

Análisis de aplicaciones empleando la computación en la nube de tipo PaaS y la metodología ágil Scrum

RECIBIDO: 15/01/14 ACEPTADO: 08/02/15

LIZ MELISSA ARANA LÓPEZ*
 MARÍA ELENA RUIZ RIVERA**
 NORA LA SERNA PALOMINO***

RESUMEN

Muchas de las aplicaciones desarrolladas para las empresas e instituciones demandan una alta inversión en costos y una planificación excesivamente prolongada en el tiempo de entrega. Para brindar una solución, agilizando los entregables y reduciendo costos, se unió conceptos de tecnologías actuales como son la Computación en la Nube y las Metodologías Ágiles.

El objetivo es demostrar que las aplicaciones se pueden desarrollar en un tiempo mucho menor a las tradicionales y aproximadamente con la mitad del costo estimado.

Para lo cual se realizó una investigación de las aplicaciones que se vienen desarrollando actualmente en la Computación en la Nube de tipo PaaS (Plataforma como Servicio), el concepto teórico y práctico de las Metodologías Ágiles y la unión explosiva de ambos para ofrecer entregables eficientes sin abandonar la calidad que demanda el mercado actual en la realización de aplicaciones.

El resultado es demostrar la eficiencia en el desarrollo de aplicaciones uniendo los conceptos de Computación en la Nube y las Metodologías Ágiles (SCRUM).

Palabras clave: computación en la nube, PaaS, metodología ágil, Scrum, aplicación, hardware, software

ANALYSIS OF APPLICATIONS USING COMPUTING PaaS CLOUD TYPE AND AGILE SCRUM METHODOLOGY

ABSTRACT

Many of the applications developed for companies and institutions demand high investment costs and an excessively long planning lead time. To provide a solution, streamlining and reducing costs deliverables, concepts of current technologies such as Cloud Computing and Agile Methodologies joined.

The objective to demonstrate that applications can be developed in a much smaller traditional and approximately half of the estimated cost time.

For which conducted an investigation of the applications that are being developed today in the Cloud Computing type PaaS (Platform as a Service), the theoretical and practical concept of Agile Methodologies and explosive bonding of both to provide efficient deliverables without leave the quality required by the current market in the development of applications.

The result is to demonstrate the efficiency of application development linking the concepts of Cloud Computing and Agile Methodologies (Scrum).

Keywords: cloud computing, agile methodology, PaaS, Scrum, application, hardware, software

1. INTRODUCCIÓN

Las metodologías tradicionales no son totalmente adecuadas para todos los desarrollos de software. Las razones son diversas, pero quizá las principales son la falta de flexibilidad de su proceso de desarrollo y la necesidad de generar una cantidad excesiva de documentación [SalesForce, 2011].

Los clientes que contratan un desarrollo de software, plantean una serie de necesidades que no se satisfacen con este tipo de metodologías. A diferencia de las metodologías denominadas ágiles, que adaptan el proceso de desarrollo al desarrollo software específico y que permiten fácilmente incorporar y contemplar cambios en los requisitos.

En resumen, el proceso adecuado para este tipo de desarrollo de software es un proceso ágil, iterativo e incremental; es decir, un proceso que se adapte al desarrollo de los diferentes tipos de aplicaciones web, permitiendo la incorporación de nuevos requisitos y sin exigir una excesiva generación de documentación.

Pero para gran parte de su historia, al desarrollo ágil le falta un componente fundamental: una plataforma de desarrollo que apoye los ciclos de desarrollo rápido que hace la metodología de trabajo. En los entornos tradicionales de software, la distribución de software nuevo requiere de parches, reinstalación, y la ayuda del equipo de apoyo. En tal ambiente, meses o incluso años son necesarios para obtener una nueva distribución en manos de los usuarios. La incorporación de sus comentarios en la próxima versión requiere de tiempos incomparables.

Es aquí donde la computación en la nube de tipo PaaS hace una diferencia sustancial, eliminando los engorrosos requisitos de distribución que puede aletargar el desarrollo ágil. No hay parches para distribuir y no son necesarias las reinstalaciones. Con la computación en la nube las nuevas distribuciones están instaladas en servidores alojados y puestos a disposición de los usuarios inmediatamente. Como resultado, es posible que la aplicación que se ejecuta hoy, se haya modificado la noche anterior. Podemos decir que la computación en la nube es el ambiente ideal para el desarrollo ágil. Le permite obtener una funcionalidad valiosa a sus clientes de forma rápida, recoger

* Ingeniero de Sistemas e Informática. Facultad de Ingeniería de Sistemas e Informática, UNMSM. E-mail: liz.meli.arana@gmail.com

** Licenciada en Computación. Profesora de la facultad de Sistemas e Informática, UNMSM. E-mail: meruiz@unmsm.edu.pe, merruri@hotmail.com

*** Doctora en Informática. Profesora de la facultad de Sistemas e Informática, UNMSM. E-mail: nlasernap@hotmail.com

información inmediata y realizar cambios rápidos basados en la retroalimentación. Este ciclo de desarrollo rápido, es una ventaja inherente de la computación en la nube, es imposible implementar en el modelo tradicional de desarrollo debido a los altos costos de distribución.

Objetivo General

Desarrollar aplicaciones empleando la computación en la nube de tipo PaaS y la metodología ágil Scrum con el objetivo de demostrar la disminución del tiempo de desarrollo y el costo de su realización.

Objetivos Específicos

- Definir las distintas metodologías de desarrollo ágil, eligiendo entre ellas la metodología ágil Scrum que se ajusta más a la solución del problema.
- Definir las plataformas de desarrollo en la nube, empleando la plataforma como servicio (PaaS) de Windows Azure que se adapta mejor a las experiencias del autor y cumple con la norma ISO/IEC 27001 en cuanto a la SGSI.
- Comprobar la factibilidad del objetivo principal desarrollando una aplicación de corto alcance que compruebe la disminución del tiempo y el costo siguiendo la norma ISO/IEC 12207 en cuanto al proceso de desarrollo de software.

2. MARCO REFERENCIAL

2.1. Metodología Ágil

Desarrollo Agile no significa FACIL, Metodologías Agiles no se refieren a una sola “práctica” o a una “metodología”. El desarrollo de software mediante métodos ágiles nos permite brindar soluciones de TI basadas en desarrollos iterativos donde los requerimientos y sus soluciones están relacionados directamente con la colaboración entre el cliente, nuestros desarrolladores y los usuarios finales. [Patricio L., +, 2011]:

El Manifiesto Ágil

Comienza enumerando los principales valores del desarrollo ágil. Según el Manifiesto se valora [Patricio L., +, 2011]:

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente.

- Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- **Desarrollar software que funciona más que conseguir una buena documentación.** La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- **La colaboración con el cliente más que la negociación de un contrato.** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo. Los principios son [Patricio L., +, 2011]:

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.

- El software que funciona es la medida principal de progreso.
- Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Scrum

La metodología desarrollada para nuestro sistema será el Scrum que define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la impredecibilidad y el riesgo a niveles aceptables. El mismo surge en 1986, de un artículo de la Harvard Business Review titulado "The New Product Development Game" de Hirotaka Takeuchi e Ikujiro Nonaka, que introducía las mejores prácticas más utilizadas en 10 compañías japonesas altamente innovadoras. A partir de ahí y tomando referencias al juego de rugby, Ken Schwaber y Jeff Sutherland formalizan el proceso conocido como Scrum en el año 1995. [Patricio L., +, 2011]

Scrum es un método iterativo e incremental que enfatiza prácticas y valores de project management por sobre las demás disciplinas del desarrollo. Al principio del proyecto se define el Product Backlog, que contiene todos los requerimientos funcionales y no funcionales que deberá satisfacer el sistema a construir. Los mismos estarán especificados de acuerdo a las convenciones de la organización ya sea mediante: features, casos de uso, diagramas de flujo de datos, incidentes, tareas, etc. El Product Backlog será definido durante reuniones de planeamiento con los stakeholders. A partir de ahí se definirán las iteraciones, conocidas como Sprint en la jerga de Scrum, en las que se irá evolucionando la aplicación evolutivamente. Cada Sprint tendrá su propio Sprint Backlog que será un subconjunto del Product Backlog con los requerimientos a

ser construidos en el Sprint correspondiente. La duración recomendada del Sprint es de un mes.

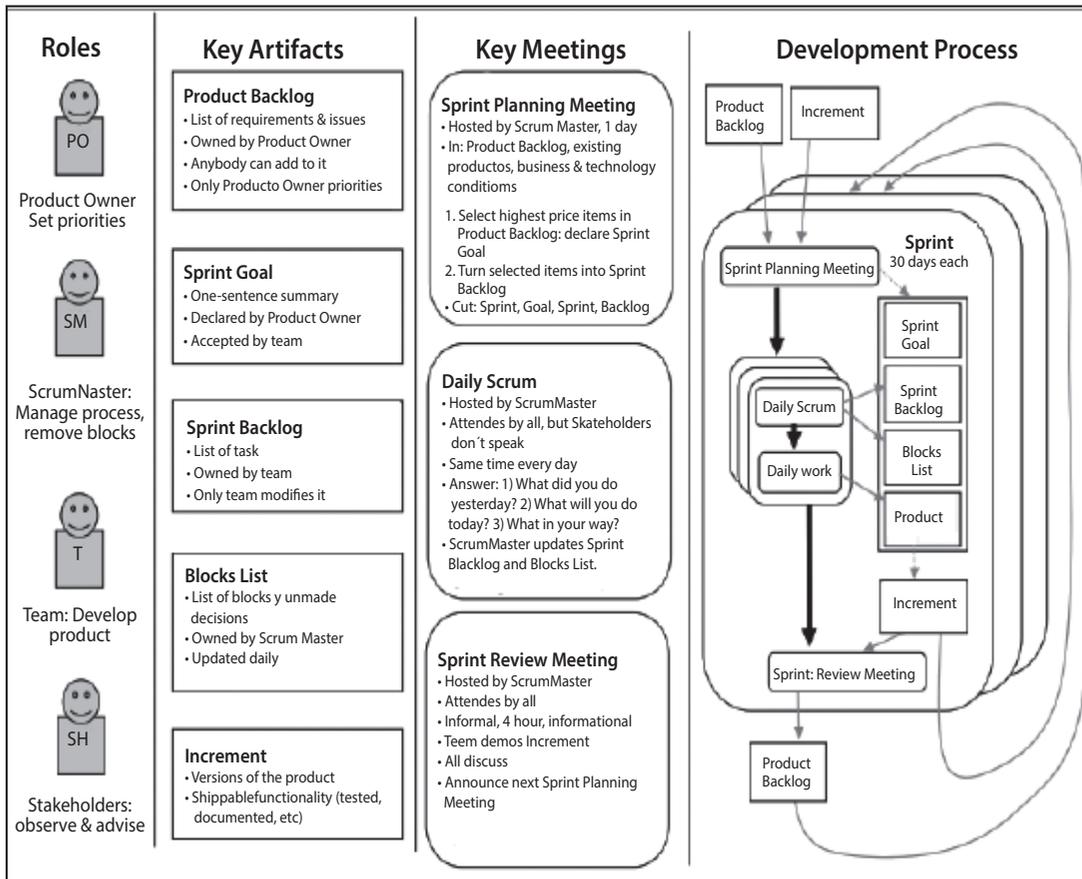
Dentro de cada Sprint el Scrum Master (equivalente al Líder de Proyecto) llevará a cabo la gestión de la iteración, convocando diariamente al Scrum Daily Meeting que representa una reunión de avance diaria de no más de 15 minutos con el propósito de tener realimentación sobre las tareas de los recursos y los obstáculos que se presentan. Al final de cada Sprint, se realizará un Sprint Review para evaluar los artefactos construidos y comentar el planeamiento del próximo Sprint. Figura 1.

El ciclo de vida de Scrum es el siguiente [Patricio L., +, 2011]:

- 1. Pre-Juego: Planeamiento.** El propósito es establecer la visión, definir expectativas y asegurarse la financiación. Las actividades son la escritura de la visión, el presupuesto, el registro de acumulación o retraso (backlog) del producto inicial y los ítems estimados, así como la arquitectura de alto nivel, el diseño exploratorio y los prototipos. El registro de acumulación es de alto nivel de abstracción.
- 2. Pre-Juego: Montaje (Staging).** El propósito es identificar más requerimientos y priorizar las tareas para la primera iteración. Las actividades son planificación, diseño exploratorio y prototipos.
- 3. Juego o Desarrollo.** El propósito es implementar un sistema listo para entrega en una serie de iteraciones de treinta días llamadas "corridas" (sprints). Las actividades son un encuentro de planeamiento de corridas en cada iteración, la definición del registro de acumulación de corridas y los estimados, y encuentros diarios de Scrum.
- 4. Pos-Juego: Liberación.** El propósito es el despliegue operacional. Las actividades, documentación, entrenamiento, mercadeo y venta.

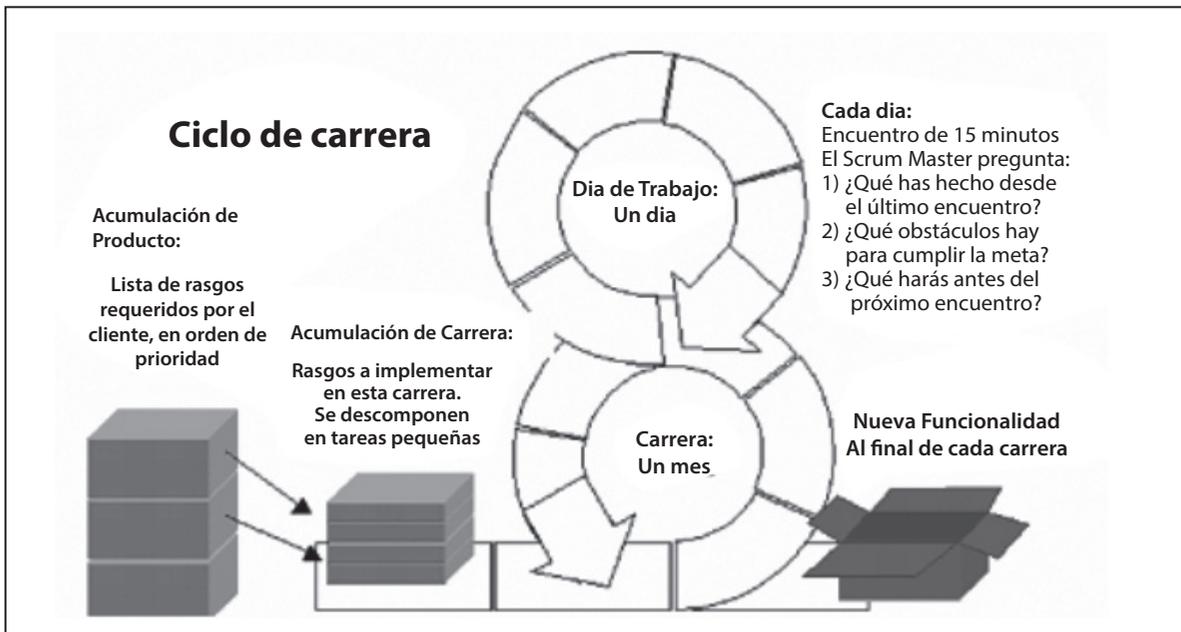
Usualmente los registros de acumulación se llevan en planillas de cálculo comunes, antes que en una herramienta sofisticada de gestión de proyectos. Los elementos del registro pueden ser prestaciones del software, funciones, corrección de bugs, mejoras requeridas y actualizaciones de tecnología. Hay un registro total del producto y otro específico para cada corrida de 30 días. En la jerga de Scrum se llaman "paquetes" a los objetos o componentes que necesitan cambiarse en la siguiente iteración.

Figura 1. Descripción de Roles, artefactos, reuniones y proceso de desarrollo de Scrum. Ciclo de Vida de Scrum.



Fuente: [William C. Wake]

Figura 2. Ciclo de Carrera o de Vida (Sprint) de Scrum.



Fuente: [William C. Wake]

2.2 Computación en la Nube

La computación en la nube o informática en la nube o nube de conceptos, del inglés Cloud Computing, es un paradigma que permite ofrecer servicios de computación a través de Internet. [Francisco M., +, 2011]

El concepto de computación en la nube empezó en proveedores de servicio de Internet a gran escala, como Google, Amazon, AWS y otros que construyeron su propia infraestructura. De entre todos ellos emergió una arquitectura: un sistema de recursos distribuidos horizontalmente, introducidos como servicios virtuales de TI escalados masivamente y manejados como recursos configurados y mancomunados de manera continua. Este modelo de arquitectura fue inmortalizado por George Gilder en su artículo de octubre 2006 en la revista Wired titulado “Las fábricas de información”. Las granjas de servidores sobre las que escribió Gilder eran similares en su arquitectura al procesamiento grid (red, parrilla), pero mientras que las redes se utilizan para aplicaciones de procesamiento técnico débilmente acoplados, este nuevo modelo de nube se estaba aplicando a los servicios de Internet.

En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles en la nube de Internet sin conocimientos (o, al menos sin ser expertos) en la gestión de los recursos que usan. Según el IEEE Computer Society, es un paradigma en el que la información se almacena de manera permanente en servidores de Internet y se envía a cachés temporales de cliente, lo que incluye equipos de escritorio, centros de ocio, portátiles, etc. Esto se debe a que, pese a que las capacidades de los PC han mejorado sustancialmente, gran parte de su potencia se desaprovecha, al ser máquinas de propósito general.

Cloud Computing es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado.

El cambio paradigmático que ofrece computación en nube es que permite aumentar el número de servicios basados en la red. Esto genera beneficios tanto para los proveedores, que pueden ofrecer, de forma más rápida y eficiente, un mayor número de servicios, como para los usuarios que tienen la posibilidad de acceder a ellos, disfrutando de

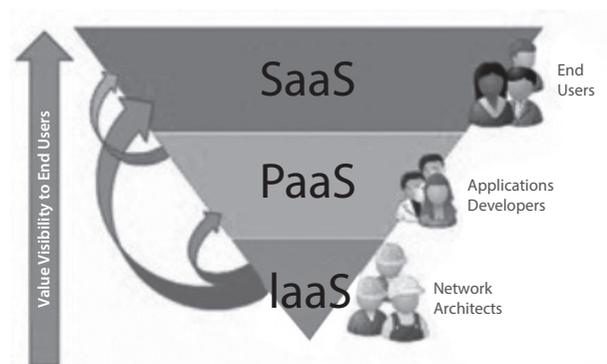
la transparencia e inmediatez del sistema y de un modelo de pago por consumo.

Computación en nube consigue aportar estas ventajas, apoyándose sobre una infraestructura tecnológica dinámica que se caracteriza, entre otros factores, por un alto grado de automatización, una rápida movilización de los recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como virtualización avanzada y un precio flexible en función del consumo realizado, evitando además el uso fraudulento del software y la piratería.

Capas

Este modelo propone una serie de capas o componentes estratificados (ver *Figura. 3*): Software como Servicio (Software as a Service — SaaS), Plataforma como Servicio (Platform as a Service — PaaS) e Infraestructuras como Servicio (Infrastructure as a Service — IaaS) [Francisco M., +, 2011].

Figura 3. Capas del paradigma Cloud Computing.



Fuente: [Francisco M., +, 2011]

El Software como Servicio se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, bajo demanda, es decir una sola instancia del software que corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes. [Francisco M., +, 2011]

La capa central, que es la Plataforma como Servicio, es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una carga de servicios.

La Infraestructura como Servicio se encuentra en la capa inferior y es un medio de proporcionar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red. Servidores,

sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran (por ejemplo a través de la tecnología de virtualización) para manejar tipos específicos de cargas de trabajo —desde procesamiento en lotes hasta aumento de servidor o almacenamiento durante las cargas pico.

Plataforma como un Servicio (PaaS)

Las soluciones PaaS (Platform as a Service) son plataformas de software para las cuales la herramienta de desarrollo en sí misma está alojada en la nube y se accede a través de un navegador web. Los desarrolladores pueden construir, analizar, desarrollar, testear, documentar y poner en marcha aplicaciones en un sólo proceso, sin tener que instalar ninguna herramienta adicional en sus computadoras. [Carla V., +, 2012]

Desde el punto de vista de Christopher Keene, las plataformas PaaS tienen evidentes diferencias funcionales respecto a las plataformas de desarrollo tradicionales. Estas diferencias incluyen: [Keene C., +, 2012]

Herramientas de desarrollo múltiples inquilinos:

Las herramientas de desarrollo tradicionales son para un único usuario – un estudio basado en la nube debe soportar múltiples usuarios, cada uno de los cuales tiene múltiples proyectos activos.

Arquitectura de despliegue de múltiples inquilinos:

La escalabilidad suele no ser una preocupación durante el esfuerzo inicial de desarrollo y se deja para que lo gestionen los administradores de sistemas cuando se despliega el proyecto. En PaaS, la escalabilidad de la aplicación y de sus capas de datos debe ser intrínseca a la solución (por ejemplo, el balanceo de carga y la disponibilidad tienen que ser elementos básicos de la plataforma de desarrollo).

Tipos de Nubes

- **Las nubes públicas** se manejan por terceras partes, y los trabajos de muchos clientes diferentes pueden estar mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos propios.
- **Las nubes privadas** son una buena opción para las compañías que necesitan alta protección de datos y ediciones a nivel de servicio. Las nubes privadas están en una infraestructura en-demanda manejada por un solo cliente que controla qué aplicaciones debe correr y dónde. Son propietarios del servidor, red, y disco y

pueden decidir qué usuarios están autorizados a utilizar la infraestructura.

- **Las nubes híbridas** combinan los modelos de nubes públicas y privadas. Usted es propietario de unas partes y comparte otras, aunque de una manera controlada. Las nubes híbridas ofrecen la promesa del escalado aprovisionada externamente, en-demanda, pero añaden la complejidad de determinar cómo distribuir las aplicaciones a través de estos ambientes diferentes. Las empresas pueden sentir cierta atracción por la promesa de una nube híbrida, pero esta opción, al menos inicialmente, estará probablemente reservada a aplicaciones simples sin condicionantes, que no requieran de ninguna sincronización o necesiten bases de datos complejas. [Francisco M., +, 2011]

2.3. Desarrollo en la nube PaaS con Metodologías Ágiles

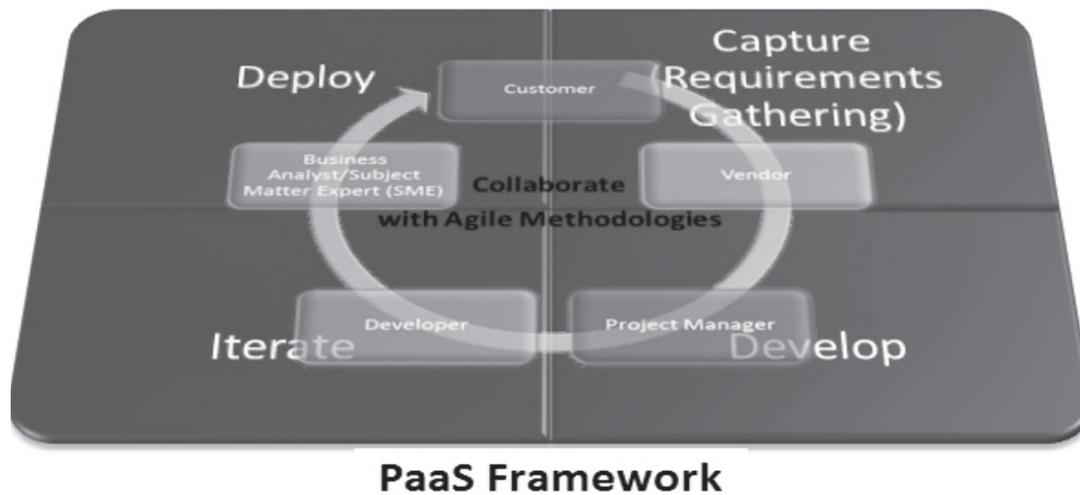
En una época donde la tendencia es superar a la competencia a través del tiempo de comercialización y agregar valor a la empresa, las metodologías ágiles de desarrollo y PaaS hacen una combinación incalculable.

Como una rama de la computación en nube, PaaS es una plataforma como servicio. Está disponible como un servicio de utilidad para desarrollar aplicaciones de software donde los desarrolladores no tienen que preocuparse acerca de la instalación del sistema operativo, la capacidad de almacenamiento o capacidades de alojamiento. Los usuarios de PaaS tienen que preocuparse sólo por el costo de pago que es, al igual que otras ramas del cloud computing, muy pequeño en comparación con los convencionales que en las instalaciones de configuración de TI. El proveedor de servicios PaaS proporciona toda la configuración necesaria para desarrollar una aplicación de software, el sistema operativo, las actualizaciones, las capacidades de alojamiento y de la red y seguridad de datos. [Mridula V., +, 2011]

PaaS ofrece a los desarrolladores ciertos componentes preestablecidos que se pueden personalizar (o editar) en función de las necesidades del cliente. Los desarrolladores no tienen que construir toda la aplicación de software desde cero. Simplemente ahorra tiempo y ayuda a las organizaciones para ofrecer las soluciones más rápido.

Con PaaS, los desarrolladores y administradores de proyectos pueden adoptar el método iterativo donde los requisitos pueden ser reunidos según las etapas de desarrollo. Las soluciones pueden

Figura 4. PaaS y Metodologías Ágiles.



Fuente: [Mridula V., +, 2011]

ser entregadas como prototipo, con un conjunto de datos y, a continuación, las iteraciones se pueden tomar desde allí para llegar a la mejor solución.

Como un resultado natural, PaaS refuerza el desarrollo ágil y metodologías de gestión que las organizaciones tratan de adoptar, a fin de mejorar la capacidad de atención al cliente y reducir el tiempo de entrega de soluciones a las partes interesadas.

También se destaca claramente que PaaS explícitamente promueve un enfoque de colaboración (que tiene sus propias ventajas cuando se combina con metodologías ágiles y cloud computing) para proyectos de desarrollo. Se requiere que los interesados colaboren, comprendan y expresen sus necesidades cada vez mayores a los desarrolladores y al director de proyecto, a fin de obtener un mejor ajuste de soluciones de negocio para lo que están gastando dinero.

Ágil con PaaS es ideal para aquellas situaciones en las que reina la incertidumbre alta y el cliente no está seguro por completo de sus necesidades o el proyecto es demasiado grande para ser documentado en su totalidad. En tal situación, si los proveedores del servicio hacen uso del PaaS, entonces pueden personalizar, incorporando los cambios en la codificación más rápido ya que no tienen que empezar desde el principio.

Windows Azure

Windows Azure es un sistema operativo de servicios en la nube que actúa como entorno de desarrollo, hospedaje y administración de servicios

de la plataforma Windows Azure. Windows Azure proporciona a los desarrolladores servicios informáticos y de almacenamiento a petición para hospedar, escalar y administrar aplicaciones web de Internet a través de los centros de datos de Microsoft.

Windows Azure es una plataforma flexible que admite varios idiomas y se integra con el entorno local existente. Para compilar aplicaciones y servicios en Windows Azure, los desarrolladores pueden usar su experiencia previa con Microsoft Visual Studio. Asimismo, Windows Azure es compatible con los estándares, protocolos y lenguajes más frecuentes, como SOAP, REST, XML, Java, PHP y Ruby. En la actualidad, Windows Azure se comercializa en 40 países.

2.4. Estudio Analítico de la Metodología Ágil con la Computación en la Nube

Artículo de la Universidad de Maharishi Markandeshwar que de India, enfatiza las ventajas de la computación en la nube y las metodologías ágiles. La computación en la nube se ha ido convirtiendo en una moda entre las empresas de todo el mundo. De acuerdo con Gartner (empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, Connecticut, Estados Unidos), la computación en la nube se convertirá en el vehículo preferido para la entrega de aplicaciones en 2015. Así, la mayoría de los CIOs están preparando la infraestructura de TI para adaptarse al entorno de la nube.

Ventajas de la computación en nube con el desarrollo ágil:

- Acortar el ciclo de desarrollo al tiempo de 75 %.
- Mayor estabilidad de cargas de trabajo.
- Una mayor utilización de carga de trabajo, es decir, el desarrollo de gran escala, sistemas de software con un número fijo de desarrolladores.
- Mayor calidad debido a comentarios continuos de los clientes.
- Mayor flexibilidad para cambiar de planes.
- Reducir el costo de mover información entre la gente.
- Colocar a las personas físicamente más cerca.

Análisis oferta-demanda de servicios de despliegue de aplicaciones en la nube (PAAS).

Resumen

Artículo que realiza un análisis comparativo de los servicios PAAS (Platform as a service) actuales orientado hacia las necesidades de las Pymes del medio local, enfocándonos en las siguientes preguntas:

¿Cuál es la problemática de las Pymes actuales con respecto al uso de servicios PAAS? ¿Cuáles son los servicios actuales y cuáles son sus similitudes y desventajas? [Carla L., +, 2012]

Análisis del estudio

Se realiza una comparativa entre tres de los más conocidos proveedores del servicio PAAS en el mercado para identificar los factores necesarios para proveer un servicio aceptable: Google App Engine, Force y Heroku. [Carla L., +, 2012]

En la Tabla 1 se muestra una comparativa de la funcionalidad para evaluar los factores más importantes:

Benchmarking

A continuación se realizara el benchmarking para la tecnología y metodología a utilizar.

Estudio Comparativo de Tecnologías Escogidas para Implementación de la Solución

La computación en la nube de tipo PaaS ofrece innumerables proveedores de servicios para el desarrollo de aplicaciones. Se pretende elegir aquel servicio en la nube que soporte la plataforma .Net, esto básicamente porque es uno de los lenguajes de programación con mayor demanda en el mercado peruano y es la plataforma que domina el autor de la tesis. En la *Tabla 2* se observa el benchmarking de las plataformas actuales en la nube.

Propiedades críticas

A continuación se muestra las propiedades críticas que se tomarán en cuenta en la realización del BenchMarking

- **Interfaz de Control:** Es muy importante que la plataforma del proveedor de servicios tenga las herramientas adecuados de uso para el desarrollar, esto garantiza el desarrollo óptimo de la aplicación sin problemas de infraestructura.
- **Garantía:** Es necesario informarse en este aspecto para tomar decisiones en caso de defectos que afecten el correcto funcionamiento del servicio.
- **Lenguajes de Programación:** Las Plataformas de Servicio en la Nube se pueden desarrollar en varios lenguajes de programación, para ello se tiene que analizar que **lenguaje** se acomoda mejor a nuestros requerimientos.
- **Costo del Plan: Por lo general:** Los costos van de la mano con los beneficios que ofrece

Tabla 1. Comparativa de funcionalidad

	GAE	FORCE	HEROKU
APIs de despliegue	Muy estables y completas	Limitadas, delegando toda la responsabilidad a su consola web	Sencillas, pero son orientadas a la terminal
Precios	Bajos y accesibles.	Bajos, pero con recursos considerablemente reducidos.	Bajos, con recursos considerables.
Integración	Con el development stack de Google.	Requiere planes adicionales.	Es delegada a los frameworks que provee.
Escalabilidad	Alta	Alta	Media
Soporte	Limitado para dispositivos móviles.	Muy bueno para dispositivos móviles.	Dispositivos móviles

Fuente: [Carla L., +, 2012]

el servicio, por este motivo el menor costo no es significado del mejor servicio, se tiene que analizar los beneficios que más se ajusten.

En análisis a estos datos se desarrollará bajo la plataforma de Windows Azure, ya que los criterios requeridos para la solución son satisfechos por esta solución.

La extensa cantidad de metodologías ágiles para desarrollar proyectos, lleva a analizar una clasificación de las más eficientes mediante en benchmarking como se muestra en la *Tabla 3*.

A continuación se muestra las propiedades críticas que se tomarán en cuenta en la realización del BenchMarking.

Características Metodológicas: Son las indicadores básicos para desarrollar un proyecto, muy importantes en el uso de metodologías.

Sistema como algo cambiante: Es un indicador para medir el grado de una metodología para adaptarse a sistemas cambiantes.

Colaboración: En las metodologías ágiles el nivel de colaboración es fundamental, es por ello se debe medir el grado de colaboración en cada una.

Análisis de Metodologías Ágiles y Tradicionales

En la *Tabla 4* se enumera las principales diferencias respecto de las metodologías tradicionales (“no ágiles”). Se recoge esquemáticamente estas diferencias que no se refieren sólo al proceso en sí, sino también al contexto de equipo y organización.

Tabla 2. Benchmarking de Plataformas Tecnológicas en la Nube

Categoría	Indicadores	Windows Azure	Salesforce	LunaCloud	Amazon EC2
Interfaz de control	Aplicación basada en web/ pan. de control	1	1	1	1
	API	1	1	1	1
	Línea de comandos	1			1
Garantía	100%			1	
Lenguajes de Programación	.Net, Java, PHP o Python.	1			
Costo del Plan	Libre		1		
Puntaje		4	3	3	3

(Elaboración propia).

Tabla 3. Benchmarking de Metodologías Ágiles

Categoría	Indicadores	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Características Metodológicas	Resultados	5	5	4	4	4	5	5
	Simplicidad	4	4	3	5	3	5	5
	Adaptabilidad	5	5	3	3	4	4	3
	Excelencia técnica	4	3	3	4	4	4	3
	Prácticas de colaboración	2	5	5	4	3	3	4
Sistema como algo cambiante		5	4	3	3	4	5	5
Colaboración		5	5	4	4	4	5	5
Media Total		4.8	4.5	3.6	3.6	3.9	4.7	4.8

Fuente: (UPV, 2011)

Tabla 4. Diferencias entre Metodologías Tradicionales y Ágiles

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

Fuente: [María E., 2011].

Tabla 5. Diferencias por las características del proyecto

Modelo de Proceso	Tamaño del Proceso	Tamaño del Equipo	Complejidad del Problema
RUP	Medio / Extenso	Medio / Extenso	Medio / Alto
ICONIX	Pequeño / Medio	Pequeño / Medio	Pequeño / Medio
XP	Pequeño / Medio	Pequeño	Medio / Alto
SCRUM	Pequeño / Medio	Pequeño	Medio / Alto

Fuente: [María E., 2011].

Diferencias por las características del Proyecto

En la Tabla 5 se presenta una comparativa de los modelos de proceso en cuanto a las características del proyecto, analizamos el tamaño del proceso, del equipo y la complejidad del problema para cada uno de los modelos. Podemos resaltar que: con un pequeño equipo de desarrollo se puede realizar grandes proyectos, de alta complejidad; es el caso de XP y SCRUM. [María E., 2011].

Diferencias por la curva de Aprendizaje

En la Tabla 6 con respecto a la curva de aprendizaje, vemos que los modelos ágiles, nos ofrecen una mayor ventaja pero con ciertas limitaciones, ya que aún no han sido explotadas a gran escala como lo es RUP que posee un alto soporte y herramientas integrales que nos guían a través del mismo, facilitando aplicar con mayor efectividad esta metodología, permitiendo aprovecharla al máximo. [María E., 2011]

Tabla 6. Diferencias por la curva de aprendizaje

Modelo de Proceso	Curva de aprendizaje	Herramienta de integración	de Soporte Externo
RUP	Lenta	Alto Soporte	Alto Soporte
ICONIX	Rápida	Algún Soporte Disponible	Algún Soporte Disponible
XP	Rápida	No mencionado	Algún Soporte Disponible
SCRUM	Rápida	No mencionado	Algún Soporte Disponible

Fuente: [María E., 2011].

3. ANÁLISIS Y DISEÑO

En el presente capítulo se introduce la contribución teórica que brinda la tesis, las herramientas y metodologías aplicadas para cumplir con los objetivos propuestos. Para demostrar la disminución del tiempo y el costo de desarrollo de aplicaciones en la nube con metodologías ágiles, se ha tomado de ejemplo un software realizado con una metodología tradicional como es el RUP y se ha desarrollado la misma aplicación con la metodología ágil en la nube de tipo PaaS de Windows Azure.

Aplicación a desarrollar: Portal de Gestión de Recursos Humanos (PGR)

La aplicación web Portal de Gestión de Recursos Humanos (PGR) actualmente se encuentra en producción con la finalidad de gestionar al personal de la empresa Servicio de Call Center del Perú. El propósito de la aplicación es organizar de manera adecuada los procesos de Administración de Personal clasificándolos en módulos, para el presente proyecto nos centraremos sólo en el módulo de captación debido a que interactúa con muchos usuarios públicos. El desarrollo de la aplicación se realizó bajo la metodología RUP, previa elaboración de un cronograma para medir los tiempos y costos que implica el desarrollo completo en las fases del RUP. En la Figura 5 se presenta dicho cronograma, establecido para todos los módulos, sin embargo se observa que para el módulo de captación se tiene un tiempo determinado de 2 meses aproximadamente con los procesos previos de captura de requerimientos, análisis y documentación.

En la Figura 6 se muestra la presentación de la página de inicio de la aplicación actual en producción, se visualiza todos los módulos que existentes pero por fines de pruebas de investigación y seguridad de la información sólo se desarrollará el módulo de captación como se ha mencionado previamente.

4. CONCLUSIONES

- Los procesos que se han seguido con la metodología ágil Scrum son una recopilación de varios autores, puesto que este tipo de metodología no tiene un estándar a seguir, cada equipo se acopla de una manera más novedosa y productiva según las necesidades que presente la situación y precisamente esa es la clave de su éxito.
- El uso de la tecnología Windows Azure como computación en la nube de tipo plataforma como servicio (PaaS), ha sido un gran facilitador por dos razones principales: primero, la plataforma de programación .net se ha acoplado sin presentar ninguna dificultad a lo largo del desarrollo, y segundo, la base de datos sql azure presenta una interfaz de usuario muy amigable para el programador, sin previo entrenamiento el avance del desarrollo fue inmediato.
- En la solución presentada se observa que los resultados concuerdas con los objetivos planteados al inicio del desarrollo. Existe una disminución en el tiempo de desarrollo en un porcentaje de 75% (de 2 meses a 2 semanas) y una reducción aproximada en el costo de desarrollo de 65% (de \$13000.00 a \$4300.00). Esto concluye que el desarrollo de aplicaciones en la nube con metodologías ágiles disminuye el tiempo de desarrollo y reduce los costos de su realización.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] William.Wake@acm.org. www.xp123.com
- [2] [SalesForce, 2011]: SalesForce.com, Agile Development Meets Cloud Computing for Extraordinary Results at Salesforce.com, 2011. http://www.developerforce.com/media/ForcedotcomBookLibrary/WP_Agile_112608.pdf
- [3] [Patricio L., +, 2011]: Patricio Letelier, Carmen Penadés, Metodologías Ágiles para el Desarrollo

Cronograma de actividades.

Figura 5

Nombre de tarea	Costo	Tiempo	Duración	Comienzo	Fin	Predece	Nombre de los recursos
1 PORTAL DE GESTIÓN DE RRHH	\$71,680.00	560 horas	65 días?	mar06/03/12	jun04/06/12		
2 Iniciación	\$100.00	16 horas	2 días?	mar06/03/12	mié07/03/12		
3 Definición de Servicio	\$0.00	8 horas	1 días?	mar06/03/12	mar06/03/12		Gestion Interna
4 Estimación de Servicio	\$0.00	8 horas	1 días?	mié07/03/12	mié07/03/12		Gestion Interna
5 Planeamiento	\$980.00	96 horas	12 días?	jué08/03/12	vié23/03/12		
6 Gestionar Alcance/Costo/Tiempo	\$240.00	8 horas	1 día?	jué08/03/12	jué08/03/12	4	Consultor de Proyecto
7 Elaboración de Propuesta Técnica de Servicio	\$240.00	8 horas	1 día?	vie09/03/12	vie09/03/12	6	Consultor de Proyecto
8 Entrega de Propuesta Técnica de Servicio	\$0.00	80 horas	10 días?	lun12/03/12	vie23/03/12	7	Usuario Funcional
9 Ejecución	\$6,700.00	444 horas	48 días?	lun26/03/12	mié29/05/12		
10 Funcional	\$1,280.00	64 horas	8 días?	lun26/03/12	mié04/04/12		
11 Análisis de requerimientos	\$1,280.00	64 horas	8 días?	lun26/03/12	mié04/04/12		
17 Técnico	\$1,280.00	64 horas	8 días?	jué05/04/12	lun16/04/12		
23 Documentación	\$320.00	32 horas	7 días?	jué05/04/12	vie13/04/12		
24 Elaboración de Manual de Análisis Funcional	\$160.00	16 horas	2 días?	jué05/04/12	vie06/04/12	16	Asistente
25 Elaboración de Manual de Especificaciones Técnicas	\$160.00	16 horas	2 días?	jué05/04/12	vie13/04/12	21	Asistente
26 Entrega de manuales de Análisis	\$0.00	0 horas	1 días?	lun16/04/12	lun16/04/12	25	
27 Desarrollo	\$3,700.00	252 horas	29 días?	mar17/04/12	vie29/05/12		
28 Autenticación	\$360.00	24 horas	3 días?	mar17/04/12	jué19/04/12		
29 Elaboración autenticación	\$120.00	8 horas	1 día?	mar17/04/12	mar17/04/12	26	Analista Programador I
30 Elaboración de Registro y Contraseña	\$240.00	16 horas	2 días?	mié18/04/12	jué19/04/12	29	Analista Programador I
31 Módulos de Seguridad	\$300.00	20 horas	2.5 días?	vie20/04/12	mar24/04/12		
32 Desarrollo del mantenimiento de bases de datos	\$60.00	4 horas	0.5 días?	vie20/04/12	vie20/04/12	30	Analista Programador I
33 Desarrollo del mantenimiento de BBDD	\$60.00	4 horas	0.5 días?	vie20/04/12	vie20/04/12	32	Analista Programador I
34 Desarrollo del mantenimiento de Objeto	\$60.00	4 horas	0.5 días?	lun23/04/12	lun23/04/12	33	Analista Programador I
35 Desarrollo de Asignación Rol	\$60.00	4 horas	0.5 días?	lun23/04/12	lun23/04/12	34	Analista Programador I
36 Desarrollo de Asignación de Objeto	\$60.00	4 horas	0.5 días?	mar24/04/12	mar24/04/12	35	Analista Programador I
37 Módulos de Adaptación	\$1,800.00	120 horas	15 días?	vie20/04/12	jué01/05/12		
38 Desarrollo de Ficha Personal	\$1,080.00	72 horas	9 días?	vie20/04/12	mié02/05/12	30	Analista Programador II
39 Desarrollo de Ficha Académica	\$360.00	24 horas	3 días?	jué03/05/12	lun07/05/12	38	Analista Programador II
40 Desarrollo de Ficha Laboral	\$360.00	24 horas	3 días?	mar08/05/12	jué10/05/12	39	Analista Programador II
41 Módulos de Administración	\$360.00	24 horas	3 días?	vie10/05/12	mar15/05/12		
42 Desarrollo del Mantenimiento de Entidades	\$360.00	24 horas	3 días?	vie11/05/12	mar15/05/12	40	Analista Programador II

Fuente: Elaboración propia.

Presentación principal de la aplicación PGR.

Figura 6



Fuente: Elaboración propia.

de Software, 2011. http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf

- [4] [Francisco M., +, 2011]: Francisco Maciá-Pérez, José Vicente Berná Martínez. Cloud Agile Manufacturing, 2011.
- [5] [Carla V., +, 2012]: Carla Villena, Juan Timoteo, Marines López, Roger Humberto, Jenny Pérez, Análisis de Oferta-Demanda de servicios de despliegue de aplicaciones en la nube (PaaS), 2012.
- [6] [Keene C., +, 2012]: What Is Platform as a Service (PaaS)? 2011. <http://www.keeneview.com/2009/03/what-is-platform-as-service-paas.html>
- [7] [Mridula V., +, 2011]: Agile Methodologies and PaaS: A Priceless Combination? 2011. <http://blog.bootstraptoday.com/2011/10/12/agile-methodologies-and-paas-a-priceless-combination/>.
- [8] [María E., 2011]: Diferencias entre metodologías ágiles y tradicionales. 2011. <http://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/>