

Revisión de las principales metodologías para la construcción de aplicaciones distribuidas en la nube

Barboza Roldán, Marco
Universidad Latina de Costa Rica
Facultad de Tecnologías de la Información y Comunicación
Docente Investigador de la Facultad de Tecnologías de la Información y Comunicación
marco.barboza@ulatina.cr



Recibido 4/jul/2018
Aprobado 24/oct/2018

Resumen:

Desde la creación de la computadora y para que su funcionamiento sea el apropiado, es necesario que una aplicación informática la haga funcionar. En el principio la construcción de las aplicaciones no era sistematizada, pero con el pasar del tiempo la disciplina tecnológica exigió definir metodologías, que permitieran concluir con éxito cada una de esas aplicaciones.

Con el advenimiento de las nuevas tecnologías, fue necesario la adaptación o creación de metodologías que permitieran satisfacer la construcción de las aplicaciones de tal manera que respondieran a las nuevas necesidades. Al día de hoy se disponen de varias metodologías, las cuales no necesariamente pueden ser utilizadas para el desarrollo de aplicaciones que funcionen bajo la tecnología de la nube, por lo que este artículo muestra algunas de las principales metodologías para desarrollo de aplicaciones, tomando como base distintas publicaciones que hacen un estado del arte relacionados con este tema. El artículo suministrará datos al lector, que le permitan

determinar la conveniencia en caso de requerir utilizar una de las técnicas que se describen, teniendo como base alguno de los aspectos que se han indicado para cada una de las metodologías analizadas.

Abstract

Since the creation of the computer and for its operation to be appropriate, it is necessary for a computer application to make it work. In the beginning the construction of the applications was not systematized, but with the passing of time the technological discipline demanded to define methodologies, which would allow to successfully conclude each one of those applications.

With the advent of new technologies, it was necessary to adapt or create methodologies that allowed satisfying the construction of applications in such a way that they responded to new needs. Today we have several methodologies, which can not necessarily be used for the development of applications that work under cloud technology, so this article shows some of the main methodologies for application development, based on different

publications that make a state of the art related to this topic. The article will provide data to the reader, allowing him to determine the convenience in case of requiring use of one of the techniques described, having as a basis some of the aspects that have been indicated for each of the analyzed methodologies

Palabras clave:

Metodologías, Aplicaciones Distribuidas, Nube, Construcción

Keywords:

Methodologies, Distributed Applications, Cloud, Construction

I INTRODUCCIÓN

Desde que la humanidad ha requerido de la construcción de aplicaciones informáticas que han permitido un mejor desarrollo de la sociedad, ha conllevado a la definición de diferentes metodologías para que desarrollo de esas herramientas informáticas. Aunado a esto, la evolución de la tecnología y con el advenimiento del concepto “nube”, ha implicado mayor demanda de estas. Existen casos fallidos de aplicaciones que no lograron terminar completamente o llegan a funcionar de manera parcial. Determinar las razones por las cuales se han presentado estas situaciones, debería empezar la búsqueda en la metodología que fue utilizada y muy importante, si es que efectivamente se utilizó alguna.

Por lo anterior, a continuación, se describen metodologías para el desarrollo de aplicaciones, que fueron diseñadas para aplicaciones tradicionales, otras que fueron adaptadas a que el resultado funcione en la nube y otras que pueden adaptarse para ambas plataformas. Ante esa diversidad de metodologías disponibles, es importante que el responsable de la construcción de desarrollos tecnológicos tenga una noción de algunas de ellas y así pueda determinar cuál sería la que mejor podría adaptarse para obtener un resultado exitoso.

II DESARROLLO

El diccionario de la Real Academia define el concepto “metodología” como “el conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal” (Española, 2014)

En este orden de ideas, ha surgido una corriente para que el desarrollo de las aplicaciones se realice de manera ágil. Para respaldar lo anterior, Eugenia Bahit describe una metodología ágil de la siguiente manera: “Así como existen métodos de gestión de proyectos tradicionales, como el propuesto por el Project Management Institute más conocido como PMI®, es posible encontrar una rama diferente en la gestión de proyectos, conocida como Agile. El desarrollo ágil de software no es más que una metodología de gestión de proyectos adaptativa, que permite llevar a cabo, proyectos de desarrollo de software, adaptándose a los cambios y evolucionando en forma conjunta con el software”. (Bahit, 2012)

Es por ello que las metodologías cumplen la función de apoyar y sistematizar la construcción de aplicaciones computacionales. Para ilustrar lo anterior se describen seguidamente algunas de las metodologías utilizadas mundialmente.

1. SCRUM: Esta metodología fue identificada y definida por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80. Alexander Menzinsky, Gertrudis López, y Juan Palacio estiman sobre esta metodología que “Scrum es un modelo de desarrollo ágil caracterizado por: Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto. Se basa más en la calidad del resultado que en el conocimiento tácito de las personas. Por otro lado, se fija en los equipos autoorganizados que, en la calidad de los procesos empleados, además en el solapamiento de las diferentes fases del desarrollo, que en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada” (Menzinsky, López, & Palacio, 2016)

2. Extreme Programming (XP): Fertilj Krešimir describe esta metodología de la siguiente manera “It is focused on the developer who makes technical decisions, while the customer makes business decisions. This is achieved by intensive customer and developer interaction” [Está enfocado en el desarrollador que toma decisiones técnicas, mientras el cliente toma decisiones comerciales. Esto se logra por clientes intensivos e interacción de los desarrolladores]. (Krešimir & Marija, 2008)

3. Hypertext Design Model (HDM) Para Franca Garzotto, la metodología tiene la siguiente orientación “Some of the most innovative features of HDM are: the notion of perspective; the identification of different categories of links (structural links, application links, and perspective links) with different representational roles; the distinction between hyperbase and access structures; and the possibility of easily integrating the structure of a hypertext application with its browsing semantics. HDM can be used in different manners: as a modeling device or as an implementation device” [Algunas de las características más innovadoras de HDM son: la noción de perspectiva; la identificación de diferentes categorías de enlaces (enlaces estructurales, enlaces de aplicaciones y enlaces de perspectiva) con diferentes roles de representación; la distinción entre hiperbase y estructuras de acceso; y la posibilidad de integrar fácilmente la estructura de una aplicación de hipertexto con su semántica de navegación. HDM se puede usar de diferentes maneras: como una metodología de modelado o como una metodología de implementación] (Garzotto, Paolini, & Schwabe, 1993).

4. Relationship Management Methodology (RMM). A este respecto, Tomás Isakowitz describe “The class of applications for which RMM is most suited exhibits a regular structure for the domain of interest, i.e., there are classes of objects, definable relationships between these classes and multiple instances of objects within each class. Many hypermedia

applications satisfy this requirement. Examples include product catalogs and hypermedia front-ends to traditional database or “legacy” applications. Since many hypermedia applications in this class have volatile data that requires frequent updating, some means to routinize and automate both the initial development and subsequent update process is needed”. [La clase de aplicaciones para las cuales RMM es más adecuada, exhibe una estructura regular para los intereses, por ejemplo, hay clases de objetos, relaciones deficientes entre estas clases y múltiples instancias de objetos dentro de cada clase. Muchas aplicaciones de hipermedia satisfacen este requisito. Entre los ejemplos se incluyen: catálogos de productos y aplicaciones para usuarios finales que contienen hipermedia a las bases de datos tradicionales o aplicaciones “heredadas”. Dado que muchas aplicaciones hipermedia en esta clase tienen datos volátiles que requieren una actualización frecuente, algunos medios por rutina y automatizar tanto el desarrollo inicial, requieren de un proceso de actualización posterior] (Isakowitz, Stohr, & Balasubramanian, 1995).

5. Enhanced Object Relationship Methodology (EORM). Nora Koch describe esta metodología como “Is defined as an iterative process concentrating on the enrichment of the object-oriented modeling by the representation of relations between objects (links) as objects” [un proceso iterativo que se concentra en el enriquecimiento del modelado orientado a objetos mediante la representación de relaciones entre objetos (enlaces) como objetos] (Koch, 1999).

6. Object-Oriented Hypermedia Design Model (OOHDM) Heeseok Lee describe esta metodología de la siguiente manera “adopts four design phases such as conceptual design, navigation design, abstract interface design, and implementation. A view based hypermedia design methodology”. [adopta cuatro fases de diseño como el diseño conceptual, diseño de navegación, diseño de interfaz abstracto y

implementación. Un diseño hipermedia basado en vistas metodología] (Lee, Lee, & Yoo, 1999).

7.Scenario-Based Object-Oriented Hipermedia Design Methodology (SOHDM). Según Heeseok Lee, estas son las características de esta metodología “Consists of six phases: domain analysis; object modeling; view design; navigation design; implementation design; and construction. Users' requirements are analyzed with a responsibility-driven technology using scenarios. Object-oriented views are generated as the result of object modeling, and then used for the subsequent navigation and implementation design. The implementation design phase deals with database schema, page structure and flow, and user interface. This methodology integrates enterprise databases with distributed hypermedia systems via Internet, Intranet, or Electronic Commerce”. [Consta de seis fases: análisis de dominio; modelado de objetos; ver diseño; diseño de navegación; diseño de implementación; y construcción. Los requisitos de los usuarios se analizan con una tecnología basada en la responsabilidad utilizando escenarios. Las vistas orientadas a objetos se generan como resultado del modelado de objetos y luego se usan para el diseño de navegación e implementación posterior. La fase de diseño de la implementación se ocupa del esquema de la base de datos, la estructura y el flujo de la página y la interfaz del usuario. Esta metodología integra bases de datos empresariales con sistemas hipermedia distribuidos a través de Internet, Intranet o Comercio Electrónico] (Lee, Lee, & Yoo, 1999).

8.Projects In Controlled Environments (PRINCE2): Para Kruger Wandí este es el criterio acerca de esta metodología: “is aimed at assisting organisations to manage their projects. It was developed by the UK Office of Government Commerce (OGC) (2009), based on a consolidation of experience from thousands of projects. PRINCE2 provides a

structured approach covering the wide variety of disciplines and activities required for effective project and resource management published in a single document - Managing Successful Projects with PRINCE2. The focus throughout PRINCE2 is on the business case, which describes the rationale and business justification for the project. PRINCE2 applies four key elements to each project: 1.Seven Principles - the guiding obligations and good practices which determine whether the project is being

managed using PRINCE2 2. Seven Processes - steps in the Project 3. Seven Themes - or aspects that must be addressed continually throughout the Project 4. Project Environment - the need to tailor PRINCE2 to a specific context [tiene como objetivo ayudar a las organizaciones a administrar sus proyectos. Fue desarrollado por la Oficina del Reino Unido de Government Commerce (OGC) (2009), basado en la consolidación de la experiencia de miles de proyectos. PRINCE2 proporciona un enfoque estructurado que cubre la amplia variedad de disciplinas y actividades requeridas para gestión efectiva de proyectos y recursos publicada en un documento único - Gestión de proyectos exitosos con PRINCE2. El enfoque en PRINCE2 está en el caso de negocios, que describe los fundamentos y la justificación del proyecto para el negocio. PRINCE2 aplica cuatro elementos clave a cada proyecto: 1. Siete principios: las obligaciones rectoras y las buenas prácticas que determinan si el proyecto se está llevando a cabo administrado usando PRINCE2 2. Siete procesos: pasos en el proyecto 3. Siete temas o aspectos que deben abordarse continuamente a lo largo del proyecto 4. Ambiente del proyecto: la necesidad de adaptar PRINCE2 a un contexto específico] (Wandi & Riaan, 2013)

9. Feature Driven Development (FDD). Bedir Tekinerdoğan indica sobre esta metodología “Los desarrollos hechos con esta metodología han tenido que adaptarse a entregas frecuentes, tangibles y con resultados. Otra característica

notable de esta metodología es que la definición de "característica", se refiere como "una funcionalidad valorada por el cliente, la cual puede ser implementada en dos semanas o menos". Las características hacen que sean fáciles de seguir y ágiles. Por eso, incluye un "factor de tiempo", que se cree tener sutilmente combinación entre los factores técnicos (por ejemplo, agilidad) y sociales factores psicológicos (por ejemplo, fomentar el valor) en el desarrollo del software (Tekinerdoğan, Moreira, Araújo, & Clements, 2004).

10. Accessibility for Web Applications (AWA). Lourdes Moreno López describe esta metodología de la siguiente manera "proporciona un espacio de trabajo para incluir el requisito de la accesibilidad en las organizaciones dedicadas al desarrollo web. AWA proporciona guías a los profesionales de la ingeniería para incorporar requisitos de accesibilidad desde distintas perspectivas: (a) en las organizaciones y empresas, integrando políticas de accesibilidad y calidad; (b) en el proceso de desarrollo siguiendo un enfoque metodológico, que aporte sistematización en la integración de la accesibilidad desde el inicio y, por último, (c) siguiendo un enfoque de Diseño Centrado en el Usuario

(DCU) que sitúe al usuario como protagonista y le haga participar del proceso de diseño (Moreno López, 2010).

11. Web Development Life Cycle (WDLC) Aaron M. French indica al respecto sobre esta metodología "El ciclo de vida de desarrollo web es una metodología que se propone específicamente para el desarrollo de aplicaciones web. El WDLC es un híbrido de dos metodologías anteriores conocidas como el ciclo de vida de desarrollo de sistemas y creación de prototipos. El WDLC usa componentes de cada metodología, combinándolas en un nuevo enfoque que disminuirá el tiempo de desarrollo, agregue estructura a un problema no estructurado y mantenga a los usuarios involucrados a lo largo

de todo el ciclo de vida de desarrollo" (French, 2011).

12. Metodología de Desarrollo de Aplicación Web Utilizando Datos Enlazados (MEDAWEDE). David Wood describe esta metodología de la siguiente manera: "los datos enlazados implican un proceso complejo con diferentes etapas no excluyentes, por lo tanto, el ciclo de vida de los datos bajo tecnologías dato enlazado no es un proceso estático sino dinámico" (Wood, Zaidman, Ruth, & Hausenblas, 2014).

III. COMPUTACION EN LA NUBE

El concepto de computación en la nube, toma auge después del 2000, cuando las redes de comunicaciones logran el soporte del tráfico de datos que demanda este concepto, así como la tecnología evoluciona de una manera acelerada, provoca un uso reiterado de las aplicaciones. Esta evolución pasa del almacenamiento de información a otros servicios complementarios que aumentan la complejidad de las aplicaciones al considerar otros aspectos de concurrencia.

El concepto de la computación en la nube, la describe Carlos A. Guerrero de la siguiente manera: "La computación en la nube es un paradigma que almacena la información en servidores de Internet y se utiliza desde clientes con almacenamiento temporal, incluye el uso de equipos de escritorio, tabletas, portátiles, etc. Es un modelo que permite al usuario acceder a servicios estandarizados para responder a sus necesidades de manera adaptativa, rápida y flexible, pagando únicamente por el consumo realizado". Agrega a la descripción anterior "en el área de desarrollo, orientado a la computación en la nube, se siguen utilizando las mismas metodologías de los procesos de desarrollo tradicionales" (Guerrero & Londoño, 2016)

Por otra parte, Sajid Ibrahim Hashmi, al hacer referencia al criterio anterior expresa lo siguiente: "With the expansion of national markets beyond geographical limits, success of

any business often depends on using software for competitive advantage. Furthermore, as technological boundaries are expanding, projects distributed across different geographical locations have become a norm for the software solution providers. Nevertheless, when implementing Global Software Development (GSD), organizations continue to face challenges in adhering to the development life cycle” [Con la expansión de los mercados nacionales, más allá de los límites geográficos, el éxito de cualquier empresa depende del software para obtener una ventaja competitiva. A medida que se expanden los límites tecnológicos, los proyectos distribuidos en diferentes ubicaciones geográficas, se han convertido en una norma para los proveedores de soluciones de software. Sin embargo, al implementar Global Software Development (GSD), las organizaciones continúan enfrentando desafíos para cumplir con el ciclo de vida de desarrollo] (Hashmi, y otros, 2011). A este respecto, Michael Armbrust considera lo siguiente, al hacer referencia a la computación en la nube: “We predict cloud computing will grow, so developers should take it into account. Regardless of whether a cloud provider sells services at a low level of abstraction like EC2 or a higher level like AppEngine, we believe computing, storage, and networking must all focus on horizontal scalability of virtualized resources rather than on single node performance” [Prevedemos que la computación en la nube crecerá, por lo que los desarrolladores deben tenerlo en cuenta. Independientemente de si una nube proveedor vende servicios a un bajo nivel de abstracción como EC2 o un nivel superior como App Engine, creemos en la informática, el almacenamiento y la red deben centrarse en la escalabilidad horizontal de virtualización de recursos en lugar de en un solo nodo actuación] (Armbrust, y otros, 2010). Con el propósito de aclarar lo anterior, los mismos autores establecen una serie de obstáculos y oportunidades respecto de la

computación en la nube. Tal y como se indica en la siguiente tabla:

Table 2. Top 10 obstacles to and opportunities for growth of cloud computing.

Obstacle	Opportunity
1 Availability/Business Continuity	Use Multiple Cloud Providers
2 Data Lock-In	Standardize APIs; Compatible SW to enable Surge or Hybrid Cloud Computing
3 Data Confidentiality and Auditability	Deploy Encryption, VLANs, Firewalls
4 Data Transfer Bottlenecks	FedExing Disks; Higher BW Switches
5 Performance Unpredictability	Improved VM Support; Flash Memory; Gang Schedule VMs
6 Scalable Storage	Invent Scalable Store
7 Bugs in Large Distributed Systems	Invent Debugger that relies on Distributed VMs
8 Scaling Quickly	Invent Auto-Scaler that relies on ML; Snapshots for Conservation
9 Reputation Fate Sharing	Offer reputation-guarding services like those for email
10 Software Licensing	Pay-for-use licenses

Tabla 3 Los 10 principales obstáculos y oportunidades para el crecimiento de la computación en la nube.

	Obstáculo	Oportunidad
1	Disponibilidad/Continuidad del negocio	Uso de múltiples proveedores de la nube
2	Datos cerrados	API (application programming interface: interface de programación de aplicaciones) estándares; compatible con software para habilitar Surge o computación en la nube híbrida
3	Confidencialidad y auditoría de los datos	Implementación de cifrado,

		redes virtuales y firewalls
4	Cuellos de botella en la transferencia de datos	Discos rápidos; switches de alto nivel
5	Imprevisibilidad del rendimiento	Soporte a máquinas virtuales mejorados; Memoria flash; horarios de máquinas virtuales
6	Almacenamiento escalado	Invencción de tiendas escalables
7	Errores en grandes sistemas distribuidos	Depuradores innovadores que dependan de las máquinas virtuales distribuidas
8	Escalar rápidamente	Innovaciones autoescalables que dependen del lenguaje de programación ; snapshots para la conservación
9	Reputación del destino compartido	Ofrecer servicios de protección de la reputación como los de correo electrónico
10	Licenciamiento de software	Licencias de pago por uso

Fuente: (Armbrust, y otros, 2010)

Precisamente, en la tabla anterior se puede observar que existe una gran cantidad de oportunidades para la construcción de aplicaciones distribuidas en la nube. No obstante, las mismas requieren de una metodología para que las aplicaciones tengan éxito.

IV. ELEMENTOS POR CONSIDERAR EN EL MOMENTO DE CONSTRUIR UNA APLICACIÓN PARA LA NUBE

En cuanto a lo que representan las aplicaciones que serán ejecutadas en internet, Jorge Luis Pérez considera “Una página web tiene la misión esencial de vincular los visitantes con el vasto mundo de la información de un modo agradable y eficiente. Los contenidos sin un formato normalizado en el web y las facilidades que brinda la hipermedia permiten, además de acceder a la información de un modo entretenido, satisfacer los requerimientos y gustos de los más disímiles usuarios” (Subirats, 2003). Tom Kirkham complementa el pensamiento anterior de la siguiente manera sobre las aplicaciones orientadas a la Web: “El uso de los servicios web para crear negocios poco vinculados con los procesos, se basa en la orquestación para lograr la funcionalidad de aplicación distribuida deseada. Por lo general, esto se hace mediante el acoplamiento de servicios estables en un flujo de trabajo estático. Sin embargo, en entornos dinámicos los servicios son a menudo no estáticos y con frecuencia cambia el estado y las características. También el medio ambiente puede ser inestable y como resultado, el flujo de trabajo tiene que adaptarse dinámicamente a los cambios y condiciones de ejecución (Kirkham & Winfield, 2011).

Otra de las actividades que mejor han aprovechado la internet, es la educación. Diana Butucea razona al respecto “Desarrollar una plataforma de software de e-aprendizaje robusta representa un desafío muy importante para tanto el director de proyecto como el equipo de desarrollo. Debido a que las funcionalidades de

estos sistemas de software mejoran y crecen cada día y deben ser tomados en consideración, por ejemplo, flujos de trabajo (Butucea & Cervinski, 2011).

Para finalizar este tema, Antonio Navarro apunta “La etapa de conceptualización es un paso clave en el proceso de construcción y mantenimiento de sitios web, pero está lejos de ser una tarea simple.” (Navarro & Fernández-Valmayor, 2014)

V. CONSIDERACIONES SOBRE LA CONCURRENCIA DE USUARIOS Y DATOS

El concepto de aplicaciones distribuidas lo define Muhammad Fraz Malik de la siguiente manera: “In distributed software system the different components located of different locations can communicate and coordinate their actions by passing the messages. All processor have equal rights to access shared memory between the distributed computing for exchanging their information between the multiple processors”. [En las aplicaciones distribuidas, los diferentes componentes son ubicados en diferentes lugares las cuales pueden comunicarse y coordinar sus acciones pasando los mensajes. Todos los procesadores tienen los mismos derechos para acceder a la memoria compartida entre los sistemas distribuidos para intercambiar su información entre los procesadores múltiples] (Malik & Khan, 2016). Para complementar lo anterior Azeem Sarwar advierte sobre la importancia de las aplicaciones en la nube, de la siguiente manera: “Before truly benefiting from cloud computing, there are several issues associated to it which need to be addressed in the first place. One of the most important aspects that needs special attention pertains to the cloud security. Cloud computing has the important component as trust management. In this paper, we look at some security services practices like authentication, confidentiality and integrity as well as the trust management”. [Antes de beneficiarse de la computación en la nube, hay

varios problemas asociados a ella que deben abordarse, en primer lugar: la seguridad en la nube. Es uno de los aspectos más importantes que requieren atención especial. La Nube tiene un componente importante cual es la administración de la confianza] (Sarwar & Ahmed Khan, 2013).

Los sistemas de información distribuidos pueden situarse dentro de un marco referencial tal y como lo propone Nalini Belaramani “Para construir sistemas de almacenamiento distribuido podrían ubicarse dentro de políticas de arquitectura, la cual tiene dos aspectos. Primero, un conjunto común de mecanismos que permiten la implementación de nuevos sistemas simplemente definiendo nuevas políticas. En segundo lugar, una estructura sobre cómo deberían especificarse las políticas” (Belaramani, y otros, 2009).

Filosóficamente B. R. Badrinath Rutgers y Krithi Ramamritham opinan respecto a la concurrencia en los sistemas de información de la siguiente manera: “La concurrencia de las transacciones que se ejecutan en tipos de datos atómicos se puede mejorar mediante el uso de información semántica de las operaciones definidas en estos tipos. Hasta ahora, la conmutatividad de las operaciones se ha explotado para proporcionar una concurrencia mejorada y evitar interrupciones en cascada” (Badrinath Rutgers & Ramamritham, 1992). Complementa el concepto anterior Peter Bailis de la siguiente manera “Para minimizar la latencia de la red y permanecer en línea durante las fallas del servidor y las particiones de red, muchos sistemas modernos de almacenamiento de datos distribuidos evitan la funcionalidad transaccional, que proporciona fuertes garantías semánticas para grupos de operaciones múltiples sobre múltiples elementos de datos” (Bailis, y otros, 2013).

Brandon Holt opina con relación a las aplicaciones distribuidas “Las aplicaciones modernas se distribuyen: desde las aplicaciones web interactivas más simples hasta las redes sociales con centros de datos masivos en todo el

mundo. Incluso, la simple distribución de las aplicaciones depende de un ecosistema complejo de servidores, bases de datos y cachés para operar. Para escalar los servicios y manejar el tráfico de Internet turbulento, los desarrolladores de aplicaciones distribuidas equilibran constantemente las compensaciones fundamentales entre el paralelismo y localidad, replicación y sincronización, consistencia y disponibilidad. Esta tarea se hace más difícil por el hecho de que cada componente opera de forma independiente por diseño, sabiendo poco sobre el propósito original de la aplicación o de sus necesidades específicas de rendimiento. Las capas de abstracción entre la aplicación y sus datos previenen el sistema se adapte para cumplir mejor los requisitos de la aplicación.” (Holt, 2016).

Considera Pressman independientemente del tipo de desarrollo que se realice, “El aseguramiento de la calidad del software (ACS), es una actividad que se aplica a todo el proceso del software. El ACS incluye procedimientos para la aplicación eficaz de métodos y herramientas, supervisa las actividades de control de calidad, tales como revisiones técnicas y las pruebas del software, procedimientos para la administración de cambio y elaboración de reportes” (Pressman, 2010). La idea anterior es reafirmada por Syed Mohsin Saif al indicar: “un aspecto muy crítico en el desarrollo de software en general y la aplicación web en particular es la estimación del esfuerzo. Es un proceso que se adapta para predecir los esfuerzos necesarios para el desarrollo de la aplicación web en las primeras etapas de su desarrollo. Los esfuerzos se estiman después de aproximar las medidas de tamaño funcional, longitud y complejidad de la aplicación web que se desarrollará” (Saif & Wahid, 2017).

VI. CONCLUSIONES

Desde que los sistemas informáticos son parte de la vida cotidiana de los seres humanos, ha sido necesario definir metodologías que

permitan apoyar los desarrollos tecnológicos con el fin de que se cumplan de manera satisfactoria los objetivos que se han planteado. Se describieron metodologías, sin importar su afinidad al desarrollo convencional, web o ambas. Con el surgir del internet y la posibilidad de que esta plataforma permita la ejecución de aplicaciones, se pensó en algún momento que era necesaria la definición de metodologías que se adaptasen a esta forma de utilizar la tecnología. Como se dio a conocer, muchas de las metodologías que se describen, se adaptan a la construcción de aplicaciones hacia la nube. Hoy en día es importante hacer notar que la mayoría de las aplicaciones que demanda la industria computacional, deben orientarse hacia el funcionamiento en la nube. En síntesis, existen diversas metodologías que se orientan para desarrollar aplicaciones en la nube; unas fueron adaptaciones a partir de desarrollos convencionales, mientras que otras, se definieron específicamente para desarrollos llamados “tradicionales”.

El punto importante en todo este tema, es que, para construir una herramienta computacional, se debe utilizar alguna metodología que sistematice dicha construcción y que así el resultado del producto, no solo sea exitoso, sino que sea tal y como fue proyectado. Por ello no se deben dejar de lado elementos que conllevan las aplicaciones que se ejecutarán en la nube, como lo son: seguridad, esfuerzo y concurrencia, los cuales resultan sensibles para este tipo de plataforma.

Referencias

- Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., . . . Zaharia, M. (2010). A View of Cloud Computing. *Communications of the acm*, 50-58.
doi:10.1145/1721654.1721672
- Badrinath Rutgers, B., & Ramamritham, K. (1992). Semantics-based concurrency control: beyond commutativity. *ACM*

- Transactions on Database Systems (TODS)*, 163-199.
doi:10.1145/128765.128771
- Bahit, E. (2012). *Scrum y eXtreme Programming para Programadores*. Buenos Aires: Creative Commons Atribución.
- Bailis, P., Davidson, A., Fekete, A., Ghodsi, A., Hellerstein, J., & Stoica, I. (2013). Highly available transactions: virtues and limitations. *Proceedings of the VLDB Endowment VLDB Endowment*, 181-192.
doi:10.14778/2732232.2732237
- Belaramani, N., Zheng, J., Nayate, A., Soulé, R., Dahlin, M., & Grimm, R. (2009). PADS: a policy architecture for distributed storage systems. *NSDI'09 Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, 59-73.
Obtenido de https://www.usenix.org/legacy/event/nsdi09/tech/full_papers/belaramani/belaramani.pdf
- Butucea, D., & Cervinski, C. (2011). Building Robust E-learning Software Systems Using Web. *INFOREC Association*, 107-118. Obtenido de <https://search.proquest.com/docview/887898237?accountid=32236>
- Cuaresma, M. J., Mejías Risoto, M., & Torres Valderrama, J. (2002). Methodologies to develop Web Information Systems and Comparative Analysis. *the European Online Magazine for the Information Technology Professional*, 25-36.
- Española, R. A. (2014). *Metodología*. Recuperado el 15 de marzo de 2018, de <http://www.rae.es/rae.html>
- French, A. M. (2011). Web Development Life Cycle: A New Methodology for Developing Web Applications. *Journal of Internet Banking and Commerce*, 11.
- Garzotto, F., Paolini, P., & Schwabe, D. (1993). HDM - a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1), 1-26.
- Guerrero, C., & Londoño, J. (2016). Revisión de la Problemática de la Calidad del Software para el Desarrollo de Aplicaciones de Computación en la Nube. *Información Tecnológica*, 61-80.
doi:doi: 10.4067/S0718-07642016000300007
- Hashmi, S., Clerc, V., Razavian, M., Manteli, C., Tamburri, D., Lago, P., . . . Richardson, I. (2011). Using the Cloud to Facilitate Global Software Development Challenges. *IEEE*.
doi:10.1109/ICGSE-W.2011.19
- Holt, B. (2016). Type-Aware Programming Models for Distributed Applications. Washington, Washington, United States. Obtenido de <http://bholt.org/gen/thesis.pdf>
- Kirkham, T., & Winfield, S. (2011). The Challenge of Dynamic Services in Business Process Management. *International Journal of Digital Society (IJDS)*, 544-551.
- Koch, N. (1999). A Comparative Study of Methods for Hypermedia Development. *LMU Technical Report 9905*.

- Krešimir, F., & Marija, K. (2008). An Overview of Modern Software Development Methodologies. *Central European Conference on Information and Intelligent Systems*, 1-7.
- Lange, D. B. (1996). An object-oriented design approach for developing hypermedia information systems. *Journal of Organizational Computing and Electronic Commerce*, 269-293.
- Lee, H., Lee, C., & Yoo, C. (1999). A scenario-based object-oriented hypermedia design methodology. *Elsevier Science B.V.* , 121-138.
- Malik, M., & Khan, M. (2016). An analysis of performance testing in distributed software applications. *International Journal of Modern Education and Computer Science*, 53-60. doi:10.5815/ijmecs.2016.07.06
- Menzinsky, A., López, G., & Palacio, J. (2016). *Scrum Manager Guía de formación*. Iubaris Info 4 Media SL.
- Moreno López, L. (Febrero de 2010). AWA, Marco metodológico específico en el dominio de la accesibilidad para el desarrollo de aplicaciones web. *TESIS DOCTORAL*. Leganés, Madrid, España. Obtenido de <https://docs.google.com/viewerng/viewer?url=https://e-archivo.uc3m.es/bitstream/handle/10016/8213/TesisDoctoral+LourdesMoreno+Feb2010.pdf>
- Navarro, A., & Fernández-Valmayor, A. (2014). Conceptualization of hybrid web sites. *Internet Research; Bradford*, 17(2), 207. doi:<http://dx.doi.org/10.1108/10662240710737059>
- Pressman, R. (2010). *Ingeniería del Software: Un Enfoque Práctico*. Mexico: Mc Graw Hill.
- Saif, S. M., & Wahid, A. (2017). EFFORT ESTIMATION TECHNIQUES FOR WEB APPLICATION DEVELOPMENT: A REVIEW. *International Journal of Advanced Research in Computer Science*, 6. Obtenido de <https://search.proquest.com/docview/1980478172?accountid=32236>
- Sarwar, A., & Ahmed Khan, M. (2013). A Review of Trust Aspects in Cloud Computing Security. *International Journal of Cloud Computing and Services Science*, 116-122. Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.869.2401&rep=rep1&type=pdf>
- Subirats, J. L. (2003). Diseño informacional de los sitios web. *ACIMED*, 11. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352003000600010&lng=es&tlng=en
- Tekinerdoğan, B., Moreira, A., Araújo, J., & Clements, P. (9 de Febrero de 2004). *Early Aspects: Aspect-Oriented Requirements Engineering and*. Obtenido de Early Aspects: <http://trese.cs.utwente.nl/workshops/early-aspects-2004/>
- Wandi, K., & Riaan, R. (2013). Strategic Alignment Of Application Software Packages And Business. *International Business & Economics Research Journal*, 1239-1259. Obtenido de <https://search.proquest.com/docview/1442472172?accountid=32236>

Wood, D., Zaidman, M., Ruth, L., &
Hausenblas, M. (2014). *Linked Data:
Structured Data on the Web*. Manning
Publications.