

Evaluación de capacidades de Calidad de Servicio de proveedores de Computación en la Nube

J. A. López-Cázar, J. F. Ibáñez-Salas, A. Díaz-Ramírez, *Miembro IEEE*, L. Villafuerte-León

Resumen—La invención de la Web a inicios de los 80's provocó el uso masivo de Internet, así como el desarrollo de aplicaciones de negocios (*e-commerce*). Recientemente, la Computación en la Nube ha potenciado las capacidades de Internet, permitiendo que puedan desarrollarse aplicaciones en muchos ámbitos, ampliando las posibilidades de negocio y de la Web. A la fecha hay muchos proveedores de servicios de computación en la nube. Sin embargo, existen muchas diferencias entre ellos, desde opciones tecnológicas hasta facilidad de uso. En este artículo, se presentan los resultados de la evaluación de algunos de los mas importantes proveedores de computación en la nube. Particularmente, se evalúa el desempeño de Azure de Microsoft, que es una plataforma propietaria, así como las plataformas OpenShift y Koding, que son de código abierto. Los proveedores son evaluados con respecto a sus capacidades de calidad de servicio (QoS), y se discuten los resultados.

Temas claves—Computación en la nube, Azure, Koding, Openshift, PaaS.

Abstract—The invention of the Web in the early 80's motivated the massive use of the Internet, and the development of *e-commerce* applications. Recently, the cloud computing technology has enhanced the possibilities of the Internet, allowing the development of applications to be used in many areas, such as *e-business* and the Web. Nowadays, there are many cloud computing services providers. However, there are differences among them, related to the model services offered, or the easy of use. In this paper, we introduce the results of the evaluation of some of the most important cloud computing providers. Particularly, we compare the performance of Microsoft Azure, and the open source providers OpenShift and Koding. Their performances are evaluated regarding their quality of service (QoS) capabilities, and the results are discussed.

Keywords—Cloud computing, Azure, Koding, Openshift, PaaS.

(josa.lopez@live.com.mx), José Alberto López Cázar. Instituto Tecnológico de Mexicali, Baja California. (pacois20@gmail.com), Juan Francisco Ibáñez Salas. Instituto Tecnológico de Mexicali, Baja California. (adiaz@itmexicali.edu.mx), Arnoldo Díaz Ramírez. Instituto Tecnológico de Mexicali, Baja California. (letyvillafuerte@hotmail.com), Leticia Villafuerte León. Instituto Tecnológico de Mexicali, Baja California.

I. INTRODUCCIÓN

Debido al gran desarrollo tecnológico, el acceso a la Web se ha utilizado como medio de comunicación, medio de entretenimiento e indispensable para el trabajo. La proliferación y disponibilidad de la Web a motivado la búsqueda nuevas formas de ofrecer servicios con mejor desempeño, facilidad de acceso y al momento de desarrollar e implementar una aplicación se necesita agilidad y rapidez. Debido a las grandes oportunidades que brinda este mercado muchas empresas se encuentran ofertando servicios de computación en la nube. Este nuevo modelo brinda muchas ventajas para diferentes tipos de usuarios como por ejemplo: los usuarios de la computación móvil, las empresas que pueden sustituir su centro de datos por los servicios de algún proveedor en la nube, o las empresas que ofrecen servicios de Tecnologías de la Información, por mencionar algunas.

Con la computación en la nube se soluciona el problema de almacenamiento, ya que la información del usuario puede ser almacenada en la nube y así liberar espacio en los dispositivos. Otra ventaja de la computación en la nube es para las empresas o desarrolladores de software ya que este modelo brinda el rápido acceso a recursos indispensables para éste tipo de usuarios.

En este artículo se presenta un análisis comparativo entre proveedores de computación en la nube comerciales y proveedores Open Source. El objetivo es identificar la mejor opción de proveedores de plataforma como servicio (PaaS) en la nube evaluando plataformas entre distribución comercial y distribución Open Source, en base al desempeño de calidad de servicios (QoS). Se analizan los resultados obtenidos del desempeño de estas plataformas en un caso experimental. Este documento está organizado de la siguiente manera. En la Sección II se muestran los trabajos relacionados. En la Sección III se presenta el marco conceptual de la computación en la nube. En la Sección IV se muestran los proveedores propuestos para este trabajo. El análisis comparativo de los experimentos para evaluar las plataformas de los proveedores se discute en la Sección V. en la sección VI se presentan los resultados. Finalmente, las conclusiones y el trabajo futuro se explican en la Sección VII.

II. TRABAJOS RELACIONADOS

Algunos trabajos comparativos realizan estudios taxonómicos sobre la descripción de la arquitectura de proveedores en la nube para después realizar una comparación [16]. Otro análisis que se realizó fue el de aplicar un algoritmo de multiplicación de matrices y variar el número de núcleos en el procesador para verificar la aceleración superlineal de Azure [6]. También se han realizado estudios centrados en los aspectos tales como la arquitectura, características y la aplicación para conocer las diferencias entre proveedores open Source y proveedores close Source en las capas SaaS, PaaS e IaaS [18]. En otro estudio se discuten en detalle las características, arquitecturas y aplicaciones de varias plataformas de computación en la nube populares, para identificar diferencias, características y ventajas de distintas plataformas [14]. En otro estudio se basó tanto en el análisis del rendimiento como en la retroalimentación de los usuarios de plataformas de computación en la nube para realizar un modelo de selección de servicios en la nube [15]. En otro artículo se hace un análisis comparativo de plataformas en la capa IaaS con un criterio de popularidad, uso de la comunidad, la modularidad, la apertura y el código abierto [19]. En este trabajo se realiza un análisis comparativo de proveedores en la capa PaaS utilizando métricas de calidad del servicio (QoS) a través de pruebas utilizando servicios web situados en la capa PaaS.

III. COMPUTACIÓN EN LA NUBE

La computación en la nube es una forma especializada de computación distribuida, que introduce modelos de utilización para la provisión de forma remota de recursos escalables y medidos [3]. En este paradigma existen seis características las cuales son las más comunes en este entorno: uso bajo demanda, acceso ubicuo, múltiples inquilinos, uso medido, flexibilidad y elasticidad.

Uso bajo demanda. El consumidor de la nube puede acceder a los recursos de la nube dando al consumidor la libertad de elegir y configurar estos recursos.

Acceso ubicuo. El acceso ubicuo representa la habilidad del servicio de la nube sea altamente accesible, normalmente a través de internet.

Múltiples inquilinos. Es la característica de un programa informático que permite a una instancia del sistema servirle a diferentes consumidores (inquilinos), por lo cual cada uno está aislado del otro.

Uso medido. El uso medido es la característica que representa la habilidad de una plataforma en la nube

de poder realizar un seguimiento del uso de sus recursos.

Flexibilidad. Computación flexible es una forma de tolerancia de error que distribuye implementaciones redundantes de recursos a través de la ubicación física.

Elasticidad. Es la habilidad automatizada de la nube de escalar de forma transparente los recursos de IT, según sea necesario, en respuesta a las condiciones de tiempo de ejecución o predeterminado por el consumidor o el proveedor de la nube.

Los modelos de entrega de servicios en la nube representan una combinación de recursos pre empaquetados ofrecidos por un proveedor. Los modelos más usuales que se ofrecen a los clientes y usuarios de la nube son: SaaS (Software as a Service), software como servicio, PaaS (Platform as a Service), plataforma como servicio e IaaS (Infrastructure as a Service), infraestructura como servicio [3].

Software como Servicio: El modelo software como servicio se refiere esencialmente al software residente (instalado) en la nube puesto a disposición del consumidor a través de un navegador web, el cliente puede acceder a este servicio a través de cualquier dispositivo con conexión a internet.

Plataforma como Servicio: el proveedor ofrece un entorno de desarrollo al consumidor del modelo, esencialmente para desarrolladores de sistemas, quienes crean sus aplicaciones y ofrecen sus servicios a través de la plataforma del proveedor.

Infraestructura como Servicio: Este modelo proporciona la infraestructura necesaria para crear, ejecutar y desplegar aplicaciones. El modelo IaaS ofrece espacio de almacenamiento, capacidad de procesamiento, servidores, sistemas operativos, esto lo hace a través de una máquina virtual.

IV. PROVEEDORES DE COMPUTACIÓN EN LA NUBE

Con la popularización de la computación en la nube, muchas empresas han surgido como proveedores de este servicio. Ejemplos son: Amazon, Microsoft, Google, Salesforce, quienes son las más conocidas [10], [13], [20]. Por ejemplo, los inventores de la computación en la nube, Amazon, destaca como proveedor de Infraestructura como Servicio [11], [21]. Microsoft, con su plataforma Azure, es un proveedor clave de computación en la nube en la capa PaaS [16]. Debido a la gran influencia que tiene Microsoft en el desarrollo de aplicaciones empresariales en México y gran parte del mundo, se ha decidido elegir a ésta plataforma en el estudio comparativo.

Con el auge de la computación en la nube también ha crecido el desarrollo de proveedores de soluciones en

la nube Open Source [20], como Cloud Foundry, Cloudify y OpenShift, siendo éste último uno de los proveedores más referenciados en la literatura [2], [7], [22]. De hecho, se consideraba OpenShift como líder dentro de los proveedores open source [18]. Por esta razón y por su relevancia en con los otros autores se propone a OpenShift como el otro proveedor a evaluar. Se decidió trabajar paralelamente con el proveedor Koding, aunque no fue citado por autores en publicaciones de investigación, ha tenido una gran aparición en blogs relacionados a la programación.

Azure es la plataforma como servicio de Microsoft que permite a los desarrolladores crear rápidamente, hospedar, y escalar aplicaciones en un entorno de nube. Azure ofrece una amplia gama de recursos que soportan diversos lenguajes de programación como Java, .NET, Python, Ruby, PHP, Node.js, y un conjunto completo de herramientas de desarrollo para aumentar la productividad del desarrollador y acelerar la entrega de aplicaciones [25].

OpenShift es la plataforma como servicio de Red Hat que permite a los desarrolladores trabajar en un entorno de nube. OpenShift ofrece una selección de lenguajes de programación como Java, Ruby, PHP, Node.js, Python y Perl [24].

Koding ofrece una nueva tendencia de desarrollo basado en la nube [17]. En el 2009 se creó la primera versión de esta plataforma. Koding es un entorno de desarrollo en línea con el objetivo de simplificar el desarrollo a sus usuarios, proporciona computación libre y desarrollo para todos. Lo hace ofreciendo máquinas virtuales (VMs) gratis para el desarrollo en la nube [23].

V. TRABAJO REALIZADO

A pesar de que se estudian proveedores de plataforma como servicio (PaaS), las evaluaciones en este trabajo se realizan en la capa software como servicio (SaaS) ya que se utilizan servicios Web para la obtención de resultados.

Las métricas de calidad de un servicio son: rendimiento, fiabilidad, integridad, accesibilidad, disponibilidad, interoperabilidad y seguridad. El rendimiento del servicio mide la velocidad de finalización de las solicitudes del servicio. El rendimiento se puede medir en términos de tiempo de respuesta, latencia, tiempo de ejecución, y algunos otros indicadores bajo esta clase [4], [8]. Dentro de estas métricas existen atributos objetivos, los cuales se pueden medir directamente y los atributos subjetivos, los cuales no se pueden medir directamente [15]. Pero para este trabajo se necesitan atributos objetivos para poder medir y así conseguir datos significativos, analizarlos detalladamente para

obtener una conclusión en base a los resultados arrojados por las evaluaciones.

La latencia es el tiempo transcurrido que hay entre el envío de la solicitud y la recepción de la respuesta. Este atributo es citado en varios artículos [8], [9]. El tiempo de procesamiento o tiempo de ejecución es el lapso tiempo que se necesita para realizar las instrucciones dentro de un proceso. Este atributo es utilizado por varios autores en sus trabajos [5], [6], [8]. Estas métricas son elegidas debido a su relevancia en investigaciones pasadas.

Para fines de este artículo y poder realizar las evaluaciones con respecto a las métricas señaladas es necesario realizar una aplicación, la cual deberá consumir servicios web alojados en las plataformas de los proveedores.

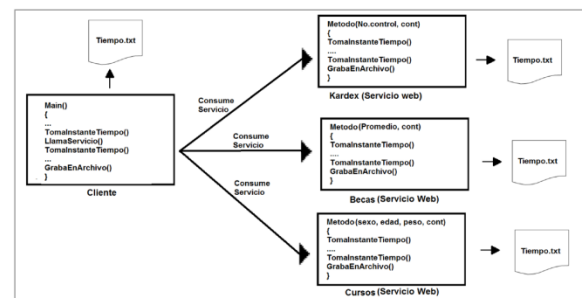


Figura 1. Diseño de la aplicación.

La Fig. 1 muestra el diseño de la aplicación, la cual consta de un cliente y 3 servicios web desplegados en las plataformas de los proveedores. Por el lado del cliente se realizó una aplicación en lenguaje JAVA. Este programa antes de enviar la solicitud al servicio web toma el instante de tiempo, la acción subsecuente es la llamada al servicio web, después de recibir la respuesta vuelve a tomar el instante de tiempo para después a través de una resta [12] obtener el lapso de tiempo que tardó el servicio en terminar de atender la solicitud. Este proceso se realiza para cada servicio y por último almacena en un archivo tiempo.txt los valores obtenidos para cada servicio. Esto con el fin de medir la latencia que tiene el proveedor en atender las solicitudes.

Por otro lado los servicios Web son aplicaciones independientes realizadas en lenguaje PHP con el apoyo de la herramienta NuSOAP para su implementación como servicio web [1]. Para cada servicio web se requieren parámetros diferentes. El servicio kardex obtiene como parámetro el número de control, y un parámetro extra para varias el contenido de materias contenidas en el vector de respuesta del servicio web, este parámetro extra se presenta en todos los servicios web y tienen el mismo funcionamiento. Al iniciar el servicio kardex se toma el instante de tiempo, en este proceso se tienen

almacenadas una lista de materias en un arreglo dependiendo del parámetro extra que se recibe en el servicio web es el número de materias que se tomarán de esta lista para devolver la respuesta a la solicitud junto con una calificación dentro del rango del 6 al 10 tomada aleatoriamente. Al terminar el proceso se vuelve a tomar el instante de tiempo, para finalmente conocer el lapso de tiempo que necesito el proveedor en procesar la información. Ya por último se almacena los datos obtenidos en un archivo tiempo.txt. El proceso de almacenar materias en el vector de respuesta y el de obtener el lapso de tiempo que necesita el proveedor para ejecutar el proceso es igual en los tres servicios, solo que en el servicio de extraescolares se reciben como parámetros la edad, el peso, el sexo y el parámetro extra. Estos parámetros sirven para cambiar la elección de la lista de materias dependiendo de la edad, el peso y el sexo. Y por último se tiene el servicio beca que recibe como parámetro el promedio y el parámetro extra. Este servicio toma una decisión dependiendo del promedio obtenido es el número de beca que regresa como parámetro y con el parámetro extra simplemente realiza un ciclo con la cantidad de veces que indique este número.

Los tres servicios web son desplegados en cada proveedor, en OpenShift es necesario descargar programas para trabajar con los repositorios que maneja este proveedor. En Azure se trabaja directamente con la plataforma del proveedor a través de un navegador pero todo bajo un costo y en Koding se trabaja en el editor de texto que ofrece el proveedor en su plataforma a través de un navegador.

VI. RESULTADOS

Para la evaluación de los servicios el cliente debe correr en una maquina con las mismas características, para esto se utilizó una computadora con sistema operativo Ubuntu 12.10, memoria RAM de 1893 MB, procesador Intel i3 3.10 GHz, en la cual serán desactivadas las funciones menos indispensables para que el cliente consuma los servicios con el mejor desempeño posible. Para de desarrollar el cliente se utilizó el programa Eclipse versión Juno. La computadora está conectada a una red con cable Ethernet y una tarjeta Gigabit Ethernet, la red se comparte con 14 nodos y una salida a Internet con una velocidad de 12 MB.

Para las pruebas el cliente solicitó 156 veces el servicio con una respuesta de 100 materias por solicitud. Se realizaron 156 pruebas para después poder eliminar espurios. Se eliminaron los tres tiempos más altos y los 3 más bajos quedando así 150 muestras. Estos resultados son el promedio de latencia en segundos por el lado del cliente.

TABLA 1.
LATENCIA EN EL SERVICIO DE KARDEX.

Proveedor	Latencia
Azure	1.000793 segundos
OpenShift	0.288627 segundos
Koding	0.53692 segundos

En la Tabla 1 se muestra la latencia que se obtuvo como promedio en el cliente para el servicio de kardex.

TABLA 2.
LATENCIA EN EL SERVICIO DE EXTRAESCOLARES.

Proveedor	Latencia
Azure	1.31736 segundos
OpenShift	0.26368 segundos
Koding	0.378927 segundos

En la Tabla 2 se muestra la latencia que se obtuvo como promedio en el cliente para el servicio de extraescolares.

TABLA 3.
LATENCIA EN EL SERVICIO DE BECA.

Proveedor	Latencia
Azure	0.529893 segundos
OpenShift	0.26778 segundos
Koding	0.266093 segundos

En la Tabla 3 se muestra la latencia que se obtuvo como promedio en el cliente para el servicio de beca. Otros resultados obtenidos fueron el tiempo que necesitaba el proveedor para procesar la información que enviaría como respuesta al cliente. Estas respuestas varían entre 20 y 200 materias que el proveedor respondería al cliente. Para estas pruebas se tomaron 106 muestras y al igual que las pruebas anteriores se eliminaron los 3 tiempos más bajos y los 3 más altos, esto con el fin de eliminar espurios.

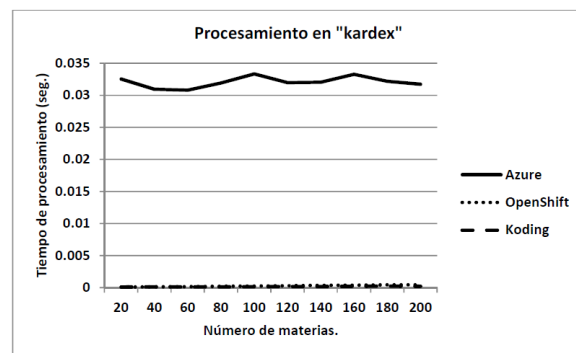


Figura 2-A. Tiempo de procesamiento para el servicio kardex.

En la Fig. 2-A se muestra la gráfica del tiempo que tarda el proveedor en atender la solicitud del servicio kardex. Debido a que Azure tomó más tiempo se creó otra gráfica para poder observar el comportamiento de los demás proveedores. Ya que al realizar un

acercamiento Azure desaparecía de la imagen. En la Fig. 2-B se realiza un acercamiento a los tiempos de los proveedores OpenShift y Koding para poder observar su comportamiento.

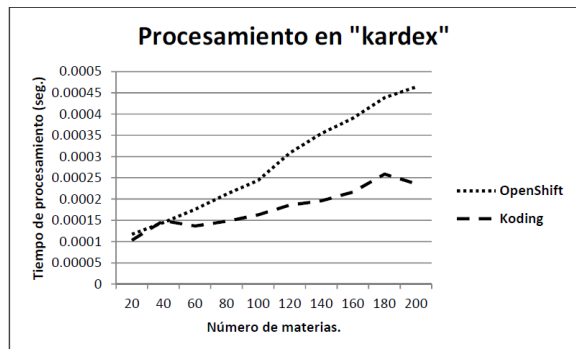


Figura 2-B. Tiempo de procesamiento para el servicio kardex.

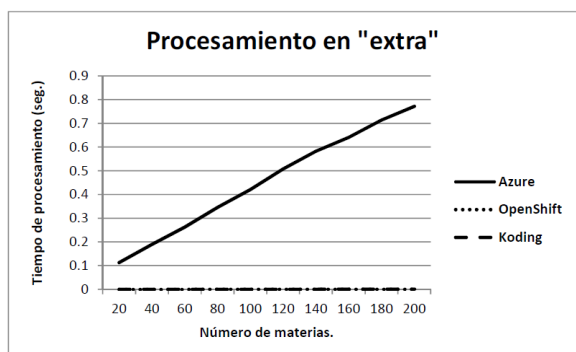


Figura 3-A. Tiempo de procesamiento para el servicio extra.

En la Fig. 3-A se muestra la gráfica del tiempo que tardo el proveedor en atender la solicitud del servicio extra. Debido a que Azure tomo más tiempo se creó otra grafica para poder observar el comportamiento de los demás proveedores. Ya que al realizar un acercamiento Azure desaparecía de la imagen. En la Fig. 3-B se realiza un acercamiento a los tiempos de los proveedores OpenShift y Koding para poder observar su comportamiento.

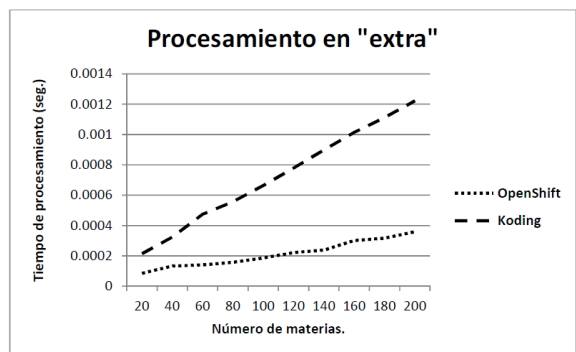


Figura 3-B. Tiempo de procesamiento para el servicio extra.

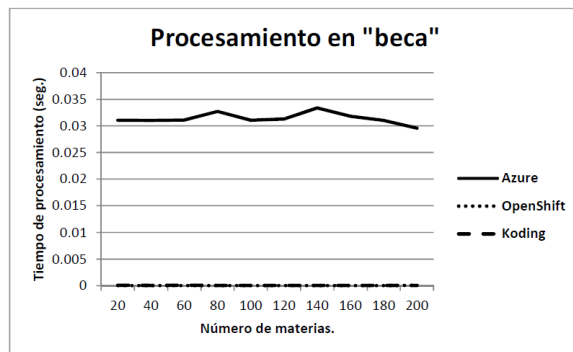


Figura 4-A. Tiempo de procesamiento para el servicio beca.

En la Fig. 4-A se muestra la gráfica del tiempo que tardo el proveedor en atender la solicitud del servicio beca. Debido a que Azure tomo más tiempo se creó otra grafica para poder observar el comportamiento de los demás proveedores. Ya que al realizar un acercamiento Azure desaparecía de la imagen. En la Fig. 4-B se realiza un acercamiento a los tiempos de los proveedores OpenShift y Koding para poder observar su comportamiento.

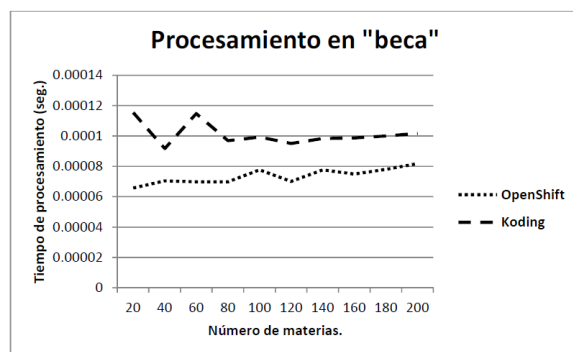


Figura 4-B. Tiempo de procesamiento para el servicio beca.

VII. CONCLUSIONES Y TRABAJO FUTURO

Tomando en cuenta los parámetros de QoS planteados en la propuesta se puede concluir que el mejor proveedor en la capa PaaS es OpenShift. En comparación de OpenShift y Koding, el primero mostro mejores resultados que el segundo, pero las características y herramientas de desarrollo son mejores en Koding ya que brinda realmente una plataforma en la nube como tal, cosa que en OpenShift es limitado. Por otro lado el proveedor Koding ofrece una serie de herramientas para el desarrollo y despliegue de aplicaciones directamente en la nube, cosa que no se puede realizar con el proveedor OpenShift. Por lo tanto se puede mencionar que Koding es una muy buena herramienta para el desarrollo y despliegue de aplicaciones debido a los resultados obtenidos, a la funcionalidad y herramientas que ofrece el proveedor

a través de su plataforma. Koding demostró un buen resultado en las pruebas que se realizaron a pesar de que es un proveedor gratuito, fue mejor que Azure. Contrariamente de que Azure es un proveedor de paga fue el que más tiempo tomó para realizar los procesos, este proveedor ofrece un amplio repertorio de herramientas para el desarrollo y despliegue de aplicaciones pero todo bajo un costo. Cabe mencionar que si el usuario o empresa cuentan con una buena solvencia económica y puede sustentar una suscripción a Azure, este podría ser una buena opción por toda la serie de herramientas que cuenta su plataforma y también al presentársele alguna serie de problemas, como Azure pertenece a Microsoft tiene un amplio respaldo en cuanto a soporte. Por otro lado si el usuario o empresa no cuentan con tanto capital y desean implementar una aplicación con un buen funcionamiento OpenShift es el indicado, o también Koding es una buena opción, tanto para este tipo de consumidores o usuarios en ambiente académico, ya que brinda una plataforma más amigable en comparación con OpenShift. Dentro de diferentes artículos referentes a este modelo de computación existe mucha inquietud referente a la seguridad, ya que los datos de los usuarios son alojados fuera del entorno del usuario. Uno de los trabajos futuros es analizar la seguridad que brindan los proveedores a través de sus plataformas.

VIII. BIBLIOGRAFÍA

- [1] Ayala, D. and Nichol, S. (2013) "NuSOAP - SOAP Toolkit for PHP". Disponible en: <http://sourceforge.net/projects/nusoap/>
- [2] Bryant, C. (2014). "A Guide to Open Source Cloud Computing Software (PaaS)". Disponible en: <http://www.tomsitpro.com/articles/open-source-cloud-computing-software,2-754-6.html>.
- [3] Erl, T. and Mahmood, Z. and Puttini, R. (2013). "Cloud Computing, Concepts, Technology & Architecture". Prentice Hall.
- [4] Geetha, E. and Suresh, T. and Rajani, K. and Ram, Ch. and Reddy, R. and Raghavendra, R. (2013). "Qos of web service: survey on performance and scalability". B M S College of Engineering, Bangalore, India M S Ramaiah Institute of Technology, Bangalore, India.
- [5] Goyal, M. and Gupta, P. and Aggarwal, A. and Kumar, P. (2012). "QoS based trust management model for Cloud IaaS". Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on, 2012, 843-847.
- [6] Gusev, M. and Ristov, S. (2012) "Superlinear speedup in Windows Azure cloud". Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on, 2012, 173-175.
- [7] Harvey, C. (2014) "60 Open Source Apps You Can Use in the Cloud". Disponible en: <http://www.datamation.com/open-source/60-open-source-apps-you-can-use-in-thecloud-1.html>
- [8] Ladan, M. I. (2011) "Web Services Metrics: A Survey and A Classification". 2011 International Conference on Network and Electronics Engineering IPCSIT vol.11, 2011.
- [9] Li, A. and Yang, X. and Kandula, S. and Zhang, M. (2011) "Comparing Public-Cloud Providers". Internet Computing, IEEE, 2011, 15, 50-53.
- [10] Lifka, D. and Foster, I. and Mehringer, S. and Parashar, M. and Redfern, R. and Stewart, C. and Tuecke, S. (2013) "XSEDE Cloud Survey Report" The Extreme Science and Engineering Discovery Environment.
- [11] Malawski, M. and Kuźniar, M. and Wojcik, P. and Bubak, M. (2013) "How to Use Google App Engine for Free Computing". Internet Computing, IEEE, 2013, 17, 50-59.
- [12] Mares, O. and Villavicencio, A. and Delgadillo, O. and Ibanez, J. and Diaz, A. (2013) "Análisis Comparativo entre .Net y Mono" Tercer Congreso Nacional de Tecnologías de la Información y Comunicación (Conatic 3.0). Tercer Congreso Nacional de Tecnologías de la Información y Comunicación (Conatic 3.0), 2013.
- [13] Panettieri, J. and Katz, A. (2013) "Top 100 Cloud Services Providers: 2013 Edition" Talkin' Cloud, 2013.
- [14] Peng, J. and Zhang, X. and Lei, Z. and Zhang, B. and Zhang, W. and Li, Q. (2009) "Comparison of Several Cloud Computing Platforms" Information Science and Engineering (ISISE), 2009 Second International Symposium on, 2009, 23-27
- [15] Qu, L. and Wang, Y. and Orgun, M. (2013) "Cloud Service Selection Based on the Aggregation of User Feedback and Quantitative Performance Assessment" Services Computing (SCC), 2013 IEEE International Conference on, 2013, 152-159.
- [16] Rimal, B. and Choi, E. and Lumb, I. (2009) "A Taxonomy and Survey of Cloud Computing Systems" INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on, 2009, 44-51.
- [17] Rigsby, J. (2012) "Koding Makes Development Social With New Browser-Based Tool". Disponible en: <http://www.cmswire.com/cms/web-experience/koding-makes-development-social-with-new-browser-based-tool-016706.php>
- [18] Nadir, K. and Zang, T. (2012) "Survey and comparison for Open and closed sources in cloud computing" International Journal of Computer Science Issues (IJCSI); Vol. 9 Issue 3, 2012, p118
- [19] Sefraoui, O. and Aissaoui, M. and Eleuldj, M. (2012) "OpenStack: Toward an Open-Source Solution for Cloud Computing" International Journal of Computer Applications (0975 - 8887) Volume 55 - No. 03.
- [20] Takako, P. and Estacio, G. and Kelner, J. and Sadok, D. (2010) "A Survey on Open-source Cloud Computing Solutions" VIII Workshop em Clouds, Grids e Aplicacoes (WCGA), Federal University of Pernambuco, Brazil, 2010.
- [21] Zhou, M. and Zhang, R. and Zeng, D. and Qian, W. (2010) "Services in the Cloud Computing era: A survey" Universal Communication Symposium (IUCS), 2010 4th International, 2010, 40-46.
- [22] Williams, A. (2014) "Survey: What are the Best Open Source Cloud Projects?". Disponible en: <http://www.linux.com/news/enterprise/cloud-computing/781062-survey-what-are-thebest-open-source-cloud-projects>
- [23] Koding (2009). ". Disponible en: <http://www.koding.com>
- [24] OpenShift (2014). Disponible en: <https://www.openshift.com>
- [25] Azure (2014). Disponible en: <http://azure.microsoft.com>

IX. BIOGRAFÍA



José Alberto López Cázarez nació en la ciudad de Mexicali, Baja California, el 3 de Enero de 1986. Egresó del Instituto Tecnológico de Mexicali, de la carrera de Ingeniería en Sistemas Computacionales, en el año 2008, y de la Maestría en Ingeniería Electrónica con especialidad Computación Ubicua en el 2014, de la misma institución

Actualmente labora como profesor de asignatura en la UNIDEP, en Mexicali, impartiendo cátedra a estudiantes de la carrera de Ingeniería en Sistemas Computacionales. Trabajó como desarrollador de software administrativo y contable en la empresa INNOVA, en el periodo 2008-2009, y como desarrollador de software administrativo en la empresa Grupo GM Transportes, en el periodo comprendido entre los años 2010 al 2012. De ente sus tema de interés destacan la computación en la nube y el desarrollo de sistemas empresariales.

El M.I. López es miembro de la Asociación de Egresados del Instituto Tecnológico de Mexicali.



Juan Francisco Ibáñez Salas nació en Durango, Mex. el 20 de agosto de 1965. Se graduó del Instituto tecnológico de Durango, y obtuvo el grado de Maestría en Ciencias en enseñanza de las Ciencias en El Centro de investigación Educativa en Querétaro, Mex.

Labora en el instituto tecnológico de Mexicali y ha ocupado puestos en el mismo como responsable del área académica de Ing. en sistemas computacionales y la licenciatura en informática, también del centro de cómputo y actualmente es responsable de la división de estudios de posgrado e investigación. Además fue responsable de la subdirección de planeación en el instituto tecnológico de Huatabampo en Sonora (1993), y de la subdirección académica del instituto tecnológico de Ensenada Baja California (1994). Sus áreas de interés incluyen, entre otras, multimedia, y aplicaciones distribuidas.

El M.C. Ibáñez tiene el reconocimiento como perfil deseable desde el año 2010 a la fecha



Arnoldo Díaz Ramírez nació en Mexicali, Baja California, México, el 1ro de Marzo de 1964. Se graduó de la carrera de Ingeniería en Ciencias Computacionales en 1988, y de la Maestría en Tecnología de Redes e Informática en 1998, ambos en el Cety Universidad, campus Mexicali. Obtuvo el

Doctorado en Ciencias de la Computación por la Universidad Politécnica de Valencia, España, en el año 2006, y llevó a cabo una estancia posdoctoral en el CINVESTAV-IPN durante los años 2008 y 2009.

Actualmente trabaja como profesor-investigador en el Instituto Tecnológico de Mexicali, en donde es Coordinador de la Maestría en Ingeniería Electrónica. Adicionalmente, es coordinador del Cuerpo Académico Informática Industrial. Sus áreas de interés incluyen, entre otras, los sistema de tiempo-real, los sistemas ciberfísicos, la computación ubicua y las redes de sensores inalámbricas.

El Dr. Díaz es miembro de la IEEE, y de las sociedades de Computación y Comunicaciones de la misma organización. Tiene el reconocimiento como perfil deseable desde el año 2007 a la fecha.



Leticia Villafuerte León, nacida en la ciudad de Morelia Michoacán, el 06 de Diciembre de 1965. Estudió la Licenciatura en la Universidad Autónoma de Baja California en la ciudad de Mexicali Baja California, México obteniendo el título de Contador Público de la Escuela de Contabilidad y Administración (1994).

Ella actualmente labora como Profesora en el Instituto Tecnológico de Mexicali, adscrita al Departamento de Ciencias Económico Administrativas donde contribuye de forma activa en la formación de futuros profesionistas en las disciplinas de las Ciencias Económico- Administrativa; Dentro de su trayectoria profesional ha prestado sus servicios de forma independiente como Asesora contable, financiera y fiscal, a diversas empresas, ha colaborado con algunas firmas de despachos contables y de auditores. Actualmente se encuentra desempeñando el cargo de Jefa del Departamento de Recursos Financieros en el Instituto Tecnológico de Mexicali, contribuyendo en la administración de los recursos financieros en la ciudad de Mexicali Baja California, México. Su interés en proyectos de investigación son las ciencias económico administrativas donde ha participado como colaboradora en el proyecto de investigación denominado “Los costos en el proceso de producción” y actualmente se encuentra como colaboradora en un proyecto de investigación educativa en la línea de investigación educación- sociedad en el ámbito de cultura financiera.

La C.P. Villafuerte se encuentra afiliada al Colegio Cachanilla de Contadores A.C. en Mexicali Baja California., es miembro activo de la Academia Estatal de Ciencias Económico Administrativas de los Institutos Tecnológicos de Baja California.