

UNIVERSIDAD NACIONAL DE SAN MARTÍN - TARAPOTO
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



TESIS

**USO DE SOFTWARE COMO SERVICIO PARA REDUCIR LOS
COSTOS TOTALES DE PROPIEDAD QUE SE INCURREN EN
LAS PYMES DEL SECTOR COMERCIO DEL DISTRITO DE
TARAPOTO**

**Para optar al Título de:
INGENIERO DE SISTEMAS E INFORMÁTICA**

Presentada por el Bachiller

Pedro Ever Vasquez Ylquimiche

Tarapoto -Perú

2017

UNIVERSIDAD NACIONAL DE SAN MARTÍN - T
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**USO DE SOFTWARE COMO SERVICIO PARA REDUCIR
LOS COSTOS TOTALES DE PROPIEDAD QUE SE
INCURREN EN LAS PYMES DEL SECTOR COMERCIO
DEL DISTRITO DE TARAPOTO**

**TESIS PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS E INFORMÁTICA**

Presentada por:

Bachiller : Pedro Ever Vasquez Ylquimiche

Asesor : Ing. Mg. Miguel Ángel Valles Coral

SUSTENTADO Y APROBADO ANTE EL HONORABLE JURADO:

Presidente : Ing. M. Sc. Miguel Angel Rengifo Arias

Secretario : Ing. José Enrique Celis Escudero

Miembro : Ing. Mg. Víctor Manuel Vallejos Monja



Firma



Firma



Firma



Firma



Firma

Declaratoria de Autenticidad

Yo, **Pedro Ever Vasquez Ylquimiche**, egresado de la Facultad de Ingeniería de Sistemas e Informática, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Universidad Nacional de San Martín – Tarapoto, identificado con DNI N°42547514, con la Tesis titulada: USO DE SOFTWARE COMO SERVICIO PARA REDUCIR LOS COSTOS TOTALES DE PROPIEDAD QUE SE INCURREN EN LAS PYMES DEL SECTOR COMERCIO DEL DISTRITO DE TARAPOTO

Declaro bajo juramento que:

1. La tesis presentada es de mi autoría.
2. He respetado las normas internacionales de citas y referencias para las fuentes consultadas. Por tanto, la tesis no ha sido plagiada ni total ni parcialmente.
3. El trabajo de tesis no ha sido auto plagiado; es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
4. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presenten en la tesis se constituirán en aportes a la realidad investigada.

De considerar que el trabajo cuenta con una falta grave, como el hecho de contar con datos fraudulentos, demostrar indicios y plagio (al no citar la información con sus autores), plagio (al presentar información de otros trabajos como propios), falsificación (al presentar la información e ideas de otras personas de forma falsa), entre otros, asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a la normatividad vigente de la Universidad Nacional de San Martín - Tarapoto.

Tarapoto, 04 de mayo del 2018.

.....
Pedro Ever Vasquez Ylquimiche
DNI N°42547514



Formato de autorización NO EXCLUSIVA para la publicación de trabajos de investigación, conducentes a optar grados académicos y títulos profesionales en el Repositorio Digital de Tesis.

1. Datos del autor:

Apellidos y nombres:	Vasquez Ylquiniche Pedro Ever		
Código de alumno :	107132	Teléfono:	976143808
Correo electrónico :	ever_vasquez306@hotmail.com	DNI:	42547514

(En caso haya más autores, llenar un formulario por autor)

2. Datos Académicos

Facultad de:	Ingeniería de Sistemas e Informática
Escuela Profesional de:	Ingeniería de Sistemas e Informática

3. Tipo de trabajo de investigación

Tesis	(X)	Trabajo de investigación	()
Trabajo de suficiencia profesional	()		

4. Datos del Trabajo de investigación

Título:	Uso de software como servicio para reducir los costos Totales de propiedad que se incurren en las PYMES del sector Comercio del distrito de Tarapoto.
Año de publicación:	2017

5. Tipo de Acceso al documento

Acceso público *	(X)	Embargo	()
Acceso restringido **	()		

Si el autor elige el tipo de acceso abierto o público, otorga a la Universidad Nacional de San Martín – Tarapoto, una licencia **No Exclusiva**, para publicar, conservar y sin modificar su contenido, pueda convertirla a cualquier formato de fichero, medio o soporte, siempre con fines de seguridad, preservación y difusión en el Repositorio de Tesis Digital. Respetando siempre los Derechos de Autor y Propiedad Intelectual de acuerdo y en el Marco de la Ley 822.

En caso que el autor elija la segunda opción, es necesario y obligatorio que indique el sustento correspondiente:

6. Originalidad del archivo digital.

Por el presente dejo constancia que el archivo digital que entrego a la Universidad Nacional de San Martín - Tarapoto, como parte del proceso conducente a obtener el título profesional o grado académico, es la versión final del trabajo de investigación sustentado y aprobado por el Jurado.

7. Otorgamiento de una licencia *CREATIVE COMMONS*

Para investigaciones que son de acceso abierto se les otorgó una licencia *Creative Commons*, con la finalidad de que cualquier usuario pueda acceder a la obra, bajo los términos que dicha licencia implica

<https://creativecommons.org/licenses/by-nc-sa/2.5/pe/>

El autor, por medio de este documento, autoriza a la Universidad Nacional de San Martín - Tarapoto, publicar su trabajo de investigación en formato digital en el Repositorio Digital de Tesis, al cual se podrá acceder, preservar y difundir de forma libre y gratuita, de manera íntegra a todo el documento.

Según el inciso 12.2, del artículo 12º del Reglamento del Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales - RENATI “**Las universidades, instituciones y escuelas de educación superior tienen como obligación registrar todos los trabajos de investigación y proyectos, incluyendo los metadatos en sus repositorios institucionales precisando si son de acceso abierto o restringido, los cuales serán posteriormente recolectados por el Repositorio Digital RENATI, a través del Repositorio ALICIA**”.



Firma del Autor

8. Para ser llenado en la Oficina de Repositorio Digital de Ciencia y Tecnología de Acceso Abierto de la UNSM – T.

Fecha de recepción del documento:

04 / 05 / 2018



Firma del Responsable de Repositorio
Digital de Ciencia y Tecnología de Acceso
Abierto de la UNSM – T.

***Acceso abierto:** uso lícito que confiere un titular de derechos de propiedad intelectual a cualquier persona, para que pueda acceder de manera inmediata y gratuita a una obra, datos procesados o estadísticas de monitoreo, sin necesidad de registro, suscripción, ni pago, estando autorizada a leerla, descargarla, reproducirla, distribuirla, imprimirla, buscarla y enlazar textos completos (Reglamento de la Ley No 30035).

** **Acceso restringido:** el documento no se visualizará en el Repositorio.

DEDICATORIA

A mis padres: **Erlinda Ylquimiche Aguilar**
y Benito Vasquez Valderrama,
por su apoyo incondicional y aliento
en los momentos difíciles a lo largo
de la realización de mi carrera
Profesional y de mi vida.

AGRADECIMIENTO

Al Ingeniero **Miguel Ángel Valles Coral**, por el apoyo esmerado y paciencia en la asesoría del desarrollo de mi proyecto de tesis.

A mis familiares y mis amigos quienes me brindaron su apoyo moral para lograr culminar con éxito este gran proceso que forma parte de mi desarrollo como profesional.

RESUMEN

La presente tesis estudia el problema de los elevados costos totales de propiedad al usar una solución de Software tradicional para administrar y controlar los procesos de negocio de las PYMES del sector comercio del distrito de Tarapoto.

El Cloud Computing conocido como la computación en la nube, es un conjunto de herramientas en la nube que nos permite alquilar una infraestructura (equipos físicos) con la adopción del “pago por uso”, teniendo un modelo similar al de pagar un servicio como el de la luz o agua, estas herramientas asociadas con algunos conocimientos de análisis, programación, base de datos y otros; permiten crear una solución de software como servicio que reduce considerablemente los costos totales de propiedad que incurren en las PYMES del sector comercio del distrito de Tarapoto.

La población y muestra involucrada fueron las PYMES del sector comercio de Tarapoto; se midió el costo total de propiedad antes y después de usar la solución de software como servicio, han experimentado un ahorro del 87%, ahorrándose en la instalación, configuración, administración y hardware, permitiendo que ese ahorro de dinero se pueda direccionar hacia nuevos activos de la PYME.

Palabras clave: Software como servicio, Computación en la nube, Sistemas de información, Multi-Tenant, Costo total de propiedad.

SUMMARY

This thesis examines the problem of high total cost of ownership when using a traditional software solution to manage and control the business processes of small and medium-sized businesses in the commerce sector of the district of Tarapoto.

Cloud Computing known as cloud computing is a set of tools in the cloud that allows us to rent an infrastructure (physical equipment) with the adoption of "pay per use", having a model similar to paying for a service such as Of light or water, these tools associated with some knowledge of analysis, programming, database and others; Allow the creation of a software-as-a-service solution that significantly reduces the total cost of ownership of small and medium-sized businesses in the Tarapoto district.

The population and sample involved were the small and medium-sized businesses of the commerce sector of Tarapoto; Total cost of ownership was measured before and after using the software as a service solution, have experienced a saving of 87%, saving on installation, configuration, management and hardware, allowing that money saving can be directed to new assets of small and medium-sized businesses.

Keywords: Software as Service, Cloud computing, Information system, Multi-Tenant, Total Cost of Ownership.

ÍNDICE

NOMENCLATURAS	11
a) Lista de tablas	11
b) Lista de figuras.....	11
c) Lista de siglas, abreviaturas y símbolos.....	12
INTRODUCCIÓN	14
CAPÍTULO I.....	16
I. EL PROBLEMA.....	16
1.1 Antecedentes del problema.....	16
1.2 Definición del problema.....	17
1.3 Formulación del problema.....	18
1.4 Justificación e importancia.....	18
1.5 Alcance y limitaciones	18
CAPÍTULO II.....	20
II. MARCO TEÓRICO.....	20
2.1 Antecedentes de la investigación.....	20
2.2 Definición de términos.....	21
2.3 Bases teóricas	24
2.3.1 SOFTWARE COMO SERVICIO.....	24
3.1.1. COSTOS TOTALES DE PROPIEDAD (TCO)	48
2.4 Hipótesis	55
2.5 Sistema de variables	56
2.5.1 Variable independiente	56
2.5.2 Variable dependiente	56
2.6 Escala de medición	56
2.7 Objetivos.....	56
2.7.1 General.....	56
2.7.2 Objetivo Especifico	56
CAPÍTULO III.....	57
III. MATERIALES Y MÉTODOS	57
3.1 Universo y muestra	57
3.2 Ámbito geográfico	59
3.3 Diseño de la investigación	59
3.3.1 Tipo de investigación.....	59
3.3.2 Nivel de investigación.....	59
3.3.3 Diseño de investigación.....	59

3.4 Procedimientos y técnicas	60
3.4.1 Procedimientos.....	60
3.4.2 Técnicas	62
3.5 Instrumentos	62
3.5.1 Instrumentos de recolección de datos.....	62
3.5.2 Instrumentos de procesamiento de datos.....	62
3.6 Prueba de hipótesis	63
CAPÍTULO IV	68
IV. RESULTADOS.....	68
4.1. Sobre el costo de hardware requerido.....	68
4.2. Costo del Software requerido	70
4.3. Costo de los servicios iniciales para la instalación	73
4.4. Costos de los servicios iniciales para la configuración	76
4.5. Costos de administración de software proporcionado	78
CAPÍTULO V	81
V. DISCUSIÓN DE LOS RESULTADOS	81
CAPÍTULO VI	83
VI. CONCLUSIONES.....	83
CAPÍTULO VII	85
VII. RECOMENDACIONES	85
VIII. REFERENCIAS BIBLIOGRÁFICAS.....	86
Bibliografía	86
IX. ANEXOS	88
ANEXO A: EMPRESAS DEL SECTOR COMERCIO	88
ANEXO B: CUESTIONARIO.....	95
ANEXO C: SOFTWARE COMO SERVICIO	96

NOMENCLATURAS

a) Lista de tablas

Tabla 1. Comparativa entre SaaS, S+S y ASP	27
Tabla 2. Comparativa entre SaaS y SaaS	29
Tabla 3. Técnicas para particionar la base de datos	46
Tabla 4. Operacionalización de variables	56
Tabla 5. Técnicas e instrumentos a utilizar en el estudio	62
Tabla 6. datos recolectados de cuestionario	64
Tabla 7. cálculo de estadígrafos(ANTES)	66
Tabla 8. cálculo de estadígrafos(DESPUÉS)	67
Tabla 9. Prueba T-Student para comparar resultados del TCO ANTES y DESPUÉS del uso de la Solución de Software como Servicio en las PYMES del sector comercio. Tarapoto – San Martín	67
Tabla 10. datos recolectados de cuestionario sobre el costo de hardware requerido.....	68
Tabla 11. datos recolectados de cuestionario sobre el costo de software requerido	71
Tabla 12. datos recolectados de cuestionario sobre los costos iniciales para la instalación.....	74
Tabla 13. datos recolectados de cuestionario sobre los costos iniciales para la configuración.....	76
Tabla 14. datos recolectados de cuestionario sobre los costos de administración de software proporcionado	79

b) Lista de figuras

Figura 1 Modelos de servicio según el público	25
Figura 2. Relación entre multi y single tenant, multi-usuario y multi-instancia	31
Figura 3. Esquemas separados	44
Figura 4. Resumen de las asignaciones de costos de una implementación de software tradicional Fuente: Jan Sysmans of WebEx Communications	53
Figura 5. Resumen de las asignaciones de costos de una implementación de SaaS.....	54
Figura 6. IDC's TCO comparacion entre software tradicional y SaaS.....	55
Figura 7: Total de PyMES por actividad en el distrito de Tarapoto	57

Figura 8: Diagrama del diseño experimental de la investigación	59
Figura 9: gráfico de PRE y POST del TCO	65
Figura 10: gráfico del promedio de TCO antes y después del SaaS	66
Figura 11: gráfico de PRE y POST del costo de hardware requerido	70
Figura 12: gráfico de PRE y POST del costo de software requerido	73
Figura 13: gráfico de PRE y POST de los costos de los servicios iniciales para la instalación.....	75
Figura 14: gráfico de PRE y POST de los costos de los servicios iniciales para la configuración.....	78
Figura 15: gráfico de PRE y POST de los costos de administración de software proporcionado	80

c) Lista de siglas, abreviaturas y símbolos.

- PYMES. Pequeña y mediana empresa.
- TCO. Costo total de propiedad.
- SaaS. Software como servicio.
- PaaS. Plataforma como servicio.
- IaaS. Infraestructura como servicio.
- CRM. Del inglés customer relationship management, sistema de gestión comercial.
- TI. Tecnologías de información.
- ROI. Retorno de inversión.
- SOA. Arquitectura orientada a servicios.
- NIST. Instituto nacional de estándares y tecnología.
- CPU. Unidad central de procesamiento.
- ASP. Del inglés application service providers, proveedor de servicios de aplicación.
- OPP. Programación orientada a objetos.
- MVC. Del inglés model view controller, modelo vista y controlador.
- UI. Interface de usuario.
- URI. Url del navegador.
- SQL. Del inglés structure query language, lenguaje estructurado de consulta.
- RDBMS. Sistema de gestión de base de datos relacional.

DBMS. Sistema de administración de base de datos.

VM. Máquina virtual.

VLAN'S. Red de área local virtual.

DHCP. Del inglés dynamic host configuration protocol, protocolo de configuración dinámica de host.

UNSM-T : Universidad Nacional de San Martín – Tarapoto

INTRODUCCIÓN

En la actualidad, la tecnología evoluciona constantemente, de forma rápida y vertiginosa, de tal forma que los ciclos de reemplazo son cada vez más cortos, caracterizándose principalmente por su reducción en tamaño, así como en costos. Esto hace que cada vez más empresas (micro, pequeñas, medianas y grandes) orienten sus recursos en adquirirlas para mejorar sus procesos de negocio.

Una de las mayores pruebas que deben superar las PYMES en la actualidad para lograr ser más competitivas es reducir sus costos. Este desafío atraviesa todas las PYMES y se ha transformado en la principal prioridad para los responsables de TI. Más aún, buscar la reducción del costo total de propiedad (TCO), como se le ha denominado, se vuelve casi una obsesión, sobre todo cuando se conoce con mayor certeza que dichos recursos son el medio por el cual el negocio va a ser más fructífero, logrando con ello la satisfacción de los clientes, la rentabilidad de la organización y ser un aporte a la sociedad.

A medida que las empresas aumentan su tamaño también crecen sus recursos informáticos. Esto genera gastos extras y una cadena de eventos como: compras de licencias de software, actualización de equipos y capacitación al personal de TI.

Sin embargo los mismos avances tecnológicos dan la oportunidad de que aplicaciones ofrecidas por internet como Software como Servicio (SaaS por sus siglas en inglés “Software as a Service”) pueden ser tan robustas como las aplicaciones instaladas en una máquina, donde la misma ofrece encargarse de toda la demanda de software, hardware y creciente procesamiento de datos que necesita la empresa a un costo razonable, ofreciéndolos en paquetes de servicios que son accedidos desde cualquier lugar con una conexión de internet teniendo una disponibilidad de las 24 horas del día los 7 días de la semana.

El trabajo de investigación se divide en cinco capítulos fundamentales:

- Capítulo I, denominado El Problema, en donde se expone los antecedentes del problema, definición del problema, la justificación, el alcance y limitaciones.

- Capítulo II, denominado Marco teórico, comprende los antecedentes de la investigación, las bases teóricas, la hipótesis, sistemas de variables y los objetivos.
- Capítulo III, denominado Materiales y Métodos, Comprende la metodología realmente aplicada, las técnicas y herramientas empleadas, donde además se hace la prueba de hipótesis.
- Capítulo IV y V, denominado Resultados y Discusión de los Resultados, respectivamente, en donde se describe el comportamiento de las variables, el resultado de ambas y contraste entre ellas.
- Capítulo VI y VII, denominado Conclusiones y Recomendaciones, respectivamente de la investigación.

CAPÍTULO I

I. EL PROBLEMA

1.1 Antecedentes del problema.

En la actualidad, la tecnología evoluciona constantemente, de forma rápida y vertiginosa, de tal forma que los ciclos de reemplazo son cada vez más cortos, caracterizándose principalmente por su reducción en tamaño, así como en costos. Esto hace que cada vez más empresas (micro, pequeñas, medianas y grandes) orienten sus recursos en adquirirlas para mejorar sus procesos de negocio.

También el uso del internet a nivel global ha aumentado exponencialmente permitiendo poder acceder a soluciones de tecnologías de Información más flexibles y versátiles que permiten a personas y PYMES hacer usos de sistemas que satisfagan cualquier tipo de necesidad desde cualquier parte del mundo.

Una de las mayores pruebas que deben superar las PYMES en la actualidad para lograr ser más competitivas es reducir sus costos. Este desafío atraviesa todas las PYMES y se ha transformado en la principal prioridad para los responsables de TI. Más aún, buscar la reducción del costo total de propiedad (TCO), como se le ha denominado, se vuelve casi una obsesión, sobre todo cuando se conoce con mayor certeza que dichos recursos son el medio por el cual el negocio va a ser más fructífero, logrando con ello la satisfacción de los clientes, la rentabilidad de la organización y ser un aporte a la sociedad.

A medida que las empresas aumentan su tamaño también crecen sus recursos informáticos. Esto genera gastos extras y una cadena de eventos como: compras de licencias de software, actualización de equipos y capacitación al personal de TI.

Sin embargo los mismos avances tecnológicos dan la oportunidad de que aplicaciones ofrecidas por internet(SaaS) pueden ser tan robustas como las aplicaciones instaladas en una máquina, donde la misma ofrece encargarse de toda la demanda de software, hardware y creciente procesamiento de datos que necesita la empresa a un costo razonable, ofreciéndolos en paquetes de servicios que son accedidos desde cualquier lugar con una conexión de internet teniendo una disponibilidad de las 24 horas del día los 7 días de la semana.

1.2 Definición del problema.

La presente investigación se realiza con la finalidad de plantear una solución que permita reducir los Costos Totales de Propiedad(TCO) que se incurren como consecuencia de la implantación de una solución de software de tecnología de información en las PYMES del sector comercio en el distrito de Tarapoto.

Entre las causas que se identifican durante la formulación del problema descrito en el párrafo anterior podemos mencionar que uno de ellos está relacionado a los elevados costos del hardware requerido, software requerido, servicios iniciales para la instalación, capacitación al personal, servicios inicial para configuración y costos de administración, que en su conjunto representan una inversión muy onerosa para las PYMES que recién se están iniciando, impidiéndoles la oportunidad de usar las tecnologías de información a beneficio propio, [consecuencia] evitando así optimizar sus procesos que les permita obtener una ventaja competitiva, como también dar un valor agregado a los productos i/o servicios que la empresa ofrece.

Otra de las causas al tomar en cuenta para la cuestión de esta investigación es el hecho de que el elevado costo total de propiedad tiene que prorratearse en los productos o servicios, pasando este costo al cliente haciéndolos menos competitivos ante la competencia a escala.

Hoy en día una compañía que no destina recursos a tecnologías de información corre una desventaja muy grande para desarrollar y expandir su negocio. Es por eso que la mayoría utiliza un software que ayude en la toma de decisiones, manejo de información y registro de transacciones, entre otras cosas. Este software la mayoría de veces se instala y se mantiene en la empresa. Esta parece ser una opción natural, Sin embargo, a la hora de evaluar el costo-beneficio, instalar un software implica adquirir una infraestructura de hardware, licencias y muchos costos indirectos.

Como se explica líneas arriba, este costo asociado debe ser asumido por la organización sin embargo la propuesta que se plantea es usar software como un servicio, de una forma similar a como se utiliza la línea telefónica, internet, agua o luz, pasando de ser un costo en activo fijo a un costo corriente.

1.3 Formulación del problema.

¿Es posible mediante el uso de software como servicio reducir los costos totales de propiedad (TCO) que se incurren en las pymes del sector comercio del distrito de Tarapoto?

1.4 Justificación e importancia.

De la conveniencia

La investigación es conveniente porque permitirá automatizar las PYMES del sector comercio del distrito de Tarapoto.

De la Relevancia Social

La investigación permitirá apreciar como el uso de las nuevas tecnologías como el Cloud Computing influye significativamente en la competitividad de las PYMES del distrito de Tarapoto. Los principales beneficiados serán los dueños de las PYMES, así como las personas que quieran iniciar en el sector comercio del distrito de Tarapoto.

De las implicancias practicas

La investigación permitirá que las PYMES que no cuentan con un software de control y administración de su negocio por ser costoso, puedan acceder a este servicio.

Del valor teórico

La investigación permitiría hacer uso de nuevas tecnologías como Cloud Computing y desarrollar estudios de como las nuevas tecnologías influyen considerablemente en la reducción de los costos total de propiedad.

De la utilidad metodológica

Gracias al uso de la metodología de investigación científica y los pasos relacionados a la operacionalización de las variables, se ha diseñado un instrumento que permite recolectar datos para su posterior análisis correlacionando la variable dependiente de la independiente, el mismo que forma parte de los anexos de este documento.

1.5 Alcance y limitaciones

El alcance son todas las PYMES del distrito de Tarapoto. Delimitando el proyecto por el costo total de propiedad del software mas no del hardware.

Sin embargo, dentro de las limitaciones identificadas para la ejecución del proyecto tenemos:

Desconocimiento de la propuesta de software como servicio por parte de los propietarios de las PYMES

Inadecuada valoración contable de los costos totales de propiedad y su efecto en los estados financieros resultantes del año.

CAPÍTULO II

II. MARCO TEÓRICO

2.1 Antecedentes de la investigación.

(Sysmans de WebEx Communications, 2006) En su investigación sobre Software como servicio; Una visión integral de la coste total de propiedad de aplicaciones de software (Software as a Service; A Comprehensive Look at the Total Cost of Ownership of Software Applications) nos dice que de acuerdo con Gartner, una firma global de investigación de TI, el costo anual de poseer y gestionar las aplicaciones de software puede ser hasta cuatro veces el costo de la compra inicial. Como resultado, las empresas terminan gastando más del 75% de su presupuesto total de TI simplemente en mantener en funcionamiento los sistemas y las infraestructura de software existentes. Con la introducción de los ordenadores, las empresas han aceptado esto como un costo de hacer negocios. El número de aplicaciones de software que puede necesitar una empresa son infinitas. Los recursos que operan estas aplicaciones, sin embargo son finitos.

(Bladh & Ström, 2008) En su tesis sobre Costo total de propiedad (Total cost of Ownership) concluye que en el estudio del costo total de propiedad se encontró que ocho elementos de costo constituían el 90 por ciento del costo total de propiedad. Todos estos elementos hacen un impacto de más de cinco por ciento cada uno. Algunos elementos de costes podrían ser excluidos ya que contribuyeron con menos del uno por ciento cada uno. Estos resultados permitieron el desarrollo de una herramienta para calcular el TCO con una disminución del número de elementos de coste a considerar. De esta manera un análisis TCO poco complejo está habilitado.

(Brivo OnAir, 2015) En su investigación sobre cómo el software como servicio (SaaS) reduce el coste total de propiedad (TCO) para los sistemas de seguridad física (How Software-as-a-Service (SaaS) lowers the Total Cost of Ownership (TCO) for physical security systems) llega a las siguientes conclusiones:

La primera es que en igualdad de condiciones - ambos grupos harían bien para calcular el costo relativo de las soluciones de seguridad física propuestas antes de tomar una decisión sobre lo que debe ofrecer a un cliente (en el caso de los integradores), o qué comprar en última instancia, (en el caso de los usuarios finales).

La segunda implicación es que los ahorros proporcionados por SaaS también pueden extenderse a otros servicios de seguridad, tales como vídeo alojado, detección de intrusiones, monitorización remota, y muchos otros. Esta es una implicación importante para la gran mayoría de los propietarios de negocios, como la mayoría de las empresas no son lo suficientemente grandes como para ser capaz de absorber el costo de las soluciones de servidor dedicado en una infraestructura de TI más grande. Lo que esto significa es que tales propietarios de negocios pueden esperar para disfrutar de los niveles de servicio de nivel empresarial a un menor TCO puntos que en cualquier momento en la historia de la seguridad electrónica.

(VMWARE, 2006) En su investigación sobre la reducción de coste total de propiedad del servidor con el software de virtualización de VMware (Reducing Server Total Cost of Ownership with VMware Virtualization Software) llega a la conclusión de que puede sorprender a los clientes aún más es el ahorro de estos clientes con experiencia en operaciones de TI. En promedio, estos clientes experimentaron ahorros de TCO en las operaciones de TI de 67%.

Todos los clientes perfilados en este documento están ampliando su infraestructura de virtualización debido al éxito de su primer proyecto de virtualización en la entrega de un rápido ROI (retorno de inversión). Claramente, hay mucho que ganar en la realización de análisis de TCO para conseguir un mayor cuadro completo de ahorro que supone una inversión en tecnología.

2.2 Definición de términos.

Dentro del contexto de la investigación cabe definir los siguientes términos para su mejor comprensión:

Software as Service (SaaS)

El software como servicio (SaaS) es una forma de entrega de aplicaciones a través de Internet, como un servicio. En lugar de instalar y mantener el software, sólo tiene que acceder a él a través de Internet, liberarse de software complejo y administración de hardware.

Las aplicaciones SaaS a veces se llaman software basado en Web, el software bajo demanda, o software alojado. Sea cual sea el nombre, las

aplicaciones SaaS se ejecutan en los servidores de un proveedor de SaaS. El proveedor gestiona el acceso a la aplicación, incluida la seguridad, disponibilidad y rendimiento.

Arquitectura orientada a servicios (SOA)

Actualmente las Arquitecturas Orientadas a Servicios (Service Oriented Architecture, SOA) están en auge, debido a los beneficios que proporcionan a la hora de desarrollar e integrar aplicaciones distribuidas o modulares. El principal concepto de SOA es el de servicio. Podemos ver un servicio como una llamada a una función, que se ejecutara local o remotamente, y que es independiente del lenguaje de programación y plataforma en la que se ejecuta. Este servicio consta de una interfaz bien definida y que depende de la tecnología que se desea utilizar para implementar SOA. El Proveedor de Servicios es un ente (nodo, clase, programa, etc.) que brinda un servicio en respuesta a una llamada o petición desde un Consumidor de Servicios. Este utiliza el ´ Publicador de Servicios para obtener información sobre los servicios que estén disponibles para su uso y sobre las interfaces (Descripción del servicio) para invocarlos.

Amazon Web Service

Amazon Web Services (AWS) ofrece recursos y servicios de informática en la nube bajo demanda conforme a un modelo de precios de pago según el uso. Por ejemplo, puede ejecutar en AWS un servidor en el que puede iniciar sesión y que puede configurar, proteger y ejecutar como si fuera un servidor que tiene delante.

El uso de recursos de AWS en lugar de sus propios recursos es como comprar electricidad a una empresa eléctrica en lugar de tener su propio generador y aporta muchas de esas ventajas: capacidad a la medida exacta de sus necesidades, solo paga por lo que usa, las economías de escala reducen los costos y el servicio lo ofrece un proveedor con experiencia en la gestión de redes a gran escala.

Windows Azure

Si usted pregunta, "¿Qué es Windows Azure?" La mejor respuesta podría ser simplemente esto:

Windows Azure puede ser lo que quiera que sea.

Como plataforma en la nube de Microsoft que proporciona una amplia gama de servicios diferentes, Windows Azure le permite crear, implementar y administrar soluciones para casi cualquier propósito que usted pueda imaginar. En otras palabras, Windows Azure es un mundo de posibilidades ilimitadas. Si usted es una

gran empresa que abarca varios continentes que se necesita para ejecutar cargas de trabajo de servidor, o un pequeño negocio que quiere un sitio web que tiene una presencia global, Windows Azure puede proporcionar una plataforma para la creación de aplicaciones que pueden aprovechar la nube para satisfacer las necesidades de tu negocio.

Software más servicios (S+S)

El término Software+Servicios (S+S) fue adoptado por Microsoft hace algunos años como parte de su visión tecnológica. En pocas palabras este concepto se puede definir como "aprovechar la riqueza de los dispositivos conectados con el poder de Internet", siendo precisamente ésta la apuesta de Microsoft como plataforma para los siguientes años.

Es por eso que S+S no es un enfoque centrado totalmente en el software tradicional (para la PC) ni tampoco centrado por completo en el navegador (servicios en Internet). Es un híbrido que combina lo mejor de los dos mundos: el poder de software que corre en dispositivos conectados más la flexibilidad y ubicuidad de servicios en la nube, de tal forma que el usuario pueda acceder a sus datos y aplicaciones desde la PC, el Web y desde sus múltiples dispositivos de una forma sencilla y homogénea.

SAP

Desde una perspectiva de software de aplicaciones de negocio, SAP es casi todas las cosas a casi todas las empresas. Cada componente de software o aplicación dentro de la familia SAP de productos y servicios responde a una necesidad particular, facilitar el día a día de la gestión financiera y de recursos (SAP Enterprise Resource Planning [ERP]), que abordan los requisitos de planificación del ciclo de vida del producto (SAP Product Lifecycle Management [PLM]), el apoyo interno de contratación de la empresa (SAP Supplier Relationship Management [SRM]), la interconexión de diferentes sistemas para aliviar los dolores de cabeza de integración (SAP NetWeaver Integración de Procesos [SAP NetWeaver PI]), la gestión de relaciones que permite al cliente (SAP Customer Relationship Management [CRM]), y así. Dividido por SAP en el SAP Business Suite (que comprende todas las aplicaciones de negocio) y SAP NetWeaver (componentes de los cuales permiten esencialmente el SAP Business Suite, como un producto de portal, herramientas de desarrollo y herramientas de inteligencia de negocio) y por

lo tanto muchas posibles soluciones de SAP pueden ser ensamblados y personalizados para casi cualquier negocio.

2.3 Bases teóricas

2.3.1 SOFTWARE COMO SERVICIO

A. Cloud Computing

Hay un gran número de definiciones para la computación en la nube. Muchas fuentes han definido computación en la nube desde diferentes perspectivas y las definiciones han sido objeto de debate. Vamos a presentar varias definiciones en esta sección.

Sin embargo, la computación en nube tiene algunos nuevos significados adicionales. Para empezar, se puede afirmar que el Cloud Computing es la culminación de Grid computing, utility computing, unified communication (UC), arquitectura orientada a servicios (SOA), Web 2.0 y muchas otras tecnologías similares.

Se ha descrito como la nueva era de la informática, la adopción de un “pago por uso” o un modelo de utilidad similar a la electricidad, agua y otros servicios públicos. Para un usuario final, el Cloud Computing es una ilusión de un conjunto de recursos informáticos infinitos bajo demanda. Estos recursos son servicios por una expansión de servidores, equipos de redes y sistemas de almacenamiento de un centro de datos. Este nuevo modelo de consumo y la entrega muestra un cambio de un gasto de capital (CAPEX) a un gasto de operación (OPEX).

A partir de las miríadas de definiciones disponibles, la definición por el Instituto Nacional de Estándares y Tecnología (NIST) ha sido ampliamente aceptada (NIST 2010)

“La computación en la nube es un modelo que permite un cómodo acceso, en demanda de la red a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente aprovisionados y liberados con el mínimo esfuerzo”.

Para entender el modelo de computación que ofrece Cloud Computing, es necesario dejar en claro a qué llamamos modelo de computación. Un modelo de computación define dónde está la capacidad de cómputo y de qué forma se accede

a ella. En la siguiente sección se enumeran los diferentes modelos de computación que surgieron a lo largo de la historia.

Fundamentos de Cloud Computing

Se describe a Cloud Computing o Computación en la nube como un conjunto de características esenciales, que se apoya en ciertos modelos de despliegue y que ofrece diferentes modelos de servicio. Las siguientes tres secciones se explaya estos conceptos.

a. Modelos de servicio

El modelo de servicio de Cloud Computing está relacionado directamente con el modelo de negocio de Cloud Computing, en donde la característica esencial es la forma de pago: “El consumidor solo paga por lo que usa” debido a esta característica, el consumidor no compra un producto, sino que renta un servicio por un tiempo determinado y sólo paga por el uso que le dé al mismo dentro del tiempo estipulado. En concordancia con esta definición, en la diversa literatura sobre Cloud Computing emergió el término “X as a Service”, donde X representa el tipo de servicio que se ofrece. La mayoría de los autores presentan el modelo de servicio como una pila de servicios, como en la Figura 1, orientados para diferente público.



Figura 1 Modelos de servicio según el público

Fuente: Nieto (2013)

IaaS - Infraestructura como servicio

Provee servicios de infraestructura IT como máquinas virtuales, almacenamiento, recursos de red, procesamiento, sistemas operativos, orientados al personal de IT. Es el servicio más básico del modelo de servicio de cloud computing. Ofrece recursos de cómputo como CPU, memoria, almacenamiento mediante máquinas virtuales, las cuales son creadas utilizando un administrador de máquinas virtuales. Este administrador es conocido por el término hypervisor, y es un software especial

dedicado para emular máquinas. Ejemplos de hypervisor son Xen y KVM. Un proveedor de IaaS tiene en su datacenter un conjunto de hypervisors que trabajan unidos, que permiten ofrecerle al usuario la capacidad de obtener recursos de forma casi ilimitada. Los proveedores más conocidos de IaaS son Amazon Web Services, con el servicio EC2, Windows Azure Virtual Machines, DynDNS, Google Compute Engine, HP Cloud, Rackspace Cloud.

PaaS – Plataforma como servicio

Este servicio ofrece herramientas de software orientado a desarrolladores de aplicaciones. Permite abstraer al desarrollador de las particularidades de la infraestructura proveyendo una capa de más alto nivel para administrar. Desde el punto de vista del desarrollador de aplicaciones, la plataforma funciona como un middleware entre su aplicación y la infraestructura cloud del proveedor. Este middleware es un conjunto de aplicaciones de software, portales web, APIs, etc., cuyo objetivo es permitir al desarrollador u organización la gestión integral de la plataforma.

SaaS – Software como servicio

Es el servicio de más alto nivel, ofrece aplicaciones para el usuario final. Según la definición del NIST, SaaS es el servicio que permite al usuario utilizar aplicaciones que están corriendo en una infraestructura cloud. Estas aplicaciones son accesibles desde diferentes dispositivos a través de un browser. El consumidor de este servicio no administra ni tiene control sobre la infraestructura cloud subyacente, como ser la red, los servidores, sistemas operativos, almacenamiento, ni siquiera sobre las características técnicas de la aplicación.

B. Software como Servicio (SaaS)

Una solución de software como servicio es considerado tanto un modelo de negocio y el modelo de distribución de aplicaciones. Como tal, abarca una amplia gama de negocios, marketing y técnicas de oportunidades, problemas y retos. La flexibilidad de las implementaciones SaaS también significa que las organizaciones pueden ser cargadas en una base de pago por uso. Una de las principales ventajas de SaaS, es que, como se trata de una aplicación alojada, los usuarios no tienen que invertir en hardware costoso. En general se elimina la necesidad de las

organizaciones para manejar la instalación, puesta en marcha y el mantenimiento diario de servidores y software.

B.1. Características de una solución SaaS

Las principales características del modelo según IDC's Worldwide Software as a Service Taxonomy son:

- El software es accesible, manejado y comercializado vía red.
- El mantenimiento y actividades relacionadas con el software se realizan desde un lugar centralizado en lugar de hacerlo en cada cliente, permitiendo a estos acceder a las aplicaciones vía la red.
- La aplicación es distribuida típicamente bajo el modelo de uno-a-muchos, incluyendo su arquitectura, management, precio y partnering.
- Generalmente se basa en un modelo de comercialización en el cual no hay un costo inicial, sino un pago por suscripción o por utilización, en el cual no se diferencia la licencia del software del alojamiento del mismo.

Estas cuatro características son fundamentales para diferenciarlas de otros modelos que ofrecen software vía red, como el modelo Software más Servicios (S+S) y el modelo Application Service Providers (ASP).

(The Art of Service, 2015), nos dice que el modelo de Software más Servicios, utiliza servicios alojados en un servidor remoto, pero la aplicación principal se instala en el ambiente local del usuario. Este modelo no aprovecha completamente la capacidad de cómputo del servidor remoto y obliga al cliente a administrar y reservar los recursos necesarios para instalar la aplicación cliente en sus instalaciones.

En la Cuadro 1, se muestra un resumen de las diferencias entre los diferentes modelos.

Tabla 1. Comparativa entre SaaS, S+S y ASP

	SaaS	S+S	ASP
z	Si	Si	Si
Administración centralizada	Si	No, ya que hay Software en el Cliente.	Si
Modelo uno a muchos	Si	No	No

Costo Inicial	Muy bajo	Alto. Cada cliente alojar el software cliente.	Alto. Cada cliente paga su infraestructura.
----------------------	----------	--	---

Fuente: Nieto, (2013).

Dado que un Software como Servicio es utilizado y comercializado vía red, sin la necesidad de que el cliente tenga que instalarse nada, este modelo difiere completamente del modelo tradicional de software ofrecido como un producto instalable en el cliente. En el modelo de Software como Producto (SaaP), los proveedores desarrollan el software y lo empaquetan para ser distribuido bajo un nombre y una versión determinada, en un soporte físico digital como un CD o DVD, o para ser descargado desde Internet. El producto está preparado para ser alojado en las instalaciones informáticas del cliente, y viene asociado con un contrato de licencia de uso que define bajo qué condiciones se puede instalar y utilizar.

Para el proveedor, desarrollar una solución SaaS en lugar de un producto, le trae varias ventajas, por ejemplo:

- Le permite tener un mejor control y administración del software, dado que el mismo reside en sus instalaciones de forma completa, y no está diseminado en todos los clientes. Esto le permite al proveedor modificar, actualizar y corregir lo que sea necesario sin la necesidad de liberar versiones de forma periódica o parches.
- El proveedor solo debe codificar el software para una única plataforma, y al ser accedido vía internet, llegar a plataformas heterogéneas.
- Dado que la aplicación está alojada en la infraestructura del proveedor, le permite tener un feedback rápido de los clientes y de los bugs que se producen en la aplicación, en lugar de esperar reportes de los clientes en forma esporádica.
- Con un producto, es el cliente el que decide siempre migrar de una versión a la otra, y el proveedor tiene la obligación de ofrecer mantenimiento y soporte para versiones anteriores. En cambio, controlar y administrar un software alojado en servidores controlados por el proveedor, supone una mayor flexibilidad para aplicar actualizaciones y correcciones que fácilmente sean accedidas por todos los clientes.

Para el cliente que consume un Software como Servicio en lugar de un producto, también le trae varias ventajas, por ejemplo:

- Tener un modelo de pago flexible y a un menor costo: El mayor costo para una empresa que necesita de software está en las licencias y en el hardware que necesita para correr dicho software. Normalmente las licencias son por un período de tiempo determinado y para cierta cantidad de usuarios o máquinas. Esta inversión inicial es costosa, ya que la mayoría de los vendedores de software otorgan licencias de un año o más. En cambio, las aplicaciones SaaS no se licencian, sino que el cliente se suscribe al servicio por un tiempo determinado, de meses a años, pudiendo dejar de utilizar el servicio cuando lo desee, ya que en general no hay contratos de permanencia, dándole al cliente una gran libertad económica.
- Tener siempre la última versión del servicio: el cliente tiene la garantía de que siempre utilizará la última versión de la solución, con las actualizaciones de seguridad, correcciones y mejoras correspondientes, sin la necesidad de esperar al lanzamiento de una versión estable, y sin la necesidad de hacer migraciones de una versión a la otra.

En la Cuadro 2, se muestra un resumen de la diferencia entre un software ofrecido como producto y un software ofrecido como servicio.

Tabla 2. Comparativa entre SaaS y SaaS

	Software como producto	Software como servicio
Delivery	Diseñado para que el cliente lo instale, administre y mantenga.	Diseñado para ser utilizado desde internet.
Desarrollo	Ciclo de desarrollo largo, múltiples codificaciones para la misma aplicación.	Diseñado para correr miles de clientes sobre un único código fuente.
Política de precios	Licencia + Mantenimiento	Suscripción (Todo Incluido)
Costos adicionales	Instalación, mantenimiento, personalización, actualizaciones.	Solo para características adicionales.
Plataforma	Múltiples versiones para diferentes plataformas.	Una única versión para diferentes plataformas.
Bugs	Mecanismos de actualización lentos y con diferentes tiempos para cada cliente.	Se arregla un problema para un cliente, y se arregla para todos.
Demo de la aplicación	Requiere instalación	Online, no requiere instalación.
Actualizaciones	Lentas, dependen del cliente	Rápidas, las controla el proveedor
Feedback del cliente	Lento y dependen del cliente el envío de las mismas.	Rápido y online, se pueden monitorear estadísticas de uso.

Fuente: Nieto, (2013).

Las características principales de SaaS, que están más bien relacionadas con el modelo de negocio, y la diferencia con otras alternativas a SaaS como SaaS, ASP y S+S, se deben implementar siguiendo ciertos requisitos técnicos. En la siguiente sección se explayan estos requisitos técnicos que se necesitan para construir una solución SaaS.

C. Requisitos técnicos de las aplicaciones SaaS

El objetivo de este apartado es explorar en detalle los requisitos técnicos de una aplicación SaaS, de forma de evidenciar las cuestiones de diseño y arquitectura que se deben tener en cuenta a la hora de construir una aplicación de este tipo.

C.1. Multi-Tenant

(Bezemer & Zaidman, 2010) En su investigación sobre “Challenges of Reengineering into Multi-Tenant SaaS Applications” nos dice que una aplicación multi tenant permite a los clientes (tenants) compartir los mismos recursos de hardware, ofreciéndoles una instancia de aplicación y base de datos compartida, permitiendo a cada cliente configurar la aplicación según sus necesidades y correrla como si estuviera en un ambiente dedicado y aislado de los demás clientes. También según nos dice que un tenant o cliente es la entidad organizacional que renta una solución SaaS multi-tenant. Típicamente un tenant agrupa a un conjunto de usuarios, que son los que utilizan la aplicación dentro de la organización. Desde el punto de vista del desarrollo de software, multi-tenant es un principio de arquitectura cuyo objetivo es brindar lineamientos de diseño para que la solución SaaS sea escalable, personalizable y maximice la utilización de los recursos. .

(Bezemer & Zaidman, 2010) En su conferencia de “Multi-Tenant SaaS Applications - Maintenance Dream or Nightmare” nos dice que existen diferentes niveles de uso compartido en las aplicaciones.

a. Aplicaciones Multi-usuario

La definición de multi-usuario corresponde a los sistemas que permiten que múltiples usuarios utilicen concurrentemente una misma instancia de aplicación. Aquí el usuario es la persona física que accede al sistema, con un usuario y contraseña, pertenezca a una organización o no. Vale aclarar que algunas aplicaciones que se clasifican como SaaS no necesariamente tienen una arquitectura multi-tenant. Por ejemplo, los servicios de mail como Hotmail y Yahoo

Mail, son aplicaciones SaaS multi-usuario, pero estos usuarios son usuarios individuales para estos servicios, ya que no están asociados a ninguna organización.

b. Aplicaciones Multi-instancia

Las aplicaciones multi-instancia son aquellas que se instalan en un servidor de aplicaciones o servidor web. Por lo general estos servidores permiten alojar diferentes aplicaciones y diferentes instancias de la misma aplicación. Para el cliente, esto es transparente, y puede simular una aplicación multi-tenant, pero la realidad es que si bien se comparten recursos, la arquitectura multi-tenant aprovecha muchísimo mejor el uso de un servidor.

C.2. Multi-Tenant vs. Single-Tenant

Como grafica la Figura 2, las aplicaciones multi-tenant y single-tenant pueden estar basadas en sistemas multi-usuario y multi-instancia, pero la principal diferencia es que las aplicaciones single-tenant están preparadas para atender a una sola organización o a usuarios individuales sin relación con alguna organización. Si se decide ofrecer una solución single-tenant a otra organización, por lo general es necesario que el proveedor instale un nuevo servidor y una nueva base de datos, de forma de aislar la aplicación y los datos del otro cliente, y que no interfiera la actividad de uno con el otro.

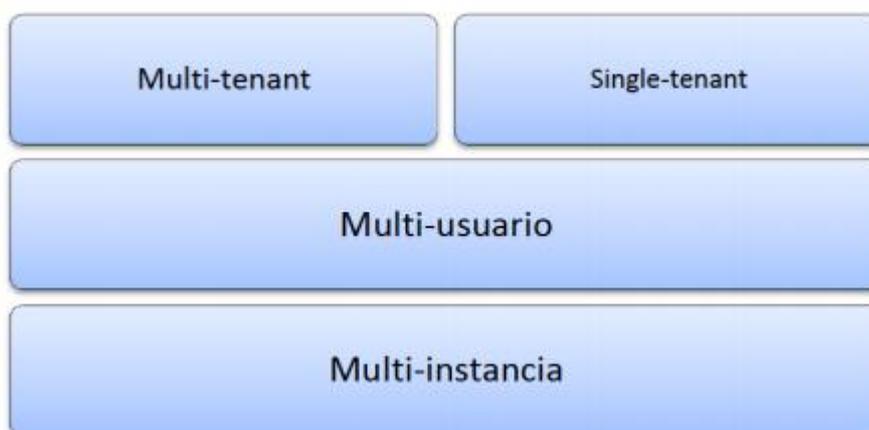


Figura 2. Relación entre multi y single tenant, multi-usuario y multi-instancia

Fuente: Nieto (2013)

Por otro lado, las aplicaciones multi-tenant representan el máximo nivel de uso compartido de una aplicación. Un sistema multi-tenant puede implementarse con

una única instancia de aplicación o con múltiples instancias. La cantidad de instancias no está relacionada directamente con la cantidad de tenants. Todos los tenants pueden correr en una única instancia, o un grupo de tenants pueden correr en una instancia y otro grupo de tenants puede correr en otra, o bien, un tenant puede tener asignada más de una instancia de aplicación. La capacidad de adecuar la cantidad de instancias de la aplicación según el uso y la cantidad de tenants es una característica importante de las aplicaciones SaaS.

C.3. Escalabilidad

El concepto de escalabilidad se podría definir como la capacidad de un sistema informático de cambiar de tamaño o configuración para adaptarse a las circunstancias cambiantes, sin perder calidad y manejando el crecimiento o decrecimiento de forma continua, sin interrupciones del servicio y de forma homogénea.

La escalabilidad hacia arriba es cuando hay un mayor volumen de uso de la aplicación y se necesita agregar mayor poder de procesamiento, de memoria, de ancho de banda, etc., para seguir brindando el servicio en óptimas condiciones, siempre de forma continua y sin interrupciones.

La escalabilidad hacia abajo es cuando hay un volumen de uso menor de la aplicación y existe capacidad ociosa de los recursos, por lo que se necesita desafectar esos recursos de la solución SaaS, siempre de forma continua y sin interrupciones.

Debido a que un tenant se puede suscribir y dar de baja en cualquier momento, la solución SaaS tiene que estar preparada para escalar en recursos de hardware y software hacia arriba o hacia abajo de forma dinámica.

La escalabilidad de una solución SaaS se logra diseñando cuidadosamente la aplicación para tal fin, de forma que el servicio se brinde sin interrupciones.

Las aplicaciones SaaS deben estar preparadas para un alto volumen de tráfico de usuarios, para ello podemos emplear la técnica de balancear la carga de trabajo de la aplicación. El balanceo de carga puede operar en tres niveles, a nivel de red, a nivel de servidor y a nivel de aplicación. En una aplicación web tradicional el balanceo de carga se hace a nivel de red o a nivel del servidor web. Por lo general

se dispone una granja de servidores relacionados, el propio sistema operativo o servidor web se encarga de monitorear la carga de trabajo y redirigir las peticiones al servidor correspondiente. Ahora bien, en aplicaciones SaaS multi-tenant necesitamos balancear la carga a nivel de aplicación, ya que para cada tenant podemos tener diferentes políticas de balanceo de carga. Este concepto implica tener en cuenta que es necesario diseñar la aplicación para soportar diferentes instancias de la misma en el mismo server, y diferentes instancias en diferentes servers, identificando siempre que tenant está haciendo el requerimiento.

También es posible balancear la carga de trabajo de la base de datos. Este punto es importante ya que dependiendo de cómo hayamos implementado la técnica de multitenant, tendremos que tener en cuenta de que tenant es el requerimiento para saber a qué base de datos dirigirlo.

C.4. Configurabilidad

(Pradeep Kumar, V., & S., 2010) En su investigación “Configurability in SaaS for an Electronic Contract Management Application”, la configurabilidad permite proveer a los clientes/tenants de una multitud de opciones y variaciones de la aplicación, usando un único código base, de tal forma que es posible que cada tenant tenga su propia configuración de software.

En una solución SaaS podemos tener configurabilidad a 2 niveles:

a. A nivel de interfaz de usuario

La configurabilidad de la interfaz de usuario implica poder adaptar el “look and feel” según las necesidades del usuario y del tenant en cuestión, tales como íconos, colores, fuentes, títulos, etc.

b. A nivel funcional

La configurabilidad a nivel funcional implica que para cada tenant puede variar el comportamiento de la aplicación, donde ciertas funciones del sistema pueden tener una lógica de negocios distinta para diferentes tenants o hasta flujos de trabajo totalmente diferentes. Además, la variabilidad de cierta funcionalidad puede ocasionar que también varíe la interface.

Para ambos niveles de configurabilidad necesitaremos almacenar los datos de configuración para poder implementar esta característica en la aplicación. Los datos

de configuración son el corazón de una aplicación SaaS y es necesario diseñar su almacenamiento y Backus con soporte para múltiples tenants.

C.6. Aprovisionamiento

El aprovisionamiento es el proceso por el cual se reservan recursos de hardware y de software en la solución SaaS para un nuevo tenant o cliente.

Permitir a un nuevo tenant utilizar una aplicación SaaS puede involucrar, por ejemplo, importar cuentas de usuario y datos, aplicar configuraciones, crear tablas o nuevas bases de datos, asignar una cantidad fija de procesadores, habilitar permisos de ejecución y reglas de acceso del firewall, etc.

El aprovisionamiento puede ser manual o automático. Es manual cuando la cantidad de tenant es muy poca, y las tareas de aprovisionamiento las puede hacer el administrador de la solución SaaS. En este caso, el cliente solicita o se suscribe al servicio SaaS, y espera un tiempo de 24 horas, por ejemplo, hasta que el administrador aprovisiona los recursos necesarios para que el tenant este operativo en la solución SaaS.

Es automático, cuando la cantidad de tenants comienza a crecer y se hace imposible dedicar una persona a las tareas de aprovisionamiento. En este caso es necesario que la solución SaaS disponga de un módulo de software específico para tareas de aprovisionamiento automático, con la posibilidad de que el administrador monitoree las mismas.

C.7. Suscripción, monetización y facturación

Los mecanismos tradicionales de cobro son por cantidad de usuarios y por acceso a ciertos módulos de la aplicación, pero en el modelo SaaS se puede cobrar por tiempo de uso de la aplicación, por la cantidad y por el tiempo de los recursos de hardware que se desea utilizar, o modelos de cobro relacionados con el negocio concreto de la solución (monetización), como por ejemplo en el caso de un CRM, donde se puede relacionar el precio con el volumen de operaciones de venta. Introducir estos mecanismos de cobro en una solución de software implica que deben ser tenidos en cuenta en el diseño y en la codificación de la aplicación.

D. Técnicas de diseño y programación para implementar la configurabilidad

Ciertos requisitos técnicos de las aplicaciones SaaS multi-tenant, como la configurabilidad a nivel funcional y la configurabilidad nivel de UI, necesitan implementarse con el soporte de técnicas modernas de diseño y programación. En este apartado se analizarán diferentes técnicas de programación para llevar a cabo estos dos requisitos y para poder entender cuál se ajusta a las necesidades de la aplicación a desarrollar, y así implementar soluciones SaaS de forma óptima, evitando diseños difíciles de mantener.

D.1. Configurabilidad a nivel funcional

a. Programación orientada a objetos (POO)

El paradigma OOP es uno de los pilares para construir aplicaciones SaaS multi-tenant altamente configurables, debido a sus principios de diseño de reusabilidad, modularidad, abstracción, encapsulamiento, ocultación de información, herencia y polimorfismo. La mayoría de los lenguajes modernos que se utilizan para desarrollar aplicaciones web, ya sean de front-end o de back-end son orientados a objetos o tienen pleno soporte del paradigma. Así, lenguajes de back-end como java, c#, visual basic, ruby, python, php, y de front-end como javascript por nombrar algunos, son indicados para desarrollar soluciones SaaS multi-tenant. Características como uso de programación paralela y multithreading, binding dinámico y tipado dinámico de los lenguajes OOP permiten escribir software altamente adaptable y configurable.

b. Arquitectura orientada a servicios (SOA)

SOA es ideal para diseñar sistemas débilmente acoplados, una característica muy importante si tenemos en cuenta cuestiones como la alta configurabilidad necesaria en una aplicación multi-tenant. Una solución SOA, con soporte multi-tenant, trae aparejado particularidades técnicas en la definición de la arquitectura que no aparecen en aplicaciones single-tenant, las cuales serán necesarias resolver para implementar con éxito tal solución, como por ejemplo, el concepto de multi-tenant utilizando un ESB, todo lo concerniente a orquestación y motores de reglas de negocio, y las diferentes técnicas de integración de aplicaciones, mecanismos de instanciación de servicios, etc.

Si bien los principios de diseño de SOA, como bajo acoplamiento, composición, interoperabilidad y federación de servicios, son afines a los principios de diseño de una aplicación SaaS multi-tenant, ésta no necesariamente debe tener una arquitectura SOA, y si bien es posible plantear la lógica de negocios del backend de la aplicación como una colección de servicios, no necesariamente significa que tengamos una arquitectura SOA.

(Pathirage, Perera, Kumara , & Weerawarana, 2011) en su publicación acerca de “A Multi-tenant Architecture for Business Process Executions”, (Mietzner , Unger, Titze, & Leymann, 2009) en su publicación “Combining Different MultiTenancy Patterns in Service-Oriented Applications” y (Mietzner & Leymann, Towards Provisioning the Cloud: On the Usage of Multi-Granularity Flows and Services to Realize a Unified Provisioning Infrastructure for SaaS Applications) en su investigación “Towards Provisioning the Cloud: On the Usage of MultiGranularity Flows andServices to Realize a Unified Provisioning Infrastructure for SaaS Applications”, se analizan diferentes patrones que pueden ser utilizados para diseñar y desarrollar aplicaciones SaaS orientadas a servicios y a procesos BPEL.

c. Programación orientada al contexto (COP)

La programación orientada al contexto, COP su abreviatura en inglés, permite definir comportamiento variable dependiendo del contexto, lo cual lo hace atractivo para utilizarlo en el desarrollo de aplicaciones multi-tenant configurables.

Los conceptos principales de COP son:

- **Variaciones de comportamiento**

Las variaciones típicamente consisten en un nuevo o modificado comportamiento. Estos pueden ser expresados como definiciones parciales de módulos en el modelo de programación subyacente, como procedimientos o clases.

- **Layers o capas**

Los layers agrupan variaciones de comportamiento relacionadas. Las layers son las entidades principales, por lo que pueden ser explícitamente referenciadas desde el modelo de programación subyacente.

- **Activación**

Las layers pueden ser activadas o desactivadas dinámicamente en tiempo de ejecución. Por código se puede decidir habilitar o deshabilitar las layers basados en el contexto actual.

- **Context**

Cualquier información que es accesible computacionalmente puede formar parte del contexto que activará las variaciones de comportamiento que dependen de él.

- **Alcance**

El alcance dentro de las cuales las layers pueden ser activadas o desactivadas puede ser controlado explícitamente.

Las layers son construidas de modo que reaccionen a la información contextual. Basado en la información disponible del contexto de ejecución actual, layers específicas pueden ser activadas o desactivadas. Los lenguajes COP o extensiones a lenguajes para soportar COP para expresar, activar y componer layers en tiempo de ejecución, pero es la aplicación la que determina cuál es la información contextual relevante.

Los mecanismos de abstracción existentes en los lenguajes orientados a objetos son suficientes para modelar el contexto.

(Truyen, et al., 2012) en su investigación “Context-oriented Programming for Customizable SaaS Applications”, se menciona cómo se puede utilizar COP para implementar la configurabilidad de una aplicación multi-tenant.

d. Inversión de control e inyección de dependencia

La inversión de control es un patrón de diseño de OOP, en el que una clase A no depende directamente de otra clase B, sino a través de un intermediario, cediendo o invirtiendo el control de la clase al intermediario, al contrario del enfoque tradicional, donde el código de la clase A, controla o define exactamente con qué otras clases necesita interactuar, definiendo una relación fuerte entre las mismas. Para esto, la clase A no utiliza las clases relacionadas directamente, sino que declara las variables utilizando interfaces o clases abstractas. El intermediario es el encargado de instanciar, por ejemplo la clase B, e inyectar la dependencia en la

clase A. Estos patrones son ideales para implementar la característica de configurabilidad de una aplicación SaaS, ya que se pueden inyectar diferentes implementaciones de las clases, dependiendo del tenant en cuestión, permitiendo adaptar en tiempo de ejecución la lógica de negocios, la lógica de presentación, y muchas otras cuestiones, de acuerdo a cada tenant. Esta técnica permitirá escribir código limpio, orientado a objetos y componentizable, evitando el código “spaguetti” y un sinnúmero de sentencias “if” que ocasionaría las diferentes configuraciones de cada tenant.

e. Sistema de control de versionado de código

Scott C. y Ben S. en su libro “Pro Git” nos dice, El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

En el proceso de construcción de una aplicación multi-tenant es primordial tener el control de todas las versiones de la aplicación desde la primera versión hasta la actual, como también GIT nos permite movernos en las diferentes versiones sin perder algún cambio de código aumentando el control del versionado de la aplicación multi-tenant.

f. El Pattern Plug-In

(Viswanathan, 2012) En su publicación “Right Engineering SaaS”, este patrón es mencionado como una buena práctica, cuya función es la de permitir agregar funcionalidad adicional una vez terminado y entregado el producto. Para ello es necesario preparar la aplicación y dejar codificado puntos de extensibilidad mediante interfaces, de forma que los componentes que respeten e implementen dichas interfaces puedan ser agregados como plug-in a la aplicación, sin la necesidad de recompilarla, y hasta sin siquiera detener su ejecución. Este tipo de patrón es ideal para ofrecer diferentes funcionalidades para diferentes tenants, y activarlos en tiempo de ejecución de acuerdo al tenant que esté accediendo a la aplicación en ese momento. También este patrón contribuye a hacer el código de la aplicación mantenible en el tiempo, ya que cada funcionalidad particular es separada en un componente diferente, y no forma parte del núcleo principal de la aplicación.

g. Colas de tareas (Task Queues)

(Shafique Hijazee, 2015) En su libro “Mastering Google App Engine” nos habla: La gente empezó a darse cuenta de que este tipo de tareas costosas computacionalmente se hacen mejor en segundo plano en lugar de poner nuestros servidores web bajo estrés y realizarlo allí mismo. Ahí es donde la idea de colas de tareas y los trabajos independientes llegó a la vida. La idea principal es que cualquier tarea costosa computacionalmente se añaden como una tarea a una cola y hay varios equipos de trabajadores que tiran de las tareas de la cola, las procesan según lo descrito por el mensaje y almacenan los resultados.

En una aplicación multi-tenant tenemos que reservar el hilo principal para los clientes (tenants) y ejecutar tareas costosas añadiéndolas a una cola de trabajo para que se ejecuten en segundo plano, por ejemplo descargar reportes, documentos, enviar correos electrónicos, etc.

D.2. Configurabilidad a nivel de usuario

A continuación, se presentan las diferentes técnicas de diseño y programación que facilitan la implementación del requisito técnico Configurabilidad a Nivel de UI.

a. Single Page Internet Application Architecture (SPIAR)

SPIAR es el acrónimo de Single Page Internet Application Architecture, o en castellano Arquitectura de aplicación de Internet de una Sola Página. Este tipo de arquitectura se popularizó con la aparición de Ajax, permitiendo que las páginas html puedan hacer invocaciones asincrónicas de javascript sin refrescar la misma. Actualmente esta tendencia se acentuó y es muy común ver aplicaciones que solo están conformadas por una única página html, que se modifica dinámicamente utilizando javascript y haciendo invocaciones al servidor de forma asincrónica.

Si bien las aplicaciones que se construyen con este tipo de arquitectura tienen un mejor tiempo de respuesta, para construir aplicaciones multi-tenant con interfaces de usuario altamente configurables será necesario replantear y tener en cuenta algunas cuestiones, como no llevar lógica de presentación por tenant al browser, ya que puede involucrar un problema de seguridad importante, y para esto se necesitará una programación mucho más cuidadosa.

b. MVC

El patrón MVC es clave para implementar una UI altamente configurable. La mayoría de los frameworks que implementan el patrón MVC permiten definir en ejecución el mapeo entre la vista y el controlador y el mapeo entre una url y una vista. Con estas dos características, estos mapeos pueden ser almacenados en una base de datos, y diferentes vistas y controladores pueden ser invocados según el tenant que este accediendo a la aplicación, permitiendo que la UI se adapte al mismo y que la lógica de presentación para cada tenant quede del lado del servidor.

c. Arquitectura dirigida por metadatos

Las características de una aplicación SaaS multi-tenant, como la configurabilidad pueden ser alcanzadas basando el producto en una arquitectura dirigida por metadatos.

Almacenando la disposición de la UI, las validaciones del cliente, las propiedades de los controles visuales, en un repositorio de metadatos, y luego interpretando estos metadatos en tiempo de ejecución hará que la aplicación tenga diferentes aspectos según las necesidades de cada tenant. La metadata puede ser usada para representar flujos de trabajo, actividades, políticas, lógica y datos. La metadata puede ser de negocio, técnica u operacional, y usualmente es almacenada en formato xml o en una base de datos. La principal ventaja de construir productos dirigidos por metadatos es la reusabilidad, la consistencia, la mantenibilidad y la alta adaptabilidad del software. Un ejemplo de aplicación enteramente basada en una arquitectura dirigida por metadatos la podemos encontrar en Salesforce.com, en (D Weissman & Bobrowski, 2011), en su publicación "The Design of the Force.com Multitenant Internet Application Development Platform" y (SalesForce, 2011) en su publicación "The multitenant, metadata-driven architecture of Databasedotcom", se realiza una buena explicación sobre su arquitectura.

Al diseñar una solución web multi-tenant, la implementación de la configurabilidad involucra tres aspectos fundamentales. El primero es la identificación del tenant en la aplicación web, ya que dado que la misma instancia de la aplicación atenderá los requerimientos http de todos los clientes, es necesario implementar algún mecanismo para diferenciar los requerimientos de un tenant y

de otro. El segundo aspecto es como diseñar y codificar la aplicación para soportar la configurabilidad tanto a nivel funcional como a nivel de UI. El tercer aspecto es cómo diseñar la base de datos de la aplicación para que soporte múltiples clientes, con diferentes datos y configuraciones. En este capítulo se abordarán estos tres aspectos, y se ofrecerán diferentes enfoques para implementarlos.

D.3. Identificación del tenant en la aplicación

Lo primero que hay que especificar para diseñar una aplicación web multi-tenant es cómo se van a identificar los tenant, y cómo se mantendrá esa información a lo largo de la ejecución de una determinada acción. Este es el punto de partida, y la forma en que lo diseñemos impactará en el diseño de los demás aspectos de la configurabilidad. Como toda aplicación web, el acceso a la misma se realiza utilizando requerimientos http a través de una URI.

(Cai, et al., 2009) En su publicación “An End-to-End Methodology and Toolkit for fine granularity SaaS-ization”, se define al TenantId como el identificador único global para identificar a un tenant, el cual se obtiene a través de la URI. El TenantId puede ser un número, un código alfanumérico, o bien puede ser el dominio de internet del cliente. Para codificar el TenantId en la URI tenemos 3 alternativas.

a. Utilizar un parámetro de la URL

Se utilizará para todos los tenant el mismo dominio, pero cada uno entraría al sistema con un parámetro de id de tenant distinto. Por ejemplo, si el dominio del sistema de alertas fuera <http://negosy.com>, se podrían codificar de la siguiente forma http://negosy.com?tenant_id=1 o <http://negosy.com/tenant1> En este caso, cada requerimiento a las siguientes páginas tiene que ser precedidas de la identificación del tenant, <http://negosy.com/tenant1/pautas> o http://negosy.com/pautas?tenant_id=1.

b. Utilizar un subdominio del dominio

Cada tenant podría identificarse como <http://primercliente.negosy.com> y <http://segundocliente.negosy.com> Cada requerimiento individual siempre se realiza utilizando el subdominio, por ejemplo <http://primercliente.negosy.com/pautas> y <http://segundocliente.negosy.com/pautas>.

c. Utilizar el dominio del cliente

Cada cliente que posea su propio dominio, puede redirigir los requerimientos de cierta url a los servidores donde está alojado el sistema. Por ejemplo, se puede configurar en el servidor de internet que la misma aplicación acepte más de un dominio, de esta manera, la misma aplicación puede aceptar requerimientos del dominio `www.primercliente.com` y requerimientos del dominio `www.segundocliente.com`. En tiempo de ejecución, antes de procesar el requerimiento, se puede analizar la URI y según el dominio determinar el tenant.

Cualquier de estas tres alternativas es válida para identificar el tenant, la elección particular de una de ellas, dependerá de las características de la plataforma como servicio donde se implemente la aplicación y de las facilidades del framework web que se utilice.

D.4. Diseño de la base de datos

El soporte de la configurabilidad para diferentes tenants, implica diseñar la base de datos específicamente para soportar ésta característica. A continuación se muestran diferentes técnicas para alcanzar este objetivo.

a. Base de datos dedicada para cada tenant

Esta aproximación consiste en dar a cada tenant su propia base de datos, donde cada tenant puede extender de acuerdo a sus necesidades. De esta forma, cada vez que un nuevo tenant es creado, se crea una nueva base de datos default, como parte del proceso de aprovisionamiento. Una vez que es creada, el tenant es libre de modificarla o extenderla, de acuerdo a sus necesidades particulares de negocio o de interface, permitiendo crear nuevos campos, nuevas consultas, o incluso nuevas tablas y relaciones.

b. Base de datos compartida, diferentes tablas por tenant

Esta aproximación utiliza la misma base de datos para todos los tenants. Cada vez que se crea un nuevo tenant, el proceso de aprovisionamiento crea las tablas específicas para el nuevo tenant, nombrando a cada tabla con un prefijo o sufijo para poder identificar a qué tenant pertenece esta tabla. Esta opción reduce los costos de administración, puesto que se comparte la misma instancia de base de datos para todos los tenants. Además, permite que cada tenant pueda extender sus tablas, agregando nuevos campos y relaciones, sin interferir con otros tenants.

Por ejemplo:

```
Select * from Pautas_Tenant_1
```

```
Select * from Pautas_Tenant_2
```

La desventaja es que hay que tener en cuenta que el código de la aplicación deberá generar todos los SQL de forma dinámica, ya que antes de hacer cualquier consulta, inserción, modificación o eliminación deberá armar el nombre de la tabla de acuerdo al tenant que ejecute la sentencia. Esta aproximación además tiene el costo de que generará un catálogo de la base de datos bastante grande, ya que se debe almacenar para cada tabla su log de transacciones, sus índices, etc. También si se necesita información estadística de todos los tenant, se necesitarán procesos de consolidación que unifiquen la información.

c. Base de datos compartida, mismas tablas, nuevo campo para distinguir los registros de cada tenant

Esta aproximación utiliza una misma instancia de base de datos y las mismas tablas para todos los tenants. Para distinguir en cada tabla, las filas correspondientes a un tenant específico, en cada tabla se agrega un campo adicional que represente el id del tenant.

Por ejemplo

```
Select * from Pautas where TenantId = 1
```

En el proceso de aprovisionamiento no hará falta crear nuevas bases de datos ni nuevas tablas, sino que solo se harán inserciones en tablas ya existentes, teniendo siempre presente de relacionar todas las entidades con el TenantId. Desde el punto de vista del código de la aplicación, siempre que se necesita ejecutar una sentencia SQL contra la base de datos, será necesario incluir una cláusula where adicional para filtrar por el ID del tenant, sin esta cláusula se estaría rompiendo el aislamiento entre los diferentes tenants. Para esta solución necesitaremos definir una nueva tabla, que contendrá la información relacionada con cada tenant.

d. Federación de base de datos

El concepto de Federación es una manera de alcanzar gran escalabilidad y performance, ya que se particiona la base de datos de forma horizontal, lo que

significa que una o más tablas de la base de datos tendrán grupos de filas en múltiples bases de datos, que forman parte de la Federación como miembros de la misma. Este tipo de particionamiento horizontal es referido como Sharding.

(Microsoft, 2013) En su publicación “Federation Guidelines and Limitation”, se puede encontrar una descripción más detallada sobre cómo funciona la Federación en Windows Azure y cómo se puede implementar para soportar una aplicación multi-tenant. Una alternativa más sofisticada es el soporte nativo del RDBMS para multi-tenant. Para un análisis del mismo consultar (Schiller, Schiller, Brodt, & Mitschang, 2011) en su publicación “Native Support of Multitenancy in RDBMS for Software as a Service”.

e. Base de datos compartida, esquemas separados

Este enfoque implica múltiples inquilinos en la misma base de datos, con cada inquilino tiene su propio conjunto de tablas que se agrupan en un esquema creado específicamente para el inquilino.

Este enfoque se puede ver en la figura 5, cada inquilino tiene su propio conjunto independiente de las tablas de una base de datos común.

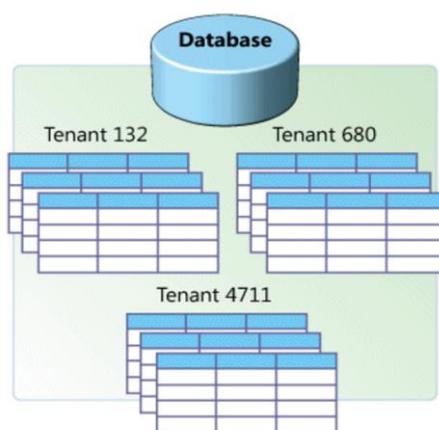


Figura 3. Esquemas separados

Fuente: Microsoft (2016)

Cuando un cliente se suscribe por primera vez al servicio, el subsistema de aprovisionamiento crea un conjunto discreto de tablas por defecto para el inquilino y lo asocia a su propio esquema del inquilino.

Al igual que el enfoque aislado, el enfoque de esquema separado es relativamente fácil de implementar, y los inquilinos pueden extender el modelo de datos tan fácilmente como con el enfoque de base de datos independiente. (Las tablas se crean a partir de un conjunto predeterminado, pero una vez que se crean que ya no tienen que cumplir con el conjunto predeterminado, y los inquilinos pueden añadir o modificar columnas e incluso tablas si lo desean.) Este enfoque ofrece un grado moderado de aislamiento de datos lógicos para los inquilinos preocupados por la seguridad, aunque no tanto como un sistema completamente aislado, y puede soportar un mayor número de inquilinos por servidor de base de datos.

Uso de Object Relational Mappings (ORMs)

Un ORM u Object Relational Mapping es el componente de software que permite utilizar las entidades de una base de datos relacional como si fueran los objetos de un modelo de objetos. Este tipo de software permite abstraerse del DBMS que se está utilizando, y permite que el desarrollador se concentre en consultar, insertar, modificar y eliminar objetos, utilizando un lenguaje de objetos, en lugar de SQL. Por lo general, un ORM es un framework independiente de la aplicación, el cual implementa toda la funcionalidad necesaria para interactuar con diferentes DBMSs y trae su propio lenguaje de consultas de objetos. Además de que el desarrollador no tiene que preocuparse de escribir código SQL, la ventaja principal es que se genera el SQL de forma dinámica, permitiendo agregar filtros antes y después de la generación del SQL. Esta característica facilita enormemente trabajar con consultas que necesitan acceder a información de cierto tenant, ya sea que estén implementadas con tablas diferentes para cada tenant, o que haya una columna en cada tabla para identificar el tenant. Para el código de la aplicación es transparente el método por el cual está particionada la base de datos, y será tarea del ORM, a través de alguna configuración o extensión, obtener el id del tenant actual, y agregar los filtros necesarios o determinar en tiempo de ejecución el nombre de la tabla a acceder según el tenant.

(Naeem Khan, 2013) En su publicación “Object-Relational Mapping Framework to Enable Multi-Tenancy Attributes in SaaS Applications”, podemos encontrar una buena discusión sobre el uso de ORMs.

El método para particionar la base de datos a elegir dependerá de la complejidad que se quiera afrontar. En la Cuadro 3 se muestra un resumen de las ventajas y desventajas.

Tabla 3. Técnicas para particionar la base de datos

Método	Ventaja	Desventaja
Diferentes bases de datos	Fácil de implementar	Consume muchos recursos de infraestructura
Misma base, diferentes tablas	Consume menos recursos, seguridad y aislamiento a nivel base de datos	Difícil para consolidar información
Misma base, mismas tablas, campo discriminador	Consume mínimos recursos. Fácil para consolidar información	Seguridad y aislamiento a nivel de aplicación.
Federación de base de datos	Partición a nivel de DMBS. Mejor performance.	Lo debe soportar el motor de DBMS. Es un servicio más caro.
Base de datos compartida, esquemas separados	Fácil de implementar	La restauración de un esquema de datos implica restaurar toda la base de datos.

Fuente: Arsalan S., Muhammad N y Ahmed K. (2013) y elaboración propia

E. Desarrollo de aplicaciones SaaS sobre IaaS

Infraestructura como Servicio es el pilar sobre el que se construye una arquitectura de computación en la nube. Con el avance de las tecnologías de las comunicaciones, la informática y dispositivos de almacenamiento, IaaS se ha convertido en una plataforma altamente eficiente para la construcción de SaaS y PaaS en la capa superior de la misma. Soluciones IaaS varían de una organización a otra.

E.1. Introducción

Infraestructura como Servicio es uno de los componentes básicos de la computación en nube. Es la pieza clave, que moderniza la solución en la nube, ofreciendo la infraestructura de TI subyacente.

IaaS en sí se compone de muchos componentes diferentes. Se proporciona aquí una lista de los componentes.

(W. Rittinghouse & F. Ransome, 2009) En su libro “Cloud Computing: Implementation, Management, and Security” nos dice los componentes que incluyen para IaaS, pero no se limitan a, lo siguiente:

- Servidores (físicos y virtuales)
- Los sistemas de almacenamiento por medio de NAS y SAN
- La segmentación de red utilizando diferentes bloques de red y VLANs

- La red de comunicación (incluyendo routers, switches, cortafuegos, equilibrador de carga, etc.)
- conectividad a Internet de alta velocidad (a menudo en OC 192 columnas vertebrales)
- entorno de virtualización de plataforma
- Acuerdos de Nivel de Servicio
- la facturación de utility computing
- Seguridad por medio de hardware o un firewall basado VM y detección de intrusiones y sistemas de prevención
- equilibrador de carga de hardware
- DNS, DHCP y otros servicios de gestión y apoyo
- sistema de recuperación de desastres de energía y refrigeración.

Según la lista de componentes, el tipo de componente que usaremos es servidores, donde instalaremos la aplicación multi-tenant. Para ello tendremos que adquirir un servidor de la empresa SO YOU START.

E.2. Herramientas y servicios esenciales

Para poder llevar a cabo un buen diseño de una aplicación SaaS multi-tenant necesariamente la plataforma debe dar soporte a una serie de cuestiones, sin las cuales no podríamos diseñar una aplicación multi-tenant, o bien, sería mucho más complejo y hasta inviable. Estas cuestiones son las que se analizarán para cada plataforma en el siguiente apartado.

a. Método de acceso

SSH (Secure SHell, en español: intérprete de órdenes seguro) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos.

Mediante el SSH podremos configurar completamente el servidor, crear la base de datos e instalar todas las librerías para poder iniciar nuestra aplicación multi-tenant.

b. Internet de banda ancha - alta velocidad

Internet de banda ancha es una de las arterias de un sistema de computación en la nube. Es la Internet que conecta miríada de dispositivos entre sí y crea una nube gigantesca. Gran ancho de banda y baja latencia en el enlace de comunicaciones a bajo costo es una característica importante para la nube. La mayoría de los dispositivos móviles y los teléfonos están ahora envía con interfaces inalámbricas. Esto ha hecho posible el uso de estos dispositivos para acceder a Internet. Con la proliferación de las tecnologías de muchas personas ahora tienen acceso a Internet en algunos aspectos. Ahora, los usuarios prefieren realizar la mayor parte de sus tareas en la web, utilizar aplicaciones web y colaborar con otros en línea. Esta tendencia se está haciendo muy popular la computación en nube.

3.1.1. COSTOS TOTALES DE PROPIEDAD (TCO)

La revolución Software as a Service (SaaS) permite a las empresas que se suscriban a aplicaciones de software y así externalizar el funcionamiento de la infraestructura de backend para el proveedor de SaaS. En la mayoría de los casos, el proveedor SaaS puede hacer esto mucho más rentable; proporcionando ahorros de costos globales de la empresa. Como resultado, las empresas pueden extender su presupuesto de TI a través de muchas más aplicaciones para apoyar y hacer crecer sus operaciones comerciales que a su vez contribuyan a la línea de fondo.

A. Gastos de capital

A.1. Software tradicional

Software y hardware, mejoras de la red de infraestructura, herramientas de monitorización y de prueba, productos de seguridad, equipos, instalaciones y otras obras de infraestructura requerida son parte de los gastos típico de adquisición de capital. En muchos casos, las actualizaciones a otras infraestructuras pueden requerir que se suma el gasto de capital adicional. Este gasto de capital es un desembolso en efectivo por adelantado.

A.2. Software como servicio (SaaS)

Con SaaS, no hay licencias de software perpetuos para comprar. La naturaleza de SaaS es que se paga por lo que usa. La mayoría de los modelos SaaS tienen una estructura de costos recurrentes. Usted paga una cuota mensual o anual de servicio durante el tiempo que se utiliza el servicio. Esta cuota de servicio por lo

general incluye el mantenimiento, soporte, capacitación y mejoras, e incluye de todo el hardware, redes, almacenamiento, bases de datos, administración y otros costos asociados con la entrega de SaaS.

B. Diseño y costos de implementación

B.1. Software tradicional

El personal y los contratos de trabajo necesarios para investigar, diseñar, integrar, probar, ajustar y puesta en marcha es un coste significativo asociado con la implementación de una solución interna. Las capacidades del servidor y la red deben ser reevaluados y aumentadas. Hardware, sistemas operativos y aplicaciones de usuario final tienen que ser evaluados para la compatibilidad con el producto de servidor seleccionado y actualizado en caso necesario. Las pruebas del sistema y puesta a punto son necesarias para asegurarse de que el rendimiento es aceptable para el lanzamiento. Será necesario capacitar a los usuarios finales y el personal de TI. Actividades de lanzamiento y pilotos, requieren recursos de TI.

B.2. Software como servicio (SaaS)

La mayoría de las aplicaciones SaaS se pueden implementar y poner en producción mucha más rápida y por una fracción del costo en comparación con una solución de software tradicional. Esto es muy importante cuando los costos de oportunidad de conseguir la aplicación son altas. Por otro lado, debido a que una aplicación SaaS es una aplicación multi-arrendatario (multi-tenant), hay menos posibilidades para personalizar la aplicación para adaptarse a los procesos de negocio.

C. Costos de infraestructura

C.1. Software tradicional

Para un funcionamiento permanente, los instrumentos de seguimiento y gestión de la red a menudo se requieren para permitir el diagnóstico de problemas en tiempo real y capacidad de respuesta. Equipos de red y ancho de banda adicional puede ser necesaria para acomodar el tráfico incremental de que no puede ser gestionado de forma eficiente en la red interna. Contratos anuales de mantenimiento y soporte de software y actualizaciones del sistema hacen una gran contribución al coste total de propiedad. Los aumentos de capacidad, múltiples

sistemas redundantes, así como conjuntos de funciones add-on aumentan aún más los costos. La reparación y sustitución de hardware y los costes ambientales recurrentes, tales como instalaciones de alta disponibilidad especializados y el consumo de energía, vienen a completar el costo en curso. Mientras que estos gastos se extienden a lo largo de la duración del servicio, deben tenerse en cuenta en un análisis completo de TCO.

C.2. Software como servicio

Aparte de las necesidades adicionales de ancho de banda de Internet, casi no hay costes de infraestructura adicionales para manejar el crecimiento de una aplicación SaaS. Dependiendo de la aplicación SaaS, la organización de TI también puede tener que implementar una aplicación de escritorio para permitir que el usuario final se comuniquen con la aplicación. Por último una API (Application Program Interface) de desarrollo puede ser necesaria para configurar la aplicación para integrar mejor con las aplicaciones empresariales existentes. El escalado de la infraestructura y los costos asociados con el crecimiento son totalmente responsabilidad del proveedor de SaaS.

D. Las actividades en marcha, operaciones y costos de soporte

D.1. Software tradicional

Las organizaciones de TI tendrán que asignar recursos para el seguimiento, soporte y mantenimiento de la aplicación. Si la aplicación es nueva para la empresa, la organización tendrá que capacitar y certificar al personal existente y/o reclutar nuevo personal con o sin certificación de aplicaciones pre-existente. La organización también será responsable de la supervisión y el mantenimiento de la aplicación y resolución de problemas de la aplicación en caso de que el tiempo de inactividad. Además, cada vez que un parche o una actualización necesita ser desplegado, se requerirá que los recursos de TI adicionales. Este suele ser el mayor costo oculto que debe tenerse en cuenta al tomar la decisión de compra de una nueva aplicación. Si este costo se estima de forma incorrecta, ningún efecto sobre los ingresos o reducción de costes puede cambiar en gran medida.

La formación inicial y permanente es otro factor de éxito para la amplia adopción de una nueva aplicación. Un proveedor puede ofrecer formación inicial o un tren de la sesión de entrenador como parte del costo inicial, pero con la mayoría de las

aplicaciones de software tradicionales es un departamento interno que se encarga de la formación inicial y permanente de los usuarios finales. Una vez más, las estimaciones incorrectas pueden afectar en gran medida los ingresos esperados o la reducción de costes.

El soporte es el factor final y más crítico para el éxito de la adopción exitosa y el uso continuo de una nueva aplicación: cada vez que los usuarios finales tienen problemas con la aplicación, esto puede llevar a una pérdida en la productividad o en el peor de los casos la negativa a utilizar la aplicación todos juntos. Esto también se conoce como software convirtiéndose en shelfware. Los problemas de los usuarios en general crecen con el uso y como tal, la carga de soporte en la organización crece también. Si se agrega el uso externo de la aplicación de esta ecuación, soporte técnico y los costos asociados con esta generalmente crecen exponencialmente. De nuevo, si este costo se estima erróneamente, ningún efecto sobre los ingresos o reducción de costes puede cambiar en gran medida.

D.2. Software como servicio

Los proveedores de SaaS son responsables de la entrega de extremo a extremo de la aplicación. La única responsabilidad de la organización de TI es asegurarse de que los puertos necesarios en el servidor de seguridad están abiertas y que no hay suficiente capacidad de acceso a Internet disponible para permitir que la base de usuarios finales para comunicarse con la aplicación.

SaaS es un servicio recurrente, por lo que para un proveedor de SaaS la venta no termina cuando se firma el contrato inicial. Si un cliente no utiliza la aplicación, simplemente pueden decidir no renovar el contrato al final de la duración del contrato. Esto se llama pérdida de clientes. Un proveedor de software tradicional no tiene que preocuparse por la rotación, ya que los clientes compran licencias de uso a perpetuidad por adelantado. Como resultado, los proveedores de SaaS tienen un interés personal en que los clientes adopten y utilicen la aplicación. Es por esta razón por la que casi todos los proveedores de SaaS se centran en hacer sus productos fáciles de usar y ofrecen la formación inicial y permanente de los usuarios finales, y este entrenamiento es en la mayoría de los casos incluidos en las tarifas de servicios.

Por último, los proveedores de SaaS también ofrecen 1ª, 2ª y 3ª línea de apoyo a sus clientes. Esto es por las mismas razones que ofrecen los servicios de formación continua; si los clientes se agitan debido a problemas de formación o de apoyo, que tendrá un impacto inmediato en el balance final del proveedor de SaaS.

E. Los costos intangibles

Mientras que los costos intangibles son más difíciles de medir y por lo tanto son más difíciles de incluir en un análisis TCO, que no son menos reales. Algunos de los factores de coste intangibles que afectan TCO incluyen:

Fiabilidad y disponibilidad: Las fallidas interacciones de tiempo de empleado perdido y la pérdida de oportunidades, pueden requerir esfuerzos de repetición para convencer a los usuarios para volver a intentar la tecnología con el aumento de la resistencia. ¿Qué nivel de servicio acuerdos (SLA) ofrece el proveedor de SaaS y cómo se comparan con el SLA interno de la organización que ofrece?

Interoperabilidad: ¿Es fácil de integrar con otras aplicaciones?

Extensibilidad: ¿Es fácil de personalizar la aplicación para adaptarse a las necesidades de la organización?

Seguridad: Los costos de una brecha de seguridad pueden ser catastróficas si la información confidencial de la empresa es robada o puesto a disposición de los competidores. ¿Cuáles son las políticas de seguridad que se encuentran en el lugar en el proveedor de SaaS y cómo se comparan con las políticas internas?

Escalabilidad: A medida que crecen las necesidades del usuario, el sistema original no puede mantener el ritmo. "las señales de ocupado" o limitaciones funcionales consumen tiempo de los empleados y la media de las oportunidades perdidas. ¿Qué tan bien puede el proveedor de SaaS adaptarse al crecimiento y cuáles son los costos asociados con el crecimiento de la aplicación interna?

Capacidad: Uso y adopción dentro de la empresa es difícil de predecir, por lo que la capacidad de manejo difícil. Las ventajas y desventajas son bajo rendimiento, por una parte, o de la infraestructura subutilizada en el otro. Con SaaS esto es más fácil de manejar cuando se compara con una aplicación interna.

Los costos de oportunidad: Los gastos de recursos y capital humano que requiere una implementación en la empresa se producen a expensas de otros proyectos o, posiblemente, podrían retrasar la puesta en marcha de nuevos productos y servicios, los cuales tienen un impacto directo en la rentabilidad de la empresa.

F. Resumen

F.1. Software tradicional

El mayor factor de coste total de propiedad de las aplicaciones de software tradicional basado en la premisa es el costo de los recursos de la gente en curso que son necesarios para controlar, mantener y actualizar la aplicación y proporcionar formación y apoyo a la base de usuarios finales. Estos costos no son citados como parte del costo de despliegue de la aplicación de software tradicional y dependiendo de la aplicación, puede ser entre el 50 y el 85% del coste total de propiedad para la application. La subestimación de estos costes puede tener un gran impacto en el predicacion general TCO.

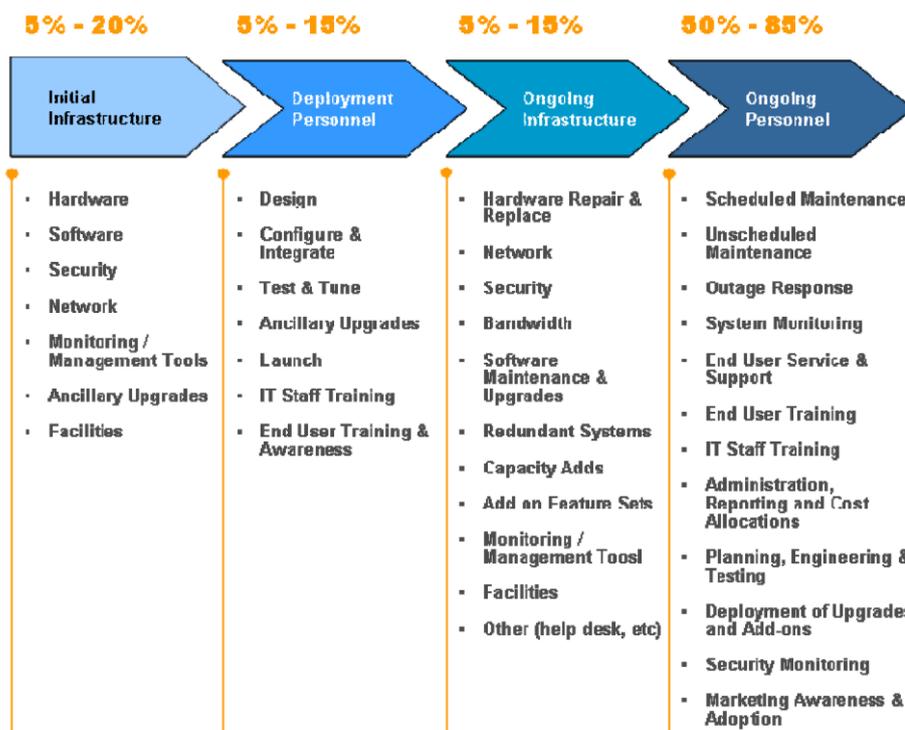


Figura 4. Resumen de las asignaciones de costos de una implementación de software tradicional
Fuente: Jan Sysmans of WebEx Communications

F.2. Software como servicio

El mayor factor de coste total de propiedad de las aplicaciones SaaS son las cuotas de suscripción que cobra el proveedor de SaaS. Estas tarifas son todo incluido, incluye el monitoreo, mantenimiento y actualizaciones de la aplicación, así como la formación y el apoyo de la base del usuario final. En comparación con los costes de personal en curso de la aplicación de software tradicional de estos costos se citan como parte del coste de despliegue de la aplicación SaaS.

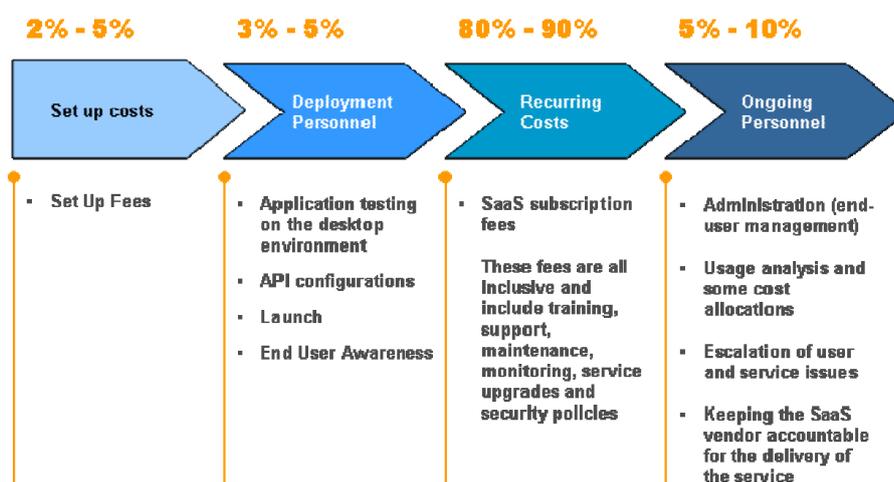


Figura 5. Resumen de las asignaciones de costos de una implementación de SaaS.

Fuente: Jan Sysmans of WebEx Communications

G. El cálculo del TCO

G.1. Alejándose del software "propietario"

Con el software tradicional, los clientes compran una licencia de uso perpetuo. Esto les da la impresión de que "poseen" el software y se pueden utilizar a voluntad ya perpetuidad. Con SaaS, en lugar de "ser dueño" de software, los clientes pagan por una suscripción al software que se ejecuta en la infraestructura de propiedad del proveedor de SaaS. El derecho del cliente a utilizar el software desaparece una vez que dejan de pagar por la suscripción. Ellos no pierden los derechos y la propiedad de los datos almacenados en la infraestructura del proveedor de SaaS.

La dinámica entre la suscripción y la propiedad se utiliza como un argumento a favor de la TCO compra de software tradicional. ¿Por qué alquilar cuando se puede comprar, sobre todo cuando el plan es utilizar la aplicación durante un largo tiempo? Por el contrario, las aplicaciones SaaS ofrecen muchas ventajas sobre el software

tradicional incluyendo evitar cuestan las enormes ocultos en el despliegue de personal, funcionamiento y mantenimiento de software tradicional.

G.2. El esquivo punto de equilibrio

Otro argumento a favor de "poseer" software ha sido que, incluso con los costos iniciales más altos, existe un punto de equilibrio donde el software tradicional se vuelve mucho más barato que el modelo de suscripción SaaS.

La firma de análisis IDC revisó varios SaaS frente a las implementaciones de software tradicionales y encontró que cuando los recursos y las personas costo de las actualizaciones se toman correctamente en consideración, este punto de equilibrio puede que nunca se realice.

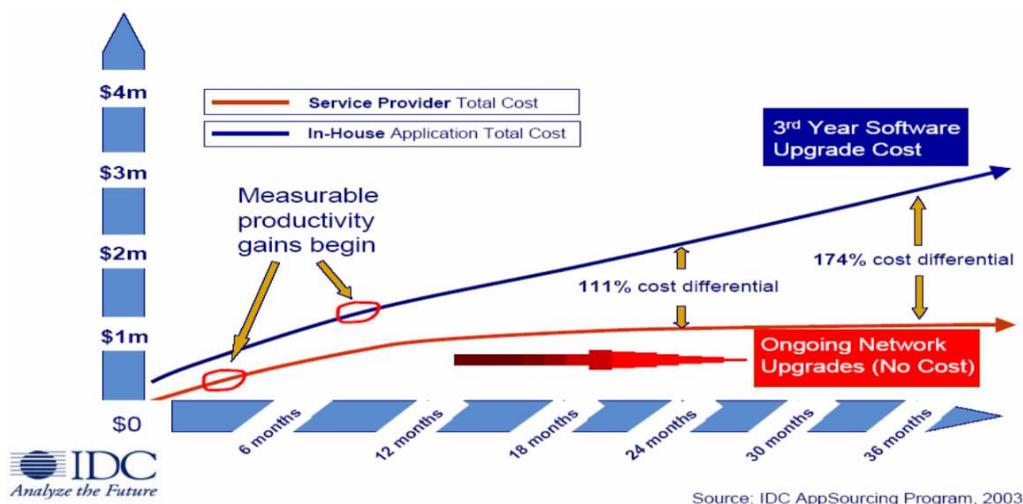


Figura 6. IDC's TCO comparacion entre software tradicional y SaaS

Fuente: Jan Sysmans of WebEx Communications

2.4 Hipótesis

Hipótesis alternativa

El uso de una Solución de Software como Servicio (SAAS) permitió reducir los Costos Totales de Propiedad (TCO) de las PYMES del sector comercio del distrito de Tarapoto.

Hipótesis nula

El uso de una solución de software como servicio (SAAS) no permitió reducir los costos totales de propiedad (TCO) de las PYMES del sector comercio del distrito de Tarapoto.

2.5 Sistema de variables

2.5.1 Variable independiente

Solución de Software como Servicio (SAAS)

2.5.2 Variable dependiente

Costos Totales de Propiedad

2.6 Escala de medición

Tabla 4. Operacionalización de variables

Tipo de Variable	Variable	Indicador	Escala de medición	Instrumento Evaluación.
Dependiente	Costos Totales de Propiedad.	-Costo del Hardware requerido -Costo del Software requerido -Costo de los servicios iniciales para instalación. -Costo de los servicios iniciales para configuración. - Costo de administración	Ordinal (cuantitativo)	Cuestionario. Estudio.
Independiente	Solución de Software como Servicio(SaaS)	Software proporcionado.		

Fuente: Elaboración propia.

2.7 Objetivos

2.7.1 General

Usar software como servicio para reducir los Costos Totales de Propiedad (TCO) que se incurren en las pymes del sector comercio del distrito de Tarapoto usando una Solución de Software como Servicio (SaaS).

2.7.2 Objetivo Especifico

- Construir una solución de software como servicio (SAAS) para la automatización de los procesos comerciales de las PYMES.
- Reducir los Costos Totales de Propiedad (TCO) que se incurren en las pymes del distrito de Tarapoto.
- Medir la reducción de los Costos Totales de Propiedad mediante la utilización de una Solución de Software Como Servicio (SAAS).

Dónde: n, tamaño de la muestra; N, tamaño de la población; Z, desviación normal; p, proporción de la población que posee la característica que se desea saber; la proporción q= 1 – p; E, margen de error que se está dispuesto a aceptar.

Para la aplicación de la fórmula 1 se puede considerar un margen de error E del 5%, y los valores correspondientes de $Z_{\alpha/2} = Z_{0.025} = 1.96$.

Se tiene:

$$n=?$$

$$N=153$$

$$z=1.96$$

$$p=0.9 \text{ (90\%, que darían la información esperada)}$$

$$p=0.1 \text{ (10\%, que no darían la información esperada)}$$

$$E=5\%$$

Aplicando la fórmula 1:

$$n = \frac{153 \times 1.96^2 \times 0.9 \times 0.1}{0.05^2 \times 153 + 1.96^2 \times 0.9 \times 0.1}$$

$$n=73$$

Comprobando, mediante la relación $n_0 = \frac{n}{N} \leq 0.1$, si es mayor se ajusta mediante:

$$n_0 = \frac{n}{1 + \frac{n}{N}}$$

Veamos: $n_0 = \frac{73}{153} = 0.477$, mayor que 0.1 o sea mayor que 10%.

Entonces la muestra corregida será: $n_0 = \frac{73}{1 + \frac{73}{153}} = 49.42 \cong 49$

En conclusión, la muestra definitiva será 49 PYMES que representan el 32% de la población.

3.2 **Ámbito geográfico**

El presente trabajo de investigación se desarrolló en ámbito de la Ciudad de Tarapoto.

3.3 **Diseño de la investigación**

3.3.1 **Tipo de investigación**

El tipo de la investigación es Aplicada o Tecnológica, ya que la investigación pretende aplicar los conocimientos ya existentes en el área de Tecnología de Información.

3.3.2 **Nivel de investigación**

El nivel de la investigación es aplicativo puesto que se pretende desarrollar una Solución SaaS en el que se aplique una serie de conceptos y metodologías orientas a reducir los costos totales de propiedad en el distrito de Tarapoto.

3.3.3 **Diseño de investigación**

El diseño será Cuasi Experimental, puesto que se aplicará la prueba de hipótesis diferencia de medias a la misma muestra, mediante muestras pareadas.

Siendo el diagrama del diseño cuasi experimental.

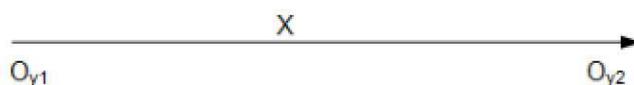


Figura 8: Diagrama del diseño experimental de la investigación

Dónde:

X = Es el experimento o aplicación de la variable independiente (Solución SaaS.)

O_{y1} = Observación de la variable dependiente antes de la aplicación del experimento.

O_{y2} = Observación de la variable dependiente después de la aplicación del experimento.

Así mismo el proyecto comprende el desarrollo de tres fases de acuerdo con el tipo y nivel de investigación correspondiente, las mismas que responden a una metodología que personaliza el desarrollo de la solución SaaS que reduce el costo total de propiedad en el distrito de Tarapoto - Región San Martín.

Primera etapa: Identificación de la Información y análisis de todos los requerimientos por las PYMES en el distrito de Tarapoto.

Segunda etapa: Diseño y modelado de la Arquitectura de la solución SaaS.

Tercera etapa: implementación y medición de los resultados de la solución SaaS. Para esta etapa se adquirirá el servicio de una plataforma IaaS e instalaremos la solución SaaS.

3.4 Procedimientos y técnicas

3.4.1 Procedimientos

A continuación, se expone sucintamente los procedimientos realizados para el desarrollo de la tesis.

Primero se realizó un levantamiento de información y análisis de requerimientos, para conocer los procesos importantes en las PYMES y automatizarlas en el SaaS. Este estudio se hizo durante 2 meses en el período julio-agosto.

Juntamente con el estudio se implementó en el servidor la solución SaaS que permite automatizar los procesos principales en las PYMES, con ello administrarlas desde cualquier lugar las 24 horas del día y los 7 días de la semana. Para la implementación de la solución SaaS se desarrolló de tal manera que podemos tener muchas PYMES con una sola instancia de Software.

La implementación en el servidor de la solución SaaS se realizó en tres etapas:

1°. Análisis

Se utilizó dos herramientas:

Lenguaje unificado de modelado (UML). - Es el lenguaje que se utilizó en el desarrollo del proyecto para poder visualizar, especificar, construir y documentar los diversos gráficos para el sistema. Se eligió a este lenguaje por que se basa en una metodología orientada a

objetos y eso permitirá expresar el modelado de la propuesta de una forma más entendible para los usuarios, además que este es independiente del lenguaje de programación y provee a los desarrolladores un lenguaje de modelado listo para utilizar. Todo esto hace que UML sea la mejor opción para el desarrollo del proyecto.

Proceso Unificado de Rational (RUP). - Es un marco de desarrollo de *software* que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental, aspecto que se adapta a las necesidades del proyecto, permitiendo así poder realizar una propuesta de acuerdo a los requerimientos de la organización.

2°. Desarrollo del sistema

Para el desarrollo del sistema se empleó el lenguaje de programación Python, por su sencillez, rapidez de desarrollo, junto a éste se optó por el sistema de gestión de bases de datos PostgreSQL, puesto que sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado.

3°. Implementación en el servidor

Para la implementación en el servidor se optó alquilar un servidor con Sistema Operativo debian, debido a la ser uno de los S.O más seguros en las distribuciones Linux.

Completado la implementación se configuro correos electrónicos y procesos que se ejecutan todos los días de manera automática, como la ejecución de crear un backup de toda la base de datos.

Posterior a la implementación se procedió ir a visitar y registrar las PYMES, luego procediendo a responder el cuestionario (ver Anexo B), este proceso se hizo durante 7 meses en el periodo de septiembre 2016 – febrero 2017.

Finalmente, se midió el impacto de la solución SaaS en las PYMES, aplicando nuevamente el cuestionario (ver Anexo B).

3.4.2 Técnicas

A continuación, se muestra las técnicas e instrumentos utilizados:

Tabla 5. Técnicas e instrumentos a utilizar en el estudio

Técnica	Justificación	Instrumentos	Aplicado en...
Entrevista	Proporciona información de los actores fundamentales.	Guía de entrevista (Anexo B)	A la muestra que es parte del estudio.
Registros	Proporciona la información del sistema.		Información de salida del sistema.
Análisis de documentos	Para obtener la información de las fuentes secundarias referentes a temas de la investigación.	Fichas bibliográficas. Subrayado.	La bibliografía necesaria para desarrollar el marco teórico y la información complementaria.

Fuente: Elaboración propia

3.5 Instrumentos

3.5.1 Instrumentos de recolección de datos

Se empleó el cuestionario diseñado durante la formulación del proyecto de tesis y que permiten recabar información en campo sobre los indicadores identificados en la operacionalización de variables.

Para el procesamiento de los datos, se plantea un enfoque cuantitativo aplicando un pre test y post test.

Debido a que el diseño será cuasi-experimental y en función de las características cuantitativas de la prueba de hipótesis a realizar, a la misma se aplicó la prueba de hipótesis diferencia de medias (T-Students) mediante muestras pareadas.

3.5.2 Instrumentos de procesamiento de datos

A continuación, se definen las herramientas estadísticas y el programa de computador que se utilizaron para el procesamiento y presentación de los datos.

Procesamiento

- Medidas de tendencia central. - Se utilizó la media aritmética.
- Medidas de dispersión. - Se usó el rango o intervalo y la desviación estándar (S).
- Prueba estadística. - Para la verificación de hipótesis se usó la prueba T-Student que permitió contrastar los datos del patrullaje antes y después de la implementación de la solución de inteligencia de negocios para constatar si existe una mejora significativa entre ambos.
- Programa de computador. - IBM SPSS Statistics, este software proporciona herramientas que permite consultar datos y formular hipótesis para pruebas adicionales de forma rápida, así como ejecutar procedimientos para ayudar a aclarar las relaciones entre variables, genera estadísticos descriptivos, identifica tendencias y realiza predicciones.

Presentación de datos

- Tablas. - Para un mejor entendimiento de los datos se presentaron en tablas distribuyéndolos en columnas y filas, agrupando los datos evitando duplicidades y superando el desorden. Servirán de ayuda visual que permitirán organizar los resultados de la investigación.
- Figuras. - Importantes para expresar la tendencia de un hecho o fenómeno. Se usaron figuras estadísticas como barras, líneas, áreas, etc.

3.6 Prueba de hipótesis

La hipótesis planteada en la presente investigación es:

H1: El uso de una Solución de Software como Servicio (SAAS) permitió reducir los Costos Totales de Propiedad (TCO).

Para realizar la prueba de hipótesis se toma como datos los resultados de los cuestionarios, a partir del cual se presentan la siguiente tabla 6 en la que las columnas del ANTES representan datos que se obtuvieron antes del uso de la Solución de Software como Servicio y la columna DESPUÉS son datos obtenidos ya con el uso de la Solución de Software como Servicio.

Tabla 6. datos recolectados de cuestionario

ANTES		DESPUÉS		DIFERENCIA
PYME	TCO(S/.)	PYME	TCO(S/.)	
01	7119	01	819	6300
02	6379	02	700	5679
03	5959	03	800	5159
04	5810	04	1250	4560
05	5449	05	700	4749
06	5989	06	780	5209
07	3579	07	750	2829
08	4819	08	830	3989
09	5579	09	750	4829
10	6149	10	700	5449
11	6369	11	810	5559
12	6549	12	750	5799
13	6559	13	750	5809
14	6589	14	700	5889
15	5630	15	850	4780
16	5009	16	800	4209
17	5439	17	750	4689
18	7040	18	810	6230
19	6239	19	750	5489
20	7199	20	830	6369
21	6929	21	800	6129
22	6829	22	1250	5579
23	6239	23	830	5409
24	5959	24	750	5209
25	6129	25	800	5329
26	6340	26	750	5590
27	6829	27	750	6079
28	7249	28	1300	5949
29	7679	29	1250	6429
30	6839	30	800	6039
31	6489	31	750	5739
32	6639	32	750	5889
33	7190	33	850	6340
34	5522	34	729	4793
35	6780	35	700	6080
36	5389	36	750	4639
37	6693	37	750	5893
38	6663	38	750	5913
39	6319	39	810	5509
40	5179	40	750	4429

41	5650	41	700	4950
42	5470	42	750	4720
43	6789	43	780	6009
44	6889	44	1250	5639
45	5344	45	750	4594
46	6259	46	830	5429
47	7019	47	1200	5819
48	6739	48	800	5939
49	6759	49	750	6009
Promedio	6250,1020	Promedio	827,7143	5422,38776
Desviación Estándar	766,34759	Desviación Estándar	164,72616	724,03959

Fuente: Elaboración propia.

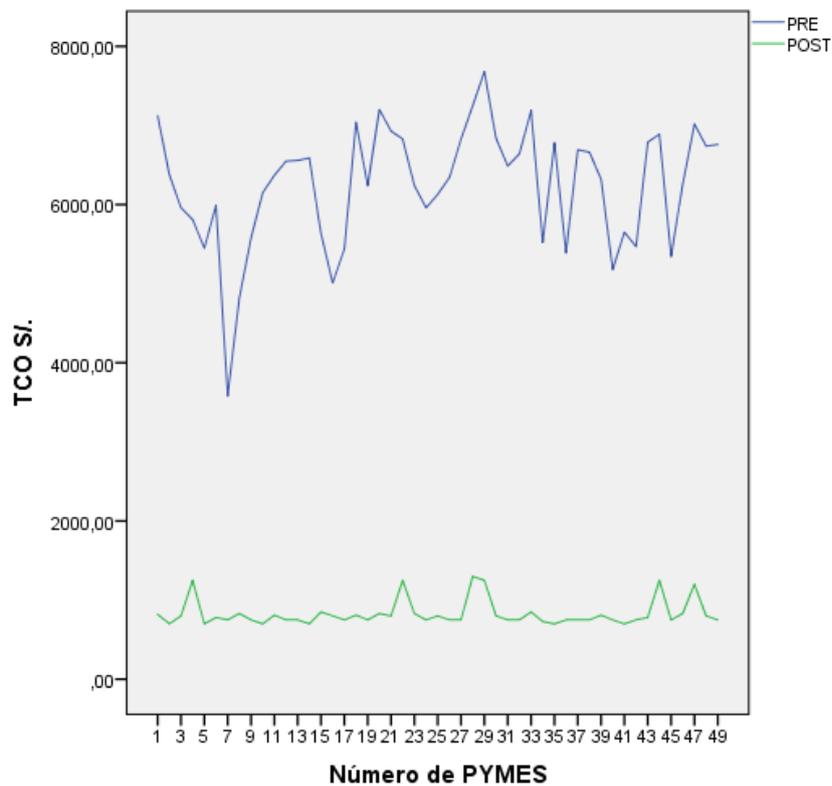


Figura 9: gráfico de PRE y POST del TCO

Fuente: SPSS

En el gráfico 10 podemos ver que el promedio antes (S/ 6250,1020) y después (S/ 827,7143) de usar la solución de software como servicio, con una reducción del 87% del TCO.

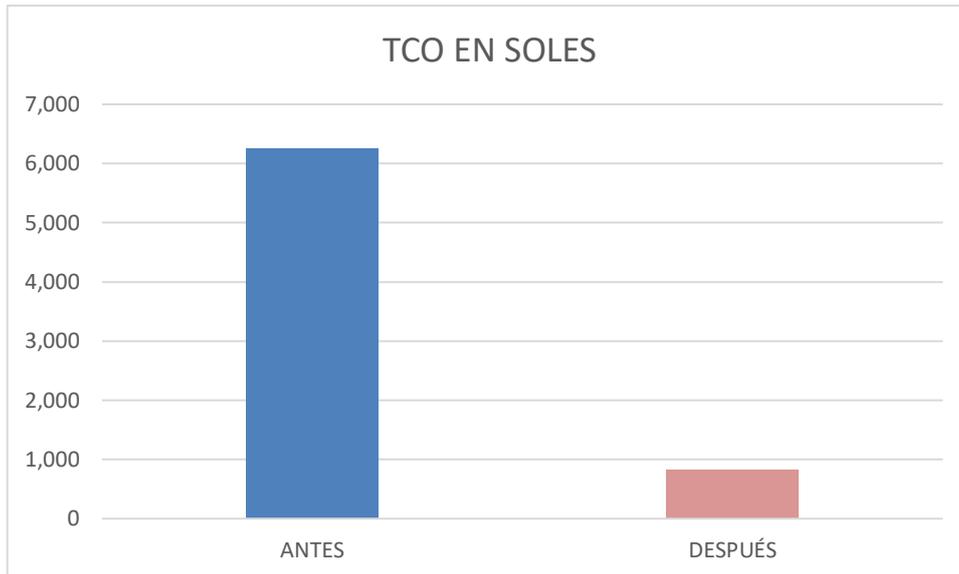


Figura 10: gráfico del promedio de TCO antes y después del SaaS

Fuente: elaboración propia

Cálculo de estadígrafos más importantes calculados a partir de los datos obtenidos ANTES de la implantación del sistema de información:

Tabla 7. cálculo de estadígrafos(ANTES)

Estadígrafo	Fórmula	Valor
Promedio:	$\bar{X} = \sum_{i=1}^N x_i$	6250,1020
Mediana:	$Me = x_{i1} + \left(\frac{\left(\frac{N_m}{2} \right) - N_i - 1}{f_i} \right) \cdot (x_{i2} - x_{i1})$	6369
Desviación estándar:	$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$	766,34759
Coefficiente de variación	$:C_v = \frac{S}{\bar{X}} * 100\%$	12.2614%

Fuente: Elaboración propia.

Cálculo de estadígrafos más importantes calculados a partir de los datos obtenidos DESPUÉS de la implantación del sistema de información:

Tabla 8. cálculo de estadígrafos(DESPUÉS)

Estadígrafo	Fórmula	Valor
Promedio:	$\bar{X} = \sum_{i=1}^N x_i$	827,7143
Mediana:	$Me = x_{i1} + \left(\frac{\left(\frac{N_m}{2} \right) - N_i - 1}{f_i} \right) \cdot (x_{i2} - x_{i1})$	750,0000
Desviación estándar:	$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$	164,72616
Coefficiente de variación	$C_v = \frac{S}{\bar{x}} * 100\%$	19.9013%

Fuente: Elaboración propia.

Tabla 9. Prueba T-Student para comparar resultados del TCO ANTES y DESPUÉS del uso de la Solución de Software como Servicio en las PYMES del sector comercio. Tarapoto – San Martín

H₀: $\mu_a - \mu_d = 0$: los Costos Totales de Propiedad (TCO) no presentan diferencias significativas. H₁: $\mu_a - \mu_d > 0$: los Costos Totales de Propiedad (TCO) ANTES es significativamente mayor al TCO DESPUÉS.

En nuestro caso se dispone de dos grupos de observaciones independientes con diferentes varianzas, la distribución de los datos en cada grupo no puede compararse únicamente en términos de su valor medio, para lo cual utilizamos la fórmula conocido como el test de Welch basada en el estadístico:

$$t = \frac{(\bar{X}_{antes} - \bar{X}_{después})}{\sqrt{\frac{S_{antes}^2}{N_{antes}} + \frac{S_{después}^2}{N_{después}}}} = \frac{(6250,1020 - 827,7143)}{\sqrt{\frac{766,34759^2}{49} + \frac{164,72616^2}{49}}} = 52,423534$$

Medidas Estadísticas	Prueba "t" datos a-pareados	Valor p Significación
$\bar{X}_{antes} = 6250,1020$	T_{calculada} = 52,423534	P = 4,9513E-44 < 0.05
$\bar{X}_{después} = 827,7143$	T_{tabular} = 1.68834	Se Rechaza H ₀
$\bar{d} = 5422,38776$	T_{calculada} > T_{tabular}	Porque el TCO ANTES de la implantación de la Solución de Software como Servicio es mayor que el TCO DESPUÉS de la implantación de la Solución de Software como Servicio
$S_{antes} = 766,34759$		
$S_{después} = 164,72616$		

Fuente: Elaboración propia.

CAPÍTULO IV

IV. RESULTADOS.

En este capítulo se muestra la parte descriptiva de las variables de la investigación para hallar qué hay en los datos y cuánto varían, así como la consecución de los objetivos propuestos.

4.1. Sobre el costo de hardware requerido

El cuestionario aplicado a las PYMES, permitió conocer que el costo total de hardware requerido antes de la aplicación del Software como Servicio obteniendo un promedio de 3200 soles(PRE) y después del uso de Software como Servicio(POST) de 828 soles.

De lo anterior se deduce que el costo de hardware requerido antes de usar la aplicación de Software como Servicio es un 386% superior al promedio después del uso de la aplicación de Software como Servicio.

En la investigación se agruparon las preguntas del cuestionario (ANEXO B) que corresponden al costo de hardware requerido, siendo:

- Servidor/Computadora
- Ticketera
- Lectora de código de barras
- Redes
- Impresora

La Tabla 10 en las columnas del ANTES representan datos que se obtuvieron antes del uso de la Solución de Software como Servicio y la columna DESPUÉS son los datos obtenidos ya con el uso de la Solución de Software como Servicio.

Tabla 10. datos recolectados de cuestionario sobre el costo de hardware requerido

ANTES		DESPUÉS		DIFERENCIA
PYME	TCO(S/.)	PYME	TCO(S/.)	
01	3519	01	819	2700
02	3679	02	700	2979

03	3259	03	800	2459
04	3510	04	1250	2260
05	2649	05	700	1949
06	3389	06	780	2609
07	1079	07	750	329
08	2619	08	830	1789
09	3279	09	750	2529
10	2949	10	700	2249
11	2769	11	810	1959
12	2949	12	750	2199
13	3319	13	750	2569
14	3349	14	700	2649
15	2830	15	850	1980
16	2569	16	800	1769
17	2999	17	750	2249
18	3440	18	810	2630
19	2999	19	750	2249
20	3599	20	830	2769
21	3329	21	800	2529
22	3529	22	1250	2279
23	3439	23	830	2609
24	3159	24	750	2409
25	3329	25	800	2529
26	3540	26	750	2790
27	3229	27	750	2479
28	3649	28	1300	2349
29	4079	29	1250	2829
30	3239	30	800	2439
31	2889	31	750	2139
32	3039	32	750	2289
33	3590	33	850	2740
34	3082	34	729	2353
35	3180	35	700	2480
36	2949	36	750	2199
37	3093	37	800	2293
38	3063	38	750	2313
39	3379	39	810	2569
40	3239	40	750	2489
41	3350	41	700	2650
42	3170	42	750	2420
43	3189	43	780	2409
44	3649	44	1250	2399
45	3204	45	750	2454
46	3459	46	830	2629

47	3419	47	1200	2219
48	3439	48	800	2639
49	3159	49	750	2409
Promedio	3200,3061	Promedio	828,7346	2371,5714
Desviación	430,6170	Desviación	164,3892	393,6823
Estándar		Estándar		

Fuente: Elaboración propia.

En la Figura 11 podemos ver esta notable diferencia en los resultados obtenidos:

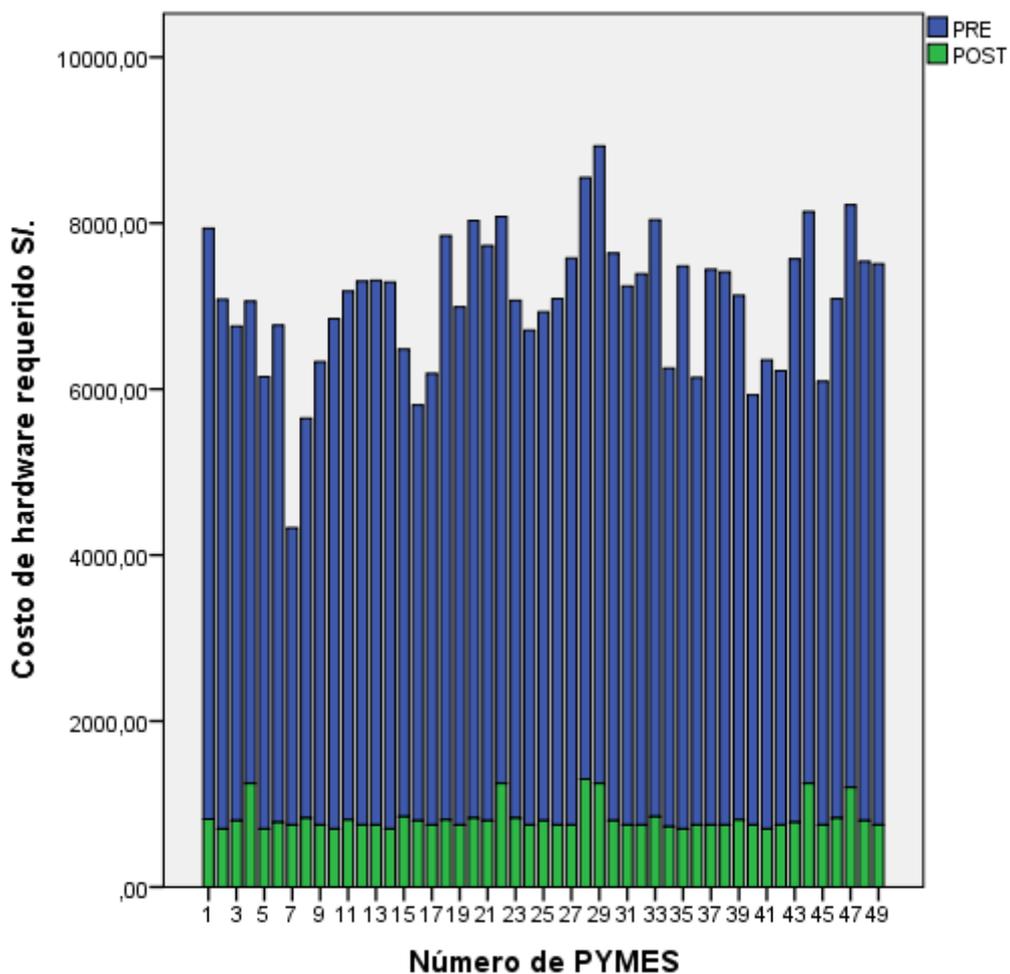


Figura 11: gráfico de PRE y POST del costo de hardware requerido

Fuente: SPSS

4.2. Costo del Software requerido

El cuestionario aplicado a las PYMES, permitió conocer que el costo total del software requerido antes de la aplicación del Software como Servicio obteniendo

un promedio de 1743 soles(PRE) y después del uso de Software como Servicio(POST) de 0 soles.

De lo anterior se deduce que el costo de software requerido antes de usar la aplicación de Software como Servicio es un 1743% superior al promedio después del uso de la aplicación de Software como Servicio.

En la investigación se agruparon las preguntas del cuestionario (ANEXO B) que corresponden al costo de hardware requerido, siendo:

- Software tradicional

La Tabla 11 en las columnas del ANTES representan datos que se obtuvieron antes del uso de la Solución de Software como Servicio y la columna DESPUÉS son los datos obtenidos ya con el uso de la Solución de Software como Servicio.

Tabla 11. datos recolectados de cuestionario sobre el costo de software requerido

ANTES		DESPUÉS		DIFERENCIA
PYME	TCO(S/.)	PYME	TCO(S/.)	
01	2000	01	0	2000
02	1500	02	0	1500
03	1500	03	0	1500
04	1500	04	0	1500
05	1700	05	0	1700
06	1800	06	0	1800
07	1800	07	0	1800
08	1200	08	0	1200
09	1500	09	0	1500
10	2100	10	0	2100
11	2000	11	0	2000
12	2000	12	0	2000
13	2000	13	0	2000
14	2000	14	0	2000
15	1500	15	0	1500
16	1500	16	0	1500
17	1500	17	0	1500
18	2000	18	0	2000
19	2000	19	0	2000
20	2000	20	0	2000
21	2000	21	0	2000
22	2000	22	0	2000

23	1500	23	0	1500
24	1500	24	0	1500
25	1500	25	0	1500
26	1500	26	0	1500
27	2000	27	0	2000
28	2000	28	0	2000
29	2000	29	0	2000
30	2000	30	0	2000
31	2000	31	0	2000
32	2000	32	0	2000
33	2000	33	0	2000
34	1500	34	0	1500
35	2000	35	0	2000
36	1500	36	0	1500
37	2000	37	0	2000
38	2000	38	0	2000
39	1800	39	0	1800
40	1000	40	0	1000
41	1000	41	0	1000
42	1000	42	0	1000
43	2000	43	0	2000
44	2000	44	0	2000
45	1200	45	0	1200
46	1500	46	0	1500
47	2000	47	0	2000
48	1800	48	0	1800
49	2000	49	0	2000
Promedio	1742,8571	Promedio	0	1742,8571
Desviación	316,2277	Desviación	0	316,2277
Estándar		Estándar		

Fuente: Elaboración propia.

En la Figura 12 podemos ver esta notable diferencia en los resultados obtenidos:

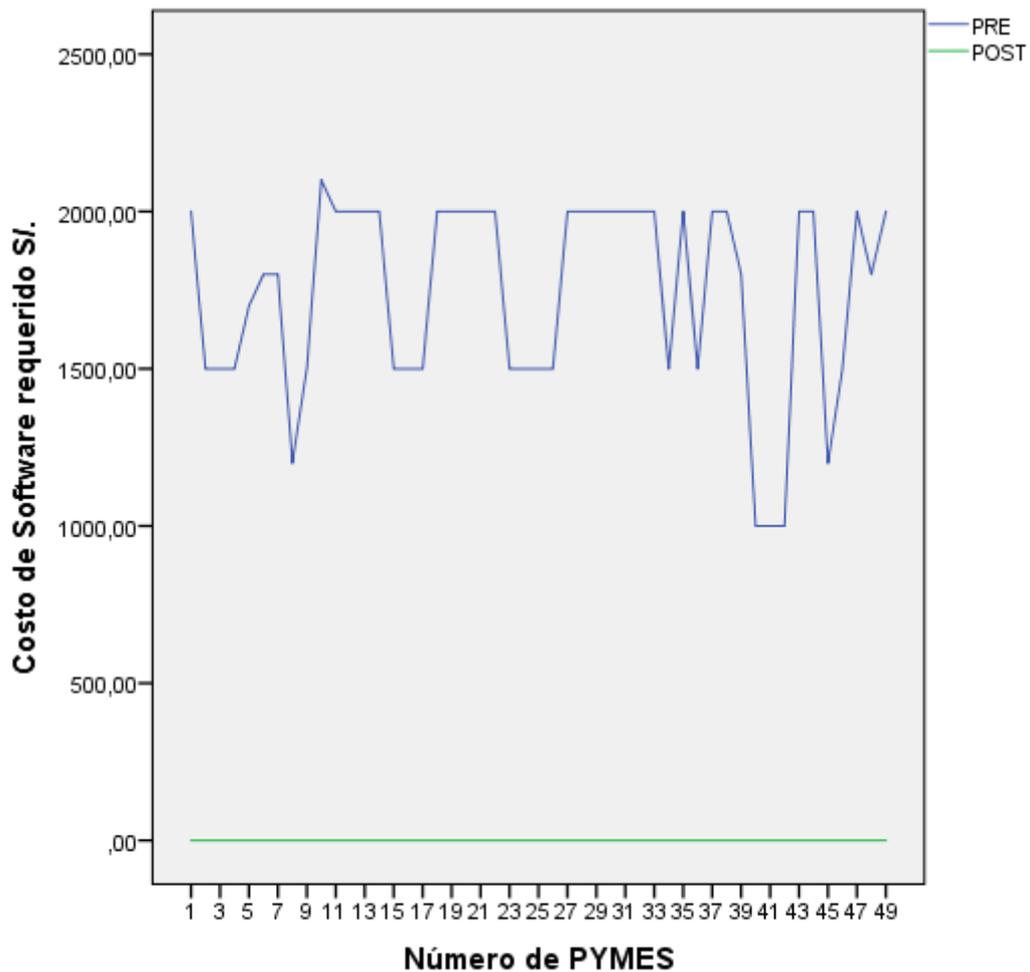


Figura 12: gráfico de PRE y POST del costo de software requerido

Fuente: SPSS

4.3. Costo de los servicios iniciales para la instalación

El cuestionario aplicado a las PYMES, permitió conocer que el costo total de los servicios iniciales para la instalación antes de la aplicación del Software como Servicio obteniendo un promedio de 661 soles(PRE) y después del uso de Software como Servicio(POST) de 0 soles.

De lo anterior se deduce que el costo de software requerido antes de usar la aplicación de Software como Servicio es un 661% superior al promedio después del uso de la aplicación de Software como Servicio.

En la investigación se agruparon las preguntas del cuestionario (ANEXO B) que corresponden al costo total de los servicios iniciales para la instalación, siendo:

- Costos de instalación

La Tabla 12 en las columnas del ANTES representan datos que se obtuvieron antes del uso de la Solución de Software como Servicio y la columna DESPUÉS son los datos obtenidos ya con el uso de la Solución de Software como Servicio.

Tabla 12. datos recolectados de cuestionario sobre los costos iniciales para la instalación

ANTES		DESPUÉS		DIFERENCIA
PYME	TCO(S/.)	PYME	TCO(S/.)	
01	800	01	0	800
02	500	02	0	500
03	500	03	0	500
04	500	04	0	500
05	800	05	0	800
06	500	06	0	500
07	500	07	0	500
08	500	08	0	500
09	500	09	0	500
10	800	10	0	800
11	800	11	0	800
12	800	12	0	800
13	800	13	0	800
14	800	14	0	800
15	500	15	0	500
16	500	16	0	500
17	500	17	0	500
18	800	18	0	800
19	800	19	0	800
20	800	20	0	800
21	800	21	0	800
22	500	22	0	500
23	500	23	0	500
24	500	24	0	500
25	500	25	0	500
26	500	26	0	500
27	800	27	0	800
28	800	28	0	800
29	800	29	0	800
30	800	30	0	800
31	800	31	0	800
32	800	32	0	800
33	800	33	0	800
34	500	34	0	500

35	800	35	0	800
36	500	36	0	500
37	800	37	0	800
38	800	38	0	800
39	700	39	0	700
40	500	40	0	500
41	500	41	0	500
42	500	42	0	500
43	800	43	0	800
44	800	44	0	800
45	500	45	0	500
46	500	46	0	500
47	800	47	0	800
48	700	48	0	700
49	800	49	0	800
Promedio	1742,8571	Promedio	0	1742,8571
Desviación	316,2277	Desviación	0	316,2277
Estándar		Estándar		

Fuente: Elaboración propia.

En la Figura 13 podemos ver esta notable diferencia en los resultados obtenidos:

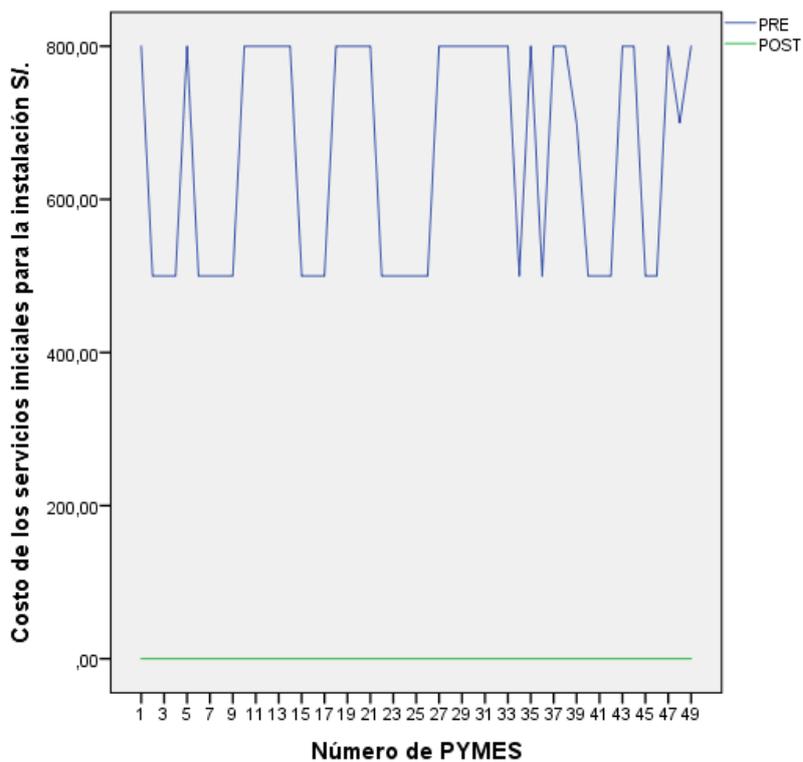


Figura 13: gráfico de PRE y POST de los costos de los servicios iniciales para la instalación

Fuente: SPSS

4.4. Costos de los servicios iniciales para la configuración

El cuestionario aplicado a las PYMES, permitió conocer que el costo total de los servicios iniciales para la configuración antes de la aplicación del Software como Servicio obteniendo un promedio de 202 soles(PRE) y después del uso de Software como Servicio(POST) de 0 soles.

De lo anterior se deduce que el costo de software requerido antes de usar la aplicación de Software como Servicio es un 202% superior al promedio después del uso de la aplicación de Software como Servicio.

En la investigación se agruparon las preguntas del cuestionario (ANEXO B) que corresponden al costo total de los servicios iniciales para la instalación, siendo:

- Costos de configuración

La Tabla 13 en las columnas del ANTES representan datos que se obtuvieron antes del uso de la Solución de Software como Servicio y la columna DESPUÉS son los datos obtenidos ya con el uso de la Solución de Software como Servicio.

Tabla 13. datos recolectados de cuestionario sobre los costos iniciales para la configuración

ANTES		DESPUÉS		DIFERENCIA
PYME	TCO(S/.)	PYME	TCO(S/.)	
01	200	01	0	200
02	200	02	0	200
03	300	03	0	300
04	200	04	0	200
05	200	05	0	200
06	200	06	0	200
07	200	07	0	200
08	200	08	0	200
09	200	09	0	200
10	200	10	0	200
11	200	11	0	200
12	200	12	0	200
13	200	13	0	200
14	200	14	0	200
15	200	15	0	200
16	200	16	0	200

17	200	17	0	200
18	200	18	0	200
19	200	19	0	200
20	200	20	0	200
21	200	21	0	200
22	200	22	0	200
23	200	23	0	200
24	200	24	0	200
25	200	25	0	200
26	200	26	0	200
27	200	27	0	200
28	200	28	0	200
29	200	29	0	200
30	200	30	0	200
31	200	31	0	200
32	200	32	0	200
33	200	33	0	200
34	200	34	0	200
35	200	35	0	200
36	200	36	0	200
37	200	37	0	200
38	200	38	0	200
39	200	39	0	200
40	200	40	0	200
41	200	41	0	200
42	200	42	0	200
43	200	43	0	200
44	200	44	0	200
45	200	45	0	200
46	200	46	0	200
47	200	47	0	200
48	200	48	0	200
49	200	49	0	200
Promedio	202,0408	Promedio	0	202,0408
Desviación	14,2857	Desviación	0	14,2857
Estándar		Estándar		

Fuente: Elaboración propia.

En la Figura 14 podemos ver esta notable diferencia en los resultados obtenidos:

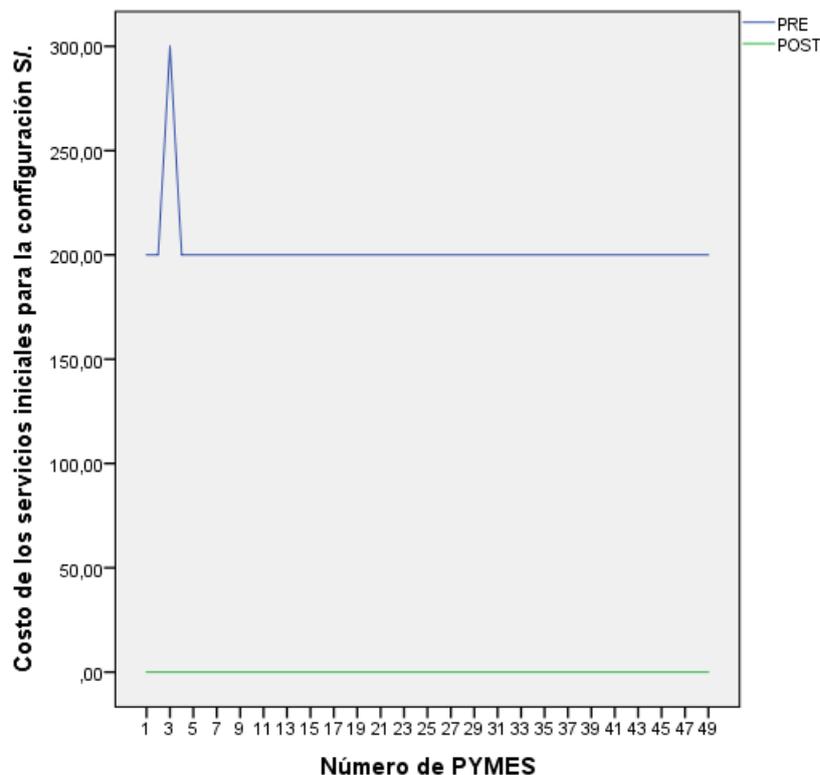


Figura 14: gráfico de PRE y POST de los costos de los servicios iniciales para la configuración

Fuente: SPSS

4.5. Costos de administración de software proporcionado

El cuestionario aplicado a las PYMES, permitió conocer que el costo total de los costos de administración de software proporcionado antes de la aplicación del Software como Servicio obteniendo un promedio de 444 soles(PRE) y después del uso de Software como Servicio(POST) de 0 soles.

De lo anterior se deduce que el costo de software requerido antes de usar la aplicación de Software como Servicio es un 444% superior al promedio después del uso de la aplicación de Software como Servicio.

En la investigación se agruparon las preguntas del cuestionario (ANEXO B) que corresponden al costo total de los servicios iniciales para la instalación, siendo:

- Gastos en soporte de software

La Tabla 14 en las columnas del ANTES representan datos que se obtuvieron antes del uso de la Solución de Software como Servicio y la columna DESPUÉS son los datos obtenidos ya con el uso de la Solución de Software como Servicio.

Tabla 14. datos recolectados de cuestionario sobre los costos de administración de software proporcionado

ANTES		DESPUÉS		DIFERENCIA
PYME	TCO(S/.)	PYME	TCO(S/.)	
01	600	01	0	600
02	500	02	0	500
03	400	03	0	400
04	100	04	0	100
05	100	05	0	100
06	100	06	0	100
07	0	07	0	0
08	300	08	0	300
09	100	09	0	100
10	100	10	0	100
11	600	11	0	600
12	600	12	0	600
13	240	13	0	240
14	240	14	0	240
15	600	15	0	600
16	240	16	0	240
17	240	17	0	240
18	600	18	0	600
19	240	19	0	240
20	600	20	0	600
21	600	21	0	600
22	600	22	0	600
23	600	23	0	600
24	600	24	0	600
25	600	25	0	600
26	600	26	0	600
27	600	27	0	600
28	600	28	0	600
29	600	29	0	600
30	600	30	0	600
31	600	31	0	600
32	600	32	0	600
33	600	33	0	600
34	240	34	0	240
35	600	35	0	600
36	240	36	0	240
37	600	37	0	600
38	600	38	0	600
39	240	39	0	240

40	240	40	0	240
41	600	41	0	600
42	600	42	0	600
43	600	43	0	600
44	240	44	0	240
45	240	45	0	240
46	600	46	0	600
47	600	47	0	600
48	600	48	0	600
49	600	49	0	600
Promedio	443,6735	Promedio	0	443,6735
Desviación	203,0055	Desviación	0	203,0055
Estándar		Estándar		

Fuente: Elaboración propia.

En la Figura 15 podemos ver esta notable diferencia en los resultados obtenidos:

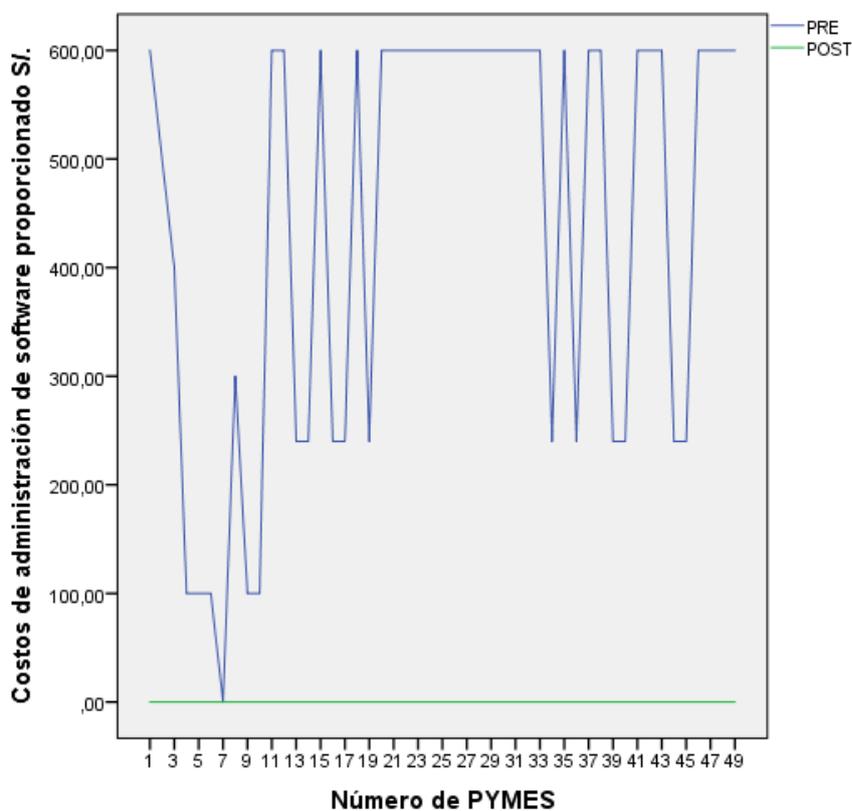


Figura 15: gráfico de PRE y POST de los costos de administración de software proporcionado

Fuente: SPSS

CAPÍTULO V

V. DISCUSIÓN DE LOS RESULTADOS

En este capítulo realizamos el análisis de los resultados obtenidos para responder las preguntas formuladas al inicio del proyecto, así como medir el logro de los objetivos propuestos.

En la investigación sobre Software como servicio; una visión integral del coste total de propiedad de aplicaciones de software, nos dice que el costo anual de poseer y gestionar las aplicaciones de software puede ser hasta cuatro veces el costo de la compra inicial, verificando los resultados obtenidos por el cuestionario **(ANEXO B)** en nuestra investigación tenemos un promedio del ANTES de S/. 6250,1020 y un DESPUÉS de S/. 827,7143 **(Tabla 6)**, con los resultados obtenidos verificamos y corroboramos que el costo es más de siete veces ANTES de aplicar el uso de Software como servicio en la PYMES del sector comercio del distrito de Tarapoto.

Bladh & Strom (2008) en su tesis sobre el Costo total de propiedad, concluye que encontró que ocho elementos de costo constituyen el 90% del costo total de propiedad. De acuerdo a nuestro cuestionario **(ANEXO B)** aplicado ANTES Y DESPUÉS a las PYMES del sector comercio del distrito de Tarapoto, afirmamos que el 89% de todas las preguntas constituyen el costo total de propiedad, siendo la única pregunta, para nuestro caso la ticketera que se usa tanto antes de usar el Software como Servicio y después de usar el Software como servicio.

Brivo OnAir (2015) en su investigación sobre cómo el software como servicio (SaaS) reduce el coste total de propiedad (TCO) para los sistemas de seguridad física, concluye que los ahorros proporcionados por el SaaS pueden extenderse a otros servicios de seguridad, tales como vídeos alojados, detección de intrusiones, monitorización remota y muchos otros, después de haber aplicado nuestra investigación verificamos que se reduce los costos totales de propiedad más de 7 veces, después del uso del software como servicio; entonces esto nos abre muchas posibilidades de poder hacer microservicios como SaaS para las PYMES del sector comercio del distrito de Tarapoto y todo el Perú (Facturas electrónicas, contabilidad, mail marketing, etc.).

VMWARE (2006) en su investigación sobre la reducción de costos totales de propiedad del servidor con el software de virtualización de VMware llegando a la conclusión de que se puede sorprender a los clientes aún más es el ahorro de estos clientes con experiencia en operaciones de TI, experimentando ahorros de TCO en las operaciones de TI de 67%; en nuestra investigación las PYMES del sector comercio del distrito de Tarapoto al no cobrar por instalación, configuración, administración y hardware experimentaron un ahorro del 87% (**Figura 10**), permitiendo que ese ahorro de dinero se pueda direccionar hacia nuevos activos de la PYME.

Con respecto al uso de Software como Servicio para reducir los costos totales de propiedad que se incurren en las PYMES del sector comercio del distrito de Tarapoto, se puede decir que influyó de manera positiva pues ahorrar un 87% con respecto al promedio y además permite que cualquier PYME pueda acceder a una solución de software por un costo bajo que mejorará la administración y control de la PYME.

La prueba de verificación de hipótesis permitió comprobar que existió una diferencia significativa entre la frecuencia del costo total de propiedad (TCO) antes y después de implementar la solución de Software como Servicio, por lo que se puede afirmar que: EL USO DE UNA SOLUCIÓN DE SOFTWARE COMO SERVICIO PERMITIÓ REDUCIR LOS COSTOS TOTALES DE PROPIEDAD DE LAS PYMES DEL SECTOR COMERCIO EN EL DISTRITO DE TARAPOTO.

CAPÍTULO VI

VI. CONCLUSIONES

Las principales conclusiones de la investigación son las siguientes:

- 1°. Se verificó la hipótesis de trabajo, por lo tanto, se afirma que: EL USO DE UNA SOLUCIÓN DE SOFTWARE COMO SERVICIO PERMITIÓ REDUCIR LOS COSTOS TOTALES DE PROPIEDAD DE LAS PYMES DEL SECTOR COMERCIO EN EL DISTRITO DE TARAPOTO.
- 2°. Se implementó satisfactoriamente una aplicación de software como servicio acorde a las necesidades de las PYMES del sector comercio del distrito de Tarapoto, aplicando tecnologías como Multi-Tenant, SaaS.
- 3°. Antes del uso de la solución de Software como Servicio, las PYMES no podían acceder a un software de administración, no lo hacían por que los costos totales de propiedad para soluciones tradicionales son exagerados para una PYME del sector comercio del distrito de Tarapoto.
- 4°. Después del uso de la solución de Software como Servicio, permitió acceder a PYMES del sector comercio del distrito de Tarapoto que nunca habían usado una solución de software para la administración de su PYME por tener unos costos elevados y también permitió reducir en un 87% el TCO de los que ya habían usado un software para la administración y control de su PYME.
- 5°. La solución de software como servicio tuvo un impacto muy positivo sobre las PYMES, esto debido a que ahorró un 87% de dinero con relación al promedio, haciendo que todo ese ahorro se pueda utilizar en activos para la PYME, como también poder administrarla desde cualquier lado, dando más seguridad al micro-empresario.
- 6°. La propuesta de tarifas para el cobro mensual de la solución de Software como Servicio es de dos modalidades mensual y anual que

van a variar de acuerdo a las especificaciones y características de los servidores alquilados para el servicio.

Para nuestro caso, se tuvo financiación con fondos de la UNSM-T a través de la Oficina de Investigación para la implementación de la presente tesis.

CAPÍTULO VII

VII. RECOMENDACIONES

Las principales recomendaciones son:

- 1°. Que las PYMES del sector comercio del distrito de Tarapoto comiencen a usar un software como servicio para la administración y control de sus procesos principales dentro de su negocio.
- 2°. Que las PYMES comiencen a usar las nuevas tecnologías como el cloud computing para aumentar la competitividad de las PYMES del sector comercio del distrito de Tarapoto.
- 3°. Que el dinero ahorrado por el uso de software como servicio se use exclusivamente para mejorar el servicio y así generar más activos para las PYMES del sector comercio de Tarapoto.
- 4°. Que la Universidad Nacional de San Martín, como institución que tiene como fin la investigación, promueva más investigaciones de este tipo en coordinación con otras instituciones del estado y así contribuir a desarrollo y bienestar de la sociedad.
- 5°. Los datos obtenidos por la investigación sirvan para futuras investigaciones sobre Cloud Computing y desarrollar estudios de como las nuevas tecnologías influyen considerablemente en la reducción de los costos totales de propiedad.

VIII. REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

- Bernabeu, R. D. (Julio de 2010). HEFESTO: Metodología para la Construcción de una Data Warehouse. Córdoba, Argentina.
- Bezemer, C., & Zaidman, A. (2010). Challenges of reengineering into multi-tenant SaaS applications.
- Bladh, A., & Ström, A. (2008). Total Cost of Ownership.
- Brivo OnAir. (2015). TOTAL COST OF OWNERSHIP.
- Bustamante Mosquera, S. L. (25 de Enero de 2011). Metodología Para El Rediseño De Los Cuadrantes Utilizados Por Carabineros De Chile En El Plan Cuadrante De Seguridad Preventiva. Santiago, Region Metropolitana, Chile.
- Cai, H., Zhang, K., Zhou, M., Gong, W., Cai, J., & Mao, X. (2009). An End-to-End Methodology and Toolkit for Fine Granularity SaaS-ization.
- D Weissman, C., & Bobrowski, S. (2011). The Design of the Force.com Multitenant Internet.
- Microsoft. (2013). Federation Guidelines and limitation.
- Mietzner , R., & Leymann, F. (s.f.). Towards Provisioning the Cloud: On the Usage of Multi-Granularity Flows and Services to Realize a Unified Provisioning Infrastructure for SaaS Applications. Honolulu, HI, 2011.
- Mietzner , R., Unger, T., Titze, R., & Leymann, F. (2009). Combining Different Multi-tenancy Patterns in Service-Oriented Applications. Auckland.
- Naeem Khan, M. (2013). Object-Relational Mapping Framework to Enable Multi-Tenancy Attributes in SaaS Applications.
- Pathirage, M., Perera, S., Kumara , I., & Weerawarana, S. (2011). A Multi-tenant Architecture for Business Process Executions. Washington, DC.
- Pradeep Kumar, A., V., V., & S., P. (2010). Configurability in SaaS for an Electronic.
- SalesForce. (2011). The multitenant, metadata-driven architecture of Databasedotcom.
- Schiller, O., Schiller, B., Brodt, A., & Mitschang, B. (2011). Native Support of Multi-tenancy in RDBMS.
- Shafique Hijazee, M. (2015). Mastering Google App Engine.

- Systemans de WebEx Communications. (2006). A Comprehensive Look at the Total Cost of.
- The Art of Service. (2015). Cloud Computing Certification Kit - Software as a Service and Web.
- Truyen, E., Cardozo, N., Walraven, S., Vallejos, J., Bainomugisha2, E., & Günthe, S. (2012). Context-oriented Programming for Customizable SaaS.
- Viswanathan, K. (2012). Right Engineering SaaS.
- VMWARE. (2006). Reducing Server Total Cost of Ownership with VMware Virtualization Software.
- W. Rittinghouse, J., & F. Ransome, J. (2009). Cloud Computing: Implementation, Management, and Security.

IX. ANEXOS

ANEXO A. EMPRESAS DEL SECTOR COMERCIO

N°	NOMBRE DE NEGOCIO	DIRECCIÓN	ACTIVIDAD	RUC
1	REPRESENTACIONES STYLOS MIGUELITO-QH E.I.R.L	JR. ALONSO DE ALVARADO N°784 JR. SAN MARTIN N° 352-BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR	20600119827
2	BOUTIQUE "MI BEBE"	CERCADO	VENTA DE ROPAS PARA NIÑOS	1125043
3	GRUPO SERVICE DEL ORIENTE S.A.C	AV. VIA DE EVITAMIENTO N° 1599-BARRIO HUAYCO	INVESTIGACION Y SEGURIDAD	20493974379
4	AGUJETAS E.I.R.L.	JR. AUGUSTO B. LEGUIA N° 402-BARRIO COMERCIO	VENTA DE ROPA DEPORTIVA	20600287151
5	BROOKE CLOTHES	JR. JIMENEZ PIMENTEL N° 625-BARRIO HUAYCO	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	10456531143
6	NOVEDADES KARY	JR. LIMA N° 495-BARRIO COMERCIO	VENTA DE ROPA	10469360607
7	NOVEDADES CUBAS	JR. ANDRES AVELINO CACERES N° 355- BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	10104537087
8	MAMITAS, NENES & NENITOS	JR. SAN MARTIN N° 383 - BARRIO CERCADO	VENTA DE ROPA EXCLUSIVA PARA BEBES Y MAMAS GESTANTES	10453733918
9	TALLER DE MODAS ELENITA	JR. JUAN VARGAS N° 193-BARRIO HUAYCO	CONFECCION DE PRENDAS DE VESTIR PARA DAMAS, CABALLEROS Y NIÑOS	10010604724
10	CENTRO COMERCIAL CUBAS	JR. DANIEL A. CARRION N°310-BARRIO COMERCIO	VENTAS DE PRENDAS DE VESTIR Y ACCESORIOS	10011288231
11	TEJIDOS "SANTA JACINTA S.R.L"	JR. PEDRO DE URZUA N° 164-BARRIO CERCADO	VENTA DE ROPAS	20107807994
12	DOLY	JR. ALFONSO UGARTE N° 321-BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR	10101894652
13	LIAM'S BOUTIQUE	JR. ALEGRIA A. DE MOREY N° 395- BARRIO SUCHICHE	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	10456531135
14	ALESSANDRA SHOPPER	JR. MANUELA MOREY N° 191 - BARRIO CERCADO	VENTA DE ROPA	10456010780
15	CONFECCIONES SILVIA	JR. ALONSO DE ALVARADO N° 738 - BARRIO COMERCIO	VENTA DE CONFECCIONES, PRENDAS DE VESTIR	10011151251
16	NOVEDADES KEYNER	JR. ALONSO DE ALVARADO N° 609 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR	10272961838
17	NOVEDADES ANGELIE	JR. ALONSO DE ALVARADO N° 551 - BARRIO COMERCIO	VENTA DE ROPA, LENCERIA Y ACCESORIOS	10445382375
18	ALESSANDRA FASHION	JR. LIMA N° 270 - BARRIO COMERCIO	VENTA DE ROPA	10103556541
19	PAÑALES KIDS E.I.R.L	JR. ALFONSO UGARTE N° 291- BARRIO CERCADO	VENTA DE ROPA EXCLUSIVA PARA BEBES Y NIÑOS	10103556542
20	PASARELLA ANGELIE	JR. LIMA N° 272 - BARRIO COMERCIO	VENTA DE ROPA, LENCERIA Y ACCESORIOS	10445382375

21	NOVEDADES JORGITO	JR. JIMENEZ PIMENTEL N°343-BARRIO CERCADO	VENTA DE ROPAS PARA DAMAS, CABALLEROS Y NIÑOS	10450104871
22	NOVEDADES TREICY	JR. LOS GIRASOLES N° 176 - BARRIO LOS JARDINES	VENTA DE ROPA	10434946684
23	VESTUARIOS FOLKLORICOS	JR. ALONSO DE ALVARADO N° 128 - BARRIO CERCADO	ALQUILER DE VESTIMENTA FOLCLORICAS	1106552
24	ALAN MARTIN D'CLAUD BOUTIQUE & ACCESORIOS	JR. BOLOGNESI N° 143 - BARRIO SUCHICHE	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS PARA DAMAS	10011495899
25	NOVEDADES CASTILLO	JR. DANIEL ALCIDES CARRION N° 359 - BARRIO PARTIDO ALTO	VENTA DE ROPAS DE DAMAS Y CABALLEROS	10730845451
26	SURFING SURF SHOP	JR. ALFONSO UGARTE N° 566 - BARRIO COMERCIO	VENTA DE VESTUARIO	10407275786
27	CONFECCIONES ANDREA	JR. ANTONIO RAYMONDI N° 537 - BARRIO HUAYCO	CONFECCION DE ROPA DEPORTIVA	10412708828
28	SUPER MODA	JR. ALONSO DE ALVARADO N° 389 - BARRIO COMERCIO	VENTA DE ROPA	10420226433
29	CORPORACION WAMA S.A.C	JR. LIMA N° 225 - BARRIO COMERCIO	COMERCIALIZACION Y VENTA DE PRENDAS DE VESTIR	20374343964
30	BAZAR BOUTIQUE Y REGALOS MCV	JR. JOSE OLAYA N° 784 - BARRIO HUAYCO	VENTA DE ROPA PARA DAMAS, CABALLEROS Y NIÑOS	10703486415
31	LOS CRISTALES	JR. LEONCIO PRADO N° 841 - BARRIO PARTIDO ALTO	CONFECCION DE PRENDAS	10011248841
32	MUAH!	JR. ALFONSO UGARTE N° 354 - BARRIO COMERCIO	VENTA DE ROPA INTERIOR	10106818164
33	CHRISMARY NOVEDADES	JR. SAN PABLO DE LA CRUZ N° 392 - BARRIO SUCHICHE	VENTA DE ROPA	10804368901
34	PAO SHOWROOM	JR. DANIEL ALCIDES CARRION N° 253 - BARRIO COMERCIO	VENTA DE ROPAS Y ACCESORIOS	10700721618
35	CERO DRAMA S.A.C	JR. JOSE OLAYA N° 765 - BARRIO HUAYCO	VENTA DE ROPA, ACCESORIOS PARA DAMAS Y CABALLEROS	20600502485
36	SAURIOS STORE	JR. JORGE CHAVEZ N° 663 - BARRIO HUAYCO	VENTA DE ROPA Y ACCESORIOS	10435853221
37	RUPAY	JR. SAN MARTIN N° 499 - BARRIO COMERCIO	VENTA DE ROPA ARTESANAL	10011311976
38	RAFAELLA	JR. CAHUIDE N° 217 - BARRIO SUCHICHE	COMPRA Y VENTA DE PRENDAS DE VESTIR	10472686262
39	FLOREZITAS	JR. ANTONIO RAYMONDI N° 264 - A - BARRIO CERCADO	VENTA DE ROPA DE VESTIR	10010607740
40	CATHYA	JR. SAN MARTIN N° 372 - BARRIO PARTIDO ALTO	VENTA DE ROPAS Y ACCESORIOS POR CATALOGO	10009543622
41	DISTRIBUCIONES MONDRAGON	JR. ALFONSO UGARTE N° 315 - BARRIO COMERCIO	VENTA DE ROPA	10467662894
42	LESFY FASHION	JR. ALONSO DE ALVARADO N° 730 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR	10407814083

43	BAZAR BOUTIQUE Y REGALOS MCV	JR. JOSE OLAYA N° 784 - BARRIO HUAYCO	VENTA DE ROPAS PARA DAMAS CABALLEROS Y NIÑOS ELABORACION DE ARTICULOS PROMOCIONALES Y VENTA DE PRENDAS DE VESTIR	10011455218
44	IDEAS COLORES S.A.C	JR. VICTOR MANUEL AREVALO DELGADO N° 149 - BARRIO PARTIDO ALTO		20572188869
45	PEKELANDIA	JR. ALONSO DE ALVARADO N° 709 - BARRIO COMERCIO	VENTA DE ROPAS Y ARTICULOS PARA NIÑOS	10436317447
46	NOVEDADES ISABEL	JR. BOLOGNESI N° 174 - BARRIO CERCADO	VENTA DE ROPA PARA DAMAS CABALLEROS Y NIÑOS	10447869212
47	NENA'S BOUTIQUE	JR. GREGORIO DELGADO N° 135 - BARRIO CERCADO	VENTA DE ROPA	10011233932
48	SHARON	JR. LIMA N° 111 - BARRIO COMERCIO	VENTA DE ROPA Y ACCESORIOS	10454645494
49	RAIZA Y RENZO MODAS	JR. GREGORIO DELGADO N° 211 - BARRIO CERCADO	VENTA DE ROPAS PARA NIÑOS, ADULTOS Y REGALOS	10081472641
50	NOVEDADES KING & TIN	JR. COMANDANTE CHIRINOS N° 436 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR	10451135053
51	NOVEDADES "BABY FASHION"	JR. MARTINEZ DE COMPAGÑON N° 291 - BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR PARA BEBES Y NIÑOS	10422978904
52	"GLAM"	JR. MARTINEZ DE COMPAGÑON N° 417 - BARRIO CERCADO	VENTA DE ROPA	10010750771
53	MAJOSE	JR. LIMA N° 634 - 2DO PISO - SATND N° 15 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR	10011011026
54	LA CHIC	JR. MARTINEZ DE COMPAGÑON N° 331 - BARRIO HUAYCO	VENTA DE ROPA Y ACCESORIOS	10447524983
55	CHIOS BOUTIQUE	JR. RAMON CASTILLA N° 222 - BARRIO CERCADO	VENTA DE ROPA AL POR MENOR	10011477742
56	LA FLORCITA	JR. LIMA N° 634 - 2DO PISO - STAND 14 - BARRIO COMERCIO	CONFECCION DE PRENDAS DE VESTIR	10010699059
57	KATTALEYA BOUTIQUE	JR. AUGUSTO B. LEGUIA N° 415 - BARRIO COMERCIO	VENTAS DE ROPA Y ACCESORIOS	10167226481
58	DESARROLLO COMERCIALES EUROAMAZONAS S.A.C.	JR. RAMON CASTILLA N° 145 - BARRIO CERCADO	VENTA DE ROPA Y ACCESORIOS PARA DAMAS	20600806093
59	NEGOCIOS UNFAU-OP E.I.R.L	JR. SANTA ROSA N° 390 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	20450131467
60	INVERSIONES GECAR E.I.R.L	JR. RAMON CASTILLA N° 294 - BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	20542357984
61	ELEGANCIA	JR. JOSE OLAYA N° 249 - BARRIO HUAYCO	ALQUILER DE VESTIDOS	10461660211
62	J Y L	JR. ALONSO DE ALVARADO N° 625 - BARRIO COMERCIO	VENTA DE ROPA	10742765968
63	I LOVE YOUR STYLE	JR. RAMON CASTILLA N° 432 - BARRIO HUAYCO	COMERCIO - VENTA DE ROPA	10459576105

64 NOVEDADES IVETH	JR. ALONSO DE ALVARADO N° 430 - BARRIO COMERCIO	VENTA DE ROPA	10479654838
65 NOVEDADES FIORELLA	JR. ALFONSO UGARTE N° 335 - BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR	10297319022
66 LIMA FASHION	JR. ALFONSO UGARTE N° 412 - BARRIO COMERCIO	VENTA AL POR MENOR DE PRENDAS DE VESTIR	10452146733
67 POLOS HADA E.I.R.L	JR. JIMENEZ PIMENTEL N° 319 - BARRIO CERCADO	FABRICACION Y COMERCIALIZACION DE ROPA DEPORTIVA	20362395063
68 MODAS NANSI	JR. ALFONSO UGARTE N° 544 - BARRIO HUAYCO	VENTA Y CONFECCION DE ROPA DEPORTIVA Y DE VESTIR	10010734377
69 CONFECCIONES JUAN ANGEL	JR. ALFONSO UGARTE N° 342 - BARRIO COMERCIO	VENTA DE ROPA	10480205150
70 WAPAS URBAN & CASUAL	JR. MANUELA MOREY N° 148 - BARRIO CERCADO	VENTA DE PREMDAS DE VESTIR Y ACCESORIOS	10406215488
71 BAZAR NOVEDADES LINDA	JR. TAHUANTINSUYO N° 179 - BARRIO COMERCIO	VENTA DE ARTICULOS DE PERFUMERIA, ROPAS Y OTROS AFINES	10010685899
72 COMERCIAL EL DORADO	JR. AUGUSTO B. LEGUIA N° 209	VENTA DE ROPA MASCULINA	10010614045
73 ATHLETIC SPORT S.A.C	JR. FRANCISCO BOLOGNESI N° 229 - BARRIO 9 DE ABRIL	FABRICACION DE PRENDAS DE VESTIR DEPORTIVA	20600950488
74 MODANNY STYLES	JR. NICOLAS DE PIEROLA N° 257 - BARRIO COMERCIO	VENTA DE ROPA	1046600504
75 "GALERÍA LLANOS CABALLERO"	JR. NICOLAS DE PIEROLA N° 345 - BARRIO COMERCIO	COMERCIO DE ROPA	10011167891
76 MODA MAS MODA	JR. LIMA N° 420	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	10009683912
77 EL PRINCIPIITO	JR. LIMA N° 244 - BARRIO COMERCIO	VENTA DE ROPA	20600677960
78 NOVEDADES "HUACHEZ"	JR. LIMA N° 281 - BARRIO COMERCIO	VENTA DE CONFECCIONES DE PRENDAS DE VESTIR	10033003159
79 LLANOS FASHION	JR. LIMA N° 314 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESTIR	10009683912
80 "DISFRACES LUANITA"	JR. AUGUSTO B. LEGUIA N° 517	VENTA DE ROPA	10447718184
81 THAIS MODA	JR. LIMA N° 307 - BARRIO COMERCIO	VENTA DE PRENDAS DE VESITIR Y ACCESORIOS	10009666261
82 INVERSIONES GAONA "DISTRIBUCIONES	LIMA N° 401 - BARRIO COMERCIO	VENTA DE ROPA	10462663116
83 MONDRAGON"	JR. DANIEL A. CARRION N° 336 - BARRIO COMERCIO	VENTA DE ROPA	10417759773
84 BELLISIMAS	JR. LIMA N° 323	VENTA DE ROPA	10469911824
85 DISTRIBUCIONES MONDRAGON MULTINEGOCIOS VIRGEN DE LA	JR. DANIEL A. CARRION N° 336	VENTA DE ROPA	10417759773
86 PUERTA	PROLONGACION LOS PINOS N° 188 JR. MANUELA MOREY N° 133 - BARRIO CERCADO	VENTA DE ROPA	10404718555
87 MONADAS COLLECTION	JR. DANIEL A. CARRION N° 351 - B	VENTA DE ROPA AL MENOR	10427706155
88 NOVEDADES GAOTICK	JR. DANIEL A. CARRION N° 351 - B	VENTA DE PRENDAS DE VESTIR	10411165685

89	BELLA MIA ACCESORIOS DE MODA	JR. AUGUSTO B. LEGUIA N° 426 - BARRIO PARTIDO ALTO	VENTA DE ROPAS Y ACCESORIOS EN GENERAL	10459446881
90	SOLO PARA ELLAS S.A.C.	JR. AUGUSTO B. LEGUIA N° 172	VENTA DE ROPA Y PASAMANERIA	20600243544
91	NOVEDADES FIORELLA	JR. ALFONSO UGARTE N° 357 - BARRIO CERCADO	VENTA DE ROPAS EN GENERAL	10297319022
92	"EL PARAISO DE LA MODA"	JR. GREGORIO DELGADO N° 132-136 - BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	10009666261
93	ELLAS MODA FASHION	JR. ALONSO DE ALVARADO N° 144 - BARRIO CERCADO	VENTA Y CONFECCION DE PRENDAS DE VESTIR	10449610241
94	WOMAN	JR. ALFONSO UGARTE N° 297 BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR Y ACCESORIOS	10470114971
95	INTIMA - BOUTIQUE	JR. CABO A. LEVEAU N° 229 - BARRIO CERCADO	ALQUILER Y VENTA DE PRENDAS DE VESTIR Y ARTICULOS DE BAZAR	10011088011
96	PEKEÑITOS BABYS & KIDS	JR. ALFONSO UGARTE N° 264 - BARRIO CERCADO	VENTA DE PRENDAS DE VESTIR PARA NIÑOS	20572162711
97	ABYMAR	JR, MATEO PUMACAHUA N° 392	VENTA DE ROPA	10011636921
98	MULTISERVICIOS RODRIGUEZ	JR. MATEO PUMACAHUA N° 191 - URB. 9 DE ABRIL	BODEGA, FOTOCOPIAS, ANILLADOS, RECARGAS, VENTA DE UTILES, MERCERIAS Y ROPAS	10011230747
99	NOVEDADES "CHE CARLITOS"	JR. ALONSO DE ALVARADO N° 770 ESQ. ALONSO DE ALVARADO CON MARTIN DE LA RIVA N° 396	VENTA DE ROPAS	10407803871
100	NEGOCIOS J.C	JR. MARTINEZ DE COMPAGÑON N° 640	VENTA DE ROPAS	10010968351
101	RUSTIC ROCK	JR. ALONSO DE ALVARADO N° 820	VENTA DE ROPA	10420744809
102	NOVEDADES "BARBY FASHION" INVERSIONES NOVEDADES	JR. ALONSO DE ALVARADO N° 820	VENTA DE ROPA	10468401130
103	RANDY	JR. MARTIN DE LA RIVA HERRERA N° 430	VENTA DE ROPA	10011455676
104	NOVEDADES L & E	JR. ALONSO ALVARADO N° 840	VENTA DE ROPA	10011650339
105	NOVEDADES BARBY FASHION	JR. ALONSO DE ALVARADO N° 820	VENTA DE ROPAS	10468401130
106	DIVERXIA	JR. AUGUSTO B. LEGUIA N° 121	VENTA DE ROPAS	20101298851
107	DISTRIBUCIONES "DIAZ"	JR. DANIEL ALCIDES CARRION N° 374	VENTA DE ROPA POR MAYOR Y MENOR	10272586247
108	NOVEDADES L & E	JR. ALONSO DE ALVARADO N° 840	VENTA DE ROPA	10011650339
109	RUSTIC ROCK	JR.MARTINES DE COMPAÑON N°640-TARAPOTO	VENTA DE ROPA	10420744809
110	RUSTIC ROCK	JR. MARTINEZ DE COMPAGÑON N° 640	VENTA DE ROPA	10420744809
111	ALTEX	JR. PEDRO DE URZUA N° 315	CONFECCION Y VENTA DE ROPAS	10806095562
112	NOVEDADES CARMENCITA - JUGUERIA CARMENCITA	JR. CENEPAN° 600 - VILLA UNIVERSITARIA	VENTA DE ROPA Y JUGUERIA	10415465217
113	TRAVIESITOS BABY	JR. TAHUANTINSUYO N° 340	VENTA DE PRENDAS DE VESTIR	10438263786
114	COMERCIAL FLOREZITAS	JR. LEONCIO PRADO N° 219	VENTA DE ROPAS	10010607740

115 LUKIDS E.I.R.L	JR. ALFONSO UGARTE N° 573 - BARRIO COMERCIO	VENTA DE ROPAS Y CALZADOS PARA NIÑOS	20600540310
116 CREDITOS "DEL AGUILA"	JR. ALFONSO UGARTE N° 1411 - BARRIO HUAYCO	VENTA DE ROPAS, CALZADOS Y AFINES	10430253889
117 ELLAS MODA FASHION	JR. ALONSO DE ALVARADO N° 144 - BARRIO CERCADO	VENTA Y CONFECCION DE PRENDAS DE VESTIR	10449610241
118 MODAS NANSI	JR. ALFONSO UGARTE N° 544 - BARRIO HUAYCO	VENTA Y CONFECCION DE ROPA DEPORTIVA Y DE VESTIR	10010734377
119 CENTRO COMERCIAL CUBAS	JR. DANIEL A. CARRION N°310-BARRIO COMERCIO	VENTAS DE PRENDAS DE VESTIR Y ACCESORIOS	10011288231
120 KATTALEYA BOUTIQUE	JR. AUGUSTO B. LEGUIA N° 415 - BARRIO COMERCIO	VENTAS DE ROPA Y ACCESORIOS	10167226481
121 FD BY FIORELLA DIAZ	JR. JUAN VARGAS N° 248 - BARRIO HUAYCO	BOUTIQUE	10445087471
122 BELLA MODA	JR. MARTIN DE LA RIVA N° 475 - BARRIO COMERCIO	BOUTIQUE	10413321994
123 CHANNELI	JR. ALEGRIA ARIAS DE MOREY N° 205 - BARRIO CERCADO	BOUTIQUE	10735081221
124 CRISS BOUTIQUE INVERSIONES Y NEGOCIOS	JR. AUGUSTO B. LEGUIA N° 438 - BARRIO COMERCIO	BOUTIQUE	10466178280
125 CREDIMAS E.I.R.L	JR. GRAU N° 205 - BARRIO PARTIDO ALTO	BOUTIQUE	20542368919
126 BOUTIQUE A & E	JR. LOS PROCERES N° 221 - BARRIO LOS JARDINES	BOUTIQUE	10462912850
127 SECRET STYLE	JR. AUGUSTO B. LEGUIA N° 436 - BARRIO COMERCIO	BOUTIQUE	10438004730
128 JOELMA - BRAZIL	JR. RAMON CASTILLA N° 236 - BARRIO CERCADO	BOUTIQUE	10056115205
129 ALEVALE BOUTIQUE	JR. ANTONIO RAYMONDI N° 282 - STAND 02 - BARRIO CERCADO	BOUTIQUE	10455908596
130 BOUTIQUE D'MON@S	JR. AUGUSTO B. LEGUIA N° 706 - BARRIO PARTIDO ALTO	BOUTIQUE	10723144022
131 A & V BOUTIQUE S.R.L	JR. ALEGRIA ARIAS DE MOREY N° 583 - BARRIO SUCHICHE	BOUTIQUE	20600973283
132 SUPER BARATITO E.I.R.L	JR. LIMA N° 477 - BARRIO COMERCIO	BOUTIQUE	20494122336
133 FASHION & STYLOS	JR. PROGRESO N° 760 - BARRIO 09 DE ABRIL	BOUTIQUE	10011619121
134 COMERCIAL "ARROYO"	JR. NICOLAS DE PIEROLA N° 344	BOUTIQUE	100708490
135 NOVEDADES NORMA	JR. DANIEL ALCIDES CARRION N° 355 - BARRIO COMERCIO	BOUTIQUE	10008692136
136 BLACK & WHITE STORE E.I.R.L.	JR. ALFONSO UGARTE N° 560 - BARRIO HUAYCO	BOUTIQUE	20600815823

137 ANITAS ROPAS Y ACCESORIOS	JR. GREGORIO DELGADO N° 425 - BARRIO CERCADO	BOUTIQUE	10011301857
138 "SOLO PARA TI"	JR. AUGUSTO B. LEGUIA N° 629	BOUTIQUE - BAZAR	10778196978
139 NUOVACERO	JR. CABO ALBERTO LEVEAU N° 223 - B AV. CIRCUNVALACION N° 411 - BARRIO PARTIDO ALTO	BOUTIQUE	10416459601
140 VALENTINA MODA - BELLEZA	JR. VICTOR ANDRES BELAUNDE N° 205	BOUTIQUES	10009497841
141 LA CASITA DEL REGALO	JR. GREGORIO DELGADO N° 180	BOUTIQUES	10720479945
142 INTIMA'S BOUTIQUE & LENCERIA	JR. DANIEL ALCIDES CARRION N° 121	BOUTIQUE	20600135872
143 CHIOS BOUTIQUE	JR. MARTINEZ DE COMPAGÑON N° 149	BOUTIQUE	10011477742
144 COCOLISO WOMAN	JR. CABO ALBERTO LEVEAU N° 540	BOUTIQUE	20531596138
145 SARA'S MODA Y ACCESORIOS	JR. ALFONSO UGARTE N° 190	BOUTIQUE	10405284907
146 NOVEDADES VICKY S.R.L.	JR. TAHUANTINSUYO N° 315	BOUTIQUE	20531420951
147 NOVEDADES JOHCE	JR. LIMA N° 441 STAND 108	BOUTIQUE	10719024942
148 LA CHIC	JR. ANDRES AVELINO CACERES N° 470	BOUTIQUE	10447524983
149 THE CLOSET ROUSS & AARON TERAPIA MANOS DE LUZ	JR. AUGUSTO B. LEGUIA N° 447	BOUTIQUE	10409686139
150 BOUTIQUE	JR. MARTINEZ DE COMPAGÑON N° 141	SALON DE MASAJES - BOUTIQUE	10008824831
151 LUCEROS	JR. GREGORIO DELGADO N° 485	BOUTIQUE	20600908899
152 DOLLY	JR. MIGUEL GRAU N° 205 - BARRIO CERCADO	BOUTIQUE	10416938381
153 D & K POLOS		VENTA DE POLOS	10467857229

Fuente: Municipalidad provincial de San Martín – Oficina de licenciamiento

ANEXO B. CUESTIONARIO

En este cuestionario, usamos la palabra empresa, para representar la unidad sobre la que nos está contestando las preguntas. Por favor sea consistente a lo largo del cuestionario sobre contestar acerca de la misma compañía, sea empresa o planta.

Datos de la empresa:

1. Nombre _____
2. Tiempo operando _____ años
3. Número de empleados _____
4. Línea de productos que maneja _____
5. Responder con un costo aproximado para los equipos necesarios de la implementación de una solución de software tradicional.

Equipo	Costo en soles
Servidor/Computadora	
Ticketera	
Software tradicional	
Impresora	
Lectora de barras	
Redes	
Costos de instalación	
Costos de configuración	
Gastos en soporte del software	

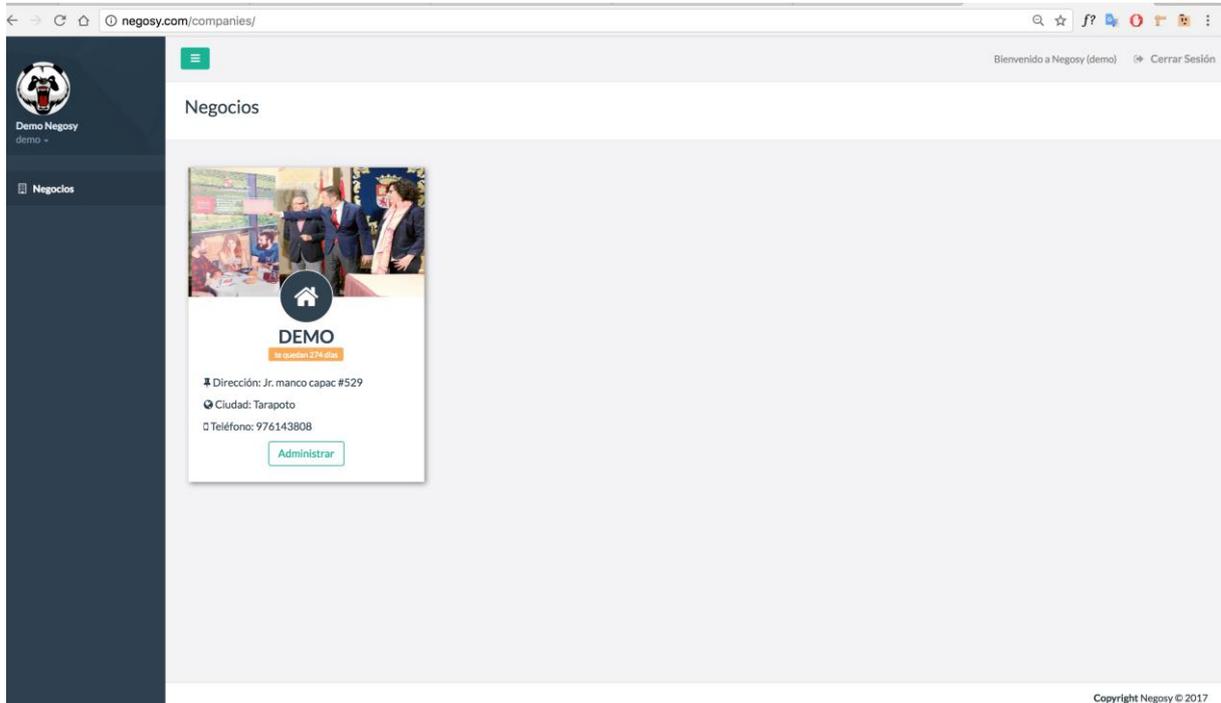
ANEXO C. SOFTWARE COMO SERVICIO

Módulos de la solución de Software como Servicio. (para más información visite <http://negosy.com>)

0. Página Web



1. Módulo de Administración



negosy.com/companies/demo/

Bienvenido a Negosy (demo) Cerrar Sesión

Administrador

Negocios / demo

Contrato con el negocio DEMO

Plan: PLAN ANUAL DE PRUEBA

Fecha de expiración: 24 Enero, 2018 te quedan: 274 días

Creado por: Demo Negosy el 25 de Enero de 2017 a las 23:38

Última actualización: 9 de Marzo de 2017 a las 01:30

Usuarios:

Última actividad

Usuarios (2)

- Demo Negosy: hoy 06:36 a.m. - 24.04.2017 hace 5 horas
Nuevo Caja: Abrió caja en DEMO
- Demo Negosy: hoy 06:36 a.m. - 24.04.2017 hace 5 horas
Edito Caja: Cerró caja en DEMO
- Demo Negosy: ayer 11:42 p.m. - 23.04.2017 hace 12 horas
Edito Stock: MEDIAS: 0.00 unid. en DEMO
- Demo Negosy: ayer 11:42 p.m. - 23.04.2017 hace 12 horas
Nuevo Transacción: Venta de productos S/11.70 en DEMO
- Demo Negosy: ayer 10:40 p.m. - 23.04.2017 hace 13 horas
Edito Stock: VESTIDO ROJO FLOREADO: 3.00 unid. en DEMO

Datos del negocio

Editar negocio

DEMO
 Jr. manco capac #529
 Tarapoto, Perú
 RUC: 10425475148
 Tel: 976143808
 Email: hola@negosy.com

negosy.com/companies/update/demo/

Bienvenido a Negosy (demo) Cerrar Sesión

Editar DEMO

Negocios / demo / Editar

Razón social* DEMO

RUC de la Tienda* 10425475148

Ciudad* Tarapoto

Dirección* Jr. manco capac #529

Telefono* 976143808

Correo Electrónico* hola@negosy.com

N° de autorización 83498920834

N° serie según SUNAT PPPPSS64638425

Logo Actualmente: company/image_content_4784782_20160505125015.jpg [Limpiar](#)
 Modificar: No file chosen

Copyright Negosy © 2017

2. Módulo de Inventario

demo.negosy.com/#!/inventario/productos

PRODUCTOS LÍNEAS CATEGORÍAS DESCUENTOS

Q search Nuevo producto Desactivar

Código	Producto	Precio	Linea	Stock	Imagen
PACC0082	POLO AMARILLO CON CUELLO	S/. 59.00	POLOS K	0	
PRCCN0081	POLO ROSADO CON CUELLO NIÑA	S/. 63.00	POLOS K	0	
PPCML0080	POLO PLOMO CON MANDA LARGA	S/. 76.00	POLOS K	3	
PCCC0079	POLO CELESTE CON CUELLO	S/. 59.00	POLOS K	4	
PMCC0078	POLO MORADO CON CUELLO	S/. 69.00	POLOS K	5	
PBA0077	POLO BLANCO ADIDAS	S/. 115.00	POLOS H	2	
PDV0076	POLO DUNKELVOLK VERDE	S/. 99.00	POLOS H	1	
PRA0075	POLO ROJO ADIDAS	S/. 120.00	POLOS H	0	
PACCD074	POLO AZUL CON CUELLO	S/. 69.00	POLOS H	0	
PAMCC0073	POLO AZUL MARINO CON CUELLO	S/. 75.00	POLOS H	3	
PRCC0072	POLO ROJO CON CUELLO	S/. 69.00	POLOS H	3	
PPR0071	PANTALÓN PALO ROSA	S/. 89.00	PANTALONES M	2	
PDR0070	PANTALÓN DRILL ROJO	S/. 69.00	PANTALONES M	1	
PJA0069	PANTALÓN JEAN AZUL	S/. 77.00	PANTALONES M	0	
PDA0068	PANTALÓN DRILL AZUL	S/. 79.00	PANTALONES M	5	

26 - 50 de 62 registros

3. Módulo de Contactos (empleados y clientes)

demo.negosy.com/#!/contact/client

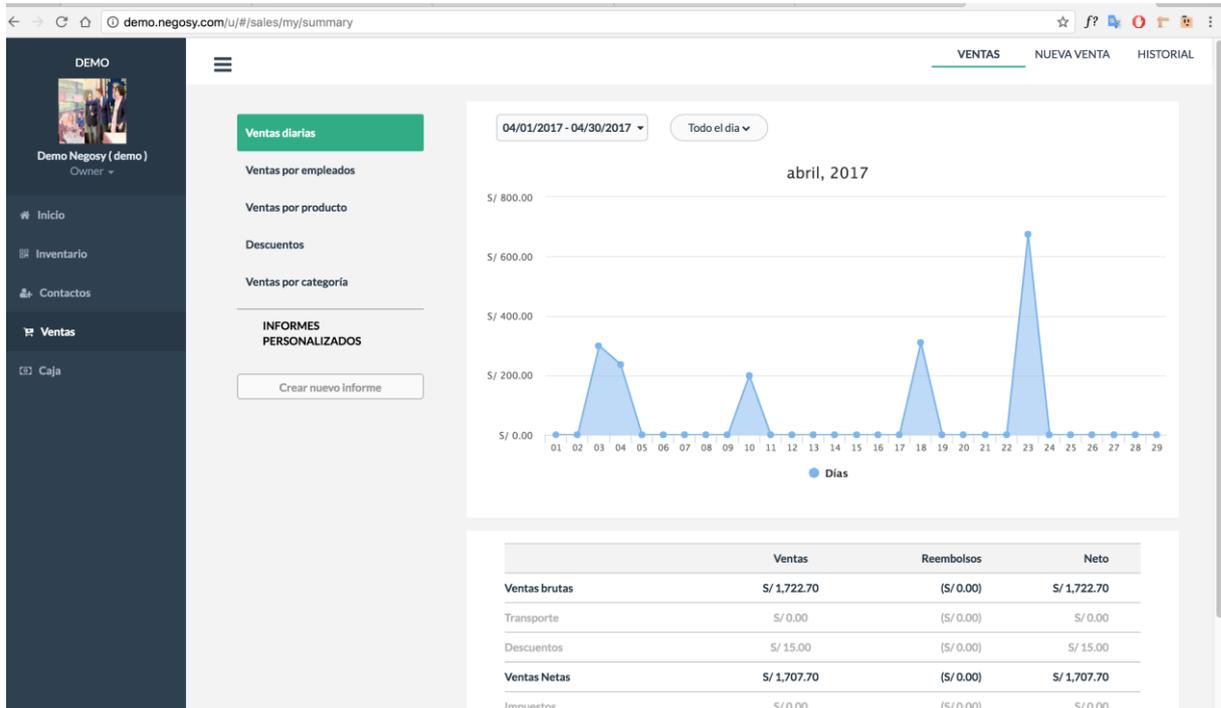
EMPLEADOS CLIENTES

Q search Nuevo Cliente Desactivar

Id	Nombre Completo	Documento	Tipo	Estado Civil
64	María		Natural	Soltero
63	Pepito		Natural	Soltero
62	Rosa		Natural	Soltero
61	Elmer		Natural	Soltero
60	Martin		Natural	Soltero
59	Willy		Natural	Soltero
58	Miguel		Natural	Soltero
57	Mariano	70624512	Natural	Soltero
56	Elias		Natural	Soltero
55	Jose Vasquez	10425241846	Natural	Soltero
54	Macela	74859623	Natural	Soltero
53	Mario Vargas Sánchez	01548714	Natural	Soltero
52	Marcos	70542163	Natural	Soltero
51	Filome		Natural	Soltero
50	bb		Natural	Soltero

1 - 25 de 35 registros

4. Módulo de Ventas (Ventas, Nueva Venta, Historial)



The screenshot shows the 'Nueva Venta' (New Sale) module interface. At the top, there's a search bar with the text 'Ingresa producto o código' and a 'Buscar' button. Below the search bar is a grid of 12 product cards, each showing a product image, name, and price. The products are:

- BLUSA TURQUESA CON ENCAJE (S/ 100.00)
- VESTIDO ROJO FLOREADO (S/ 89.00)
- VESTIDO VERDE CON MANGA (S/ 110.00)
- VESTIDO CHINO FLOREADO (S/ 87.00)
- VESTIDO NEGRO, MELÓN Y BLANCO (S/ 75.00)
- POLO AMARILLO CON CUELLO (S/ 59.00)
- VESTIDO MELÓN NIÑA (S/ 95.00)
- POLO ROSADO CON CUELLO NIÑA (S/ 63.00)
- POLO ROJO ADIDAS (S/ 120.00)
- POLO AZUL CON CUELLO (S/ 69.00)
- BLUSA ROSADA CUELLO V (S/ 75.00)
- PANTALÓN JEAN AZUL (S/ 77.00)

At the bottom of the grid, it says 'productos encontrados 24 unid.'. On the right side, there's a checkout summary area. At the top, it shows 'S/. 0.00'. Below this is a table with columns: 'Producto', 'Cant.', 'PU', 'SubTotal', and 'Acción'. The table is currently empty. Below the table, there's a 'SubTotal' row showing 'SubTotal S/ 0.00'. At the bottom, there's a 'Cliente' section with a search bar and a 'Buscar Cliente' button. Below that, there's a 'Descuento' section with a dropdown menu set to '0'. There are also fields for 'T. venta' (set to 'Contado'), 'Empleado' (set to 'Elaccer Fernandez G'), and 'Entrega' (set to 'Ninguno'). At the bottom, there are two buttons: 'Vender (Ctrl+Enter)' and 'Cancel'.

5. Módulo de Caja (Apertura, Caja, Transacciones)

demo.negosy.com/uj/#/finanzas/caja

APERTURA CAJA TRANSACCIONES

Arqueo Cerrar Caja

Caja al día 24 Abr. 2017

S/ 100.00

Caja Inicial: S/ 100.00

0 Transacciones

Tipo de documento

Aún no hay ventas por documentos

Ingresos

AÚN NO TIENE NINGÚN INGRESO

Egresos

AÚN NO TIENE NINGÚN EGRESOS

Monto de apertura de caja del día 24 Abr. 2017				S/ 100.00
Formas de Pago	Transacciones	Ingreso(0)	Egreso(0)	Total
Efectivo	0	S/ 0.00	S/ 0.00	S/ 0.00
Visa	0	S/ 0.00	S/ 0.00	S/ 0.00
Checke	0	S/ 0.00	S/ 0.00	S/ 0.00
Deposito a cuenta	0	S/ 0.00	S/ 0.00	S/ 0.00
Total Neto				S/ 100.00
Total en caja (solo transacciones en efectivo)				S/ 100.00

Copyright Negosy © 2017

demo.negosy.com/uj/#/finanzas/transaction

Enviar recibo Procesar reembolso

04/19/2017 - 04/24/2017

9 TRANSACCIONES

S/ 675.70 TOTAL RECIBIDO

domingo, 23 de abril del 2017

11:42 pm	(Varios) 1 x MEDIAS, (Varios) 1 x CEVICHE
10:40 pm	(Varios) 1 x VESTIDO ROJO FLOREADO
10:38 pm	(Varios) 1 x VESTIDO ROJO FLOREADO
10:38 pm	(Varios) 1 x VESTIDO ROJO FLOREADO
10:36 pm	(Varios) 1 x PANTALÓN DRIL ROJO
10:36 pm	(Varios) 1 x NUEVO PRODUCTO
10:34 pm	(Varios) 1 x VESTIDO ROJO FLOREADO
10:33 pm	(Varios) 1 x camisa guinda
10:29 pm	(Varios) 1 x VESTIDO ROJO FLOREADO

S/ 89.00 (Pagado)

Cliente: Varios
23 Abr. 2017 10:40 pm

VESTIDO ROJO FLOREADO x 1 (S/ 89.00) Cod. VRF00B5	S/ 89.00
Subtotal	S/ 89.00
Impuesto incluido (S/ 0.00)	
Total	S/ 89.00
Efectivo	S/ 89.00
Sin documento	

¡Listo!