



## **INGENIERÍA DE SOFTWARE Y COMPUTACIÓN EN LA NUBE. CONCEPTOS BÁSICOS QUE DEBEN SER ENSEÑADOS EN EL CURSO DE INGENIERÍA DE SOFTWARE I, SOFTWARE ENGINEERING AND CLOUD COMPUTING BASIC**

**Por:** Juan Jose Saldaña-Barrios , Yuraisma Moreno | Dept. of Software Engineering  
Technological University of Panama

**e-mail:** Juan.saldana@utp.ac.pa, yuraisma.moreno@utp.ac.pa

**Recibido:** Diciembre de 2016

**Aceptado:** Diciembre de 2016

### **Resumen**

La ingeniería de software es una disciplina relativamente nueva que provee procesos de ingeniería, arquitectura, patrones de diseño y controles para alcanzar calidad, seguridad, mantenimiento, interoperabilidad y escalabilidad en las aplicaciones. La computación en la nube es un término que constantemente escuchamos en el ámbito empresarial, tecnológico y educativo. Este paradigma brinda servicios como infraestructura, plataformas de desarrollo y aplicaciones que se ejecutan en los servidores en la nube, con tecnologías de virtualización y redes distribuidas, mas no así en nuestras computadoras. La intención de este trabajo es presentar conceptos básicos que necesitan ser impartidos en los cursos de ingeniería de software para incrementar el conocimiento de los estudiantes de este nuevo paradigma.

**Palabras claves:** computación en la nube, ingeniería de software, software como servicio, infraestructura como servicio, plataforma como servicio.

### **Abstract**

Software engineering is an engineering discipline that provides the requirements gathering process, software development process, architectural, design patterns and testing process models required to provide quality, security, maintainability, interoperability, scalability and availability of the applications. Cloud Computing and On the Cloud are terms that today we can see everywhere in the technology field. This new and emerging paradigm provides software and services that run over the Internet using distributed networks and virtualization technology. The intention of this paper is to present basic concepts that need to be taught in the software engineering courses in order to increase, in the student, the understanding of this new paradigm.



**Keywords:** cloud computing, software engineering, software as a Service, infrastructure as a Service, and platform as a service.

## Introducción

La ingeniería de software es una disciplina que es cada día más requerida por los profesionales en el área de desarrollo de software en nuestro país. En las universidades, la cátedra es dictada bajo constante revisión y actualización, pero en algunas ocasiones, el mercado de la tecnología avanza más rápido que los procesos de actualización de nuestros programas académicos. Esto hace necesario que los docentes que imparten la materia constantemente deban adicionar en sus clases, estos temas de actualización.

Uno de los principales auges dentro del contexto tecnológico es el paradigma de la computación en la nube, que ha emergido con la intención de proveer recursos, aplicaciones y servicios a los clientes o usuarios finales de una manera más fácil, económica, rápida y escalable. Los programas de computadora deben tener la capacidad de crecer con base en la demanda, interactuar con otros programas y, además, ser fáciles de mantener.

La arquitectura de software —una sub área de la ingeniería de software, intenta alcanzar estos beneficios promoviendo la abstracción, modularidad y el desarrollo basado en componentes. Los modelos de procesos de desarrollo de software prescriptivos y ágiles que encontramos en la mayor parte de la literatura no fueron

desarrollados bajo este nuevo paradigma. Aspectos como calidad, verificación, validación, portabilidad y niveles de abstracción pueden tener algunos cambios al desarrollar programas que se implementaran en una nube.

Los estudiantes que actualmente están tomando cursos de ingeniería de software necesitan comprender cómo este nuevo paradigma revoluciona el campo del desarrollo de software.

En este documento se explican algunos temas de alta relevancia en el campo de la computación en la nube, se revisarán algunos trabajos relacionados con el tema de computación en la nube e ingeniería de software y se analizarán ideas comunes, problemas y relaciones entre ambos.

El resto del documento está estructurado de la siguiente forma: Sección 2 un resumen de la ingeniería de software y sus principales pilares. Sección 3 presenta un resumen de computación en la nube. Sección 4 es una revisión de algunos trabajos relacionados. Sección 5 presenta un análisis de las relaciones entre los trabajos revisados. Finalmente en sección 6 se presentan las conclusiones y trabajos futuros.

## Ingeniería de software

La ingeniería de software es una disciplina de la ingeniería aplicada al proceso de desarrollo de software.



[3]. El corazón de la ingeniería de software radica en el ciclo de vida del desarrollo del software (SDLC), ya que éste establece los pasos a seguir en ese proceso de desarrollo, empezando por la obtención y análisis de los requerimientos hasta llegar al retiro de la aplicación del mercado.

Hay diferentes modelos de procesos de desarrollo de software basados en el tipo y contexto de aplicación. No es lo mismo desarrollar un formulario de contacto de una página web que desarrollar una aplicación que controlará un robot para cirugías o un sistema de control automático de un avión. La ingeniería de software por su naturaleza permite desarrollar productos no tangibles [3], por lo que muchos de los problemas no pueden detectarse a simple vista hasta que la aplicación sea implementada. Por esta razón, esta ingeniería incluye un conjunto de actividades sombrillas o paraguas que ayudan a mitigar el riesgo, teniendo un mayor control sobre el proyecto y así permitir alcanzar niveles de calidad superiores en el producto final.

Los procesos de desarrollo estándares o prescriptivos son altamente afectados por los cambios en los requerimientos que demandan modelos de procesos más prácticos. Los modelos de procesos ágiles utilizan menos documentación, se implementan en tiempos muchos menores, con menor cantidad de trabajo y totalmente funcionales.

Para ser ágiles, el equipo de desarrolladores deben basarse en simplicidad, constantes cambios y

reutilizar componentes existentes para agregar funcionales adicionales en el menor tiempo posible. Una pregunta sería: ¿Qué pasa cuando estos paquetes están localizados en la nube?

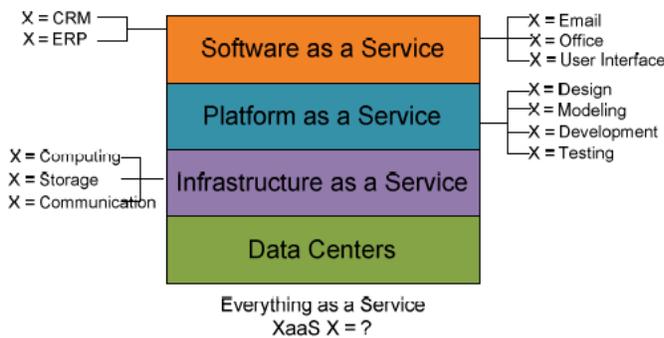
### Computación en la nube

La computación en la nube [1], puede clasificarse utilizando una visión jerárquica basándose en las capas de producto y servicio a saber: Infraestructura como servicio, Plataforma como servicio y Programa como servicio (tal y como se muestra en la figura 1)

- **Infraestructura como Servicio (IaaS):** A través de internet, el usuario o cliente tiene acceso a una serie de recursos de infraestructura tecnológica como lo son almacenamiento, capacidad de procesamiento y transferencia en la red. Cada uno de estos recursos pueden ser incrementados a medida que la demanda de los mismos aumenta. Como ejemplo de proveedores de IaaS podemos mencionar Amazon EC2, Open Stack, IBM y Rackspace.
- **Plataforma como Servicio (PaaS):** En esta capa, el usuario tiene acceso a plataformas de desarrollo. El usuario puede seleccionar el lenguaje de programación y el sistema de administración de base de datos; puede desarrollar y realizar las pruebas; también, implementar y monitorear la aplicación en ambiente de producción. Como ejemplos de proveedores tenemos JBoss OpenShift de Red

Hat, Heroku, Google App Engine y Windows Azure.

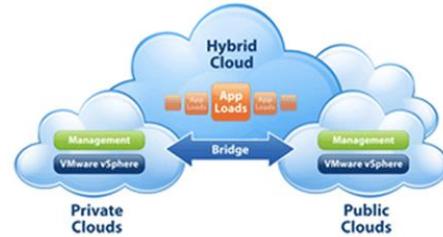
- Software como Servicio (SaaS): En esta capa los proveedores ofrecen aplicaciones que brindan diferentes tipos de servicios a los usuarios finales o a la lógica de negocio de compañías, como lo son Google, Docs, Adobe, Connect, Dropbox entre otras.



**Figura 1** : Capas de la Computación en la nube

Las nubes también pueden clasificarse con base en la accesibilidad de sus recursos [2] en:

- Nubes privadas: En este tipo de nube, todos los recursos se encuentran localizados dentro de una organización y los programas corren solo dentro de esta nube computacional.
- Nubes públicas: Los recursos computacionales son de acceso público y compartidos por todos los usuarios que los requieran
- Híbridos: En este tipo de nubes, algunos recursos son de acceso público mientras que otros de acceso limitado como se muestra en la figura 2. [7].



**Figura 2** : Nube Híbrida

### Trabajos relacionados

Luego de haber revisado algunos términos y definiciones importantes relacionados sobre ingeniería de software y computación en la nube, se presentará una revisión y comparación de trabajos publicados en este campo por otros autores.

El artículo *Retos para la Migración al Paradigma de la Computación en la Nube* [2] presenta un caso de estudio, con éxito, sobre la migración de un sistema legado de tamaño empresarial a la nube. Los autores trabajaron bajo un proceso centrado en modelos y un marco de trabajo denominado REMICS basado en principios de ingeniería de software. La metodología original fue adaptada para habilitarla al nuevo ambiente computacional.

El artículo *Computación en la Nube con Arquitectura Abierta* [4], muestra un modelo que integra diversos principios arquitectónicos que enlazan la computación en la nube con la Arquitectura Basada en Servicios (SOA). En SOA, existen tres elementos fundamentales: SOAP, UDDI y WSDL.

SOAP es el acrónimo de Simple Object Acces Protocol el cual es un protocolo usado para invocar servicios remotamente entre redes. UDDI acrónimo de Universal



Description, Discovery and Integration es un servicio que permite a los clientes descubrir y conectarse con otros servicios disponibles en la arquitectura y así brindar servicios más complejos y por último WSDL o Web Service Description Language el cual es un lenguaje común entre servicios el cual permite la comunicación y sincronización entre los mismos. La comunicación se basa principalmente en información relacionada al tipo de servicio ofrecido y los requerimientos para la sincronización entre los mismos.

La arquitectura de computación en la nube orientada a servicios, propuesta por Tsai [1], durante la Conferencia Internacional de Tecnologías de Información del año 2010, intenta incluir principios de la arquitectura SOA con la computación en la nube. En ese artículo el autor comenta importantes problemas que comúnmente presenta el paradigma de la computación en la nube. Por ejemplo, muchos sistemas en la nube se encuentran altamente acoplados, tienen poco soporte SLA y, las interfaces de los clientes, no son tan amigables. La arquitectura orientada a servicios (SOA), como paradigma para diseñar y desarrollar sistemas distribuidos presenta 4 capas:

1. Capa de Proveedor: Esta capa contiene los proveedores que brindan los servicios. Cada proveedor necesita tener sus propios servicios de red, computación y almacenamiento de manera independiente.
2. Mapeo de Ontologías en la Nube: Esta capa define la información relacionada con el procesamiento, almacenamiento y comunicación de la nube.
3. Corredor de Servicio: Esta capa define los agentes corredores en SOA que están a cargo de publicar la información perteneciente a la nube, calificar al proveedor y permitir una negociación y aprovisionamiento dinámico.
4. Capa SOA: Esta capa utiliza la tecnología de SOA para ofrecer flexibilidad al momento de elegir la nube que proporcionará los servicios.

Utilizando esta arquitectura en su prototipo, lograron proveer una excelente funcionalidad, interoperabilidad y calidad de servicio a los clientes que accedían a su infraestructura en busca de una nube computacional.

El artículo, *Roystone- A Cloud Computing System With Pluggable Component Architecture* [5] propone una arquitectura flexible utilizando componentes altamente integrables. El autor presenta tres componentes principales: El administrador de máquinas virtuales, quien se encarga de controlar las tareas que se ejecutan en las máquinas; el administrador de subsistemas, controla el hardware y el middleware y el coordinador, quien controla y sincroniza al administrador de máquinas virtuales con el administrador de subsistemas y graba todas las actividades entre los componentes. Dicha arquitectura también proporcionó muy buenos resultados a los proveedores de



servicio en la nube que la implementaron.

### Comparación y análisis

Luego de revisar algunos artículos relacionados en esta sección se analizan algunas diferencias entre ellos y algunas brechas aún no muy bien definidas entre la computación en la nube y la ingeniería de software

Los proveedores de servicios en la nube ofrecen tres capas principales que son: software, plataforma y arquitectura. Cuando se selecciona alguno de estos servicios, en muchas ocasiones el usuario se verá atado a trabajar con ese proveedor de allí en adelante, ya que entre proveedores existen incompatibilidades. Por ejemplo, en IaaS, si eliges el servicio de computación, es necesario también utilizar los recursos de almacenamiento y comunicación que el proveedor ofrece.

En la ingeniería de software, la arquitectura promueve la abstracción en los elementos del sistema y trabaja con módulos de alta cohesión y bajo acoplamiento. De esta forma, si se necesitara realizar algún cambio en busca, por ejemplo, de un mejor desempeño, calidad o algún nuevo requerimiento, el proceso de cambio será más rápido y alcanzable.

Varios de los artículos presentan marcos de trabajo que permiten utilizar principios de arquitectura de software dentro del campo de la computación en la nube. Roystone, en su caso, aplica los principios dentro de la capa de infraestructura como servicio. En este marco de

trabajo, el usuario es capaz de interactuar entre diferentes proveedores de servicios de computación, comunicación y almacenamiento, y de seleccionar el que más cubra sus necesidades. El siguiente ejemplo se basa en el mismo principio, donde abarca un nivel más amplio.

La arquitectura CCOA promueve la cohesión entre diferentes niveles de computación en la nube, iniciando desde la infraestructura, la plataforma, el aprovisionamiento de servicios y la lógica del negocio hasta la pantalla de interacción con el cliente final.

La intención principal es adaptar los principios de la Arquitectura orientada a servicios en todos los niveles de la computación en la nube. SOA [1] es una arquitectura que intenta proveer flexibilidad, habilitando al cliente el poder seleccionar, entre una variedad de proveedores de servicios, el apropiado.

Uno de los campos de la ingeniería de software de alta importancia son las pruebas de software, cuya finalidad es verificar y validar que el producto final cuente con la debida calidad. Para ello, existen modelos de procesos como el Modelo de Desarrollo basado en Pruebas en los cuales un conjunto de pruebas es primeramente desarrollado para luego construir el sistema con base en las pruebas a realizar.

No es fácil realizar pruebas en un software integrado y mucho más aun cuando el software se encuentra compuesto por componentes desarrollados y funcionales en la nube.



Proceso de Desarrollo: Los procesos de desarrollo tradicionales [6] fueron creados para construir o desarrollar los programas desde cero. Nuevos paradigmas como los procesos ágiles en donde los requerimientos cambian constantemente, adaptan estos procesos de desarrollo de una manera más rápida, implementando más grupos pequeños de códigos, pero todos funcionales, utilizando componentes ya desarrollados.

Obviamente, cuando se utilizan módulos desarrollados, el proceso de prueba necesita adaptarse a estos nuevos pasos. Ahora en computación en la nube, el proceso de desarrollo de software cambia nuevamente, porque el software es construido utilizando servicios de negocio, aplicaciones, plataformas y una infraestructura de diferentes proveedores de servicio.

El desarrollo de software debe incluir nuevos mecanismos para buscar y seleccionar los servicios apropiados para el desarrollo de software, así como dónde pueden encontrarse todos estos elementos y si estos elementos pueden ser calificados.

Primero, la arquitectura necesita ser diseñada y construida para poder proveer diferentes servicios en la nube, poder ser descubiertos y ser calificados. En resumen, poder

### Conclusiones y trabajo futuro

En este documento se han revisado los conceptos básicos que necesitan ser impartidos en las clases de ingeniería de software para preparar a los estudiantes a comprender la importancia de aplicar procesos y metodologías en el campo del desarrollo de software en la nube.

aplicar SOA en todos los elementos de la computación en la nube.

- La ingeniería de software también define la reingeniería de procesos. La reingeniería es la acción de aplicar ingeniería inversa para analizar y entender los requerimientos de un software existente, para posteriormente poder aplicar ingeniería para mejorar las características o funcionalidades del producto.

El artículo [2] presenta un caso de estudio sobre la migración de sistemas heredados a una plataforma en la nube. Este paradigma necesita nuevos cambios en los procesos de reingeniería de software.

- La calidad de software en la ingeniería de software no solo está relacionada con la cantidad de errores que la aplicación puede presentar. La calidad también incluye importantes factores como la escalabilidad, la disponibilidad y el mantenimiento de la aplicación. En la computación en la nube, estos aspectos son difíciles de evaluar porque ellos dependen de los proveedores de servicios y están fuera del alcance de los desarrolladores.

Como trabajo futuro, se aplicarán técnicas de aprendizaje activo en el proceso de enseñanza de estos cursos de manera que podamos mejorar el conocimiento adquirido por los estudiantes e incrementar la sinergia en el salón de clases.

## Referencias

- [1] Tsai, W.-T., Sun, X., & Balasooriya, J. (2010). Service-Oriented Cloud Computing Architecture. *2010 Seventh International Conference on Information Technology: New Generations*, 684–689. doi:10.1109/ITNG.2010.214
  - [2] Mohagheghi, P., & Sæther, T. (2011). Software Engineering Challenges for Migration to the Service Cloud Paradigm Ongoing Work in the REMICS Project. doi:10.1109/SERVICES.2011.26
  - [3] Guha, R., & Al-Dabass, D. (2010). Impact of Web 2.0 and Cloud Computing Platform on Software Engineering. *2010 International Symposium on Electronic System Design*, 213–218. doi:10.1109/ISED.2010.48
  - [4] Zhang, L., & Zhou, Q. (2009). 2009 IEEE International Conference on Web Services CCOA: Cloud Computing Open Architecture. doi:10.1109/ICWS.2009.144
  - [5] Yen, C., & Yang, J. (2011). Roystonea: A Cloud Computing System with Pluggable Component Architecture. doi:10.1109/ICPADS.2011.111
  - [6] Scacchi, W. (2001). Process Models in Software Engineering, (May), 1–24.
- Nolan, J. (n.d.). Understanding Public, Private and Hybrid Cloud Infrastructure. *Thursday, October 7th, 2010*. Retrieved January 10, 2012, from [http://www.eci.com/blog/images/Hybrid\\_Could](http://www.eci.com/blog/images/Hybrid_Could)

**Facultad de Economía**  
**UNACHI**

**Oferta Académica**

**Especializaciones y Maestrías:**

- Estadística Aplicada
- Tecnología y Sistemas de Información Empresarial
- Formulación, Evaluación y Administración de Proyectos
- Diseño de Aulas Virtuales
- Gestión de la Educación Virtual

UNACHI - Facultad de Economía  
[www.unachi.ac.pa/feconomia](http://www.unachi.ac.pa/feconomia)  
730-5300 ext 6601/6602

Universidad Autónoma de Chiriquí  
Institución Acreditada