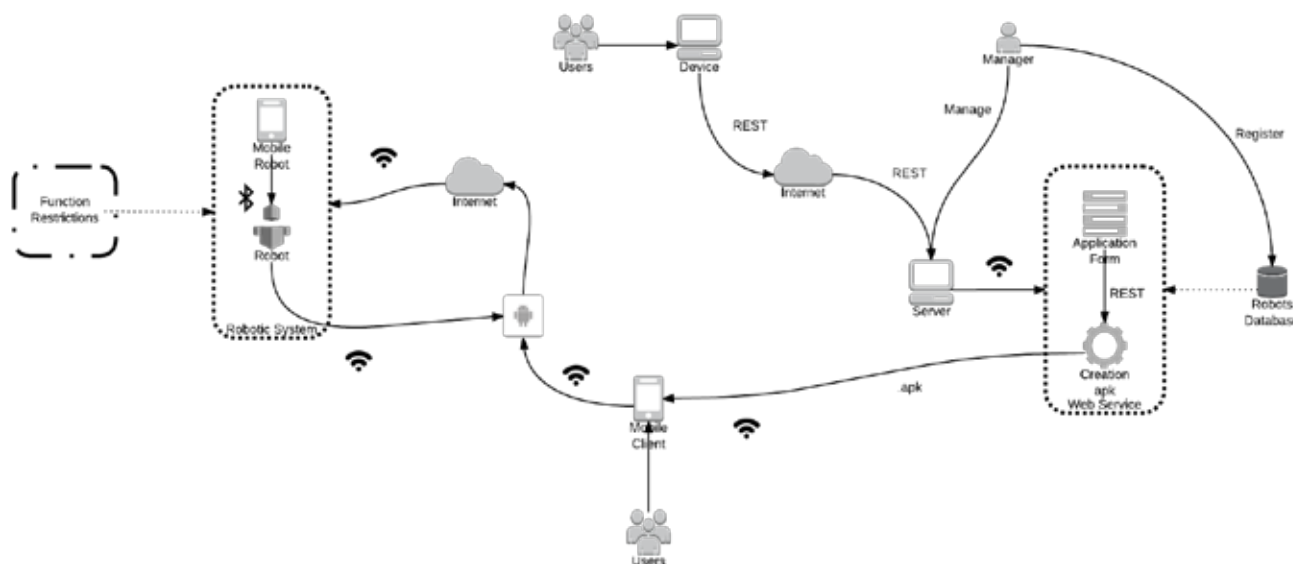


Figura 1. Arquitectura propuesta

sobre HTTP, así como el acceso con escaso consumo de recursos para sus operaciones por el limitado número de operaciones y el esquema de direccionamiento unificado [9], [10].

3. ARQUITECTURA PROPUESTA

Después de analizar las arquitecturas presentadas en la sección anterior, se determinó que comparten varias similitudes, entre las que se resaltan: i) el enlace entre el robot y un dispositivo (interno o externo). ii) la conexión que existe entre el dispositivo que es conectado al robot y el servidor. iii) el uso funcional del dispositivo, como mecanismo receptor de comandos del servidor y medio de transmisión al robot. Tales características, junto a las cualidades

descritas en [8, 11], sobre enfoque de servicios web, permiten proponer el modelo de gestión de cómputo en la nube, apoyado por servicios web RestFul, tal como se observa en la Figura 1.

Como se muestra en la Figura 1, el usuario actúa como eje central de la arquitectura y puede acceder a través de un dispositivo móvil, mediante la invocación de un servicio web (Restful) a algunas de las tareas que se encuentran disponibles en el servidor. Para ello, el usuario debe identificar el tipo de robot que tiene sobre una interfaz de acceso al servidor. Como paso siguiente, el servidor habilita el acceso al usuario a una base de datos generalizada.



Figura 2. Tracción del robot

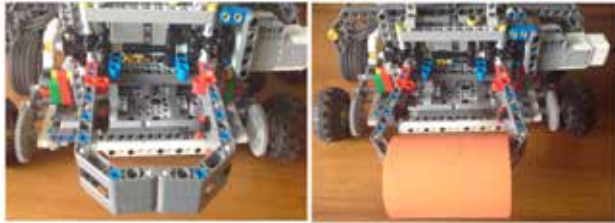


Figura 3. Garra del robot

Esta base, contiene acciones (tareas) propias de un robot (ir capturar imagen, girar adelante, girar a la derecha, girar a la izquierda, etc.) almacenadas bajo una taxonomía genérica de identificación robots. Las características de cada robot identificado en la taxonomía pueden ser enriquecidas por el usuario, a través de un acceso (limitado, por cuestiones de seguridad) de información al servidor. Sin embargo, cuando se habilita el acceso al usuario, este podrá acceder únicamente a las tareas que se encuentran disponibles para su tipo de robot. De esta manera la identificación del robot del usuario, actúa como un filtro de tareas. Una vez se ha identificado el robot, el usuario selecciona la tarea de su interés y ordena su ejecución, a través de otro servicio web. Para ello, la orden debe ser convertida a un APK (el cual corresponde a un archivo, comprensible por el sistema operativo Android, el cual está presente en la mayoría de dispositivos móviles). Una vez, el archivo es descargado sobre el dispositivo, la tarea puede ser ejecutada sobre el robot. En cada paso se espera tener habilitada la conexión wifi, para otorgar acceso a algunas librerías, driver o cualquier otro componente que lleve a buen fin la ejecución de la acción en el robot.

Con el fin de brindar un mayor detalle del caso de estudio desarrollado, a continuación se expone el diseño físico del robot utilizado, así como la descripción del dispositivo móvil implementado.

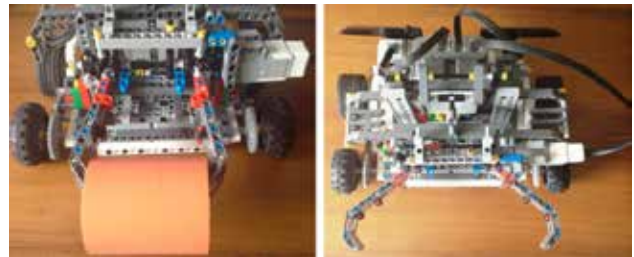


Figura 4. Vista aérea del robot

La Tabla 3 muestra el resumen de las conexiones de los motores y los puertos del robot construido.

Tabla 3.

CONEXIONES DEL ROBOT

Motor	Puerto
Motor de la garra	A
Motor izquierdo	B
Motor derecho	C

3.1. Caracterización física del robot

Para la implementación física del robot se utilizaron piezas del kit Lego Mindstorms NXT Education y del kit Lego Mindstorms Ev3. La característica más sobresaliente es que el ladrillo del robot es un NXT y los motores son de un Ev3.

El robot posee tres motores; uno pequeño (para la garra) y dos grandes para las cuatro llantas, así, cada llanta trasera está unida a una llanta delantera por medio de engranajes para permitir que se dé una doble tracción en las cuatro llantas, tal y como se muestra en la Figura 2.

Finalmente, el robot utiliza una garra para coger o soltar objetos para trasladarlos de un punto a otro, como se observa en la Figura 3.

En la Figura 4 se puede apreciar el robot en su dimensión física completa.

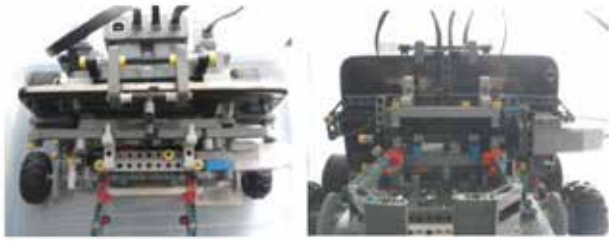


Figura 5. Robot con tablet Samsung Galaxy Tab 3

3.2. Dispositivo móvil

El dispositivo móvil del usuario está representado por una tablet Galaxy Tab 3, la cual dispone de una app que provee una interfaz de control del robot y una visualización directa de lo que es capturado por el robot; la app fue desarrollada en código nativo de Android y dispone de unos botones para tele-operar al robot y que son accionados de forma táctil.

Para lograr una completa compatibilidad, al robot también se le incorporó una tablet Samsung Galaxy Tab 3, a través de la cual se permite la integración de los comandos que llegan vía wifi al robot. Al mismo tiempo que se puede utilizar su cámara para guiar al usuario en el manejo remoto del robot, como se observa en la Figura 5.

3.3. Implementación

La aplicación que implementa la arquitectura de computo en la nube fue desarrollada en el IDE Unity [17] junto al SDK Vuforia [18], que dispone de un plugin para este IDE. El trabajo en este entorno es sencillo y ofrece muchas posibilidades, por lo que se puede compilar la aplicación para distintas plataformas móviles como Android, iOS, PC, entre otras.

Vuforia realiza el trabajo de forma transparente al desarrollador, a través de la cámara en el dispositivo busca y detecta los *markers* (secuencia gráfica codificada) en el entorno real y, ya reconocidos, superpone sobre la vista del mundo real elementos virtuales predefinidos en el desarrollo de la app, es decir, se construye así una realidad aumentada.

El robot muestra la visión del robot y los objetos 3D en la pantalla de la tablet que está sobre el robot. Por ello, se pensó retransmitir la pantalla de la tablet como un *streaming* a la tablet del usuario y se eva-

luaron varias opciones, entre ellas *Screen Stream Mirroring*, *Webkey* y *Mobizen*, pero la que mejor desempeño tuvo fue *Mobizen* en su versión 2.7.0.4 puesto que provee un video continuo y muy cercano al tiempo real, razón por la cual se instaló en la tablet del robot para generar un *streaming* de la pantalla a través de la red local a la cual se puede acceder desde una página web (alojada en el servidor), mediante el propio servicio.

Cuando se presiona un botón se envía un comando mediante un socket a la tablet robot, a la par que muestra el *streaming* de video proveído por *Mobizen* a través de una página web incrustada en un *WebView*; cabe resaltar que es ideal que la app usuario corra en una versión de Android 4.4.x en esta versión se añadieron características que mejoran la versatilidad del *WebView* para acceder al *streaming* sin problemas, además, se debe modificar el *User Agent* del *WebView*, pues la página web de *Mobizen* no deja ser visualizada desde dispositivos móviles.

4. CASO DE ESTUDIO

Con el ánimo de comprobar la funcionalidad de la arquitectura, se valoró la tele-operación de un robot de forma remota, dentro de una locación de prueba (ver Figura 6), que tiene dispuesto *markets* en diferentes espacios que deben ser retransmitidos entre el robot y el usuario. El objetivo principal consiste en navegar con el robot operado remotamente, realizar un recorrido de un espacio físico y retransmitir las imágenes en tiempo real. Tales pruebas se hicieron desde dos perspectivas: i) sin incorporar elementos funcionales de la arquitectura y ii) ingresando a servicios de visualización de realidad aumentada y transmitir las imágenes. Ambos resultados fueron recopilados, analizados y valorados.

La Figura 6, también señala el recorrido que se espera realice el robot en un tiempo mínimo y traslade con su garra objetivos del punto A a los posibles puntos B.

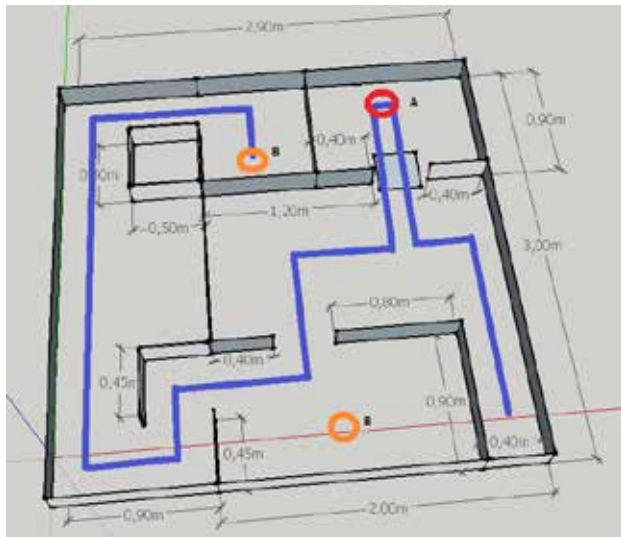


Figura 6. Mapa físico y recorrido objetivo del robot

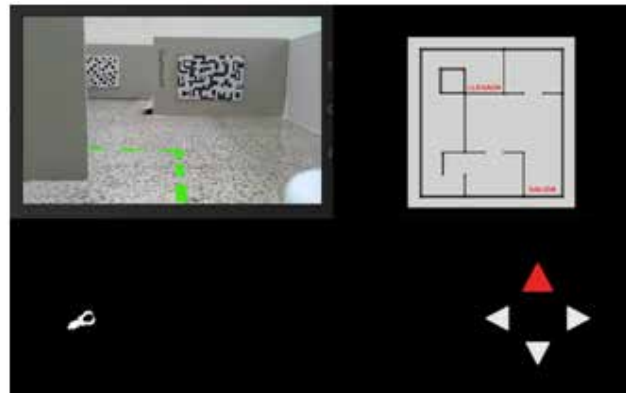


Figura 8. Interfaz de usuario sin realidad aumentada

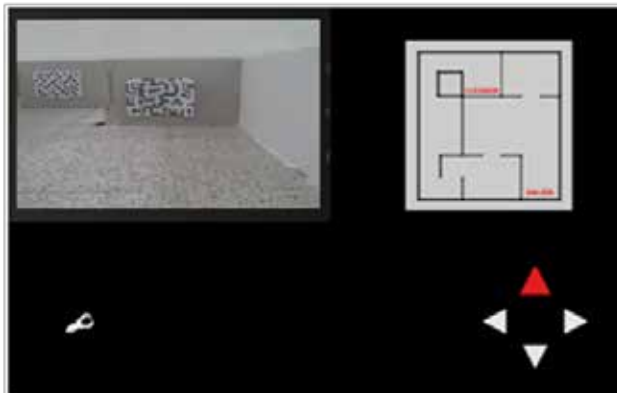


Figura 7. Interfaz de usuario sin realidad aumentada

Para evaluar la eficiencia de la plataforma se valoró primero la prueba sin incorporar la realidad aumentada. En la Figura 7 se registra el resultado de la imagen transmitida desde el robot al usuario.

Por su parte, la Figura 8, muestra el registro de la imagen transmitida al usuario con la prueba con la incorporación de la realidad aumentada.

Los resultados de los tiempos obtenidos con ambas pruebas fueron valorados por una muestra poblacional de 11 personas, quienes realizaron ambas pruebas (sin realidad aumentada –Sin RA- y con realidad aumentada –Con RA-). El registro de los tiempos generados como resultado fue tabulado en la Figura 9.

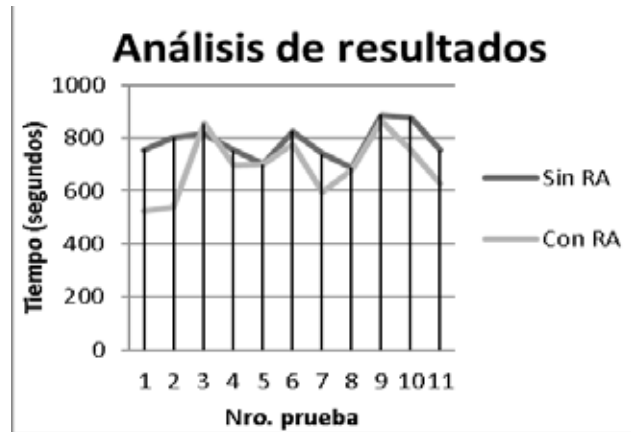


Figura 9. Análisis de los resultados obtenidos en las pruebas

Según los resultados obtenidos, se puede concluir que el tiempo en el que un robot realiza el recorrido completo es menor si se utiliza la realidad aumentada en comparación a si sólo se hace con una vista plana del video que capta el robot. Esto demuestra que el tiempo del proceso de conexión y acceso al servidor resulta despreciable. Se añaden, además, los evidentes beneficios logrados en la calidad de la imagen que resultan más claros en la transmisión Con RA.

5. TRABAJOS FUTUROS

Como trabajo futuro se propone la implementación de la plataforma e incorporación de otras tareas funcionales dentro del servidor y con la categorización de otros modelos robóticos. Por ahora sólo ha sido considerado con el dominio Lego.

6. CONCLUSIONES

Para obtener un mejor rendimiento y aprovechar el potencial de los robots teleoperados (Lego por ahora), resulta útil considerar arquitecturas de cómputo en la nube que, mediante servicios web, habiliten la integración de usuario-robot y servidor.

La realidad aumentada juega un papel relevante en la tele-operación de robots ya que si bien es cierto que el mismo recorrido se puede realizar sin la necesidad de una realidad aumentada (sólo con inspección visual), resulta más preciso y efectivo hacerlo con una visión aumentada de la realidad.

El modelo de tele-operación basado en servicios web RestFul, está fundamentado en los criterios de escalabilidad de software y versatilidad. Además, resulta compatible con el sistema operativo Android.

7. AGRADECIMIENTOS

Los autores de este documento reconocen las contribuciones del grupo de investigación Sintelweb, de la Universidad Nacional de Colombia (sede Medellín), en la creación y propuesta de un modelo capaz de tele-operar un robot, por medio de interfaces de usuario que usan realidad aumentada para facilitar la tele-operación del robot. Se destaca el aporte significativo de César Camilo Parra Marino, miembro del grupo que contribuyó con el despliegue del trabajo anteriormente mencionado. Además de brindar un especial reconocimiento al proyecto “Modelo de gestión en la nube para la manipulación de sistemas robóticos utilizando dispositivos móviles” aprobado por el Departamento Administrativo de Ciencia, Tecnología e Innovación (Colciencias) en su convocatoria 617 del 2013 llamada “Convocatoria para conformar bancos de elegibles para formación de alto nivel para la ciencia, la tecnología y la innovación (semilleros y jóvenes investigadores, doctorados nacionales y en el exterior)”. Este proyecto está asociado particularmente con el módulo (1) semilleros de investigación de dicha convocatoria.

REFERENCIAS

- [1] E. A. Castro Estrada, “Manual de prácticas de robótica móvil programadas en Java utilizando el Kit Lego Mindstorms.” 2011.
- [2] O. Mejia, “Computación en la nube”. *ContactoS*, Vol. 80, pp. 45-52, 2011.
- [3] B. Rosero and J. Guzmán, “Computación en la nube como modelo distribuido para la interacción de plataformas robóticas”. *Vínculos*, Vol. 10, No 1, pp. 227-235, 2013.
- [4] A. Khamis, F. Rodríguez, and M. Salichs, “Remote interaction with mobile robots”. *Autonomous Robots*, Vol. 15, No 3, pp. 267-28, 2003.
- [5] S. Jia and K. Takase, “Internet-based robotic system using CORBA as communication architecture”. *Journal of Intelligent and Robotic Systems*, Vol. 34, No 2, pp. 121-134, 2002.
- [6] R. Arumugam and V. Enti, “DAVinCi: A cloud computing framework for service robots”. *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, pp. 3084-3089, 2010.
- [7] J. Du, U. Witkowski, and U. Rückert, “Teleoperation of a mobile autonomous robot using web services”. En *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*. Springer Berlin Heidelberg, 2006. p. 55-60, 2006.
- [8] M. Murazza, F. Millán, N. Rodríguez, D. Segura, and D. Villafañe, “Desarrollo de aplicaciones para Cloud Computing”. En *XVI Congreso Argentino de Ciencias de la Computación*, 2010.
- [9] W. Brogden, “REST versus SOAP—the REST story,” Website. Last accessed Novemb. Vol.20. 2009.
- [10] R. Enríquez and A. S. C., “RESTful Java Web Services Security”. Packt Publishing Ltd. 2014.
- [11] L. Richardson, M. Amundsen, and S. Ruby, “RESTful Web APIs”. O’Reilly Media, Inc. 2013.
- [12] A. B. Alonso, I. F. Artime, M. Á. Rodríguez, and R. G. Baniello, “Dispositivos móviles”. EPSIG Ing. Telecomunicación Universidad de Oviedo.

- [13] Samsung, "Samsung Galaxy S5," 2014. [Online]. Available: <http://www.samsung.com/es/offer/galaxys5/>. [Accessed: 16-Oct-2014].
- [14] Apple, "iPhone 5s," 2014. [Online]. Available: <https://www.apple.com/la/iphone-5s/specs/>. [Accessed: 16-Oct-2014].
- [15] Y. Bermúdez and J. López, "Análisis comparativo entre sistemas operativos de dispositivos móviles Android, Iphone y Blackberry", Universidad Tecnológica de Pereira, 2011. [Online]. Available: <http://repositorio.utp.edu.co/dspace/handle/11059/2474>. [Accessed: 15-Oct-2014]
- [16] L. Delía, N. Galdamez, P. Thomas, P. Pesado, "Un análisis experimental tipo de aplicaciones para dispositivos móviles", XVIII Congreso Argentino de Ciencias de la Computación, 2013. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/32397>. [Accessed: 15-Oct-2014].
- [17] Unity Technologies, "Unity," 2014. [Online]. Available: <http://unity3d.com/unity/download>. [Accessed: 22-Oct-2014].
- [18] Qualcomm Connected Experiences, "Vuforia Developer Portal," 2014. [Online]. Available: <https://developer.vuforia.com/>. [Accessed: 22-Oct-2014].