

“Multi Alquiler”

“Multi – Tenancy”

Wayner Xavier Bustamante Granda

Universidad Internacional del Ecuador, Ecuador

Autor para correspondencia: wabustamantegr@internacional.edu.ec

Fecha de recepción: 5 de diciembre de 2016 - Fecha de aceptación: 10 de enero de 2017

Resumen: Este trabajo de investigación inicia con una introducción de Cloud Computing, para luego presentar el estudio de un principio de arquitectura llamado “Multi-Tenancy”; desde sus conceptos básicos, sus características, funcionamiento, niveles y comparativas. Finalmente se identificarán puntos claves para que los clientes puedan determinar si los proveedores de estos servicios ofrecen soluciones en la nube con Multi-Tenancy.

Palabras Clave: Tenancy, Multi-Tenancy, SaaS, Cloud Computing, API.

Abstract: This research begins with an introduction of Cloud Computing, and then presents the study of an architecture principle called "Multi-Tenancy"; from its basic concepts, characteristics, performance, levels and comparative. Finally key points will be identified so that customers can determine if the providers of these services offer solutions in the cloud with Multi-Tenancy.

Key Words: Tenancy, Multi-Tenancy, SaaS, Cloud Computing, API.

Introducción

Con el alto crecimiento de los volúmenes de información, se han originado nuevas tecnologías como Cloud Computing; cuyo principal atributo es la Elasticidad (incremento y decremento de recursos bajo demanda; y la compartición de recursos). Con Cloud Computing se puede empezar a trabajar en pocos días, representa ahorro de adquisición, mantenimiento, administración de recursos; además se resulta que son más adaptables, más seguras y más fiables que la mayoría de aplicaciones del mercado. Finalmente se basan en una Arquitectura Multiusuario. Una aplicación multiusuario es una aplicación que comparten todos los clientes, esto significa que las aplicaciones son Elásticas, de modo que sus aplicaciones obtienen mejoras (Seguridad y Rendimiento) automáticamente a esto también se le llama Multi-Tenancy.

El presente trabajo de investigación hace referencia:

En la sección II a conceptos básicos de Cloud Computing, tipos de servicios que ofrece Cloud Computing como: Infraestructura, Plataforma y Software. En la sección III se hace énfasis a un principio de arquitectura de software llamado “Multi-Tenancy, sus conceptos básicos, características y pilares en los que se basa. En la Sección IV se presentan los enfoques de Multi-Tenancy, conceptos, características, y también se hace un análisis comparativo entre los tres enfoques existentes. Finalmente en la sección V, se presenta un análisis de las características que los proveedores de Cloud Computing deben tener, para poder ofrecer soluciones informáticas con Multi-Tenancy.

Cloud Computing

En esta sección se realizará la introducción a este tipo de tecnología, en donde se abordarán conceptos referentes a Cloud Computing, a los servicios que se ofrecen en la nube, con la finalidad de ayudar a los lectores a agregar nuevos conocimiento en el mundo de las Tecnologías de la Información.

Existen varios términos usados para Cloud Computing entre ellos: computación en la nube, servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, es un paradigma que permite ofrecer servicios de computación a través de Internet. En la siguiente figura se puede apreciar una aproximación a Cloud Computing.

Figura. 1. Cloud Computing.



Fuente: (Janbask, 2015)

Cloud Computing dicho de otra forma es la informática que aprovecha la conexión a internet para proporcionar servicios a empresas en todo el mundo. Además ofrece las mismas oportunidad de aprovechar todo los potenciales tecnológicos, recursos informáticos (seguros, fácil de acceder, On Demand, software, etc.) tanto para empresas grandes o pequeñas.

Otra ventaja competitiva de Cloud Computing es que proporciona a las empresas mayor flexibilidad con sus datos y su información, a los que es posible acceder en cualquier momento y desde cualquier lugar. (Salesforce, 2016).

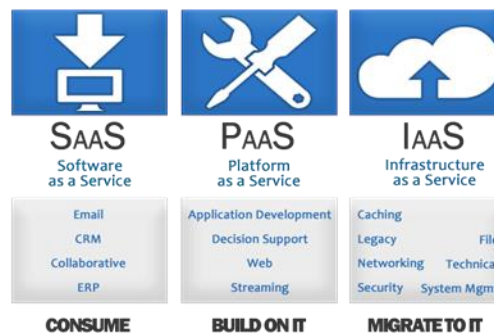
Las tecnologías en que suele apoyarse el Cloud Computing son virtualización, infraestructura, Multi-Tenant y escalabilidad.

“Los entornos multiusuario o Multi_tenancy corresponder a una distinción arquitectónica clave que diferencia el Cloud Computing de la simple tercerización o los antiguos modelos de proveedor de servicio de aplicaciones”. (Salesforce, 2016).

El presente trabajo de investigación se enfoca en Multi-tenancy como el atributo clave y común para los tipos de Cloud Computing (públicas y privadas); además Multi-tenancy se aplica a las tres modelos de servicios que se ofrecen, como son:

- IaaS (Infraestructura como Servicio).
- Paas (Plataforma como Servicio).
- SaaS (Software como servicio).

Figura 2. Modelos de Servicios de Cloud Computing.



Fuente: (NIST, 2009)

A continuación se describirán cada uno de los modelos de servicios que Cloud Computing ofrece:

A. IaaS

Infraestructura como servicio constituye el nivel más bajo de abstracción de los recursos de los clientes, el concepto se centra en la posibilidad de ofrecer a las compañías todos los recursos de hardware necesario para sus operaciones en la Nube, desde el espacio de almacenamiento hasta la capacidad de proceso de cómputo.

La principal ventaja es que las empresas pueden utilizar el modelo IaaS para construir nubes privadas, y/o nubes privadas virtuales con la finalidad de reducir gastos, aumentando performance de sus sistemas de información críticos. (Fernández E., 2013)

Es un Modelo de hardware compartido, que provee aislamiento, seguridad, procesos de migración eficiente para las bases de datos de los clientes, con un rendimiento aceptable.

B. PaaS

Plataforma como servicio provee un alto nivel de abstracción de los recursos de los clientes; es básicamente un ambiente de desarrollo en donde se pueden crear otras aplicaciones que hagan uso de las características del Cloud Computing. Por lo general utilizan un API de almacenamiento de datos¹, en donde los modelos Multi-Tenancy de tabla compartida y proceso compartido pueden cumplir los requerimientos para el funcionamiento de la plataforma. (Jiménez J. 2013)

C. SaaS

Software como servicio, posee el más alto nivel de abstracción en la que un cliente utiliza un servicio para realizar una tarea específica.

Se caracteriza por brindar On Demand una aplicación, corriendo en la denominada "Multi-Tenancy ", es decir, una sola instancia de la aplicación que se ejecuta en los servidores del proveedor y es capaz de operar y brindar servicios a múltiples compañías simultáneamente.

Las características de SaaS son:

- Los cambios son superficiales ya que los flujos de trabajo y modelos de datos son definidos por el proveedor del servicio.
- Las restricciones a los datos y a los procesos, y la imposibilidad de acceder a la capa de datos a través de un navegador, hacen que el proveedor sea el encargado de tener el control de cómo los clientes interactúan con el almacenamiento de datos.
- Modelo de tabla compartida.

La relación que existe entre Multi-Tenancy y Cloud Computing es que son temas de arquitectura de software que le permitirán a un Arquitecto de Software, poder tomar decisiones que no solo van a afectar a la tecnología que utiliza, sino que también puede ayudar a optimizar nuestros recursos (Elasticidad de Recursos ²) y suponer un ahorro de tiempo y dinero.

En la actualidad muchas empresas están usando soluciones en la nube por las ventajas que estas le ofrecen entre ellas la Elasticidad de Recursos, que contempla la posibilidad de crecer

¹ Un API de almacenamiento de datos, sirve como herramienta para la gestión integrada de recursos en la nube y aplicaciones

² La elasticidad es una propiedad de la arquitectura de nuestra plataforma.

o decrecer de forma automática según una serie de condiciones predefinidas y principalmente porque se caracteriza por: (Jiménez J. 2013)

- Funcionar en ambos sentidos: ampliación o reducción de los recursos de la plataforma.
- Es instantánea y automática: No requiere nuestra intervención.

¿Qué es un Tenancy?

Es un grupo de usuarios que comparten el uso de una sola aplicación ya sea desde el acceso a los datos, la configuración, la gestión de usuario, las propiedades funcionales y no funcionales. (Krebs R. Momm1 C. Kounev S. 2012)

Es una arquitectura en la que una sola instancia de una aplicación de software y la infraestructura de apoyo sirve a un cliente. En el modelo SaaS, al cliente se lo llama inquilino.

¿Qué es un Multi - Tenancy?

Se refiere a un principio de arquitectura de software, donde un sistema comparte varios clientes, es decir que trabaja sobre una misma instancia de software, la cual se ejecuta desde un servidor, a este tipo de arquitectura se la llama arquitectura multipropietario o multiusuario.

Otra forma de conceptualizar Multi– Tenancy; se refiere a la habilidad para ubicar varios clientes dentro de una única instancia del software que se ejecuta en un servidor.(Plan Avanza. 2011) En un entorno Multi-Tenant, todos los clientes y sus usuarios consumen el servicio desde la misma plataforma tecnológica, desde el modelo de datos, servidores y las capas de base de datos.

Con una arquitectura Multi-Tenancy, una aplicación de software está diseñada para particionar sus datos y la configuración de manera que cada cliente trabaje con una instancia de la aplicación virtual personalizada.

Multi-Tenancy (Multiusuario) hace referencia al modo de operación del software en varias instancias independientes de uno o múltiples aplicaciones las cuales están operando en un entorno compartido. (Plan Avanza. 2011) Los casos (Tenancy) están aislados lógicamente, pero físicamente integradas.

En la figura 3 se puede apreciar la diferencia entre Tenancy y Multi-Tenancy.

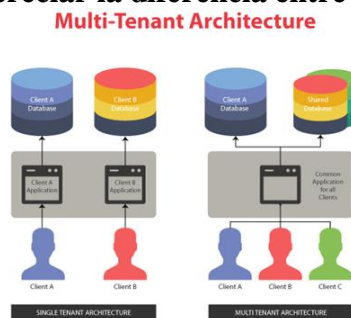


Figura 3. Arquitectura Multi-Tenancy. Fuente: (Goyal S. 2016)

Las arquitecturas Multi-Tenant (multipropietario) son cada vez más utilizadas entre los proveedores de SaaS (Software as a Service), considerándose fundamentales para Cloud Computing. Para lograr una distribución de aplicaciones Multi-Tenant hay que centrarse principalmente en tres aspectos fundamentales: (Carvajal S. 2011)

1. **El mecanismo para la distribución de recursos:** que tiene como finalidad reducir costos de hardware, de software y de la administración de cada Tenant.
2. **El mecanismo de aislamiento de seguridad:** que tiene como finalidad evitar posibles accesos no válidos, conflictos e interferencias entre los Tenants.
3. **El mecanismo de personalización:** que tiene como finalidad dar soporte al modelo de UI (Interfaz de Usuario) por Tenant, de control de acceso, de proceso y de datos a través de los enfoques de configuración.

Así mismo una Arquitectura Multi-Tenancy se basa en 4 pilares: (Carvajal S. 2011)

1. **Disponibilidad:** Al ser una arquitectura compartida, se debe proporcionar redundancia integrada de forma que los recursos necesarios de informática, red y almacenamiento sigan estando disponibles en cualquier momento; ya que un posible fallo podría afectar a muchos clientes a los que se ofrece estos servicios.
2. **Separación segura:** Al ser una arquitectura compartida, se debe tomar en cuenta el aislamiento de los usuarios y la seguridad en el almacenamiento de la información.
3. **Garantía de servicios:** De igual manera se debe aislar y garantizar el rendimiento del equipo, de la red y del almacenamiento en cualquier momento ya sea durante el funcionamiento normal, cuando se produzcan fallos o cuando determinados servicios generan problemas.
4. **Gestión:** La capacidad de administrar, gestionar y supervisar rápidamente todos los recursos de forma íntegra, brindándoles a los clientes poder acceder a la gestión de sus servicios de manera inmediata y transparente.

Enfoques De Multi-Tenancy

Compartir recursos en diferentes niveles de abstracción, tomando en cuenta el aislamiento de los usuarios y seguridad en el almacenamiento de la información, dan como resultado los diferentes modelos Multi-Tenancy o Multi-Cliente. (Elmore A. Das S. Agrawal D. and Abbadi A. 2011)

A continuación se dará a conocer los tres enfoques de la gestión de datos para Multi-Tenancy, tal como se muestra en la figura 4.

Figura 4. Niveles de Multi-Tenancy



Fuente: (F. Chong, G. Carraro, and R. Wolter. 2006)

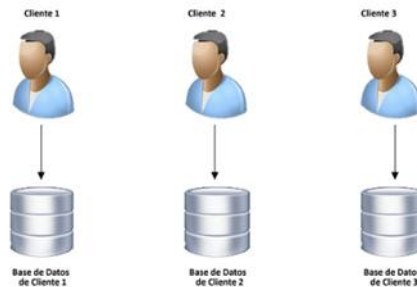
1. Base de Datos por separado:

En este enfoque los datos de cada cliente se almacenan en bases de datos separadas. Las bases de datos pueden estar en el mismo servidor o pueden ser divididas a través de servidores de bases de datos múltiples. Este enfoque proporciona aislamiento máximo de los datos de los clientes, como lo muestra en la Figura 5. (Castillo C. 2012)

Desde el punto de vista del Hardware y de los recursos computacionales, este esquema es conocido como Hardware Compartido, en donde cada cliente tiene su propio proceso independiente de la base de datos pero se comparte el hardware de la máquina (recursos como RAM, Memoria, CPU) en donde se encuentran alojados los clientes.

La virtualización es la forma de utilizar recursos de una misma máquina, donde se crean máquinas virtuales (VM) que sirven para aislar a los clientes, cada uno de estos cuentan con una base de datos instalada exclusivamente para la máquina virtual. En este modelo no se comparten los recursos de base de datos, únicamente los recursos físicos (RAM, Memoria, CPU) del servidor donde se alojan las VM. (Elmore A. Das S. Agrawal D. and Abbadí A. 2011)

Figura 5. BD diferentes para cada cliente.



Fuente: (Castillo C. 2012)

2. Base de Datos Compartidas, Esquemas Separados.

Este enfoque consiste en que cada cliente tiene su propio conjunto de tablas que se agrupan en un esquema creado específicamente para el cliente. Con este enfoque puede tener una sola base de datos con un esquema para cada cliente, como se indica en la Figura 6.

Igualmente desde el punto de vista del Hardware, a este enfoque se lo conoce como Proceso Compartido, porque en este modelo los usuarios utilizan un esquema de tablas individual, pero estos comparten el mismo proceso de base de datos, con lo cual permite

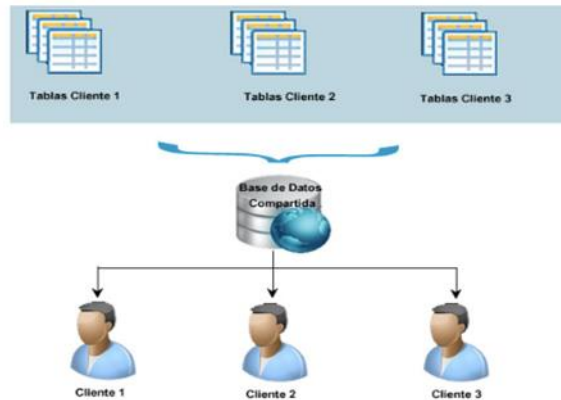
aprovechar de manera óptima la utilización de recursos del servidor donde se encuentran alojados los Tenants.

La utilización de un solo motor de base de datos permite a los administradores realizar tareas de migración, respaldo o recuperación de la información. (Jacobs D. Aulbach S. and MÃnchen T. 2007)

Otra ventaja de este modelo es la optimización en el uso del pool de memoria, el cual permite almacenar un mayor número de esquemas de base de datos (por cliente). Por ejemplo: para manejar 10.000 instancias, se requiere de 79 MB a 2061 MB, dependiendo del motor de base de datos. (Jacobs D. Aulbach S. and MÃnchen T. 2007)

El modelo de proceso compartido se ajusta al modo de compartición, esto quiere decir que el nivel de asilamiento de los clientes se da a nivel de Base de Datos y del Esquema.

Figura 6. Una sola BD con diferentes tablas.



Fuente: (Castillo C. 2012)

3. Base de Datos Compartidas, Esquemas Compartidas.

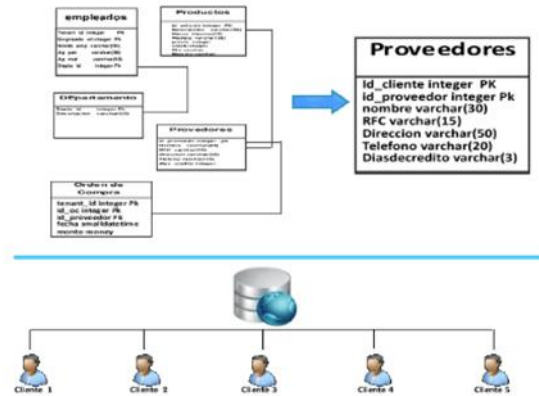
En este enfoque sólo tenemos una única base de datos y un esquema único. Cada Tabla debe de referirse a un id de cliente.

De los tres enfoques que se explican el enfoque del esquema compartido, tiene sus ventajas y desventajas las cuales son:

- Este enfoque tiene los más bajos costos de hardware y de copia de seguridad, ya que le permite servir al mayor número de clientes por base de datos del servidor.
- Para que este modelo funcione, los clientes deben compartir el mismo de tipo de información.
- Un punto de fallo está en que para hacer la migración de los clientes, esta tarea es realizada directamente a la base de datos funcional.

- El tema de seguridad es otro punto de fallo, debido a que los privilegios de acceso pueden ser asignados a diferentes filas en una misma tabla; para garantizar que los clientes no puedan acceder a los datos de otros clientes, incluso en caso de errores inesperados o ataques.

Figura 7. Una sola BD con tabla compartida.



Fuente: (Castillo C. 2012)

Finalmente una vez ingresado al mundo de los conceptos de Cloud Computing, en este trabajo de investigación se realiza un análisis de cómo impacta Multi-Tenancy en los tres niveles gestión de datos. En la tabla 1, se muestra este análisis.

Esquema	BD por Separado	BD compartida, Esquema Separados	BD compartida, Esquema Compartidos
Criterios			
Tiempo de desarrollo	+ Tiempo de Desarrollo.	- Tiempo de Desarrollo	-- Tiempo de Desarrollo
Costos de Hardware	+ Cara, cada cliente requiere su propio Base de Datos, su propio H/W, Mantenimiento, etc.	+ Barata porque se usa una sola Base de Datos y se crean esquemas diferentes para cada cliente.	+ Mucho más barata porque se usa una sola Base de Datos y un esquema único para todos los clientes.
Aplicación y rendimiento de la Base de Datos.	El rendimiento del cliente no se ve afectado por cualquier otro cliente.	El rendimiento de un cliente puede verse afectado por las actividades de los clientes que comparten el servidor.	El rendimiento de un cliente puede verse afectada por las actividades de los otros clientes. El rendimiento de las consultas tendrá que ser examinados cuidadosamente para garantizar los índices adecuados existe.

Seguridad	Debido al total aislamiento de otros clientes, los datos de cada cliente pueden ser muy bien resguardados. Impidiendo que cualquier cliente de forma accidental o malintencionada quiera acceder a los datos a otros Clientes.	El DBMS ³ debe asegurar que su estructura de permisos sea tal que cada cliente sólo está disponible para usuarios autorizados. Además, la aplicación debe seleccionar o restringir el esquema que se accede por un usuario u por otro.	La aplicación debe utilizar código de consultas especialmente para seleccionar o restringir los datos basados en el cliente. Las pruebas sólidas deben utilizarse para garantizar que un usuario no es capaz de ver los datos de los otros clientes. Encriptación de datos únicos para cada cliente no es posible.
Requisitos Personalización	Cada cliente tiene su propia base de datos, por lo que es más fácil de personalizar.	Cada cliente tiene su propio esquema, así que es fácil de personalizar para las diferentes necesidades de cada cliente.	Todos los clientes comparten el esquema, por lo que es mucho más difícil permitir la individualización.
Número de Clientes	Este enfoque hace que sea mucho más difícil de manejar un gran número de clientes.	Este enfoque es capaz de manejar más clientes que el anterior modelo, pero todavía requieren una cierta cantidad de la administración. La migración de los clientes a una base de datos independiente puede ser un desafío en función de las utilidades que se ofrecen por el DBMS.	Este modelo es capaz de manejar más clientes que los modelos anteriores. La migración de los clientes requieren un mejor rendimiento o la capacidad puede ser un reto, ya que los datos tendrán que ser extraídos de cada caso en operaciones separadas.

Fuente: (Castillo C. 2012)

Proveedores De Cloud Computing Con Multi-Tenancy

En el presente trabajo se indicarán los puntos clave para la selección de aplicaciones de negocio SaaS con Multi-Tenancy. Para ello la aplicación SaaS debe cumplir con:

- Poder ser compartida por diferentes clientes.
- Ser capaz de adaptarse y evolucionar con los diferentes requerimientos de cada cliente.
- Y al mismo tiempo ser viable, técnica y económicamente, para el proveedor.

La pregunta que todo empresario debe hacerse ¿Cómo podemos verificar con nuestro proveedor que su aplicación de gestión es Multi-Tenancy?:

³ Significa “Data Base Management System”, que significa “Sistema de Gestión de Base de Datos”.

1. Todos los clientes deberían correr la misma versión del código, sin “customizaciones” de código.
2. Los clientes no deberían preocuparse de tener que actualizar el software, para ellos debería ser transparente, en un extremo ni enterarse.
3. No debería haber versiones en el sentido tradicional sino frecuentes mejoras incrementales (un ejemplo serían las aplicaciones de Google).
4. De alguna forma la arquitectura True SaaS debe poder distinguir entre cada cliente.

A continuación se detalla proveedores que ofrecen SaaS (Software como Servicio) con Multi-Tenancy:

- SalesForce: CRM: Sales.Cloud.
- Correo: AppExchange.
- Conversación: Chatter.

Conclusiones

El modelo de Instancia Compartida (BD Compartida y Esquema Compartido) es el más adecuado para tener aplicaciones con un nivel equilibrado de consumo de recursos y aislamiento de clientes, ya que a nivel de mantenimiento, migración, es más flexible a tiempos de inactividad e interrupción del servicio.

La elección del proveedor adecuado depende estrictamente del tipo de aplicación que el cliente desea implementar, ya que cada uno de estos ofrece un servicio orientado a los diferentes paradigmas existentes en las tecnologías Cloud Computing.

El nivel de abstracción más costoso y que más tiempo demora en implementarlo, pero que es más seguro, tiene mayor rendimiento y es más fácil de personalizar, es el Enfoque Base de Datos Separados.

A medida que se comparten la información en las base de datos, se abaratan costos de implementación pero se ponen en riesgo la seguridad de los datos y la personalización de los mismo, además influye en el rendimiento por el hecho de estar compartiendo recursos con otros usuarios.

Cuando evalúes una aplicación SaaS, la pregunta correcta no es si es o no Multi-Tenancy, sino en qué grado lo es en relación al conjunto de aplicaciones evaluadas.

Bibliografía

- Alvigini A., (2014), Una mirada legal al Cloud Computing, Recuperado el 15 de Septiembre, de <http://www.encuentrojuridico.com/2014/03/una-mirada-legal-al-cloud-computing.html>
- Carvajal S, (2011), TFC-WEB Semántica memoria final, Recuperado el 20 de Agosto de 2016, de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/11021/8/scarvajalTFC0112.pdf>
- Castillo C., (2012), Arquitectura Multi – Tenancy, Recuperado el 17 de Junio de 2016, de http://www.gravitar.biz/index.php/tecnologia_negocios/arquitecturas-multi-tenant/

- Cloud Computing, (2012), Que es “Computación en la Nube”, Recuperado el 06 de Junio de 2016, de <https://computacionube.wordpress.com>
- Elmore A. Das S. Agrawal D. and Abbadi A., (2011), Towards an elastic and Autonomic multitenant data base, Recuperado el 27 de Agosto de 2016, de <http://research.microsoft.com/en-us/um/people/srikanth/netdb11/netdb11papers/netdb11-final8.pdf>
- F. Chong, G. Carraro, and R. Wolter, (2006), Multi – Tenancy Data Architecture, Recuperado el 28 de Junio de 2016, de <http://msdn.microsoft.com/en-us/library/aa479086.aspx>
- Fernández E., (2013), Multitenancy al Desnudo, Recuperado el 18 de Julio de 2016, de <http://blog.meta4.es/?p=1223>
- Goyal S, (2016), 5 Reasons Why You Should Choose Multi-Tenant Architecture for Your SaaS Application, Recuperado el 15 de Agosto de 2016, de <https://www.netsolutionsindia.com/blog/5-reasons-why-you-should-choose-multi-tenant-architecture-for-your-saas-application/>
- Jacobs D. Aulbach S. and MÄnchen T., (2007) “Ruminations on Multi-Tenant databases”, Recuperado el 26 de Julio de 2016, de <http://www.db.in.tum.de/research/publications/conferences/BTW2007-mtd.pdf>
- Janbask, (2015), Cloud Based CRM Solutions, Recuperado el 20 de Mayo de 2016, de <http://www.janbask.com/cloud-computing/>
- Jimenez J., (2013), Bases de Datos Multi-Tenancy para proveer Software como Servicio.
- Krebs R. Momm1 C. Kounev S, (2012), Architectural Concerns in Multi-Tenant SaaS Applications, Recuperado el 17 de Julio de 2016, de <http://sdqweb.ipd.kit.edu/publications/pdfs/KrMoKo2012-closer-multitenant-sass.pdf>
- Plan Avanza, (2011), Multi Tenancy, Recuperado el 20 de Julio de 2016, de <http://paas.enncloud.com/multi-tenancy.htm>
- R. Blaisdell, (2012), Multi-tenancy in the Cloud: Understading its benefits, Recuperado el 07 de Septiembre de 2016, de <http://www.enterprisecioforum.com/en/blogs/rickblaisdell/multi-tenancy-cloud-understanding-its-be>
- Salesforce, (2013), Multitenant Architecture, Recuperado el 02 de Agosto de 2016, de http://www.developerforce.com/media/ForcedotcomBookLibrary/Force.com_Multitenancy_WP_101508.pdf
- Salesforce, (2016), ¿Qué es Cloud Computing?, Recuperado el 13 de Junio de 2016, de <http://www.salesforce.com/mx/cloud-computing/>