

**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**OPTIMIZACION DE CLOUD COMPUTING POR MULTI-TENANT
PARA EMPRESAS PROVEEDORAS DE TELECOMUNICACIONES**

AUTOR:

Pablo Andrés Jaramillo Abril

Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Bastidas Cabrera, Tomas Gaspar

Guayaquil, Ecuador

3 de Marzo del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. **Jaramillo Abril, Pablo Andrés** como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

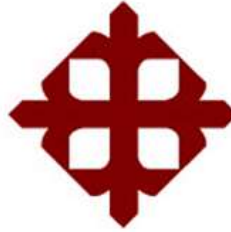
TUTOR

M. Sc. Bastidas Cabrera, Tomás Gaspar

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, 3 de marzo del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Jaramillo Abril, Pablo Andrés**

DECLARÓ QUE:

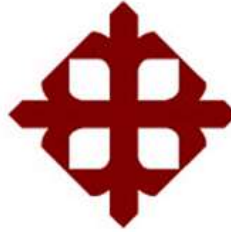
El trabajo de titulación: **“Optimización de Cloud Computing por multi-tenant para empresas proveedoras de telecomunicaciones”**, previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 3 de marzo del 2020

EL AUTOR

JARAMILLO ABRIL, PABLO ANDRES



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Jaramillo Abril, Pablo Andrés

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Optimización de Cloud Computing por multi-tenant para empresas proveedoras de telecomunicaciones”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 3 de marzo del 2020

EL AUTOR

JARAMILLO ABRIL, PABLO ANDRES

REPORTE DE URKUND



The screenshot shows the URKUND interface with the following details:

- Documento:** [TESIS PABLO JARAMILLO ABRIL 6 de febrero.docx](#) (D63662114)
- Presentado:** 2020-02-09 15:39 (-05:00)
- Presentado por:** fernandopm23@hotmail.com
- Recibido:** edwin.palacios.ucsg@analysis.orkund.com
- Mensaje:** Revisión TT Jaramillo Pablo [Mostrar el mensaje completo](#)

A summary box indicates: 3% de estas 37 páginas, se componen de texto presente en 10 fuentes.

The interface includes a toolbar at the bottom with icons for search, zoom, and navigation.

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA: Optimización De cloud computing por multi-tenant para empresas proveedoras de telecomunicaciones

AUTOR: Pablo Andrés Jaramillo Abril

Trabajo de Titulación previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES

TUTOR: M. Sc. Bastidas Cabrera, Tomas Gaspar

Guayaquil, Ecuador

6 de Noviembre del 2019

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Ing. Tomas Bastidas Cabrera MSc.

DEDICATORIA

Esta tesis va principalmente dedicada a Dios que me ha permitido llegar a esta etapa de mi vida profesional. A mis padres que siempre han estado junto a mi para apoyarme y darme fuerzas para seguir adelante, siendo pilares fundamentales en mí vida. Finalmente agradezco a la Universidad Católica Santiago de Guayaquil por darme la oportunidad de formarme con los conocimientos necesarios para desarrollarme en mi vida profesional a un gran nivel.

EL AUTOR

JARAMILLO ABRIL PABLO ANDRES

AGRADECIMIENTO

Agradezco principalmente a Dios por darme fuerzas y bendiciones para poder llegar a esta meta que tanto anhelaba conseguir en mi vida. Agradezco a mis padres que siempre estuvieron para apoyarme en todo lo que necesitaba, brindándome sus consejos y dándome la motivación que necesitaba para seguir este largo camino. Agradezco a mi tutor que gracias a su guía y conocimientos se ha cumplido con la meta.

EL AUTOR

JARAMILLO ABRIL, PABLO ANDRES



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. _____

M. Sc. ROMERO PAZ, MANUEL DE JESÚS
DECANO

f. _____

M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO
COORDINADOR DE ÁREA

f. _____

M. Sc. HERAS SANCHEZ, MIGUEL ARMANDO
OPONENTE

Índice General

Capítulo 1: Descripción General	2
1.1. Introducción	2
1.2. Antecedentes.....	3
1.3. Definición del Problema.....	4
1.4. Justificación del Problema	4
1.5. Objetivos del Problema de Investigación.....	4
1.5.1. Objetivo General.....	4
1.5.2. Objetivos Específicos	4
1.6. Hipótesis.....	5
1.7. Metodología de Investigación	5
Capítulo 2: Fundamentación Teórica	6
2.1. ¿Qué es Cloud Computing?	6
2.2. Características	6
2.2.1. Auto servicio bajo demanda (On-Demand Self Service).....	7
2.2.2. Amplio acceso a la red.....	7
2.2.3. Puesta en común de recursos.....	7
2.2.4. Rápida elasticidad	7
2.2.5. Servicio medido.....	8
2.3. Modelos de servicios	8
2.3.1. Infraestructura como servicio (IaaS).....	8
2.3.2. Plataforma como servicio (PaaS)	8
2.3.3. Software como servicio (SaaS).....	9
2.4. Modelos de implementación.....	9
2.4.1. Nube privada.....	9
2.4.2. Nube comunitaria	10
2.4.3. Nube pública.....	10
2.4.4. Nube Híbrida	10

2.5.	Software como servicio (SaaS)	11
2.6.	Potencial del Cloud Computing	12
2.7.	Empresas de telecomunicaciones como proveedores de servicios de Cloud Computing	12
2.8.	Empresas de telecomunicaciones como adoptantes de los servicios de Cloud Computing	13
2.9.	Ventajas del Cloud Computing.....	13
2.10.	Proveedores de Cloud Computing y sus servicios	14
2.11.	UIT sobre los servicios de Cloud Computing	17
2.12.	Roles de las telecomunicaciones en Cloud Computing.....	18
2.13.	Impacto del Cloud Computing en las empresas	19
2.14.	Adopción del Cloud Computing en el mundo	20
2.15.	Modelo Multi-tenant	21
2.16.	Tipos de Multitenencia.....	23
2.16.1.	Nada compartido (Nothing Shared).....	23
2.16.2.	Hardware compartido (Shared hardware).....	24
2.16.3.	Sistema Operativo compartido (Shared OS).....	24
2.17.	Base de datos compartida (Shared Database).....	24
2.17.1.	Base de datos independientes e instancias independientes. ..	24
2.17.2.	Tablas independientes e instancias de datos compartida.	26
2.17.3.	Tablas compartidas e instancias de datos compartidas.	27
2.18.	Multi-tenant vs Single-tenant.....	28
2.19.	Gestión de los recursos en la nube mediante algoritmos predictivos ..	29
2.19.1.	Regresión lineal.....	30
2.19.2.	Filtro de Kalman	30
2.19.3.	Cadenas de Markov	31
2.20.	Variabilidad y Multi-tenant.....	31
2.21.	Arquitecturas y enfoques de desarrollo en la nube.....	32
2.21.1.	Arquitecturas orientadas a servicios (SOA)	32
2.21.2.	Model View Controller (MVC).....	33
2.21.3.	Model Driven Architecture (MDA).....	34
2.22.	Transformaciones de modelos.....	35
CAPÍTULO 3: RESULTADOS OBTENIDOS		37

3.1.	Introducción	37
3.2.	Preguntas de Investigación	37
3.3.	Proceso de búsqueda	38
3.4.	Criterios de búsqueda	38
3.5.	Extracción de datos	39
3.6.	Artículos analizados	40
3.7.	Ventajas principales que se obtienen en la adopción de un modelo multi-tenant	41
3.8.	Problemas que se han encontrado en el modelo multi-tenant	43
3.9.	Herramientas y técnicas aplicables que dan soporte a un estable modelo multi-tenant.....	48
3.10.	Tipos de enfoque de optimización	49
3.11.	Soporte para la implementación del modelo multi-tenant	54
	3.11.1. Estilo Arquitectónico.....	54
	3.11.2. Capa de implementación.....	57
	3.11.3. Herramientas para la implementación	60
3.12.	Análisis de los resultados	62
	3.12.1. Estado actual de las investigaciones del modelo multi-tenant	62
	3.12.2. Tendencias a futuro en el modelo multi-tenant.....	64
	CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.	67
4.1.	Conclusiones.....	67
4.2.	Recomendaciones.....	68

Índice de Figuras

Figura 2.1: Tipos de Cloud Computing	10
Figura 2.2: Usuarios y proveedores de Cloud Computing.....	11
Figura 2.3: ¿Que es Salesforce?	14
Figura 2.4: El mundo de Amazon Web Services.....	15
Figura 2.5: Ejemplos de Backups.	15
Figura 2.6: Bases de datos IBM.....	16
Figura 2.7: Mundo Multicloud de Datrium.	16
Figura 2.8: Tipos de Cloud Computing y sus ramas.....	17
Figura 2.9: Evolución de Cloud Computing en principales países.....	21
Figura 2.10: Multi-tenant vs Single-tenant.....	22
Figura 2.11: Estructura del modelo multi-tenant.....	23
Figura 2.12: Modelo single-tenant, cada uno usa sus propios recursos.....	23
Figura 2.13: Este enfoque usa una base de datos para cada cliente.	25
Figura 2.14: Clientes con base común y conjunto de tablas separadas.	26
Figura 2.15: Estrategias de bases de datos.....	28
Figura 2.16: OMG's Model Driven Architecture.....	34
Figura 3. 1: Ventajas del modelo multi-tenant.....	43
Figura 3. 2: Problemas presentes en el modelo multi-tenant	47
Figura 3. 3: Organización de datos independientes.	51
Figura 3. 4: Clasificación de estilos arquitectónicos.....	57
Figura 3. 5: Capa de implementación según enfoque.....	58
Figura 3. 6: Atributos de calidad por capa de implementación.....	59
Figura 3. 7: Enfoque propuesto.	61

Índice de Tablas

Tabla 3. 1: Fuentes de búsqueda	38
Tabla 3. 2: Criterios de inclusión y exclusión	38
Tabla 3. 3: Caracterización de los artículos	39
Tabla 3. 4: Artículos revisados.....	40
Tabla 3. 5: Motivaciones presentadas en los artículos.....	42
Tabla 3. 6: Clasificación de los artículos de acuerdo al problema	48
Tabla 3. 7: Clasificación de artículos por soporte que brindan.....	49
Tabla 3. 8: Esquema de almacenamiento XML.....	52

Resumen

El presente trabajo de titulación realiza un estudio sobre como el Cloud Computing ha generado una gran reducción para las empresas que delegan el aprovisionamiento de diversos recursos computacionales, como servidores, plataformas, redes, aplicaciones de software y servicios. Este modelo que provee servicios y aplicaciones es conocida como Software as a Service (SaaS), que juntos a sus semejantes, Infrastructure as a Service (IaaS) y Plataforma as a Service (PaaS), forman el mundo de oportunidades que ofrece la computación en la nube. Este trabajo también analiza que a medida que ha crecido el modelo de servicios por la nube, se han creado nuevos tipos de arquitecturas para llevar el negocio a una economía de escala, haciendo que los proveedores de servicios de computación en la nube ofrezcan una misma instancia de software a varios clientes, de manera que estos comparten bases de datos, sistemas operativos y hardware en algunos casos. Se hace especial énfasis en la arquitectura multi-tenant, en la cual se analiza el beneficio que ofrece tanto a los proveedores como a los clientes el gran ahorro de recursos que se busca con el servicio de la nube, junto a las falencias presentes en la arquitectura, las mismas que están relacionadas con la seguridad en la compartición de recursos, la disponibilidad del servicio y la escalabilidad. En el último capítulo se busca subsanar estos problemas, de manera que se ofrezca un correcto funcionamiento y gran fiabilidad del servicio para los clientes de la computación en la nube.

Palabras claves: IAAS, SAAS, PAAS, HARDWARE, SOFTWARE, ESCALABILIDAD.

Abstract

This thesis carries out a study on how Cloud Computing has generated a great reduction for companies that delegate the provision of various computing resources, such as servers, platforms, networks, software applications and services. This model that provides services and applications is known as Software as a Service (SaaS), which together with their peers, Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), form the world of opportunities offered by computing in Cloud. This paper also analyzes that as the cloud service model has grown, new types of architectures have been created to bring the business to a scale economy, making cloud computing service providers offer the same instance software to several clients, so that they share databases, operating systems and hardware in some cases. Special emphasis is placed on the multi-tenant architecture, in which the benefit offered to both providers and customers is analyzed by the great saving of resources sought with the cloud service, together with the shortcomings present in the architecture, which are related to security in the sharing of resources, service availability and scalability. The last chapter seeks to correct these problems, so as to offer a correct operation and high reliability of the service for customers of cloud computing.

Key Words: IAAS, SAAS, PAAS, HARDWARE, SOFTWARE, SCALABILITY.

Capítulo 1: Descripción General

1.1. Introducción

La computación en la nube ha ido ganando acogida entre los usuarios y empresas porque ha permitido reducir costes operativos a las empresas que delegan a proveedores el aprovisionamiento de un conjunto de varios recursos computacionales como lo son las redes, plataformas, almacenamiento, servidores y hasta aplicaciones de software. Este modelo de negocio es considerado como innovador ya que genera un valor tanto para la empresa adoptante como para la proveedora. Este tipo de computación permite a las empresas dar prioridad a sus actividades empresariales delegando las plataformas de desarrollo, infraestructura y ciertos servicios.

La computación en la nube está compuesta por tres modelos de servicios diferentes: Plataforma (Platform as a Service, PaaS), Infraestructura (Infrastructure as a Service, IaaS) y aplicaciones de software (Software as a Service, SaaS). El modelo de SaaS (Software as a Service) en general cumple una propiedad fundamental, la cual es la multitenencia. Las aplicaciones que son orientadas a servicio tradicionales, manejan un modelo por tenant, que se llama al conjunto de usuarios que comparten un grupo de necesidades y requisitos, como por ejemplo una organización. En sus inicios los proveedores adoptaron el modelo single-tenant, lo que era muy efectivo a la hora de brindar sus servicios a diferentes clientes, pero había poco aprovechamiento de los recursos, hasta que apareció el modelo multi-tenant, que tiende a ser adoptado hoy en día por los proveedores de SaaS, ya que mediante este modelo en todas estas instancias se comparten o se podrían compartir una misma estructura, incluyendo desde base de datos, sistema operativo, hasta los recursos de hardware. En este trabajo se centra en el modelo multi-tenant aplicado para las empresas proveedoras que ofrezcan servicios de Cloud Computing.

Las empresas que adoptan el servicio de Cloud Computing se han visto beneficiadas ya que optimizan su infraestructura y organización de manera que, desde un levantamiento de la empresa, no se planifica gastar en

construcción y adecuamiento de cuartos con bases de datos, computadoras potentes cuando con ayuda de los servicios en la nube se puede tener dos computadoras trabajando en una sola con programas por medio de la nube.

Todos estos servicios brindados por parte de los proveedores requieren de un óptimo funcionamiento de su infraestructura, donde se ha aplicado un modelo multi-tenant para poder aprovechar mejor los recursos, pero existen algunos problemas con este modelo que afectan directamente a los clientes y que requieren una optimización para no caer en la falta de buen servicio con disponibilidad para los usuarios.

1.2. Antecedentes

La tecnología avanza cada día más con el fin explorar las posibilidades y afrontar nuevos retos con respecto a la comunicación. Esto lleva a la resolución de los problemas del diario vivir en donde una o más personas se juntan para encontrar una solución y mediante ese trabajo, poder mejorar la calidad de vida de las personas y mejor desarrollo de la sociedad.

El avance en el sector de las comunicaciones se ha dado desde hace décadas, llegando poco a poco a la era de las telecomunicaciones, donde la palabra comunicaciones adquiere el prefijo tele (distancia) para evolucionar a un entorno donde la comunicación e interacción pueda ser más real sin necesidad de haber alguna presencia.

Esta evolución ha llegado hasta el punto de la virtualización y vida en la nube, donde ya cada vez se requiere menos de objetos físicos para poder realizar tareas o cumplir necesidades del día a día. El Cloud Computing abarca un gran mundo de posibilidades para los usuarios, de manera que cada vez se logra más con menos recursos y se van acabando algunas barreras que impedían el progreso tanto de una sola persona como de toda una organización.

Estos modelos han tenido tanto éxito que ya cada vez más empresas lo adoptan y a su vez, cada día nacen nuevos proveedores de estos servicios,

algunos ampliando su portafolio de servicios como lo son las empresas de telefonía, televisión, entre otras, donde expanden sus servicios hasta dar toda la gran cantidad de productos que ofrece la nube.

1.3. Definición del Problema

Hoy en día el modelo adoptado por los proveedores de Cloud Computing es el multi-tenant, y aunque los beneficios tanto como para clientes (tenants) o proveedores son muy claros, aún hay problemas en la adopción de este modelo de multitenencia a la hora de dar el servicio. Estos problemas están relacionados con la disponibilidad, seguridad de datos, rendimiento, mantenimiento y evolución de este modelo.

1.4. Justificación del Problema

El presente trabajo realizará una investigación sobre los problemas del modelo multi-tenant en las empresas de telecomunicaciones proveedoras de servicios de Cloud Computing, identificando las falencias que presente el modelo a la hora de que un tenant requiera una nueva funcionalidad que no esté contemplada en las opciones predefinidas, donde se debe medir el impacto a los demás tenants, analizando diferentes técnicas y herramientas para dar soporte al modelo multi-tenant.

1.5. Objetivos del Problema de Investigación

1.5.1. Objetivo General

Analizar el tipo de herramientas para la optimización y mejora en el rendimiento del modelo multi-tenant en los servicios de Cloud Computing que brindan las empresas proveedoras de telecomunicaciones.

1.5.2. Objetivos Específicos

- Identificar las principales ventajas del modelo por medio de una comparativa para conocer las bondades que brinda la adopción de la arquitectura multi-tenant en servicios de Cloud Computing.
- Analizar los inconvenientes presentados por el cliente en el modelo multi-tenant para su posterior resolución.
- Comparar diferentes técnicas y herramientas que optimizan el modelo multi-tenant en el área de computación en la nube.

1.6. Hipótesis

Por medio de la computación en la nube es posible dejar de depender de varios elementos físicos, que dan la posibilidad de optimizar recursos y alcanzar los objetivos empresariales de manera más rápida. Buscando mejorar la calidad de servicio frente a los problemas que presenta el modelo en los diferentes servicios.

1.7. Metodología de Investigación

El presente trabajo tiene un enfoque cuantitativo y en base a la revisión bibliográfica realizada su alcance es de tipo descriptivo ya que se analizarán características, propiedades, rasgos, etc. del modelo de multi-tenant para la optimización y mejora de los servicios de Cloud Computing prestados por empresas de telecomunicaciones.

En la primera parte del presente trabajo se buscará entender el tema y cómo funciona el servicio de la computación en la nube, diferentes tipos, arquitecturas aplicadas a este servicio, ventajas de cada una y aplicaciones según el entorno donde se requieran para entender al sistema, y en base a técnicas y herramientas, conseguir un mayor desempeño del servicio.

Una vez finalizada la etapa de recopilación de información y de datos, se procede a la investigación descriptiva en donde se indagará sobre los problemas en el modelo multi-tenant encontrados en el camino y como afectan al servicio de la computación en la nube, hallando de esta manera el camino a seguir para la adecuada optimización del mismo y el planteamiento de las soluciones y como enfrentar las posibles situaciones en la implementación de estos procedimientos.

En suma, el trabajo presentado contiene una recopilación de información y de bibliografía apropiada, teniendo en cuenta los avances que se han dado entorno a este tema y los beneficios que conlleva el acoplamiento de este modelo para el cumplimiento de los objetivos trazados por los usuarios, resolviendo los objetivos planteados en esta tesis.

Capítulo 2: Fundamentación Teórica

2.1. ¿Qué es Cloud Computing?

La computación en la nube es un modelo que permite a través de una red, usualmente internet, ofrecer servicios de computación. A medida que el software migra de las PC locales a servidores de Internet distantes, tanto usuarios como desarrolladores van de paseo. (Hayes, 2008). Este modelo de prestación de tecnología y servicios de negocio, permite al usuario acceder a un catálogo de servicios estandarizados y usarlos para cubrir las necesidades de su negocio, de forma adaptativa y flexible, pagando solo por el consumo efectuado o inclusive de forma gratuita en algunos casos de proveedores de este servicio que se financian mediante organizaciones sin fines de lucro o publicidad.

La computación en la nube es considerada como un modelo innovador, que permite el abastecimiento de servicios de TI que generan valor para las empresas que lo adopten. Para la mayoría de las aplicaciones, toda la interfaz de usuario reside dentro de una sola ventana en un navegador web (Hayes, 2008). La computación en la nube es un modelo para permitir el acceso de red, conveniente y bajo demanda a una red compartida, conjunto de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se puede aprovisionar y liberar rápidamente con un mínimo esfuerzo de gestión o interacción del proveedor de servicios (Mell & Grance, 2011).

La computación en la nube da la facilidad a las empresas de concentrarse en sus principales actividades empresariales y de esta manera, aumentar su productividad. La adopción de Cloud Computing está creciendo rápidamente debido a la escalabilidad, flexibilidad, agilidad y simplicidad que ofrece a las empresas. (El-Gazzar, 2014).

2.2. Características

Este modelo de nube se compone de cinco características esenciales, tres modelos de servicio y cuatro modelos de implementaciones.

2.2.1. Auto servicio bajo demanda (On-Demand Self Service)

Un consumidor puede aprovisionar unilateralmente capacidades informáticas, como hora del servidor y almacenamiento en red, según sea necesario, automáticamente sin necesidad de un ser humano interactuando con cada proveedor de servicios.

2.2.2. Amplio acceso a la red

Las capacidades están disponibles a través de la red y se accede por medio del mecanismo estándar que promueve el uso de plataformas heterogéneas de cliente ligero o grueso (p. ej., teléfonos móviles, tabletas, computadoras portátiles y estaciones de trabajo).

2.2.3. Puesta en común de recursos.

Los recursos informáticos del proveedor se agrupan para servir a múltiples consumidores utilizando un modelo multi-inquilino, con diferentes recursos físicos y virtuales dinámicamente asignado y reasignado de acuerdo con la demanda del consumidor.

Existe una sensación de ubicación independiente en la que el cliente generalmente no tiene control o conocimiento sobre la ubicación exacta de los recursos proporcionados, pero puede llegar a ser capaz de especificar la ubicación en un nivel superior de abstracción (por ejemplo, país, estado o centro de datos). Ejemplos de recursos incluyen almacenamiento, procesamiento, memoria y ancho de banda de red.

2.2.4. Rápida elasticidad

Las capacidades se pueden aprovisionar y liberar elásticamente, en algunos casos automáticamente, para escalar rápidamente hacia afuera y hacia adentro de acuerdo con la demanda. Al consumidor, las capacidades disponibles para el aprovisionamiento, a menudo parecen ser ilimitadas y pueden ser apropiadas en cualquier cantidad en cualquier momento.

2.2.5. Servicio medido

Los sistemas en la nube controlan y optimizan automáticamente el uso de recursos, aprovechando una capacidad de medición para el tipo de servicio (por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas). El uso de recursos puede ser monitoreado, controlado e informado, proporcionando transparencia para el proveedor y consumidor del servicio utilizado (Mell & Grance, 2011).

2.3. Modelos de servicios

La computación en la nube cuenta con varios modelos de servicios:

2.3.1. Infraestructura como servicio (IaaS)

La capacidad proporcionada al consumidor es la provisión, procesamiento, almacenamiento, redes y otros recursos informáticos fundamentales donde el consumidor puede implementar y ejecutar el software parcialmente, que puede incluir operaciones, sistemas y aplicaciones.

Ofrece servicios de almacenamiento, servidor, red y virtualización para permitir la utilidad a los usuarios. Este hardware se suministra a través de una interfaz virtualizada o Hypervisor. El Hypervisor también se puede utilizar para la supervisión de máquinas virtuales. Comparte tanto las imágenes del sistema operativo como su software de aplicación en la infraestructura en la nube. Algunas empresas proveedoras de IaaS son: Microsoft Azure, HP, Google Compute Engine, Cloud Rackspace, IBM, Amazon EC2, DynDNS, Joyent, LeaseWeb, entre otras.

2.3.2. Plataforma como servicio (PaaS)

La capacidad proporcionada al consumidor es la de desplegarse en la nube mediante aplicaciones creadas o adquiridas por el consumidor de infraestructura, creadas con lenguajes de programación, bibliotecas, servicios y herramientas compatibles con el proveedor. El consumidor no administra ni controla la infraestructura de nube subyacente, incluida la red, los servidores, sistemas operativos o almacenamiento, pero tiene control sobre las

aplicaciones implementadas y posiblemente ajustes de configuración para el entorno de alojamiento de aplicaciones

Ofrece servicios de plataforma como lo son: sistema operativo, middleware, base de datos, servidor web y entorno de ejecución. El PaaS actúa como una capa intermedia entre IaaS y SaaS. La plataforma como servicio ofrece además hospedaje de aplicaciones, desarrollo, pruebas y entorno de despliegue. Entre las empresas proveedoras de este servicio se encuentran: Salesforce.com, Microsoft Azure, Google App Engine, AWS ElasticBeanstalk, etc.

2.3.3. Software como servicio (SaaS)

La capacidad que se brinda al consumidor es la de utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura en la nube. Se puede acceder a las aplicaciones desde varios dispositivos del cliente a través de una interfaz de cliente ligero, como un navegador web (por ejemplo, correo electrónico basado en web) o una interfaz de programa. El consumidor no gestiona ni controla la infraestructura de nube subyacente que incluye red, servidores, sistemas operativos, almacenamiento o incluso capacidades de aplicaciones individuales, con la posible excepción de ajustes de configuración de aplicación específicos de usuario limitados.

Ofrece servicios de aplicación y de software a través de la web, es decir que los datos y la aplicación de software se alojan en el servidor de Cloud. Los usuarios acceden a la aplicación con el uso de conexión de red, así dejando de necesitar de un hardware o software adicional. Entre los ejemplos de proveedores de SaaS se tiene a: Dropbox, Outlook, Google Docs, Skydrive, Google Analytics, entre otros.

2.4. Modelos de implementación

2.4.1. Nube privada

La infraestructura de la nube está aprovisionada para uso exclusivo de una sola organización, que comprende múltiples consumidores (por ejemplo, unidades de negocios). Puede ser adquirido, administrado y operado por la

organización, un tercero o alguna combinación de ellos, y puede existir dentro o fuera de las instalaciones.

2.4.2. Nube comunitaria

La infraestructura de la nube está aprovisionada para uso exclusivo de una determinada comunidad de consumidores de organizaciones que comparten las mismas preocupaciones e intereses. (por ejemplo, misión, requisitos de seguridad, políticas y consideraciones de cumplimiento). Puede ser adquirido, administrado y operado por una o más de las organizaciones de la comunidad, un tercero, o alguna combinación de ellos, y puede existir dentro o fuera de las instalaciones.

2.4.3. Nube pública

La infraestructura de la nube está aprovisionada para uso abierto por el público en general. Puede ser adquirido, administrado y operado por una organización comercial, académica o gubernamental, o alguna combinación de ellos. Existe en las instalaciones del proveedor de la nube.

2.4.4. Nube Híbrida

La infraestructura de la nube híbrida es una composición de dos o más nubes distintas (privadas, comunitarias o públicas) que siguen siendo entidades únicas, pero están vinculadas juntos por una tecnología estandarizada o patentada que permite la portabilidad de datos y aplicaciones (por ejemplo, explosión de nubes para equilibrar la carga entre nubes).

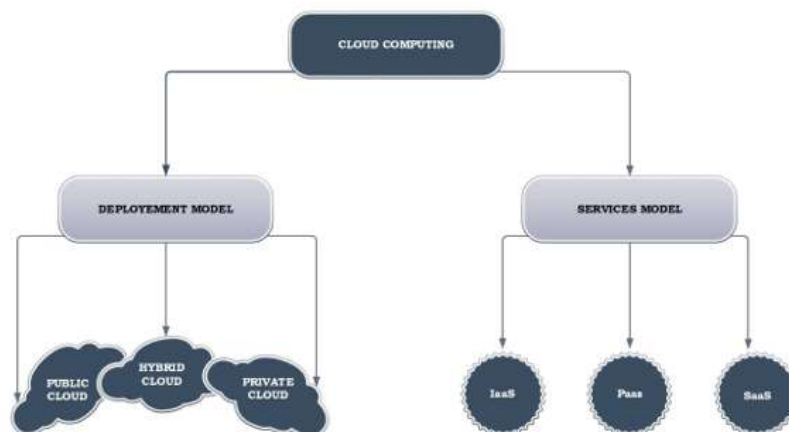


Figura 2.1: Tipos de Cloud Computing

Fuente: (Ali, 2019)

2.5. Software como servicio (SaaS)

No hay duda que el software as a service (SaaS) es el tipo de tecnología de infraestructura que facilita este tipo de entrega y cobro por software, el Cloud Computing, donde se está convirtiendo en la nueva plataforma para empresas y personal de computación (Cusumano, 2010). El software como servicio es un modelo de distribución de software donde los datos y el soporte lógico se albergan en los servidores de una compañía proveedora de tecnologías de información y comunicación (TIC) a los que el cliente accede vía internet.

Este tipo de computación en la nube ofrece una sola aplicación a través del navegador a miles de clientes que utilizan una arquitectura multi-tenant (Knorr & Gruman, 2008). Por el lado del cliente, significa que no hay una inversión inicial en servidores o licencias de software; en el lado del proveedor, con tiene una aplicación para mantener, los costos son bajos en comparación con el alojamiento convencional. La empresa de TIC se encara del mantenimiento, operación y del soporte del software usado por el usuario.



Figura 2.2: Usuarios y proveedores de Cloud Computing.

Fuente: (Javed, 2017)

En las nubes SaaS, la carga de trabajo suele ser compleja. y comprende aplicaciones con varios grados de paralelismo y prioridad. El mayor desafío es hacer frente al caso donde llegan aplicaciones de una sola tarea en tiempo real de alta prioridad y tener que interrumpir otras solicitudes en tiempo real para cumplir con sus plazos (Georgios & Karatza, 2016).

2.6. Potencial del Cloud Computing

La incursión de la tecnología de Cloud Computing ofrece un nuevo camino en el mundo empresarial, brindando nuevas posibilidades en el campo de logística y en los nuevos canales de desarrollo en procesos logísticos. La computación en la nube ha recibido un creciente interés por parte de las empresas desde sus inicios (El-Gazzar, 2014). Mediante una sola conexión a la red se dispone de los servicios como el proceso masivo de datos, servicios web y almacenamiento de la información que viajan a través de servidores y dispositivos conectados a internet.

Poder proporcionar al cliente un servicio eficiente y eficaz a través de un servicio ágil, inmediato y seguro es el objetivo de este modelo en la nube. Mediante la adopción de este modelo se asegura el alcance de los objetivos propuestos y el mayor aprovechamiento de los recursos.

2.7. Empresas de telecomunicaciones como proveedores de servicios de Cloud Computing

El servicio de Cloud Computing ha generado una nueva dimensión y varias oportunidades a los proveedores de telecomunicaciones para el avance en la cadena de valor de las tecnologías de información y de la comunicación (TIC). Las empresas de telecomunicaciones proveedoras de este servicio tienen dos ventajas en comparación a las otras industrias en materia de las nubes.

La primera es que la industria de las TIC proporciona conectividad en términos de domesticidad y de conectividad, pero en lo que respecta a redes, nadie está más cerca del consumidor que estos proveedores, de esta manera, como proveedor se puede instanciar la funcionalidad de la nube para los usuarios con la menor latencia que pueden usar.

El segundo punto es que los proveedores pueden crear a medida servicios dependiendo de las necesidades del consumidor, como cuando se movilizan los canales disponibles; ejemplo serían las tuberías de ancho de banda que conectan al usuario con los centros de datos, donde se podría asignar un periodo justo de tiempo a pedido del cliente/consumidor.

2.8. Empresas de telecomunicaciones como adoptantes de los servicios de Cloud Computing

En la otra cara, se tiene a las empresas de telecomunicaciones implementando la computación en la nube para el funcionamiento de su empresa y mediante este medio poder gestionar toda operación interna. Los proveedores operan en un entorno operativo complejo, donde dependen de las tecnologías para ejecutar sus redes y sustentar la prestación de los servicios. Estas han adoptado el servicio de la nube para poder aprovechar al máximo su uso y de esta manera dar un mayor potencial a la empresa para atender a sus clientes, donde es tan beneficioso tanto implementarlo, como proveer el servicio de la computación en la nube, generando ahorro de costos y ganancia eficiente.

2.9. Ventajas del Cloud Computing

Se destacan algunas ventajas importantes que otorga el uso de la computación en la nube, como, por ejemplo:

- Ofrece multilocación del servicio, esto quiere decir que no están anclados a ejecutar un software en un solo ordenador o servicio local, si no que desde el dispositivo que se quiera, siempre y cuando tenga conexión a internet.
- Se paga solo por el servicio que se usa en el momento, así se deja a un lado el adquirir un software personalizado que después de un tiempo quedaría obsoleto.
- No se requiere de dispositivos de gran capacidad de almacenamiento, ya que la misma nube proporciona una cantidad de espacio ilimitado donde se puede operar acorde a cada necesidad.
- Ofrece seguridad a tu trabajo de tres maneras diferentes: realiza copias de seguridad automáticas así no hay pérdida de trabajo por olvidarse de guardar; en caso de un fallo del ordenador no se verían afectados los datos virtuales; y si se llegase a bloquear un ordenador se puede acceder a la información desde otro dispositivo ya que todo está en la nube.

2.10. Proveedores de Cloud Computing y sus servicios

Salesforce

Siendo un conocido proveedor de SaaS, Salesforce es una herramienta de CRM que ayuda a las empresas a gestionar las relaciones con los clientes y generar oportunidades de ventas. Sales Cloud combina inteligencia artificial y datos de clientes para ayudar a los equipos de ventas a identificar posibles clientes potenciales y cerrar ventas más rápido. La plataforma también tiene nubes separadas para servicio al cliente y marketing (Daley, 2019). Se ha nombrado múltiples veces a Salesforce como el líder de SaaS, basado en la flexibilidad que ofrecen las soluciones basadas en su nube.

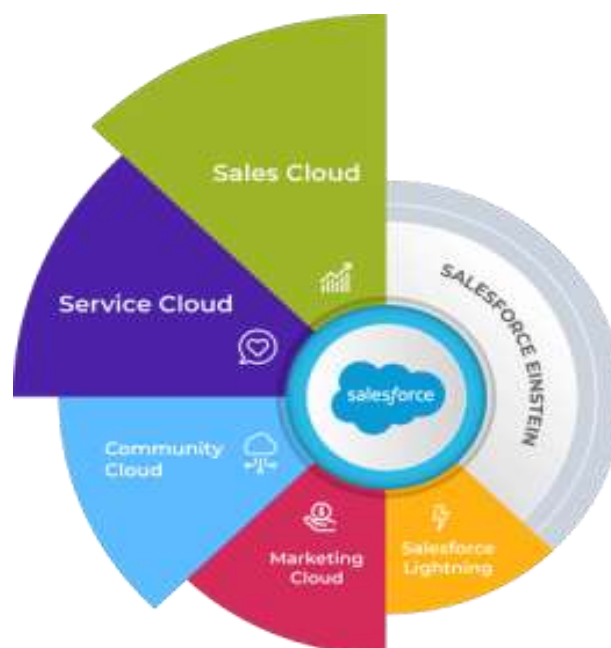


Figura 2.3: ¿Que es SalesForce?

Fuente: (Khanduri, 2019)

Amazon Web Service (AWS)

AWS tiene muchos más servicios y funciones que cualquier otro proveedor de la nube, desde tecnologías de infraestructura como computación, bases de datos y almacenamiento, como también tecnologías emergentes, como machine learning, inteligencia artificial, lagos de datos y análisis e internet de cosas. Todo esto hace que sea más rápido, más fácil y más rentable mover sus aplicaciones existentes a la nube y construir casi cualquier cosa mediante las herramientas que el servidor ofrece (Amazon Web Services, 2019). Las herramientas de AWS Lambda, que ofrece Amazon,

es usada por diversas industrias como The Coca Cola Company, The Seattle Times y iRobot.



Figura 2.4: El mundo de Amazon Web Services.
Fuente: (Goyal, 2019)

Digital Ocean

Digital Ocean es una plataforma que ayuda a los equipos de productos a implementar, administrar y escalar nuevos productos. Bajo esta plataforma de IaaS, los usuarios tienen la posibilidad de crear múltiples máquinas virtuales en segundos, e incluso pueden escalar en función del almacenamiento de datos y el tráfico entrante (Daley, 2019).

Example Backup System

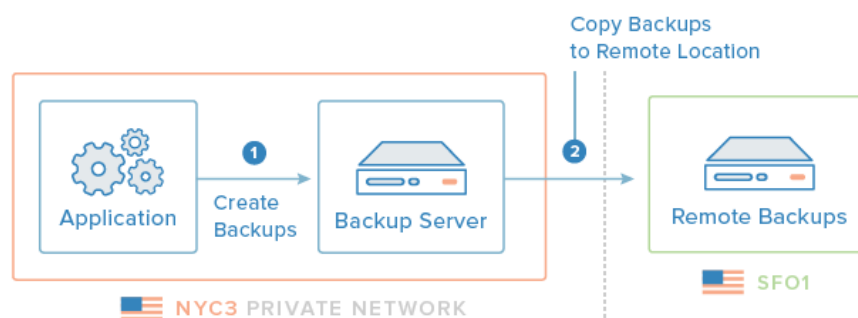


Figura 2.5: Ejemplos de Backups.
Fuente: (Anicas, 2015)

IBM

Los servidores IaaS de IBM se pueden implementar tanto en unos minutos como en unas pocas horas, dependiendo del tipo de nube. Los servidores de IBM tardan unas horas en implementarse y pueden acomodar

proyectos de hasta tres terabytes (Daley, 2019). Los servidores virtuales se implementan en cuestión de minutos y son útiles para proyectos escalables que requieren flexibilidad. El servicio ofrecido por la empresa es usado por grandes empresas como American Airlines y Applift para el hacer sus productos basados en la nube más eficientes.



Figura 2.6: Bases de datos IBM.
Fuente: (The Poughkeepsie Journal, 2014)

Datrium

Datrium es una empresa híbrida de gestión de datos y computación en la nube. La nube IaaS de la compañía ofrecen herramientas mejoradas de ciberseguridad como verificación general, respaldo de datos integrado y medidas de seguridad de datos (Daley, 2019). Actualmente la empresa, junto a Samsung hizo una inversión de \$60 millones para la búsqueda de nuevas soluciones en el campo de IaaS.

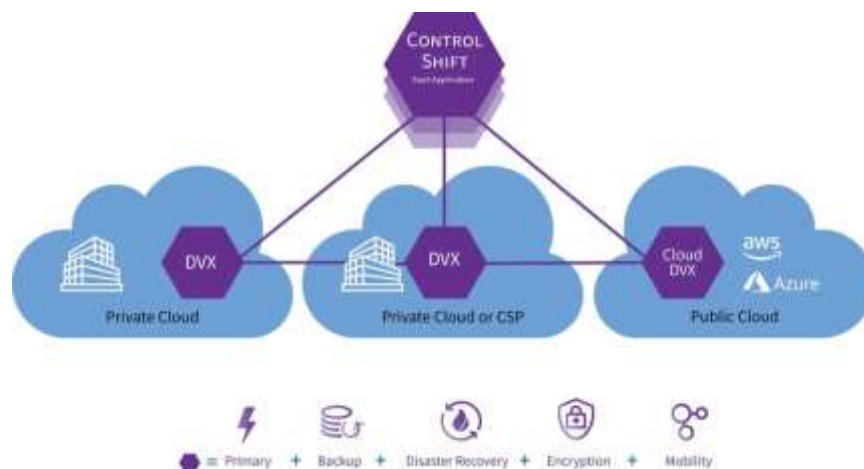


Figura 2.7: Mundo Multicloud de Datrium.
Fuente: (CISION, 2019)

Slack

Esta es una herramienta de colaboración tanto como para equipos o compañías. Esencialmente los canales de Slack son mensajes grupales y son organizados por: individuo, equipo, proyecto o tema, entre otras. De esta manera se asegura que todos estén en la misma página al momento de la comunicación sobre un tema. Esta compañía de cloud SaaS da la posibilidad de un video chat con compartición de PDF, donde integra a otras compañías de Cloud Computing como Dropbox y Salesforce. Numerosas organizaciones, incluyendo la NASA, Airbnb y Target hacen uso de Slack para la colaboración y comunicación.

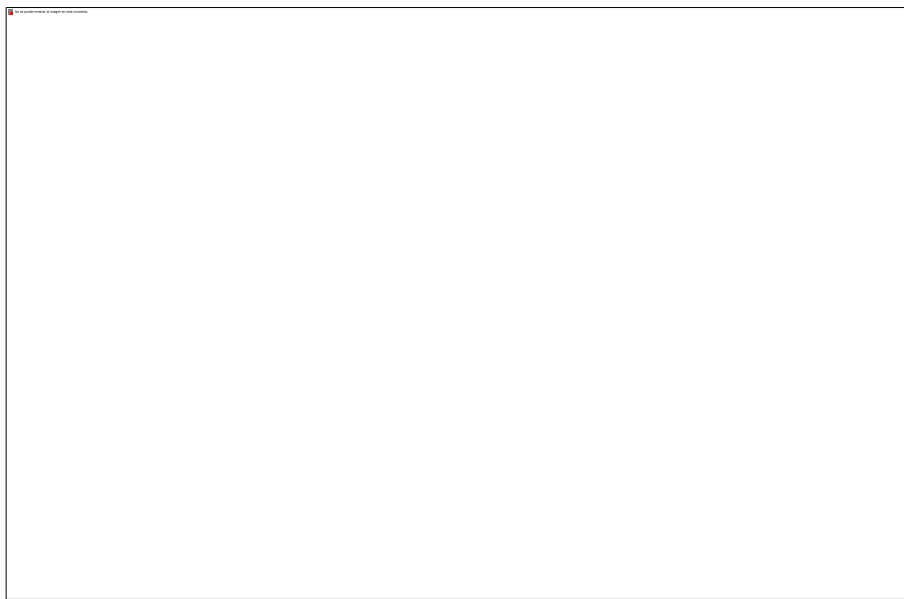


Figura 2.8: Tipos de Cloud Computing y sus ramas.
Fuente: (Mitchell, 2019)

2.11. UIT sobre los servicios de Cloud Computing

Según la Unión Internacional de Telecomunicaciones (UIT), los servicios de Cloud Computing cubren la mayor parte de las comunicaciones y servicios de colaboración en la red de telecomunicaciones, como también la tecnología informática (IT) y las aplicaciones de software.

De acuerdo a la taxonomía y al reciente desarrollo del mercado de la computación en la nube, el Cloud/Network Computing debería ser considerado como una importante oportunidad para los proveedores de telecomunicaciones para de esta manera incrementar significativamente su

negocio en el mercado de las TIC (Tecnologías de la información y comunicación) que está actualmente representado por los vendedores de IT y los de internet (International Telecommunication Union, 2012).

Los beneficios del Cloud Computing pueden ser varios de acuerdo a la perspectiva de los involucrados en el ecosistema de la nube: proveedores, asociados y usuarios.

De acuerdo a la UIT los beneficios de adoptar Cloud Computing son:

- Considerar el modelo de entrega de la nube como una plataforma para entregar IT y servicios de comunicaciones por medio de cualquier red (móvil, de cobertura global), y usada por cualquier usuario final conectado a dispositivos (PC, TV, Smartphone, maquinas, etc) (International Telecommunication Union, 2012).
- Entregar un rico conjunto de servicios de comunicaciones (creación de contenido, espacio de trabajo, transmisión, videollamadas, audio, conferencias web, comunicaciones unificadas), de acuerdo a un modelo basado en consumo-uso en un entorno cloud multi-tenant.
- Considerar los servicios de red (conectividad L2-L3, VPN y servicios de red L4-L7) como tubos inteligentes para servicios de transporte cloud e interconexión cloud (inter-cloud) para de esta manera garantizar una segura y de alto desempeño, de fin a fin, calidad de servicio (QoS) para los usuarios finales (International Telecommunication Union, 2012).

En añadidura a estos beneficios, algunos otros beneficios pueden ser considerador de parte del proveedor del servicio, asociado y perspectiva del usuario, como también mejoras en la seguridad del Cloud Computing (International Telecommunication Union, 2012).

2.12. Roles de las telecomunicaciones en Cloud Computing

De acuerdo con la UIT, los involucrados en las telecomunicaciones tiene un importante rol en el crecimiento del mercado y ecosistema Cloud. La red de telecomunicaciones es una parte central para que la arquitectura multi-

tenant cloud entregue multiservicios a múltiples usuarios con alto nivel de QoS y una óptima localización de recursos. Este rol es considerado como la única posición para entregar un alto grado de servicios cloud integrando plataforma, servicios, bases de datos e infraestructura de redes IP garantizando de fin a fin, los SLA. Los roles de las telecomunicaciones en el Cloud Computing son considerados desde el punto de vista operacional y de aspectos técnicos (International Telecommunication Union, 2012).

Los principales roles operaciones de las telecomunicaciones son:

- Control de red: Es considerado como el principal conductor en la entrega de los servicios cloud. Las telecomunicaciones/ICT tiene un alto nivel de experticia en el despacho de fin a fin del manejo de las SLA de todas las redes públicas y privadas.
- Operación y mantenimiento: Esto incluye el soporte al cliente, aseguramiento de cumplimiento y facturación a gran escala para las redes mundiales y regionales de suscriptores.
- Experiencia de usuario y la relación con el cliente: Los involucrados en las telecomunicaciones han creado una buena experiencia de usuario para el consumidor y grandes mercados empresariales.
- Asociados de confianza: Los proveedores de telecomunicaciones son considerados asociados de confianza para los consumidores por la seguridad de la información, integridad y privacidad con una presencia local. La mayor parte de proveedores de telecomunicaciones/ICT han implementado algunos estándares ICT (de ISO y UIT-T), guía y frameworks (International Telecommunication Union, 2012).
- Intermediarios Cloud: Esto permite el control de la entrega de las IT, redes y servicios de comunicaciones cloud para federaciones Inter-Cloud o Private-Public Cloud.

2.13. Impacto del Cloud Computing en las empresas

Las empresas cada día confían más en la tecnología como herramienta para su negocio y, cada vez utilizan servicios y soluciones tecnológicas más

variadas que ayudan a diferentes áreas de la compañía, entre los que destacan las soluciones cloud (Tecnología para tu Empresa, 2019).

Las empresas que compran soluciones en la nube han crecido, volcando cada vez más información y datos de sus diversas áreas de negocios hacia la red. El 73,5% de las empresas con 10 o más empleados contrataron un servicio de e-mail mientras que el 72,4% contrataron servicios de almacenamientos, el 47,2% hicieron la adquisición de servicios de software de ofimática, Un 32,3% aplicaciones para tratar información de los clientes y un 30,6% contrató aplicaciones de software financiero y contable. Aparte de esas, el 71,5% de las empresas contrataron un servicio de Cloud Computing brindado desde servidores compartidos (multi-tenant), mientras que el 46,7% compro un servicio entregado desde servidores reservados para su empresa (Tecnología para tu Empresa, 2019).

2.14. Adopción del Cloud Computing en el mundo

El Cloud Computing inicialmente emergió en los Estados Unidos. Desde el año 2015, los Estado Unidos ha sido líder en la adopción de este servicio y ha marcado el paso a la adopción en el mundo (Goasduff, 2019). No obstante, no muchos países están a la par con los Estados Unidos, de hecho, para el 2022 la adopción del Cloud Computing de los demás países estará atrasada de uno a siete años con respecto a Estados Unidos. También en el 2022 en Estados Unidos, el total del gasto en IT será en servicios cloud.

Esta búsqueda en el gasto y crecimiento por país muestra donde la adopción cloud está dándose más rápido y cuáles son los países más lentos en adoptarlo. Los países que van por detrás de los Estado Unidos se los categoriza como países en ruta e incluyen a Reino Unido y Países Bajos. Su gasto en estos servicios está previsto a aumentar 2.8 y 3.2 por ciento del 2019 a 2022, respectivamente (Goasduff, 2019), como muestra La Figura 2.9 el grafio de crecimiento de la computación en la nube en los países que más han adoptado el modelo.

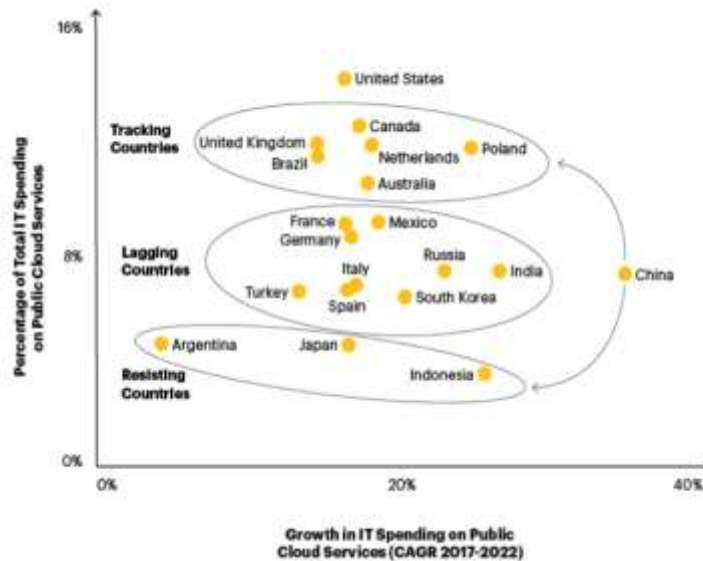


Figura 2.9: Evolución de Cloud Computing en principales países.
Fuente: (Goasduff, 2019).

Los Centros de datos en Europa fueron establecidos en primer lugar en Irlanda, Reino Unido y Países Bajos, lo que llevó a la temprana adopción de los servicios cloud dado que las organizaciones locales tomaron ventaja de su gran elasticidad, autoaprovisionamiento y un modelo *pago por uso* flexible.

A pesar de la velocidad de crecimiento del 34,8% hasta 2022, China sigue siendo un país en demora. Después del 2023, se predice que China será un país en ruta. Hasta 2022, Japón será posicionado como uno de los países resistentes por un lapso de uno a siete años atrás de los Estados Unidos (Goasduff, 2019).

2.15. Modelo Multi-tenant

La multitenencia se define como una propiedad cloud que se implementa en un modelo basado en el concepto de tenant. Un tenant es un conjunto de varios usuarios que comparten necesidades y requisitos, por ejemplo, una organización.

Las aplicaciones orientadas a servicios tradicionales, en primeras instancias dedican una aplicación por cada usuario o tenant, es decir, se crea en base a sus necesidades una instancia personalizada. Esto es llamado

single-tenant, que resulta bastante efectivo en un modelo de cloud, pero quedan bastantes recursos sin aprovechar. Sin embargo, hoy en día, las empresas de telecomunicaciones proveedoras de SaaS tienden a adoptar la arquitectura multi-tenant, ya que todas estas instancias comparten la misma estructura, desde el sistema operativo, hasta recursos de hardware, bases de datos o lógica de negocio.

Una arquitectura multi-tenant donde múltiples usuarios comparten la aplicación permite economías de escala a medida que los usuarios comparten recursos (Figura 2.10), pero todo esto a costa de una aplicación más compleja que tiene que manejar varios usuarios independientes.

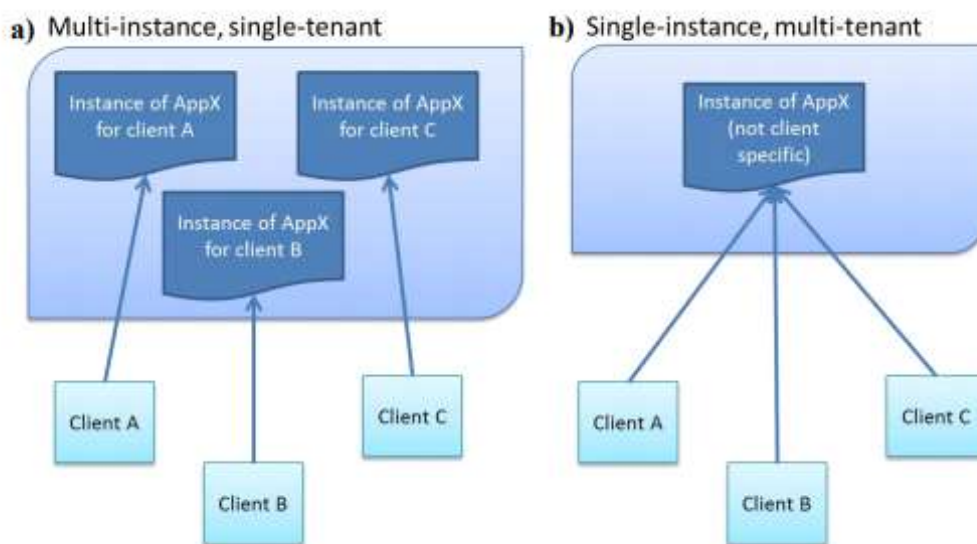


Figura 2.10: Multi-tenant vs Single-tenant
Fuente: (Microsoft, 2014)

Entonces, el modelo multi-tenant es un modelo de prestación de servicios de software que se encuentra alojado en la nube, cuya característica principal es que una sola instancia de software, pueda ser usada por varios tenants, lo que optimiza el uso de los recursos. Es vital diferenciar a los usuarios de los tenants, donde los usuarios son aquellos que acceden a la aplicación y que son parte de una organización conocida como tenant que es la que contrata el servicio (Figura 2.11).

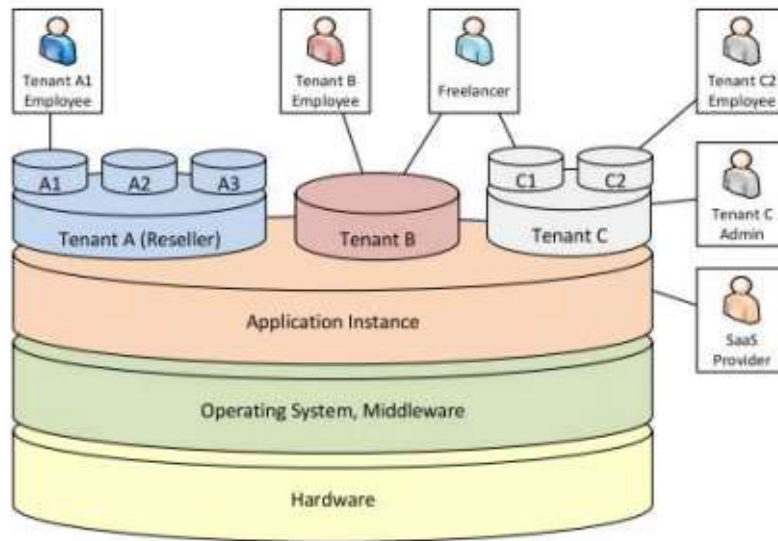


Figura 2.11: Estructura del modelo multi-tenant
Fuente: (Maenhaut, 2014)

2.16. Tipos de Multitenencia

2.16.1. Nada compartido (Nothing Shared)

Esta arquitectura de implementación provee servidores independientes por cada tenant y asigna una separación tanto de hardware como de software para cada tenant. Por lo tanto, se trata de un modelo single-tenant, en el cual cada uno de los clientes tiene su base de datos independiente e instancia del software (Figura 2.12). Con esta función, no hay nada que compartir, cada uno tiene sus propios recursos separados de todos los demás.

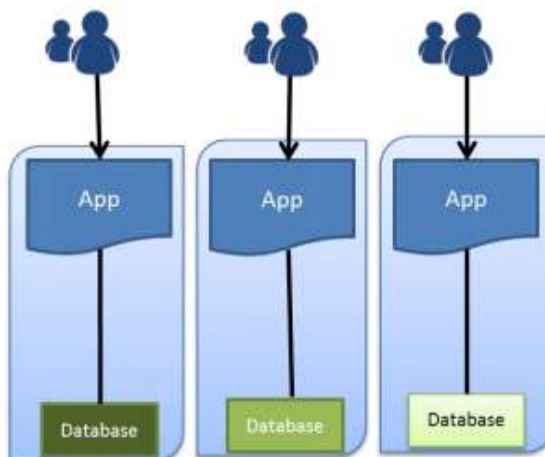


Figura 2.12: Modelo single-tenant, cada uno usa sus propios recursos.
Fuente: (Macias, 2010)

2.16.2. Hardware compartido (Shared hardware)

Se basa en el concepto de virtualización, por lo tanto, es una virtualización en la nube donde únicamente el hardware es lo compartido. En esta tecnología de virtualización, el uso de un hypervisor para ejecutar las máquinas virtuales en un equipo es esencial, donde esta máquina física se llamará host. Cada uno de estos sistemas operativos de las máquinas virtuales compartirán recursos de hardware y obtendrán soporte a través del mecanismo de reenvío dado por los recursos de hardware de capas inferiores.

2.16.3. Sistema Operativo compartido (Shared OS)

La tecnología de virtualización que se basa en máquinas virtuales o contenedores, es una diferente a nivel de sistema a la tecnología de virtualización que es basada en hypervisor. Es un mecanismo virtual que utiliza el kernel del sistema operativo para completar la gestión y aislamiento de recursos. De esta manera, cada máquina virtual es considerada más una aplicación independiente que se ejecuta en el sistema operativo.

2.17. Base de datos compartida (Shared Database)

Esta se basa en el concepto de la multitenencia de datos. Los tres diferentes enfoques son:

2.17.1. Base de datos independientes e instancias independientes.

El primer enfoque que se debe tener en cuenta para diseñar una base de datos multi-tenant es la de bases de datos independientes e instancias independientes de base de datos (Figura 2.13). En esta instancia, el proveedor creará una base de datos específica para dar servicio a cada tenant, lo que resulta en que cada tenant puede almacenar las tablas de origen en su propia base de datos (Macias, 2010). Esta instancia es fácil de desplegar y también puede ser construido directamente encima de cualquier base de datos actual. Este enfoque tiene buen aislamiento de seguridad y datos, pero su costo de mantenimiento es muy caro.

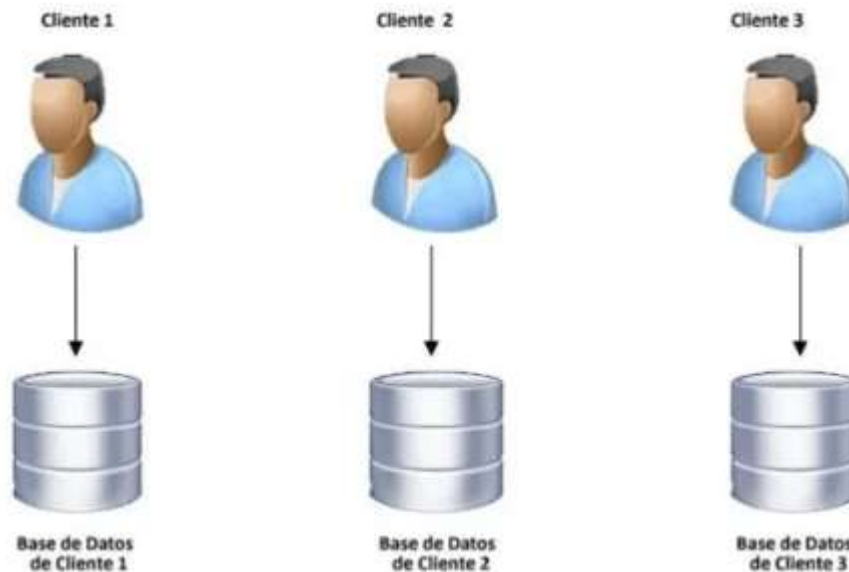


Figura 2.13: Este enfoque usa una base de datos para cada cliente.
Fuente: (Macias, 2010)

Consideraciones:

- **Tiempo de desarrollo:** Este modelo de nada compartido requiere más tiempo de desarrollo mínimo en comparación a la arquitectura estándar.
- **Aplicación y rendimiento de la base de datos:** Este rendimiento es bastante predecible, dado que el mismo no se ve afectado por los demás clientes.
- **Seguridad:** Los datos de cada cliente pueden ser protegidos de mejor manera ya que están aislados de los demás clientes.
- **Costo de hardware:** Debido a que cada cliente requiere de su propio servidor de hardware, este tipo de arquitectura es la más cara y nada se comparte. Aparte, se sumará al costo cualquier prueba a lo largo del servicio.
- **Personalización:** Dado que cada cliente posee su propia base de datos, es más fácil de que cada uno la personalice de acuerdo a sus necesidades.
- **Número de clientes:** Con este enfoque se vuelve más difícil el manejo de grandes números de clientes.

2.17.2. Tablas independientes e instancias de datos compartida.

El segundo enfoque de diseño del esquema de datos multi-tenant es el de las tablas independientes. De acuerdo a este enfoque los tenants además del hardware, comparten la instancia de la base de datos (Figura 2.14). El proveedor utiliza la misma instancia de base de datos para almacenar las tablas de origen de varios tenants, (Macias, 2010). Como las tablas de origen de diversos tenants pueden poseer el mismo nombre, se precisa añadir un ID de tenant para distinguir cada tabla de diferente tenant. Este enfoque del modelo puede reducir el costo de mantenimiento en comparación al enfoque anteriormente visto y este proporciona una mayor escalabilidad.

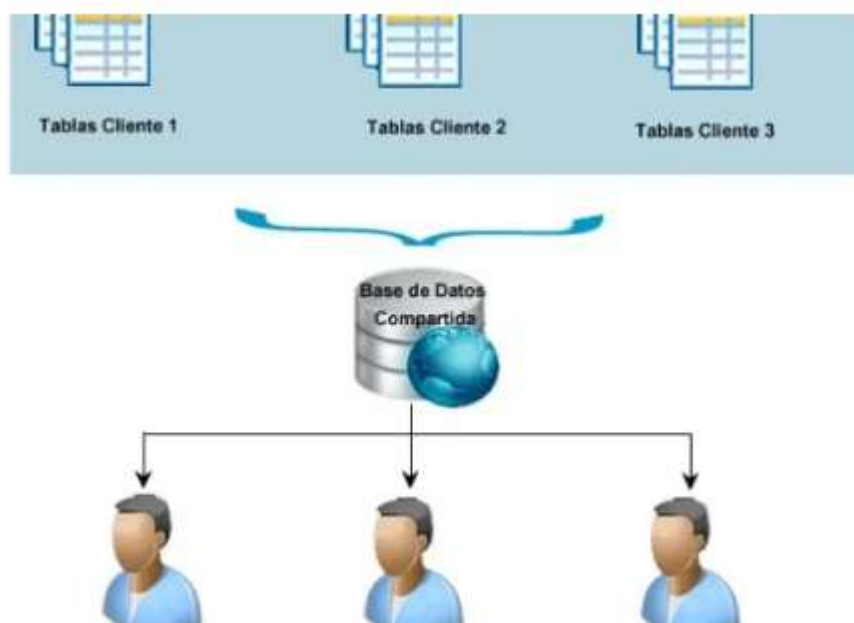


Figura 2.14: Clientes con base común y conjunto de tablas separadas.

Fuente: (Macias, 2010)

Consideraciones:

- **Aplicación y rendimiento:** El rendimiento de un cliente puede ser afectado por las actividades que realizan los demás clientes que comparten el servidor
- **Personalización:** Cada uno de los clientes tiene su propio esquema, de esta manera es fácil personalizarlo según las necesidades de cada cliente.
- **Seguridad:** El DBMS debe cerciorar de que su estructura de permisos de cada cliente solo esté disponible para usuarios autorizados.

También la aplicación debe seleccionar o restringir el esquema que se accede por un usuario mediante el uso de comandos.

- **Número de clientes:** Este modelo puede manejar más clientes que el anterior, pero aun requiere de una buena administración.

2.17.3. Tablas compartidas e instancias de datos compartidas.

Este tercer y último enfoque de diseño de esquemas de base de datos multi-tenant es el de las tablas compartidas e instancias compartidas. En este, los distintos tenants comparten, además de la instancia de la base de datos, también las tablas (Macias, 2010). El proveedor utiliza las tablas principales para administrar los datos de los distintos tenants. Las particularidades de la tabla principal son la unión de los atributos de todos los tenants. Todas las tuplas de cada uno de los tenants se consolidan en la tabla principal.

Al usar este modelo, el proveedor de los servicios puede minimizar el costo de mantenimiento en una gran parte ya que el número total de las tablas principales solo se determina por el número máximo de las tablas de origen de distintos tenants (Macias, 2010). De esta manera se obtiene mejor escalabilidad que otros enfoques.

No obstante, tiene algunas limitaciones, como, por ejemplo: si algunos atributos son compartidos por un grupo pequeño de tenants, esto implicara un numero grande de NULLs y grandes cantidades de espacio serán desperdiciadas. En otro punto, inclusive si muchas bases de datos son capaces de tratar con los NULL de una manera eficiente, se degrada el rendimiento cuando las tablas de la base de datos se esparcen demasiado.

Consideraciones:

- **Aplicación y rendimiento:** El rendimiento de un cliente puede ser afectado por las actividades que realizan los otros clientes. El rendimiento deberá ser examinado para garantizar que existen los índices adecuados para el buen funcionamiento colectivo.

- **Seguridad:** La aplicación debe seleccionar o restringir los datos basado en el cliente mediante un código de consultas. No es posible realizar la encriptación de datos únicos para cada cliente. Se debe garantizar con pruebas sólidas que un usuario no puede ver los datos de los otros clientes.
- **Personalización:** Es más difícil permitir una individualización a que todos los clientes compartan el mismo esquema. Existen varios enfoques que pueden ser utilizados para proporcionar una personalización, cuyo planteamiento sería permitir a un numero de columnas en cada tabla genérica que se pueda usar por cada cliente de diferentes maneras.
- **Número de clientes:** En este modelo de se puede manejar muchos más clientes que en los modelos anteriores. Una migración de clientes que requieran mayor capacidad o mejor rendimiento puede llegar a ser un reto, ya que los datos tendrían que ser extraídos de cada caso en diferentes operaciones.

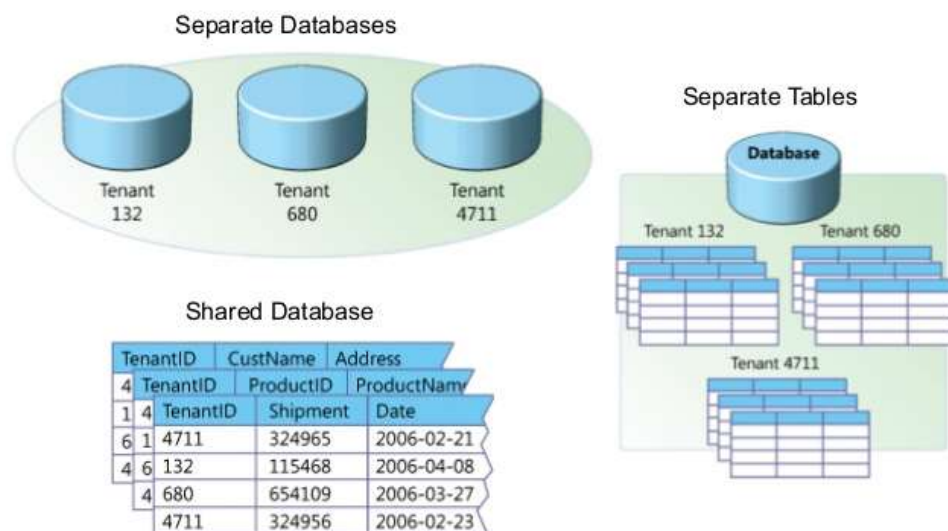


Figura 2.15: Estrategias de bases de datos.

Fuente: (Swamy, 2014)

2.18. Multi-tenant vs Single-tenant

Vistas ya las diferentes características y ventajas que presentan los modelos mencionados, donde la alternativa a la arquitectura de la nube de

multi-tenant es la single-tenant, la cual solo aloja un cliente o inquilino, el cual tiene acceso exclusivo, tiene una clara ventaja para el cliente donde le brinda mayor control sobre múltiples capacidades, que incluye datos, el rendimiento, el almacenamiento y la seguridad.

Entonces, ¿cómo el multi-tenant sobresale sobre su anterior, el single tenant?

Estos son los beneficios directos para las empresas:

- Hará que una mayor variedad de recursos esté disponible para una mayor cantidad de personas en la organización, todo esto sin una ralentización de las aplicaciones o generando un obstáculo en la privacidad como la seguridad.
- Proporciona una cantidad mayor de almacenamiento y acceso en comparación con un modelo single-tenant, que tiene parámetros de seguridad y acceso limitado.
- Virtualización de ubicaciones de almacenamiento que se basan en la nube, donde se ofrece una mayor flexibilidad y un acceso más fácil desde cualquiera que sea la ubicación del dispositivo.

2.19. Gestión de los recursos en la nube mediante algoritmos predictivos

La capacidad de predecir los problemas que está teniendo un servicio en las redes informáticas y a su vez como poder responder a estos problemas a través de la aplicación de acciones correctivas, es una forma de estar siempre un paso adelante y al tanto de todo lo que puede estar generando problemas, que trae múltiples beneficios al sistema. Los algoritmos predictivos pueden desempeñar un papel crucial en la gestión de los sistemas ya que alertan de fallos (Vilalta, Hellerstein, Weiss, Ma, & Apte, 2002).

Los casos más estudiados en materia de la predicción de fallas en los sistemas de computación son: la predicción a largo plazo sobre variables de funcionamiento del sistema (la utilización del disco), predicción a corto plazo sobre un comportamiento anormal (violaciones de seguridad), y predicciones de eventos del sistema (fallos en el router). Los resultados muestran que los

algoritmos predictivos pueden ser usados exitosamente en la estimación de las variables de rendimiento y la predicción de eventos críticos (Vilalta, Hellerstein, Weiss, Ma, & Apte, 2002).

La detección de fallos del sistema en muchos servidores se ha visto que genera un impedimento en la propagación de los fallos a toda la red, donde, por ejemplo, un tiempo de respuesta bajo da la posibilidad a escalar los problemas técnicos a todos los nodos que se comunican con el servidor. En segundo lugar, la predicción se puede utilizar para que una provisión continua de los servicios se pueda asegurar, a través de una correcta implementación de acciones correctivas automatizadas.

En el caso de la computación en la nube y del modelo multi-tenant más concretamente, se necesita conocer la cantidad de recursos que requerirá cada tenant en las diferentes horas del tráfico, esto se necesita saber para la toma de decisiones de acuerdo a los SLA de cada tenant, que es realizado por los algoritmos predictivos como la regresión lineal o el filtro de Kalman.

2.19.1. Regresión lineal

El análisis de regresión lineal es utilizado para prever el valor de una variable con base a otra variable (IBM, 2017). Este análisis es el más básico y comúnmente utilizado. Las estimaciones de regresión se utilizan para describir dato y explicar una relación entre una variable dependiente y una o más variables independientes (Statistics Solutions, 2013). Al conocer los recursos requeridos y el tiempo de ejecución de una solicitud anterior en una aplicación multi-tenant, se puede estimar el monto de recursos requeridos por una petición similar con el uso de este algoritmo.

2.19.2. Filtro de Kalman

Este filtro es un algoritmo que provee estimados de algunas variables desconocidas de acuerdo a las medidas y parámetros observados a lo largo del tiempo (Bang & Kim, 2018). En una aplicación multi-tenant este algoritmo calcula el consumo de una petición de manera dinámica, en un tiempo de acuerdo a las solicitudes que se procesaron con anterioridad para de esta

manera estimar el consumo de la petición actual. Se debe saber primero que se está realizando para controlar un sistema dinámico, ya que para estas aplicaciones no es siempre posible medir todas las variables que se desea controlar y este filtro genera un medio para conseguir inferir en la información y las fallas. El filtro de Kalman tiene forma relativamente simple y requiere una cantidad pequeña de energía computacional (Bang & Kim, 2018).

2.19.3. Cadenas de Markov

El método de las cadenas de Markov da la posibilidad a calcular posibles retardos en los servicios de la nube. En este proceso, el resultado de un experimento dado puede afectar el resultado del siguiente experimento (Dartmouth, 2017). Este tipo de proceso se llama cadena de Markov, que permite predecir el consumo de recursos en entornos multi-tenant.

2.20. Variabilidad y Multi-tenant.

La multitenencia y la variabilidad están ligados de manera que la multitenencia se basa en el hecho de compartir recursos de software, de manera que una instancia de software única sea compartida entre varios tenant. De manera que es una dificultad que todos los tenants posean las mismas necesidades y requisitos de software, se necesita brindar un grado de personalización y configurabilidad. La gestión de variabilidad sobre un conjunto de productos se encarga de la personalización de software de manera masiva (Pohl, Bockle, & Van der Linden, 2005).

Mediante el uso de la virtualización y la compartición de recursos, las nubes abordan con un solo conjunto de recursos físicos una gran base de usuarios con diversas necesidades (Iosup, Yigitbasi, & Epema, 2011). Las nubes tienen el potencial para proveer a sus dueños los beneficios de una economía a escala y, al mismo tiempo, ser una alternativa para la industria y para la comunidad científica en ambientes paralelos. En el dominio de la computación, existen otros tipos de variabilidad descritos a continuación:

- **Variabilidad de la aplicación:** Existen requisitos tanto de comportamiento como de funcionalidad que requiere a cada tenant

de SaaS, además de su aplicación principal. Por otro lado, los usuarios de los tenants quizá precisen de una apariencia diferente de interfaz de usuario y de una personalización mayor en la estructura.

- **Variabilidad del proceso empresarial:** Las necesidades y el flujo de trabajo del negocio van a variar de acuerdo a los requisitos específicos que tengan los tenants en la relación empresarial.
- **Variabilidad de la plataforma:** Las plataformas como servicio que incluyen sistemas operativos, lenguajes de programación y pilas de soluciones.
- **Variabilidad de aprovisionamiento:** Redes, máquinas virtuales, hardware. Todo lo que concierne a IaaS.
- **Variabilidad de despliegue:** Tipos de nubes, ya sean públicas, privadas, híbridas o comunitarias.
- **Variabilidad de proveedor:** Servicios de web como Salesforce, Google Application Engine o Amazon AWS, entre otros.

El modelo multi-tenant presenta varios servicios que comparten los tenants, donde se debe manejar una variabilidad para la construcción de aplicaciones eficientes.

2.21. Arquitecturas y enfoques de desarrollo en la nube

La nube está relacionada con arquitecturas orientadas a servicios (SOA), arquitecturas orientadas a eventos (EDA), arquitectura orientada a recursos (ROA) y arquitectura hipermedia (HOA). La mayoría de las aplicaciones contienen funcionalidades comunes, se pueden citar entre otras: almacenamiento en memoria estática, comunicaciones, tratamiento de la interfaz usuario, errores, transacciones, mantenimiento de la seguridad (Romero, 2005).

2.21.1. Arquitecturas orientadas a servicios (SOA)

Este tipo de arquitectura ha sido descrito en la industria de tecnologías de la información de varias maneras a través de los años. Desde el 2001, los estándares adoptados de la Edición Empresarial de Java y los Servicios Web

han habilitado un nuevo nivel de interoperabilidad y arquitecturas orientadas a servicios. De esta manera, se ha llegado a un enfoque donde los procesos se llegan a automatizar entre diferentes arquitecturas y llegan a ser vistos como servicios autónomos que interactúan entre si a través de mensajes (Romero, 2005). Además de tener lógica y datos propios, cada servicio da a conocer claramente la interfaz basada en mensajes para accederlo y así constituir una aplicación independiente y autónoma que de la posibilidad de comunicarse con otras aplicaciones.

Cada una de las aplicaciones de este enfoque conoce cuales son los servicios necesarios y como deberán ser utilizados para que los resultados finales a los usuarios sean satisfactorios. Hay varias características de SOA que permiten la mejora de los sistemas, lo cual llama la atención en el día de hoy, incluyendo al acoplamiento, agilidad, flexibilidad, autonomía, abstracción de la lógica de negocio y reducción de costos.

El principal propósito de la utilización de una arquitectura SOA es mejorar la comunicación empresarial para que de esa manera los objetivos de la empresa puedan ser realizados de manera más rápida. SOA proporciona una metodología y un marco de negocio para documentar las capacidades de negocio y a su vez, puede dar soporte a las actividades de integración y consolidación (Orantes, Gutierrez, & Lopez, 2009).

Los beneficios que pueden obtener las compañías que adoptan SOA son:

- Facilidad para evolucionar a los modelos basados en tercerización.
- Mejora en los tiempos de cambios en procesos
- Facilidad para integrar tecnologías disimiles.
- Facilidad para abordar modelos de negocios basado en colaboración.

2.21.2. Model View Controller (MVC)

En términos orientados a objetos, consistirá en el conjunto de clases que modelan y apoyan el problema subyacente, y que, por lo tanto, tenderán a ser

estables y duraderos como el problema en sí mismo (Deacon, 2009). Este patrón de diseño es muy útil para la arquitectura de softwares donde la idea clave es separar las interfaces de usuario de los datos que son representados por la interfaz de usuario (Gomezcoello Yopez, 2017). En MVC, la vista da la información al usuario y en conjunto al controlador que procesa lo que el usuario hace, comprendiéndose la interfase en la aplicación del usuario.

Este modelo es la sección de la aplicación donde se contiene la información representada por la vista tanto como la lógica, que altera la información respuesta a la interacción que tiene el usuario. La aplicación puede tener varios lenguajes o diferentes conjuntos de permisos. Los controles y el modelo se comunican con la vista vía eventos. Los eventos proveen un mecanismo de desacoplamiento que permite comunicación, con dependencia mínima (Deacon, 2009).

2.21.3. Model Driven Architecture (MDA)

MDA es una arquitectura que se emplea en la gestión de la variabilidad, que facilita el desarrollo y la evolución de las aplicaciones de manera dinámica a las aplicaciones multi-tenant. Está definida por el Object Management Group (OMG) que proporciona soluciones reutilizables que abordan el inicio y el fin de un desarrollo. El núcleo de la arquitectura se basa en los estándares de modelado de OMG: UML (Unified Modeling Language), el MOF (Meta-object Facility) y CWM (Common Warehouse Metamodel) (Soley, 2015).

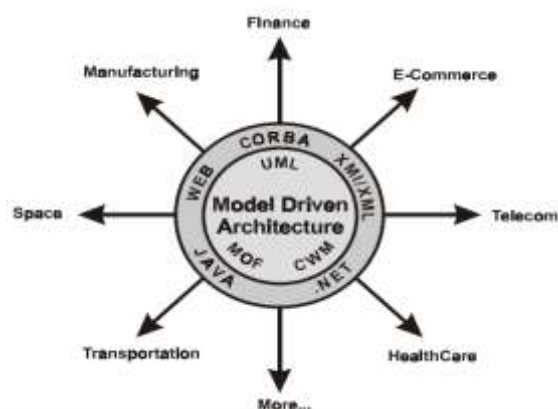


Figura 2.16: OMG's Model Driven Architecture.
Fuente: (Soley, 2015)

UML, siendo el lenguaje de modelado unificado, da la apertura que un modelo sea construido, desarrollado, visto y maniobrado de una manera estándar en el análisis y el tiempo de diseño. Los modelos UML dan la posibilidad a que un diseño de aplicación sea criticado después de una evaluación, antes de ser codificado, cuando es más fácil corregir y hacer cambios que después pueden salir más costosos de hacer (Soley, 2015). El modelo de CWM estandariza como representar modelos de bases de datos. transformaciones de esquemas, minería de datos, entre otros. El modelo MOF normaliza una fácil gestión de los modelos en un repositorio.

Independiente de la plataforma, las descripciones de las aplicaciones basadas en estos estándares de modelado se pueden implementar utilizando cualquier plataforma abierta o propietaria, incluyendo plataformas CORBA, Java, .NET, XMI / XML y Web (Gomezcoello Yopez, 2017), mediante transformaciones modelo a modelo o modelo a código.

2.22. Transformaciones de modelos

Existen cuatro pasos definidos por MDA para pasar del diseño de alto nivel a la realización de software.

- **Primer paso:** Modelo Independiente de Ordenador (CIM), que es una representación de la lógica del negocio desde un punto de vista independiente de la computación. Este modelo brinda una ayuda al momento de entender exactamente lo que el sistema va a realizar independientemente de cómo sea implementado.
- **Segundo paso:** Se arma un modelo del sistema de software que no depende de ninguna tecnología de implementación. A este tipo de modelo se lo conoce como Modelo Independiente de Plataforma (PIM).
- **Tercer paso:** Un PIM es transformado en uno o más Modelos Específicos de Plataforma (PSMs), con la implementación de una estrategia de correspondencia particular. Un PSM detalla un sistema que utiliza construcciones de implementación que se encuentran disponibles, por ejemplo, en una plataforma .NET, que es una tecnología de implementación específica.
- **Cuarto paso:** Se transforman en código los PSMs.

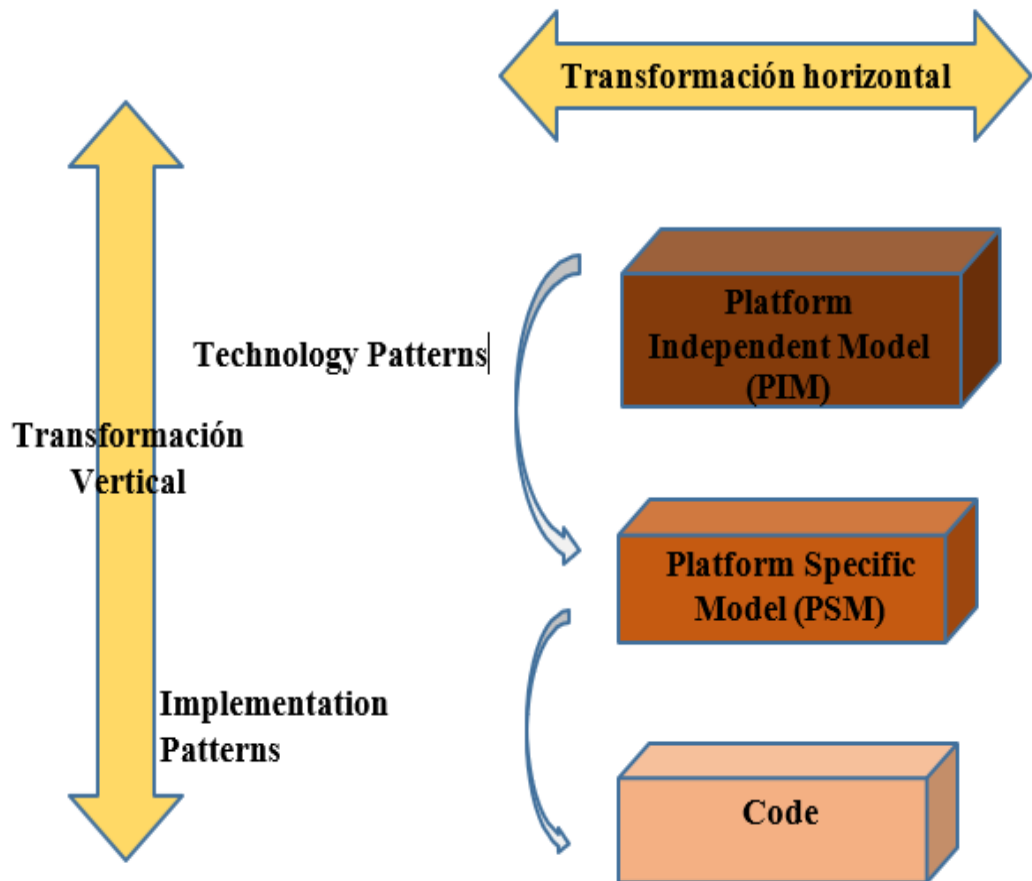


Figura 2.17: Enfoque de transformación MDA
Fuente: (Gomezcoello Yopez, 2017)

CAPÍTULO 3: RESULTADOS OBTENIDOS

3.1. Introducción

El presente trabajo se basa en el análisis y estudio de los factores, herramientas técnicas y requerimientos para alcanzar una optimización de los servicios de Cloud Computing por medio del sistema multi-tenant para las empresas proveedoras de estos servicios. Debido a esto se usó como base la característica cualitativa, de manera que los datos generados y resultados obtenidos no se considera como un ejemplo definitivo, sin embargo, si se puede utilizar como un medio valioso y confiable para la realización de trabajos o investigaciones a fines al tema.

Inicialmente se realizó una revisión bibliográfica de trabajos de Cloud Computing, multi-tenant y sus ramificaciones, realizados en varias partes del mundo, donde se fue recopilando la información para después ser analizada y esquematizada de manera que se llegue a una alternativa viable para los proveedores de los servicios ya anteriormente mencionados. Se busca dar una perspectiva de mejora en el Cloud Computing mediante diversas técnicas aplicadas de acuerdo a los mayores retos y problemas encontrados en este modelo, dando respuestas y soluciones tentativas con la finalidad de avanzar hacia una plataforma más fiable que está cambiando al mundo de las empresas.

3.2. Preguntas de Investigación

A continuación, se plantean las preguntas de investigación basada en los objetivos y motivaciones que este estudio tiene y propone.

- ¿Cuáles son las ventajas principales que se obtienen en la adopción de un modelo multi-tenant?
- ¿Cuáles son los problemas que se han encontrado en el modelo multi-tenant?
- ¿Cuáles son las herramientas y técnicas aplicables que dan soporte a un estable modelo multi-tenant?

A continuación, se procede a responder las interrogantes previamente definidas, cuyas respuestas están basadas en los artículos investigados, donde se ha propuesto algún tipo de enfoque o solución para dar soporte al modelo multi-tenant.

3.3. Proceso de búsqueda

Se requiere una estrategia de búsqueda formal para encontrar toda la población de artículos científicos que son relevantes para dar una respuesta a las preguntas de investigación que se mencionaron. La búsqueda consiste en definir el espacio de búsqueda tanto de bases de datos electrónicas como de actas de conferencias que se consideren claves para el objetivo de este trabajo, mismas identificadas en La Tabla 3.1. Las búsquedas se realizaron en bases de datos electrónicas científicas.

Tabla 3. 1: Fuentes de búsqueda

Fuente de datos	Documentación
Bases de datos electrónicas	Springer
	IEEE Xplore
	Science Direct
	Web of Science

Elaborado por: Autor.

3.4. Criterios de búsqueda

Los criterios de búsqueda establecidos tanto de exclusión como de inclusión tienen la finalidad de obtener los artículos más relevantes que nos permitan contestar las preguntas de la investigación (Tabla 3.2).

Tabla 3. 2: Criterios de inclusión y exclusión

Tipo	Criterio	Descripción
Inclusión	Tipo de estudio	Material que tenga el rigor científico necesario de acuerdo al tema
	Tiempo	Material publicado hasta julio del 2016
Exclusión	Solo se menciona la multitenencia	No se propone método, técnica o herramienta de soporte
	No abordan modelo multi-tenant	Artículos que proponen soluciones, pero sin aplicar el modelo multi-tenant
	Argumentos pobres	Artículos basados en opiniones con argumentos pobres

Elaborado por: Autor.

3.5. Extracción de datos

Tabla 3. 3: Caracterización de los artículos

Tema	Opción	Descripción
Atributo de calidad objeto de mejora	Propuestas de mejora	Atributos de calidad que han motivado plantear soporte al modelo multi-tenant.
Enfoque de optimización	Adopción de un nuevo paradigma de desarrollo	Técnicas para conseguir mejoras del modelo multi-tenant mediante la adopción de nuevas prácticas de desarrollo software.
	Optimización de la gestión y diseño de BBDD	Técnicas de diseño para optimizar el acceso a los datos o un gestionamiento más seguro de acceso a los datos.
	Estimación de recursos	Técnicas o algoritmos para estimar los recursos requeridos en las peticiones de los tenants.
	Versionamiento de software tiempo de ejecución.	Procedimientos para mejorar el versionamiento de software en tiempo de ejecución.
	Optimización de la gestión de recursos virtuales	Técnicas para conseguir mejoras del modelo multi-tenant mediante la gestión optimizada de los recursos virtuales
Soporte para la mejora	Estilo de arquitectura	Tipo de arquitectura que usan los estudios en el desarrollo de sus enfoques del cual se desprenden las técnicas usadas para la implementación de la solución planteada.
	Capa de implementación	Capa del sistema multi-tenant en la cual se implementa la solución planteada.
	Herramienta de soporte	Herramientas software para implementar las soluciones planteadas.
Método de Evaluación	Prueba matemática	Dados ciertos parámetros y mediante el uso de fórmulas matemáticas se puede predecir el comportamiento de la solución planteada.
	Caso de estudio	Aplicación de la solución planteada en ambientes reales.
	Experimento controlado	Prueba experimental de la solución aislando las variables de interés y midiendo sus resultados.
	Aplicación de ejemplo	Desarrollo de una aplicación de pequeña escala para probar la solución.
	Análisis teórico	Revisión detallada de los posibles beneficios de adoptar una solución en base a fundamentos teóricos.

Elaborado por: Autor.

En La Tabla 3.3 se presenta una caracterización de los artículos. Cada fila describe un ítem de clasificación, siendo que hay cuatro ítems que nos permiten clasificar los artículos en: atributos de calidad y objetivos de la mejora; enfoque de optimización que se refiere al mecanismo con el cual se pretende conseguir mejores prestación del servicio del modelo multi-tenant; soporte para la mejora, que es el mecanismo por el cual se implementa la mejora mediante herramientas de soporte; finalmente, método de evaluación, que se refiere al método mediante el cual se valida el enfoque y soporte a la multitenencia.

3.6. Artículos analizados

Tabla 3. 4: Artículos revisados

Numero	Título	Fuente	Referencia
M1	SaaS Dynamic Evolution Based on Model-Driven Software Product Lines	WOS	(Mohamed, 2014)
M2	Multitenant SaaS model of cloud computing: Issues and solutions	IEEE	(Saraswathi, 2014)
M3	Characterizing the performance of tenant data management in multi-tenant cloud authorization systems	IEEE	(Maenhaut, 2014)
M4	Continuous Evolution of Multi-tenant SaaS Applications: A Customizable Dynamic Adaptation Approach	IEEE	(Gey F. V., 2015)
M5	Multi-tenant Service Composition Based on Granularity Computing	IEEE	(Cai, 2014)
M6	Multi-tenant Services Monitoring for Accountability in Cloud Computing	IEEE	(Masmoudi, 2014)
M7	Resource Usage Control in Multi-tenant Applications	IEEE	(Krebs, 2014)
M8	Towards Software Product Lines Based Cloud Architectures	IEEE	(Matar, 2014)
M9	Adaptive Performance Isolation Middleware for Multi-tenant SaaS	IEEE	(Walraven S. D., 2015)
M10	Design and implementation of a Cloud SaaS framework for Multi-Tenant applications	IEEE	(Morakos, 2014)
M11	HLA-Based SaaS-Oriented Simulation Frameworks	IEEE	(Xiong, 2014)
M12	Middleware for Customizable Multi-staged Dynamic Upgrades of Multi-tenant SaaS Applications	IEEE	(Gey F. V., 2015)
M13	Modeling and Analysis of Availability for SaaS Multi-tenant Architecture	IEEE	(Su, 2014)
M14	A new architecture of server environment of SaaS system for easier migration and deployment	IEEE	(Shen, 2015)
M15	Adaptive Database Schema Design for Multi-Tenant Data Management	IEEE	(Ni, 2014)
M16	Classifying Requirements for Variability Optimization in Multitenant Applications	IEEE	(Saleh, 2014)
M17	Management of customizable Software-as-a-Service in cloud and network environments	IEEE	(Moens, 2015)

Elaborado por: Autor.

Cada uno de los artículos mencionados en La Tabla 3.4 propone algún enfoque o solución que da soporte al modelo multi-tenant. Los artículos fueron consultados en los canales científicos previamente citados.

3.7. Ventajas principales que se obtienen en la adopción de un modelo multi-tenant

En esta sección se recopila un listado de las ventajas o beneficios que se mencionan en los artículos, los cuales son considerados a la hora de adoptar el modelo multi-tenant.

Uso eficiente de los recursos

Al compartir servidores y la misma instancia de las aplicaciones, se pueden aprovechar de manera más eficiente los recursos de hardware y de software.

Ahorro

Cuando se comparte el mismo hardware y software entre los tenant, se produce un gran beneficio, ya que el proveedor reduce la compra de espacio físico y servidores para ubicar centros de datos, aparte de necesitar de menos desarrollo de software al compartir una instancia única entre los tenants, consiguiendo una reducción importante en los costos de prestación de servicios y así llegando a una economía a escala, donde los mayores beneficiados son los tenants que pueden conseguir el servicio a menor costo.

Mantenimiento

Al evitar monitorear y corregir múltiples aplicaciones que puedan surgir en la funcionalidad, los proveedores permiten un mantenimiento menos complicado. Las actualizaciones o en su lugar las correcciones de algún problema se las realiza a una sola instancia, de manera que se evita realizar la misma tarea por cada problema existente.

Flexibilidad

Las aplicaciones son capaces de adaptarse a los problemas o cambio que sean requeridos en el entorno con una mayor facilidad, debido a que las

correcciones o cambios en la instancia se lo realizan a todos los tenant, dando paso así a las aplicaciones multi-tenant flexibles.

Escalabilidad

En este modelo se tiene una mayor escalabilidad de las aplicaciones debido a que se pueden agregar funcionalidades compartidas para cada uno de los tenant en un solo versionamiento, dando así facilidades para que las aplicaciones puedan crecer más rápidamente.

Interoperabilidad

Se puede tener un cambio de información de los servicios de manera interna para el uso de las aplicaciones multi-tenant sin necesidad de desarrollar múltiples servicios para un mismo fin (Gomezcoello Yopez, 2017).

En La Tabla 3.5 se muestran agrupados los artículos analizados de acuerdo a las motivaciones que presentan a la hora de adoptar el modelo multi-tenant.

Tabla 3. 5: Motivaciones presentadas en los artículos

Motivación	Artículos
Ahorro	M2, M3, M4, M6, M7, M9, M10, M12, M14, M15, M16, M17
Uso eficiente de recursos	M1, M2, M3, M4, M9, M12, M13
Mantenimiento	M2, M5, M13
Escalabilidad	M3, M8, M11
Flexibilidad	M8, M17
Interoperabilidad	M5

Elaborado por: Autor.

En La Figura 3.1 se muestra el porcentaje de distribución de las motivaciones que se tiene principalmente para adoptar el modelo multi-tenant descritos en los artículos investigados, siendo el ahorro por mucho el primer motivo por el cual se adopta este modelo con un 50%. En segundo lugar, está el uso eficiente de los recursos con un 19% como la segunda mayor motivación para adoptar el modelo. Se tiene a la estabilidad con casi un 10% en los motivos para adoptar el modelo. Sigue el mantenimiento con un 8,66%

como cuarta motivación, La flexibilidad con un 6,67% sigue con el quinto lugar y el sexto motivo es la interoperabilidad con un 3,33%.

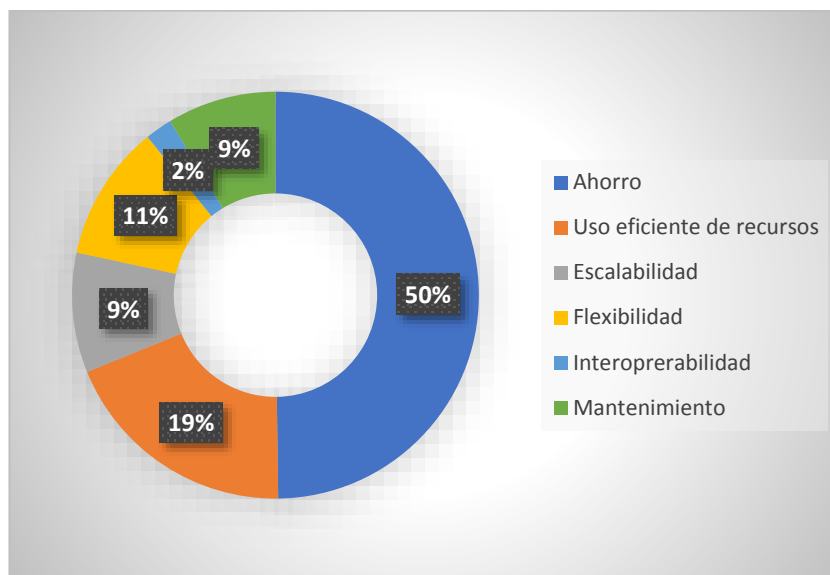


Figura 3. 1: Ventajas del modelo multi-tenant.
Elaborado por: Autor.

3.8. Problemas que se han encontrado en el modelo multi-tenant

A pesar de que los beneficios son claros en el modelo multi-tenant, los cuales a su vez motivación a su adopción, no se encuentra excepto de problemas. Estos problemas han sido identificados en los artículos analizados y se detallan a continuación.

Escalabilidad

Este es un problema que se encuentra presente en las aplicaciones multi-tenant debido a que, al poseer servicios compartidos, las correcciones o mejoras que se realicen en cualquiera de ellos, va a afectar directamente a la funcionalidad del resto de tenants, impidiendo que el modelo pueda escalar o evolucionar fácilmente, esto genera más bien que las aplicaciones permanezcan estáticas.

En la investigación, los artículos M1 Y M8 coinciden en la necesidad de llevar un manejo de la variabilidad en aplicaciones multi-tenant para poder optimizar la escalabilidad, dado que al escalar a una nueva instancia con nuevos tenants, se va a necesitar una evolución en la instancia en ejecución, dando soporte a la personalización de cada uno de los tenants. En los

artículos M4 y M12 se analiza al escalado mediante el versionamiento gradual de nuevos servicios con el fin de unir los cambios de cada tenant poco a poco. Se tiene como objetivo final evitar una interrupción del servicio.

Poca configurabilidad

La poca cantidad de opciones que poseen los tenants al momento de configurar las aplicaciones según los requerimientos de negocio de cada tenant, es otro problema grande que presenta este modelo. Al tener muchas opciones requiere de más desarrollo para los proveedores, los que significa un coste superior a los tenants, cosa que contradice y anula a la principal ventaja vista anteriormente en el modelo multi-tenant que es el ahorro.

Si se enfatiza en este problema, de acuerdo a los artículos M1 y M8 se ve la necesidad de que las aplicaciones multi-tenant puedan exhibir varios niveles de variabilidad (arquitecturas, nivel de requisitos, etc). Esto deriva en la necesidad de una gestión eficiente de la variabilidad por medio de los diferentes niveles, o incluso las diferentes vistas de un artefacto, como las diferentes vistas de la arquitectura anteriormente ya vistas.

Los niveles de variabilidad que pueden aplicarse se describen como: proceso de negocio, aplicación, infraestructura como servicio (IaaS), plataforma, despliegue de nube (comunitaria, híbrida, privada, pública) y de los proveedores (Google, Salesforce, Amazon, Microsoft, entre otros). Esta complejidad es la que hace inviable una gestión manual de la variabilidad en las aplicaciones multi-tenant de Software como servicio (SaaS), por esto se proponen técnicas nuevas y avanzadas de ingeniería de comunicaciones y software.

Ahorro

A pesar de que el principal motivo para hacerse con los servicios de un modelo multi-tenant es el ahorro de recursos, se ha planteado una mejora aun mayor para los consumidores y proveedores. Existen algunos costos altos cuando se requiere una personalización más profunda de una instancia, por lo cual al mejorar algunos tributos como son la escalabilidad o la

configurabilidad estos pueden conllevar a un mayor ahorro. El artículo M10 ofrece una herramienta de desarrollo que permite a los tenants crear sus propias aplicaciones de forma ágil.

Se puede ofrecer a los tenants herramientas que les permitan crear sus propias aplicaciones de una forma veloz y eficaz, generando una disminución en los costes.

Disponibilidad

Un problema se genera al compartir recursos virtualizados, debido a que en ocasiones a horas de alto tráfico pueden generarse bloqueos de memoria cuando aún hay una alta concurrencia de peticiones por los tenants. Esto lleva a que se vea interrumpido el servicio, se incumplan los SLA, generando pérdida de clientes o a su vez, desconfianza en el servicio. Los artículos M6, M7 y M13 coinciden en la necesidad de monitorizar constantemente las peticiones de los tenants con la finalidad de estimar los recursos requeridos o detectar posibles peticiones incorrectas.

Seguridad

Los problemas relacionados con la privacidad que se da con la virtualización de los recursos y que al momento de ser parte de un modelo compartido como el multi-tenant se agudizan siendo de vital importancia. Por esto, los artículos que se enfocan en detectar las violaciones o falencias en la privacidad que se llevan a cabo en el transcurso de la ejecución de un servicio, buscan satisfacer la privacidad de los tenants y garantizarle tranquilidad en lo que respecta a sus datos.

No es sorpresa que este problema sea de los que más preocupan a los tenants, ya que sus datos están en la misma base de datos de todos los demás, lo que genera incertidumbre a los clientes y usuarios por un posible mal uso o bloqueo de sus datos. Las diferentes opciones para implementar en este modelo y generar una mayor seguridad se analiza en entorno a la autenticación, autorización y al no intrusismo de los datos ajenos.

Mejor rendimiento

Debido al alto tráfico o incorrecto uso de los recursos, el rendimiento de los servicios multi-tenant puede verse afectado, impidiendo que los tiempos de respuesta sean los esperados de acuerdo a los SLA. Esto se ha convertido en un problema de difícil gestión debido a que las capas de ejecución responsables de controlar el uso de los recursos, como lo es el sistema operativo, no tienen conocimiento muchas veces sobre las entidades definidas a nivel de aplicación, generando que no pueda distinguir entre tenants. En el artículo M3 se caracteriza el rendimiento de la gestión de datos y se han planteado soluciones para distribuir a los usuarios y a los datos por medio de las instancias de bases de datos, optimizando y estructurando de esta manera el almacenamiento de los datos de los tenants a verse expuestos a grandes entornos.

Flexibilidad

Este es otro problema presente en el modelo multi-tenant ya que las aplicaciones no se pueden adaptar a los cambios continuos que sufren en el entorno tecnológico de una manera fácil, debido a que un cambio útil para un tenant podría afectar a los demás tenants. En este caso, en el artículo M5 se enfatiza la necesidad de técnicas que den la posibilidad de construir aplicaciones por medio del ensamblado de componentes de forma ágil y eficiente. También se estudia técnicas avanzadas de composición de servicios para la construcción de aplicaciones de manera más flexible.

Mantenimiento

El mantenimiento es una de las partes más importantes al momento de mantener un servicio, ya que, si no existe un control adecuado para todas las versiones expuestas de los servicios, se vuelve un reto mejorar el control de los servicios. Se expone en el artículo M4 la dificultad de modificar o alterar una aplicación de SaaS en el modelo multi-tenant después de su desarrollo y despliegue, sin afectar al servicio y su continuidad ya que cualquier cambio podría afectar al SLA, comprometiendo a los tenants y a sus usuarios. Se advierte sobre el incremento de la complejidad que derivara de la gestión de múltiples versiones de un componente en existencia al mismo tiempo.

Interoperabilidad

En el intercambio de recursos en grandes sistemas, la interoperabilidad se vuelve un problema debido a la complejidad y la heterogeneidad. Muchos problemas del sistema pueden salir a la luz, incluyendo códigos incompatibles, escalabilidad débil, enlaces ineficientes, mala utilización, que puede afectar un sistema de simulación. En el artículo M11 se caracteriza la complejidad de la interoperabilidad por medio de simulación en varias plataformas basadas en servicios web. En este trabajo está un análisis de requisitos, diseño de marco de simulación, integración de modelos de datos, pruebas, generación de código, análisis de resultados, optimización y retroalimentación.

En La Figura 3.2 se pueden visualizar los problemas identificados por los estudios distribuidos por su porcentaje. Se expone que la escalabilidad es el principal problema detectado con un 18,42%. Sigue con un 16% la disponibilidad de las aplicaciones junto a las aplicaciones poco configurables con el mismo porcentaje. También con un 16% está el mejor rendimiento de las aplicaciones sienta todos estos los problemas más importantes detectados en los ambientes multi-tenant.

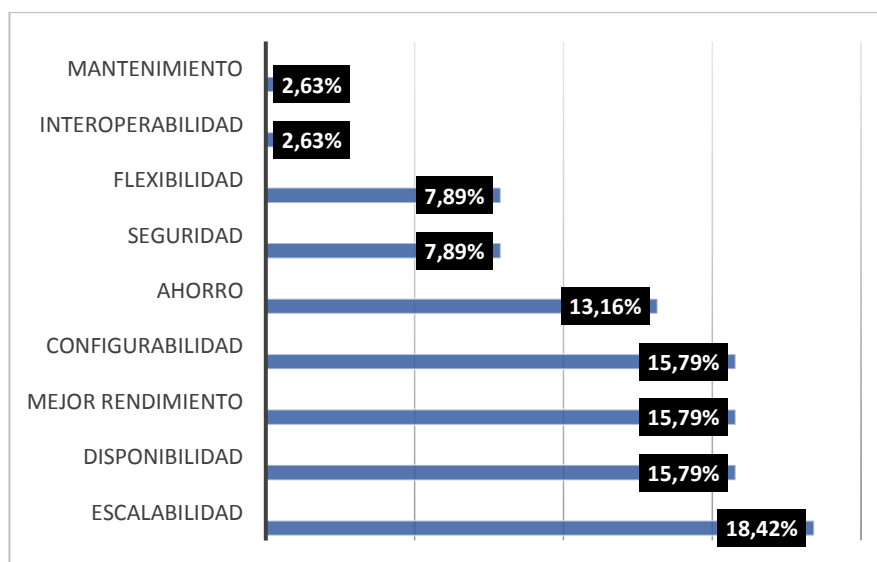


Figura 3. 2: Problemas presentes en el modelo multi-tenant
Elaborado por: Autor.

En el siguiente bloque de problemas está el ahorro, ocupando el quinto lugar con un 13,16% a pesar de ser el mayor motivo de la adopción de este sistema. En el sexto lugar esta con casi 8% la seguridad y la flexibilidad

compartiendo el porcentaje de problemas presentados en el modelo. Finalmente, está la interoperabilidad con un 2,63% en el noveno lugar y en último, pero con el mismo porcentaje se encuentra el mantenimiento.

3.9. Herramientas y técnicas aplicables que dan soporte a un estable modelo multi-tenant

La respuesta a esta interrogante está ligada con respuestas que se basan en el tipo de estilos arquitectónico, capas de implementación para la solución, herramientas de soporte y los diferentes enfoques de orientación descritos en los artículos analizados según La Tabla 3.6.

En los artículos revisados se ha encontrado enfoques viables para la solución de algunos problemas presentes en el modelo multi-tenant, los cuales ya han sido detallados anteriormente, donde gracias al análisis de cada uno se ha podido encaminar hacia alguna solución la cual tiene como único objetivo mitigar los problemas presentes en el modelo. En La Tabla 3.7 se encuentran clasificados los artículos de acuerdo al problema que los ha llevado a plantear una solución.

Tabla 3. 6: Clasificación de los artículos de acuerdo al problema

Objetivo del artículo	Artículos
Escalabilidad	M1, M3, M4, M5, M8, M12, M15
Disponibilidad	M3, M6, M7, M10, M13, M15
Altamente configurable	M1, M5, M8, M11, M16, M17
Mejor rendimiento	M2, M4, M7, M9, M11, M15
Ahorro	M4, M8, M10, M14, M17
Flexibilidad	M2, M9, M14
Seguridad	M2, M3, M7
Interoperabilidad	M11
Mantenimiento	M5

Elaborado por: Autor.

Las soluciones van desde técnicas de desarrollo avanzadas, como lo es MDA, ya revisada anteriormente, hasta técnicas de desarrollo propuestas que buscan la reutilización de servicios mediante un manejo de la composición y de la variabilidad de servicios existentes.

Tabla 3. 7: Clasificación de artículos por soporte que brindan.

ID	Tipo de Enfoque	Estilo arquitectónico	Capa de Implementación	Herramienta de soporte
M1	Paradigma de Desarrollo	Orientado a servicios	Middleware	
M2	Optimización de la gestión y diseño de BBDD	Estilo agnóstico	BBDD	Kerberos
M3	Optimización de la gestión y diseño de BBDD	Estilo agnóstico	BBDD	PUMA
M4	Versionamiento de software	Orientado a servicios	Middleware	
M5	Paradigma de Desarrollo	Orientado a servicios	Middleware	
M6	Estimación de recursos	Orientado a aspectos	Middleware	MT2MAC
M7	Estimación de recursos	Estilo agnóstico	Middleware	
M8	Paradigma de Desarrollo	Orientado a servicios	Middleware	OSGi
M9	Estimación de recursos	Estilo agnóstico	Middleware	
M10	Paradigma de Desarrollo	Modelo-Vista-Controlador	SaaS	Web IDE
M11	Optimización de la gestión de recursos virtuales	Orientado a servicios	Todas	
M12	Versionamiento de software	Orientado a servicios	Middleware	OSGi
M13	Estimación de recursos	Estilo agnóstico	Middleware	
M14	Optimización de la gestión de recursos virtuales	Estilo agnóstico	Todas	
M15	Optimización de la gestión y diseño de BBDD	Estilo agnóstico	BBDD	
M16	Paradigma de Desarrollo	Orientado a servicios	Middleware	
M17	Optimización de la gestión de recursos virtuales	Orientado a servicios	Todas	

Elaborado por: Autor.

3.10. Tipos de enfoque de optimización

De acuerdo a lo revisado en los diferentes artículos sobre el modelo multi-tenant, se describe a continuación los enfoques de desarrollo, que se basan en los enfoques de paradigmas de desarrollo, detalles de gestión y

optimización de las bases de datos, enfoques de estimación de recursos, versionamiento de software y la revisión de técnicas que optimización a los recursos virtuales.

Paradigma de Desarrollo

El artículo M1 propone el uso de un conjunto de reglas de validación de coherencia que se aplican en las transformaciones para garantizar la coherencia de los modelos generados. Una arquitectura basada en modelos (MDA) permite capturar los requisitos del sistema utilizando un modelo independiente de plataforma (PIM). Luego, integra las especificaciones de la plataforma a un PIM para generar un modelo único de plataforma funcional (PSM) para todos los tenants. Este enfoque permite obtener de forma dinámica una única instancia compartida para múltiples tenants y guía en el proceso de escalabilidad utilizando un motor basado en reglas para preservar la consistencia de la instancia evolucionada.

El artículo M5 usa un enfoque conocido como SGS (*service granularity space*) que aplicado en un middleware permite colocar los servicios desordenados en jerarquías y se ordenan los servicios agrupándolos, además hace que sea fácil y eficiente desarrollar aplicaciones multi-tenant combinando la personalización y la escalabilidad de acuerdo a los requisitos de los tenants. En pocas palabras, SGS puede ser visto como un mediador que contiene la información acerca de servicios de sus miembros y satisface a los tenants de acuerdo a la composición de los servicios configurados.

El artículo M10 presenta un framework que ofrece servicios que facilitan el proceso de desarrollo y gestión de las aplicaciones multi-tenant. Esencialmente, el objetivo de este artículo es destacar una característica importante de la nube como la composición de servicios. Este artículo propone una herramienta *Web IDE* que permite a los desarrolladores construir servicios y aplicaciones en la parte superior del framework SaaS a

través del suministro de bibliotecas apropiadas que implementan la arquitectura MVC para construir aplicaciones y servicios rápidamente.

El artículo M16 se centra en cómo construir una arquitectura de sistemas multi-tenant altamente configurable para soportar diferentes arquitecturas de organización y que cumpla múltiples objetivos del sistema, con diferentes atributos de objeto y estilos de presentación de datos. Además, proporciona pautas clave a los arquitectos y desarrolladores de software para implementar una capa de configuración en una arquitectura de multitenencia basada en el manejo de la variabilidad.

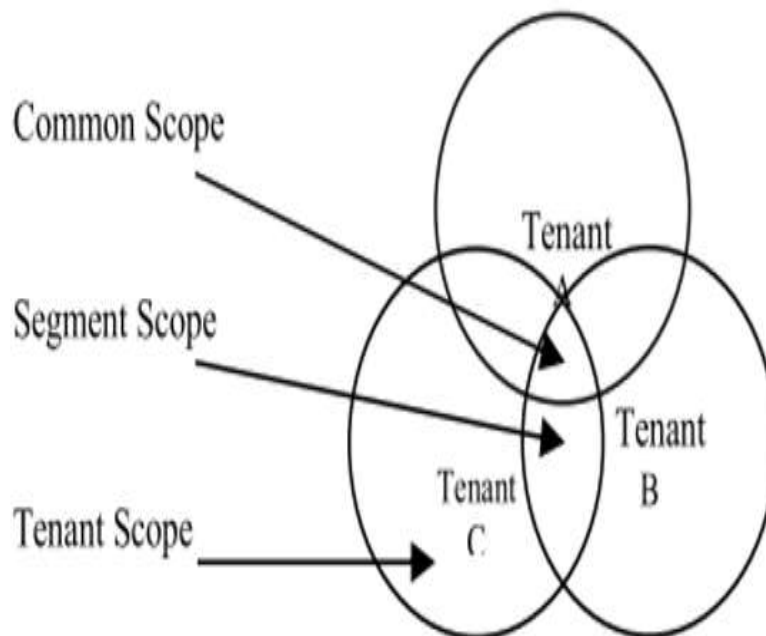



Figura 3. 3: Organización de datos independientes.
Fuente: (Saleh, 2015)

Optimización del diseño y la gestión de las bases de datos.

El artículo M2 propone una gestión del esquema de base de datos por medio de tablas que contienen metadatos, que dan la posibilidad de asociar fácilmente los registros y a su vez permite crear nuevas tablas. Además, se propone el uso del protocolo Kerberos, que carga los datos en formato XML para consumir menos espacio y de esta manera mejorar la seguridad de los datos controlando la autenticación y la autorización.

Tabla 3. 8: Esquema de almacenamiento XML

Tenant_id	Record_id	Xml_ext	Table_id
546	6732		1
546	5124		2
548	2431		5
548	7654		4



```
<status> gold </status>
<date> 2011-06-25 </date>
```

Fuente: (Saraswathi, 2014)

El artículo M15 propone rediseñar los esquemas de las bases de datos para que sean accesibles de forma fácil sin mermar el rendimiento por medio de la creación de tablas compartidas aplicadas en los campos más importantes que normalmente están presentes en la mayoría de tenants y a su vez, la creación de tablas suplementarias para los campos que no suelen compartirse, los menos comunes.

El artículo M3 propone el almacenamiento de los datos de forma jerárquica, iniciando su búsqueda de abajo hacia arriba, para así evitar que se formen los cuellos de botella en el repositorio principal y tener la información de forma asilada para la mejora de la seguridad.

Estimación de recursos

Mediante un conjunto de estudios se busca la mejora del servicio de las aplicaciones multi-tenant resolviendo la disponibilidad. Los artículos M6, M7, M9 y M13 proponen mediante el uso de algoritmos matemáticos o predictivos poder estimar los recursos requeridos para atender las solicitudes de los tenants, para alcanzar un mejor rendimiento y la disponibilidad de los diferentes servicios, todo esto por medio de los algoritmos para calcular los recursos.

En el artículo M6 se propone el uso de los algoritmos de regresión lineal o el filtro de Kalman para calcular el consumo dinámicamente en tiempo de

ejecución, para ver si existe alguna petición que sobrepase los recursos de su SLA.

Otra propuesta dada por el artículo M9 es la de dar un peso a las peticiones realizadas, donde el criterio será la importancia del SLA en cada tenant y así se podrá despachar las tareas que se consideren más importantes en primer lugar.

El uso de las cadenas de Márkov

para la estimación de la disponibilidad del servicio también es otra solución propuesta en el artículo M13, donde se basa la disponibilidad de acuerdo a la cantidad de tenants y el tiempo de respuesta que tendrá la base de datos en sus transacciones.

Versionamiento de software

En los artículos M4 y M12 se propone el versionamiento de software por medio de la aplicación gradual de los servicios nuevos que permitan una mejor escalabilidad del software sin tener que detener todos los procesos a la vez para hacer algún versionamiento. Esto se logra con la activación de versiones nuevas o corregidas de los servicios de forma gradual, aplicando los cambios a un tenant por activación, mientras coexiste con la versión actual. Así se puede validar que la funcionalidad del nuevo servicio sea correcta mientras poco a poco los demás tenants se acoplan al nuevo servicio.

En el artículo A4, se da un enfoque de adaptación dinámico en el que aborda requisitos clave como los son el soporte para la continuidad del servicio, alta automatización del proceso de actualización, aislamiento de la actualización del tenant, control de interesados. En este enfoque se maneja la dificultad de varias versiones coexistentes de componentes con la finalidad de aprovechar la mayor continuidad de servicio para cada tenant a lo largo de la evolución, mientras que las nuevas versiones ya prestan servicio a los demás tenants.

Mejora en la gestión de recursos virtuales.

En la computación en la nube es cosa normal el uso de recursos virtuales. En el modelo de multitenencia, aparte de las instancias de aplicaciones, se comparten a la vez recursos virtuales cuya buena gestión da la posibilidad de mejorar la calidad del modelo en las prestaciones.

En el artículo M14 se propone, el uso de la técnica de virtualización basada en contenedores, la cual consiste en el uso de máquinas virtuales que utilizan un kernel compartido haciéndolas más ligeras, lo cual permite tener entornos multi-tenant más flexibles a cambios, alteraciones o migraciones ya que los recursos son más ligeros y más fáciles de reemplazarlos.

También se propone el uso de redes virtuales de función (NFV) en el artículo M17 para distribuir las solicitudes de la aplicación en múltiples redes de necesitarse mayores tasas de transferencia de datos.

3.11. Soporte para la implementación del modelo multi-tenant

En esta sección se describe de forma detallada el soporte a la implementación del modelo de multitenencia, describiendo estilos arquitectónicos y posteriormente la revisión de la capa de implementación de las soluciones anteriormente propuestas. Finalmente se describen las herramientas de soporte de las cuales se revisó en los estudios primarios de este trabajo de titulación.

3.11.1. Estilo Arquitectónico

Los estudios primarios usan un estilo arquitectónico el cual muestra el tipo de arquitectura usada y sus características para la implementación de su modelo multi-tenant. Se procede a analizar los diferentes estilos arquitectónicos

Estilo agnóstico

Este tipo de estilo establece que no existe un estilo prefijado a seguir. Por esto, la solución planteada puede ser desarrollada con el estilo que el proveedor crea más ideal para implementar una determinada solución o a su

vez son soluciones que no utilizan un estilo de desarrollo arquitectónico. Para la clasificación de los estudios primarios, en esta opción se han englobado aquellos estudios que proponen el rediseño de la base de datos y la optimización de recursos virtuales que no requieren de un estilo arquitectónico en concreto para su aplicación (M2, M3, M7, M9, M13, M14, M15).

Arquitectura orientada a servicios

Este estilo es el más usado por los artículos revisados (M1, M4, M5, M8, M11, M12, M16, M17) como lo mejor para una optimización del modelo multi-tenant. Esta arquitectura da la posibilidad de exponer funciones o servicios reutilizables, donde sus principales ventajas son la interoperabilidad, dada su facilidad al momento del intercambio de información entre servicios y la escalabilidad, ya que pueden adaptar nuevos servicios conforme sean los requisitos de los tenants. Estas ventajas han sido de gran soporte para los estudios ya que está presente principalmente con la composición de servicios, la misma que se realiza mediante una transformación modelo a modelo que acepta el modelo independiente de la plataforma (PIM) del core, como entrada y la configuración del tenant; esto genera un PIM específico del tenant como salida. Mas adelante se fusiona con el PIM multi-tenant para acoplar los nuevos requisitos teniendo finalmente una única instancia compartida para todos los tenants. Esta ventaja la utilizan para sus enfoques los estudios (M1, M5, M8, M11).

Al tener muchos servicios reutilizables, se pueden ofrecer más características de servicios que puedan usar los tenants, consiguiendo una alta capacidad de configuración más pegada a los requerimientos y solicitudes de ellos.

Otra de las ventajas que presenta la arquitectura SOA es el bajo acoplamiento, lo que da la posibilidad de tener servicios independientes que no afectan directamente, en caso de ser actualizados o cambiados, a otros servicios, no obstante en el modelo multi-tenant, el cambio de un servicio puede provocar una interrupción de otro servicio, por esto se han presentado los diferentes enfoques (M4, M12) que permiten un cambio gradual al mismo

tiempo que coexiste con la versión actual del servicio y con la nueva versión, con el único objetivo de que no hayan interrupciones.

Modelo Vista Controlador

El modelo vista controlador (MVC) es un patrón que tiene como característica separar los datos y la lógica de negocio de una aplicación. El módulo es encargado de gestionar eventos, interfaz de usuario y las comunicaciones, con el fin de convertir una aplicación en un paquete modular fácil de mantener. Aparte, el patrón MVC mejora la rapidez de desarrollo. Lo que se ha propuesto en el artículo M10 para explotar esta característica, es el uso de un framework para el desarrollo de aplicaciones multi-tenant al usar las ventajas del modelo de vista controlador. Esto principalmente aprovecha su modularidad, lo que permite desarrollar o cambiar módulos de manera separada sin afectar al resto de la aplicación. Cada tenant es capaz de configurar una instancia del modelo concorde a los módulos desarrollados y así satisfacer las necesidades o intereses que cada uno presente.

Arquitectura orientada a aspectos

Este estilo de arquitectura orientada a aspectos busca dividir los conceptos que componen una aplicación en entidades, donde se elimina las dependencias entre los módulos, lo que da la posibilidad de tener módulos definidos claramente y más comprensibles. En artículos estudiados en este trabajo se usa este estilo para sacar las características de las solicitudes de los tenants con el uso de un middleware, entre el front del tenant y la capa SaaS. Este middleware monitorea el tiempo de ejecución al implementar la programación orientada a aspectos y hace uso de puntos de referencia para escoger un tipo de parámetros que se quiere monitorear; controla un conjunto de condiciones, usa consejos para envolver el código y decidir si en la traza de eventos observados existe una violación de la responsabilidad del servicio para poder ejecutar acciones apropiadas. El pointcut o punto de corte da facilidades para la selección de un evento único y la verificación de un conjunto de condiciones en el patrón.

En La Figura 3.4 se muestra la clasificación de estudios de estilos arquitectónicos y su utilización según la popularidad y aceptación de cada uno.

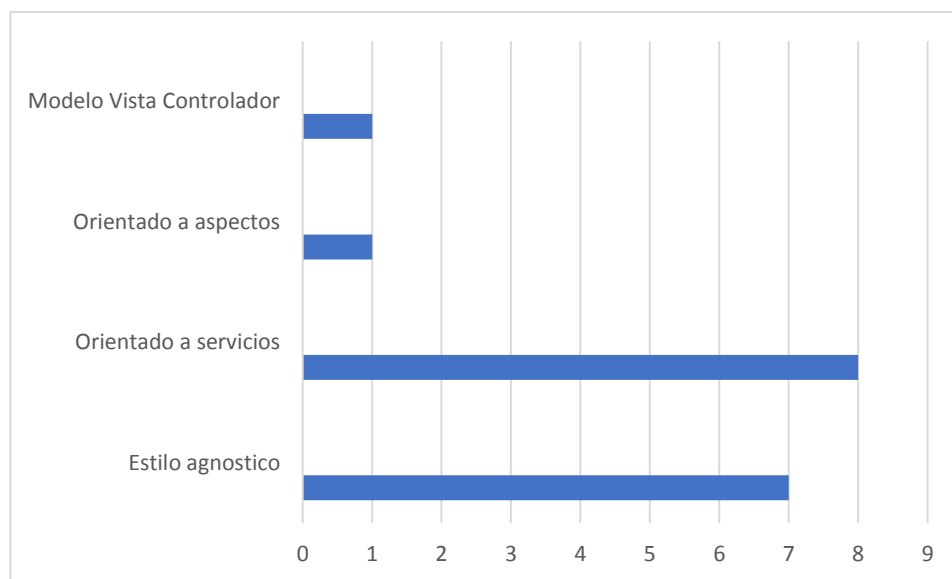


Figura 3. 4: Clasificación de estilos arquitectónicos.
Elaborado por: Autor.

3.11.2. Capa de implementación

En esta sección se analiza la capa con respecto al porcentaje de aplicación de los enfoques vistos. Esta capa es el lugar físico o virtual dentro del sistema donde se propone implementar algún enfoque en el modelo multi-tenant.

Según La Figura 3.5, de las soluciones analizadas, el 59% se da mediante la implementación de un middleware, siendo una capa entre la aplicación del tenant y los servicios mostrados para estas aplicaciones, donde se implementan estimación de recursos (M6, M7, M9, M13), paradigmas de desarrollo (M1, M5, M8, M16) o versionamiento de software (M4, M12). Otros enfoques (M2, M3, M15) que implementan soluciones en la base de datos abarcan el 17,65%. Distintos enfoques abarcan el 17,65% presentando mejoras más globales, por lo que tienen impacto en todas las capas de la arquitectura cloud, como es el caso de M11, M14 y M17 que requieren de recursos de hardware para la virtualización y simulación de los recursos para alcanzar la mejora propuesta. El 5,88% esta acaparado por el enfoque de implementar la solución en la capa de servicios expuestos a los tenants

(SaaS) propuesto en el artículo M10, ya que precisamente se trata de una aplicación a nivel de usuario para el desarrollo de aplicaciones multi-tenant.

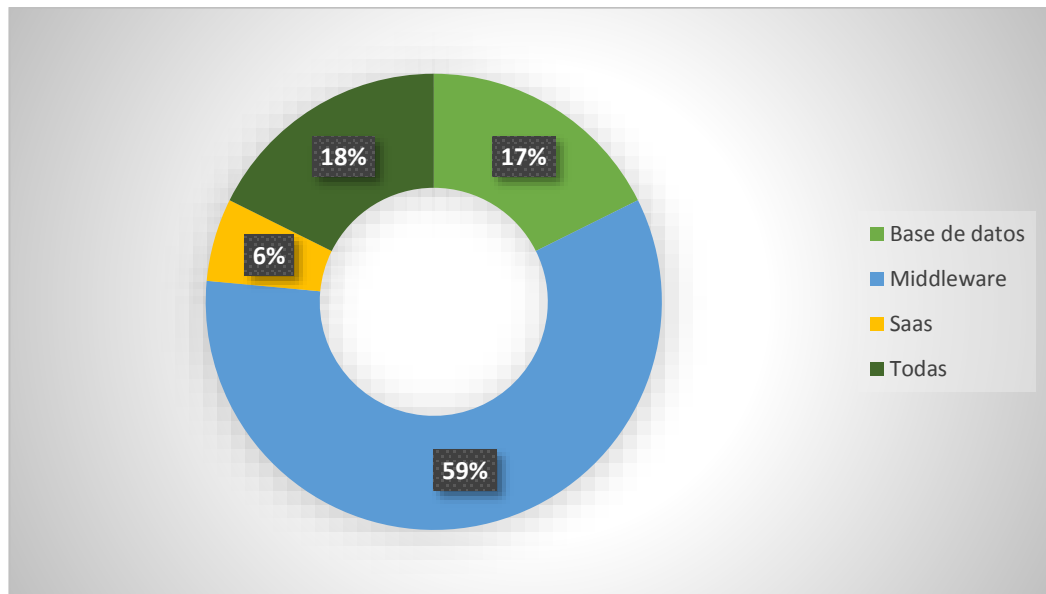


Figura 3. 5: Capa de implementación según enfoque.
Elaborado por: Autor.

En La Figura 3.6 se tiene agrupados a los atributos de calidad a optimizar por la capa donde se implementan. En la capa de implementación de base de datos se encuentra la seguridad (M2, M3) y la escalabilidad (M15), lo que demuestra que compartir los datos es algo que compromete directamente a la seguridad, por lo que una mejor gestión de la base de datos significara una mejoría en la seguridad.

En la capa de middleware, que agrupa la mayor cantidad de soluciones planteadas (M1, M4, M5, M8, M12), tiene como principal atributo de mejorar a la escalabilidad, seguido por la disponibilidad de las aplicaciones. También se busca mejorar el rendimiento (M7, M19) y la alta capacidad de configuración (M16).

En la capa de SaaS, se plantea una única solución (M10), donde se busca mejorar el ahorro para los tenants mediante el desarrollo de aplicaciones multi-tenant de forma ágil por medio del uso de un framework de desarrollo para las aplicaciones del modelo que provee una biblioteca de

servicios reutilizables y con un diseño MVC para desarrollos rápidos y eficientes.

Por último, se tiene la solución que abarca a todas las capas para tener aplicaciones altamente configurables, aparte de una mayor flexibilidad. Esto se da al momento que se optimizan los recursos virtuales.

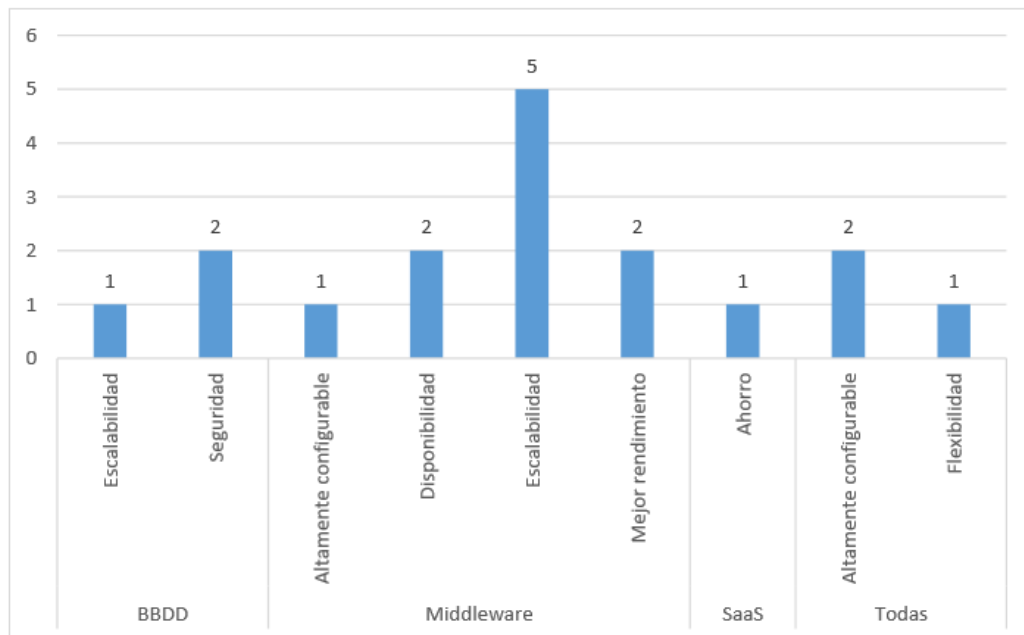


Figura 3. 6: Atributos de calidad por capa de implementación.
Elaborado por: Autor.

Se puede llegar a la conclusión de que, si se pretende mejorar la seguridad, la tendencia es que en la base de datos sea implementadas las soluciones y mejoras, mientras que cuando se requiere que el software adquiera más escalabilidad, las mejoras se realizan con implementaciones a través de un middleware.

A pesar de que no se puede llegar a una conclusión clara, se denota que, en SaaS, se dan soluciones para la mejora en el ahorro. Finalmente, cuando la implementación se realiza a nivel de todas las capas se busca tener un sistema de alta configurabilidad y flexible.

3.11.3. Herramientas para la implementación

En los estudios revisados, algunos de ellos presentan el uso de herramientas para poder implementar las soluciones, a continuación, se detallan las soluciones.

Kerberos

Esta es una herramienta usada en el artículo M2 con el fin de mejorar la seguridad de los datos. Creada por MIT (Instituto Tecnológico de Massachusetts), es un protocolo de seguridad que usa una criptografía de claves simétricas para validar usuarios con los servicios de red, dejando a un lado la necesidad de enviar contraseñas a través de la red, de esta manera al ser validados los usuarios de red por medio de Kerberos, los usuarios no autorizados se ven frustrados en sus intentos de interceptar contraseñas de red.

PUMA

En la propuesta de gestión de la seguridad en el artículo M3, con el fin de una mejor gestión de los datos, se propone el uso de PUMA. Esta es una herramienta que permite el desarrollo de soluciones de seguridad escalables para una mejor gestión y aplicación de los permisos de usuario para aplicaciones de SaaS.

MT2MAC

El MT2MAC (Multitenant Monitoring Middleware for Accountability in Cloud Computing) es un middleware para mejorar la disponibilidad de los servicios multi-tenant. Este middleware permite por medio de la monitorización en tiempo real de las peticiones de los tenants, identificar violaciones del SLA en las peticiones que puedan afectar a la disponibilidad de los servicios y al correcto funcionamiento.

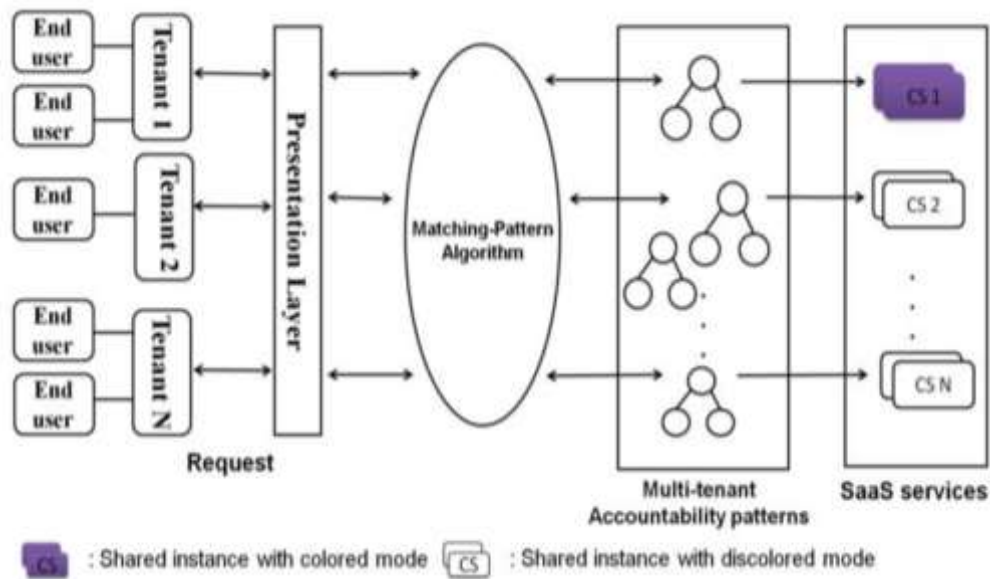


Figura 3. 7: Enfoque propuesto.
Fuente: (Masmoudi, 2015)

OSGi Framework

En búsqueda de una mejor escalabilidad, se propone en los artículos M8 y M12 el uso del framework OSGi, el cual es un middleware que da la posibilidad de soportar la activación configurable de las actualizaciones que tiene el software en el tiempo de ejecución, lo que consigue al aplicar internamente algunos pasos, con la finalidad de que, en segundo plano, las nuevas versiones se implementen mientras que las antiguas van quedando en desuso. OSGi framework tiene como objetivo principal que el servicio sea continuo y que no requiera de una interrupción a la hora de ejecutar las actualizaciones.

Web IDE

Este framework de desarrollo se propone en el artículo M10 para las aplicaciones multi-tenant, permitiendo a que los desarrolladores sean capaces de construir aplicaciones y servicios desde el lado del usuario, por medio del suministro de servicios reutilizables por el lado del usuario. El patrón MVC usa esta herramienta para construir aplicaciones de manera rápida y de forma modular.

3.12. Análisis de los resultados

3.12.1. Estado actual de las investigaciones del modelo multi-tenant

Este trabajo da a conocer los problemas a los que el modelo multi-tenant se enfrenta actualmente y los diferentes enfoques que pueden ayudar a solventar estos problemas. Las principales tendencias a la hora de dar soporte y ayudar a la evolución del modelo para cubrir las necesidades actuales son descritas a continuación.

Composición de servicios

La composición de los servicios ha sido una solución planteada recurrentemente por los estudios. Con el uso de técnicas SPL (Software Product Lines) se puede crear un software que comparta las características previamente dadas por el proveedor y que, mediante un proceso de selección de las características del tenant y mediante la transformación de modelos, permita la construcción de una única instancia compartida. Estas técnicas dan la posibilidad de componer nuevos servicios de una forma más dinámica, creando una nueva instancia mejorada, lo que permite mejorar la escalabilidad del SaaS para los entornos multi-tenant. Viéndolo desde el punto de vista económico, se ve beneficiado el proveedor de servicios porque está simplificando el proceso de desarrollo de una manera considerable y a su vez permite controlar inconsistencias en el proceso de configuración, abriendo así la posibilidad de realizar un fácil mantenimiento de las instancias generadas, lo que permite a los tenants contratar servicios con a menor costo.

Es seguro concluir en este punto que la composición de servicios es implementada por medio de un middleware que está encargado de establecer reglas de composición que relacionan los servicios de manera adecuada sin perder la consistencia, con el fin de tener una útil y evolucionada aplicación para todos los involucrados en su uso, o sea los tenants.

Uso de algoritmos predictivos

Tanto para los proveedores como para los consumidores, una preocupación constante es la de la disponibilidad de los servicios, ya que el alto tráfico y los cuellos de botella suelen presentarse en servicios

compartidos por una cantidad grande de tenants, lo que viola los SLA y produce un impacto negativo en el negocio, por lo que se han planteado soluciones para este problema.

Las soluciones propuestas son principalmente dadas por el uso de algoritmos predictivos que dan la posibilidad de estimar la demanda de recursos requeridos y de esta forma habilitar los recursos necesarios en los casos que se requiera, donde existe una priorización en casos de tenants con mayor SLA. Para lograr este cometido, se plantea la monitorización constante, por medio de un middleware, el cual que cada vez que un tenant realice una solicitud o petición, se aplicara un algoritmo para la toma de decisiones optimas y oportunas para así mantener una alta disponibilidad de los servicios a toda hora.

Bases de datos con esquemas híbridos

Los datos y su gestión es un tema que genera incertidumbre para los tenants dado que las aplicaciones multi-tenant principalmente se comparten una misma base de datos para el almacenamiento de información de todos los tenants. Para eliminar este problema de inseguridad en los usuarios se ha presentado varias soluciones, donde principalmente se basan en el uso de herramientas de terceros para generar un control en la autorización y autenticación de las transacciones de los usuarios. No obstante, el mayor aporte que realizan estos estudios va de la mano con el diseño de bases de datos híbridos que buscan mejorar la seguridad y el rendimiento.

En este trabajo se describen las ventajas y desventajas de cada modelo de base de datos. Los enfoques de las bases de datos híbridos buscan usar las ventajas de cada diseño, dado que plantean compartir datos comunes en tablas compartidas, mientras que a los datos menos comunes o más relevantes se los almacenan en tablas únicas para cada tenant, dando así una mejora en la seguridad mediante el aislamiento de información sensible, manteniendo el rendimiento ya que los datos menos sensibles se encuentran con fácil acceso en una tabla compartida. Los enfoques híbridos han sido

diseñados para que sean de fácil escalabilidad, así facilitando la creación de campos personalizados para cada tenant.

Virtualización de los recursos

La virtualización de los recursos tiene varios fines, como la simulación de entornos multi-tenant, composición de servicios de red o facilitar las migraciones de proveedor. Esta virtualización se puede tratar de una forma global, ya que da la posibilidad de mejorar las prestaciones de Cloud Computing y particularmente de las aplicaciones multi-tenant, dado que con virtualización no solo se componen servicios de software, sino que también de hardware compartido. Aparte, el que estos servicios no estén ligados a un entorno físico, genera la posibilidad de cambiar de proveedor migrando solo los recursos virtuales, que son ligeros, a el nuevo repositorio. De esta manera se ha planteado la virtualización de recursos para la simulación de entornos multi-tenant con el uso de librerías de componentes virtuales, que autoricen la prueba de forma más real de los nuevos cambios o desarrollos de los servicios en el entorno de multitenencia.

3.12.2. Tendencias a futuro en el modelo multi-tenant

A pesar de haber ya analizado y planteado varios enfoques para generar unas soluciones a los problemas que presente el modelo actualmente, estos al mismo tiempo dejan entrever algunos temas que al plantear algún enfoque, podrían producirse, aparte de una vista a ciertas tendencias que en el futuro podrían presentarse como mejores soluciones a los enfoques propuestos actualmente.

Aplicaciones altamente escalables y configurables

En este trabajo se presentaron posibles soluciones a sistemas poco configurables y escalables, donde se busca una mejora grande. Para conseguir estos objetivos, la principal tendencia es el desarrollo de middlewares que estén basados en SPL, de esta manera permiten correlacionar servicios y generan una mejora en la escalabilidad de las aplicaciones mediante un estudio que muestre la variabilidad que tengan.

A pesar de estas soluciones propuestas, poco se analizó a los problemas que puede generar una plataforma demasiado configurable, donde pueden darse casos de explosión combinatoria, que den como resultado una ola de problemas relacionados con el rendimiento, aparte que tener muchas soluciones a estos problemas puede generar un gasto de manteniendo bastante elevado para el proveedor. Estos problemas deben investigarse más a detalle y probarse en entornos reales con varios casos con muchos tenants y una alta variabilidad, donde se sabrá mejor que tanta es su utilidad.

Seguridad

Dado que todos los datos son alojados en un mismo repositorio virtual o físico de un mismo proveedor, el tema de la seguridad es una de las mayores preocupaciones en el entorno de la computación en la nube. Debido a la compartición de tablas de bases de datos con otros tenants, se ve más acentuada esta preocupación y el beneficio del ahorro puede verse opacado con la incertidumbre de la seguridad que los datos puedan tener, por esto los estudios han desarrollado soluciones enfocadas en el aislamiento de los datos para la seguridad de cada tenant, no obstante, se ha logrado con el uso de herramientas de terceros, pero no se han dado investigaciones específicas con orientación a los servicios multi-tenant, lo que podría ser de gran ayuda para una futura eliminación de la duda que hay con respecto a la seguridad.

Inteligencia artificial

El uso de la IA es bastante frecuente en los sistemas informáticos, donde en los últimos años ha conseguido posicionarse como la herramienta del futuro, siendo curioso que no se esté investigando este tipo de soluciones en el modelo multi-tenant. El uso de la IA dejaría a un lado el uso de la dependencia de algoritmos básicos para de esta manera generar una estimación de recursos más inteligentes, de acuerdo a varios factores que no se pueden lograr con algoritmos simples.

Las redes neuronales también podrían servir para predecir el tráfico de la red de una mejor forma, problemas y requerimiento de hardware y software en el futuro para los tenants. La IA está cada vez más presente en todas las

áreas de la computación por lo que debería también ser investigada para en el futuro dar soporte al modelo multi-tenant.

Alta demanda de recursos

Es un hecho que la prestación de servicios del modelo multi-tenant beneficia tanto a los proveedores por el uso eficiente de sus recursos como a los consumidores, debido al ahorro que consiguen al ser usuarios del modelo, por lo que se proyecta que en los próximos años tenga un crecimiento exponencial, donde el crecimiento va a ir de la mano con nuevos problemas al proveedor de no tener como afrontar la gran demanda que tendrá.

El desarrollo que debe realizar el proveedor con respecto a este servicio, es la preparación de los equipos y conexiones para afrontar un alto volumen de tráfico, de la mano con la implementación de un software de alta escalabilidad y eficiencia, donde la correcta implementación de las soluciones hará que sea más fácil el afrontar una alta demanda de tenants al mismo tiempo, que es la proyección del modelo a futuro. El paradigma de las funciones virtualizadas de red (NFV) apunta como un gran soporte para la gran demanda de servicios que se avecina.

La virtualización de funciones de red quizá sea el mayor reto que se presenta en la puerta de los proveedores multi-tenant, ya que su origen se da justamente para la solución de le problema de falta de escalabilidad del hardware para las redes. El virtualizar los servicios de red es un paradigma dentro del campo de la computación en la nube, que permitirá disponer de unas mayores tasas de transferencia y de esta manera ofrecer servicios a más tenants sin comprometer el rendimiento y la disponibilidad de las aplicaciones multi-tenant. La aplicación de NFV en el modelo multi-tenant, debe ser la investigación con mayor énfasis en los próximos años.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones

En el presente trabajo se llevó a cabo una revisión en el área de la computación en la nube, más específico, una revisión del modelo multi-tenant, que es el característico de los servicios de software y aplicaciones que son puestos en la nube. Tecnologías como la de la computación en la nube dan lugar a un mundo nuevo de posibilidades tanto para las personas, como para las empresas, en su trayecto de evolución y desarrollo; gozando así, a bajo costo y bajo recurso, un abanico de servicios que ahora son parte del mundo en la nube. Después de analizar los diferentes criterios de la información recopilada, en este análisis se puede concluir lo siguiente.

En primer lugar, se identificaron las principales razones por las que se adopta el modelo multi-tenant en las organizaciones por medio de comparativas para conocer los beneficios que brinda a las empresas proveedoras, donde se destaca como principal razón de su adopción al ahorro de coste que genera en los recursos, tanto virtuales como físicos.

En segundo lugar, fueron analizados los problemas presentes aun en el modelo multi-tenant, donde destacan entre los problemas principales la escalabilidad y la adaptabilidad, dado a que todo cambio, modificación o actualización en el modelo, afecta directamente a todos los tenants. La disponibilidad del servicio también es uno de los grandes problemas, ya que esta se ve afectada en horas de mucho tráfico. Estos problemas han sido estudiados y abordados con posibles soluciones como la implementación de middlewares que pretenden reducir o eliminar el impacto que tienen estos problemas en el modelo.

En tercer lugar, se han comparado varios enfoques de cara a estos problemas para abordar estos inconvenientes presentes en el modelo multi-tenant. Las soluciones propuestas están basadas principalmente en el uso de técnicas de ingeniería de software como Software Product Lines (SPL) y MDA, virtualización de componentes y algoritmos matemáticos para la estimación

de recursos. Varias de las técnicas mencionadas como los algoritmos predictivos son las de las soluciones propuestas, que van de la mano de herramientas que van a ayudar a la implementación y mantenimiento de las mismas.

Finalmente, se identificaron las tendencias de cara a futuras investigaciones con respecto al modelo multi-tenant, donde se vislumbra una mayor optimización y fluidez de este modelo que apunta a ser el futuro en las telecomunicaciones. Estas tendencias giran en torno a un manejo de peticiones en grandes cantidades, la gestión de una alta variabilidad de las aplicaciones multi-tenant y como se llevará el impacto generado en el mantenimiento de la misma, el posible uso de inteligencia artificial a medida que esta se vaya desarrollando más en el futuro, con una orientación a la estimación de recursos para una mejora en el manejo y preparación correcta de los recursos que soliciten los tenants.

4.2. Recomendaciones

- Sugerir a los proveedores probar las soluciones y enfoques expuestos en este trabajo para ver si satisfacen sus necesidades en diferentes ambientes y posteriormente ser implementados para un correcto funcionamiento del modelo mientras se brinda los servicios de Cloud Computing. Que este sea fluido y con una baja tasa de inestabilidad. Todas las soluciones deben ser simuladas como si se estuviese en un ambiente real para tener lo más cercano a las soluciones planteadas.
- Incentivar a los estudiantes a seguir la investigación en el campo de la computación en la nube, sus modelos y futuras soluciones, dado que se proyecta un crecimiento exponencial en la adopción de la computación en la nube a futuro, llegando así al mundo virtualizado del que tanto se espera, donde estos temas tratados, tanto el multi-tenant como el Cloud Computing serán los principales pilares de la era de la virtualización.

Bibliografía

- Ali, S. A. (18 de Julio de 2019). *What brings Cloud Computing*. Obtenido de <https://itsolutionsbox.com/cloud-computing-brings-to-you/>
- Amazon Web Services. (2019). *What is Amazon Web Services*. Obtenido de <https://aws.amazon.com/what-is-aws/>
- Amies, A., Feng Wu , C., Cai Wang, G., & Criveti, M. (29 de Octubre de 2012). *Gestión de redes en la nube*. Obtenido de <https://www.ibm.com/developerworks/ssa/cloud/library/cl-networkingtools/index.html>
- Anicas, M. (1 de Junio de 2015). *Building for Production: Web Applications — Backups*. Obtenido de <https://www.digitalocean.com/community/tutorials/building-for-production-web-applications-backups>
- Arora, R., & Parashar, A. (4 de Julio de 2013). *Secure User Data in Cloud Computing Using Encryption*. Obtenido de Semantic Scholar: <https://pdfs.semanticscholar.org/9799/a9f9bec6cf85715ca236035b5d89204b326a.pdf>
- Bang, H., & Kim, Y. (26 de Abril de 2018). *Introduction to Kalman Filter and Its Applications*. Obtenido de Intechopen: <https://www.intechopen.com/books/introduction-and-implementations-of-the-kalman-filter/introduction-to-kalman-filter-and-its-applications>
- Cai, H. (2014). Multi-tenant Service Composition Based on Granularity Computing. *IEEE International Conference* (págs. 669-676). IEEE.
- CISION. (15 de Mayo de 2019). *Datrium Unveils Automatrix Platform to Power Instant Outcomes in a Multicloud World*. Obtenido de <https://www.prnewswire.com/news-releases/datrium-unveils-automatrix-platform-to-power-instant-outcomes-in-a-multicloud-world-300850476.html>
- Cusumano, M. (Abril de 2010). Cloud Computing and SaaS as New Computing Platforms. *Technology Strategy and Management*, págs. 27-29. Obtenido de http://ebusiness.mit.edu/research/papers/2010.04_Cusumano_Technology%20Strategy%20and%20Management_273.pdf

- Daley, S. (19 de Septiembre de 2019). *Built In*. Obtenido de 18 Examples Of How Cloud Computing Services Keep The World At Our Fingertips: <https://builtin.com/cloud-computing/cloud-computing-examples>
- Dartmouth. (s.f.). Markov Chains. En Darmotuh.
- Deacon, J. (Mayo de 2009). *Model-View-Controller (MVC) Architecture*. Obtenido de <https://www.rareparts.com/pdf/MVC.pdf>
- El-Gazzar, R. F. (Febrero de 2014). *Springer Link*. Obtenido de A Literature Review on Cloud Computing Adoption Issues in Enterprises: https://link.springer.com/chapter/10.1007/978-3-662-43459-8_14#
- Essays. (Noviembre de 2013). *Cloud Computing In The Telecommunications Sector Information Technology Essay*. Obtenido de <https://www.uniassignment.com/essay-samples/information-technology/cloud-computing-in-the-telecommunications-sector-information-technology-essay.php?vref=1>
- Gartner. (19 de Agosto de 2019). *Cloud Adoption: Where Does Your Country Rank?* Obtenido de <https://www.gartner.com/smarterwithgartner/cloud-adoption-where-does-your-country-rank/>
- Georgios, S. L., & Karatza, H. (Junio de 2016). *Research Gate*. Obtenido de Scheduling Different Types of Applications in a SaaS Cloud: https://www.researchgate.net/profile/Georgios_Stavrinides/publication/300301927_Scheduling_Different_Types_of_Applications_in_a_SaaS_Cloud/links/57c2a02b08aeda1ec38f4cfa/Scheduling-Different-Types-of-Applications-in-a-SaaS-Cloud.pdf
- Gey, F. (2015). *Middleware for Customizable Multi-staged Dynamic Upgrades of Multi-tenant SaaS Applications*.
- Goasduff, L. (19 de Agosto de 2019). Obtenido de Cloud Adoption: Where Does Your Country Rank?: <https://www.gartner.com/smarterwithgartner/cloud-adoption-where-does-your-country-rank/>
- Gomezcoello Yepez, A. (2017). *Multitenencia Cloud: Una revision sistematica de la literatura*. Madrid: Universidad Politecnica de Madrid. Obtenido de http://oa.upm.es/44935/3/TFM_Adan_Gomezcoello_Yepez.pdf

- Goyal, A. (14 de Octubre de 2019). *The Amazing World of Amazon Web Services (AWS)*. Obtenido de <https://blog.bitsrc.io/the-amazing-world-of-amazon-web-services-aws-edc34614041a>
- Hayes, B. (1 de Julio de 2008). *Amazonaws*. Obtenido de https://s3.amazonaws.com/academia.edu.documents/30550235/news-cloud-computing.pdf?response-content-disposition=inline%3Bfilename%3DCloud_computing.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20191107%2Fus-east-1%2Fs3%2F
- IBM. (2017). *IBM*. Obtenido de ¿Que es la regresion lineal?: <https://www.ibm.com/es-es/analytics/learn/linear-regression>
- International Telecommunication Union. (2012). *Cloud computing benefits from Telecommunications and ICT Perspectives*. Telecommunications Stadarization Sector of ITU. Obtenido de https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-CLOUD-2012-P7-PDF-E.pdf
- Iosup, A., Yigitbasi, N., & Epema, D. (Enero de 2011). *On the Performance Variability of Production Cloud Services*. Obtenido de http://www.st.ewi.tudelft.nl/iosup/tech_rep/cloud-perf-var10tr.pdf
- Javed, D. (Junio de 2017). *Cloud Computing*. Obtenido de <https://www.scribd.com/presentation/213992517/09-Cloud-Computing>
- Khanduri, T. (15 de Septiembre de 2019). *What, Why and How Salesforce*. Obtenido de <https://salesforcegeek.in/what-why-and-how-salesforce/>
- Knorr, E., & Gruman, G. (Abril de 2008). *What Cloud Computing Really Means*. Obtenido de Sky Solutions: http://skysolutions.co.zw/docs/What_Cloud_Computing_Really_Means.pdf
- Krebs, R. (2014). Resource Usage Control In Multi-Tenant Applications. *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (págs. 122-131). Chicago: IEEE.
- Macias, N. (5 de Octubre de 2010). *Arquitecturas Multi-Tenant*. Obtenido de Gravatar: <https://gravitar.biz/tecnologia-negocios/arquitecturas-multi-tenant/>

- Maenhaut, P.-J. (9 de Mayo de 2014). *Characterizing the performance of tenant data management in multi-tenant cloud authorization systems*. Obtenido de <https://ieeexplore.ieee.org/document/6838232/authors#authors>
- Masmoudi, F. (12 de Febrero de 2015). *Multi-tenant Services Monitoring for Accountability in Cloud Computing*. Obtenido de <https://ieeexplore.ieee.org/document/7037725>
- Matar, M. A. (2014). Towards Software Product Lines Based Cloud Architectures. *IEEE International Conference on Cloud Engineering* (págs. 117-126). IEEE.
- Mell, P., & Grance, T. (Septiembre de 2011). *Winthrop*. Obtenido de <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>
- Microsoft. (3 de Junio de 2014). *Hosting a Multi-Tenant Application on Windows Azure*. Obtenido de [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh534478\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh534478(v=pandp.10)?redirectedfrom=MSDN)
- Mitchell, B. (20 de Diciembre de 2019). *What Is Cloud Computing?* Obtenido de <https://www.lifewire.com/what-is-cloud-computing-817770>
- Moens, H. (2016). Management of customizable Software-as-a-Service in cloud and network environments. *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. Istanbul: IEEE.
- Morakos, P. (2014). Design and implementation of a Cloud SaaS framework for Multi-Tenant applications. *5th International Conference on Information, Intelligence, Systems and Applications (IISA)* (págs. 273-278). IEEE.
- Orantes, S., Gutierrez, A., & Lopez, M. (2009). *Arquitecturas empresariales: gestión de procesos de negocio vs. arquitecturas orientadas a servicios ¿se relacionan?* Tecnura. Obtenido de <https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/6676/8259>
- Pohl, K., Bockle, G., & Van der Linden, F. (2005). *Software Product Line Engineering*. Alemania: Springer. Obtenido de https://books.google.com.ec/books?id=J4GqT4OUuSMC&printsec=frontcover&redir_esc=y#v=onepage&q&f=false

- Romero, P. A. (2005). *Arquitectura de Software, Esquemas y Servicios*. La Habana: Universidad Tecnológica de La Habana José Antonio Echeverría. Obtenido de <http://rii.cujae.edu.cu/index.php/revistaind/article/view/95/74>
- Saleh, A. I. (Febrero de 2015). *Classifying Requirements for Variability Optimization in Multitenant Applications*. Obtenido de https://www.researchgate.net/publication/283102549_Classifying_Requirements_for_Variability_Optimization_in_Multitenant_Applications
- Saraswathi, M. (19 de Diciembre de 2014). *Multitenant SaaS model of cloud computing: Issues and solutions*. Obtenido de <https://ieeexplore.ieee.org/document/7062719>
- Soley, R. (27 de Noviembre de 2000). *Object Management Group*. Obtenido de *Model Driven Architecture* : http://www.geocities.ws/pravin_suman/Resources/00-11-05.pdf
- Statistics Solutions. (2013). *What is Linear Regression?* Obtenido de <https://www.statisticssolutions.com/what-is-linear-regression/>
- Su, W. (2014). Modeling and Analysis of Availability for SaaS Multi-tenant Architecture. *2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE)* (págs. 365-369). IEEE.
- Swamy, A. (6 de Octubre de 2014). *Multi-Tenancy Design Consideration*. Obtenido de <https://www.codeproject.com/Articles/809851/Multi-Tenancy-Design-Consideration>
- Tecnología para tu Empresa. (19 de Enero de 2019). *Cada vez más empresas utilizan la modalidad cloud de pago por uso*. Obtenido de <https://tecnologiaparatuempresa.ituser.es/cloud/2019/01/cada-vez-mas-empresas-utilizan-la-modalidad-cloud-de-pago-por-uso>
- The Poughkeepsie Journal. (9 de Enero de 2014). *IBM's Watson supercomputer gets its own division*. Obtenido de <https://www.poughkeepsiejournal.com/story/news/2014/01/09/ibms-watson-supercomputer-gets-its-own-division/4387871/>
- Vilalta, R., Hellerstein, J., Weiss, S., Ma, S., & Apte, C. (2002). Predictive algorithms in the management of computer systems. En Vilalta, *IBM System Journal* (págs. 461-474). Obtenido de <https://www.uh.edu/~rvilalta/papers/ibm02.pdf>

- Walraven, S. (2016). Adaptive Performance Isolation Middleware for Multi-tenant SaaS. *IEEE/ACM 8th International Conference on Utility and Cloud Computing* (págs. 112-121). Cyprus: IEEE.
- Xiong, W. (2014). HLA-Based SaaS-Oriented Simulation Frameworks. *IEEE 8th International Symposium on Service Oriented System Engineering* (págs. 376-383). Oxford: IEEE.



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Jaramillo Abril, Pablo Andrés** con C.C: # 092070494-7 autor del Trabajo de Titulación: **Optimización De Cloud Computing Por Multi-Tenant Para Empresas Proveedoras De Telecomunicaciones**, previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 3 de marzo del 2020

f. _____
Nombre: Jaramillo Abril, Pablo Andrés
C.C: # 092070494-7



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Optimización de cloud computing por multi-tenant para empresas proveedoras de telecomunicaciones.		
AUTOR(ES)	JARAMILLO ABRIL, PABLO ANDRES		
REVISOR(ES)/TUTOR(ES)	M. Sc. BASTIDAS CABRERA, TOMAS GASPAR		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TITULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	3 de Marzo del 2020	No. DE PÁGINAS:	74
ÁREAS TEMÁTICAS:	Comunicaciones inalámbricas, Telemática.		
PALABRAS CLAVES/KEYWORDS:	IAAS, SAAS, PAAS, HARDWARE, SOFTWARE, ESCALABILIDAD.		
RESUMEN/ABSTRACT:			
<p>El presente trabajo de titulación realiza un estudio sobre como el Cloud Computing ha generado una gran reducción para las empresas que delegan el aprovisionamiento de diversos recursos computacionales, como servidores, plataformas, redes, aplicaciones de software y servicios. Este modelo que provee servicios y aplicaciones es conocida como Software as a Service (SaaS), que juntos a sus semejantes, Infrastructure as a Service (IaaS) y Plataforma as a Service (PaaS), forman el mundo de oportunidades que ofrece la computación en la nube. Este trabajo también analiza que a medida que ha crecido el modelo de servicios por la nube, se han creado nuevos tipos de arquitecturas para llevar el negocio a una economía de escala, haciendo que los proveedores de servicios de computación en la nube ofrezcan una misma instancia de software a varios clientes, de manera que estos comparten bases de datos, sistemas operativos y hardware en algunos casos. Se hace especial énfasis en la arquitectura multi-tenant, en la cual se analiza el beneficio que ofrece tanto a los proveedores como a los clientes el gran ahorro de recursos que se busca con el servicio de la nube, junto a las falencias presentes en la arquitectura, las mismas que están relacionadas con la seguridad en la compartición de recursos, la disponibilidad del servicio y la escalabilidad. En el último capítulo se busca subsanar estos problemas, de manera que se ofrezca un correcto funcionamiento y gran fiabilidad del servicio para los clientes de la computación en la nube.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593993260320	E-mail: pabloandress12@hotmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez, Edwin Fernando		
	Teléfono: +593-9-67608298		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			