



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

DESARROLLO DE SOFTWARE
ORIENTADO A PRESTACIÓN DE SERVICIOS (MODELO SAAS)

Víctor Alejandro Álvarez Contreras
Asesorado por el Ingeniero Julio Ernesto Contreras Sierra

Guatemala, noviembre de 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DESARROLLO DE SOFTWARE ORIENTADO A PRESTACIÓN DE
SERVICIOS (MODELO SAAS)**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR:

VÍCTOR ALEJANDRO ÁLVAREZ CONTRERAS

ASESORADO POR EL INGENIERO JULIO ERNESTO CONTRERAS SIERRA

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DESARROLLO DE SOFTWARE ORIENTADO A PRESTACIÓN DE SERVICIOS (MODELO SAAS),

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en mayo de 2008.

Víctor Alejandro Álvarez Contreras

Guatemala, 23 de septiembre de 2009

Señores

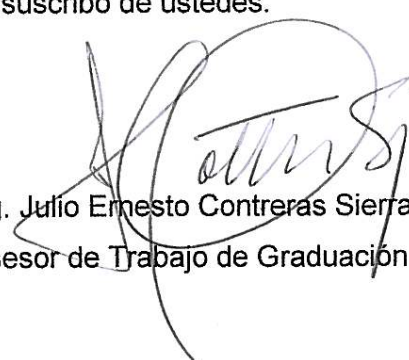
Escuela de Ciencias y Sistemas

Facultad de Ingeniería

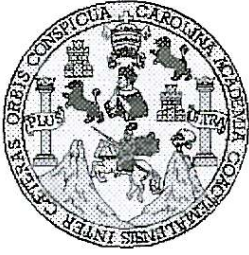
Respetables señores:

Atentamente me dirijo a ustedes para hacer de su conocimiento que he revisado el trabajo de graduación titulado: "DESARROLLO DE SOFTWARE A PRESTACION DE SERVICIOS (MODELO SAAS)", realizado por el estudiante VICTOR ALEJANDRO ALVAREZ CONTRERAS, con carné número 200212212, considerando a mi criterio que dicho trabajo cumple con todos los objetivos propuestos para su desarrollo, según el protocolo.

Sin otro particular, me suscribo de ustedes.



Ing. Julio Ernesto Contreras Sierra
Asesor de Trabajo de Graduación



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 14 de Octubre de 2009

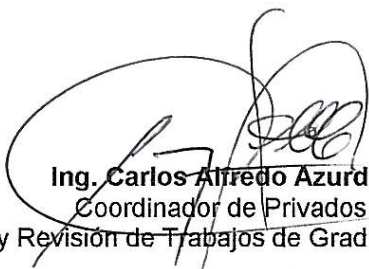
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **VICTOR ALEJANDRO ALVAREZ CONTRERAS**, titulado: **“DESARROLLO DE SOFTWARE ORIENTADO A PRESTACION DE SERVICIOS (MODELO SAAS)”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado “DESARROLLO DE SOFTWARE ORIENTADO A PRESTACIÓN DE SERVICIOS (MODELO SAAS)”, presentado por el estudiante VÍCTOR ALEJANDRO ÁLVAREZ CONTRERAS, aprueba el presente trabajo y solicita la autorización del mismo.

“ID Y ENSEÑAD A TODOS”



M. Marlon Antonio Pérez Turk
Ing. Marlon Antonio Pérez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 04 de noviembre 2009



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **DESARROLLO DE SOFTWARE ORIENTADO A PRESTACIÓN DE SERVICIOS (MODELO SAAS)**, presentado por el estudiante universitario **Víctor Alejandro Álvarez Contreras**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

A handwritten signature in black ink, enclosed within a hand-drawn oval border.

Ing. Murphy Olimpo Paiz Reinos
DECANO



Guatemala, noviembre de 2009

/gdech

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
GLOSARIO	IX
RESUMEN	XIII
OBJETIVOS	XV
INTRODUCCIÓN	XVII
1. CONCEPTOS GENERALES	1
1.1 ¿Qué es el modelo SaaS?	1
1.2 Modelo tradicional vrs. Modelo SaaS	4
1.2.1 Conceptos generales del modelo tradicional	4
1.2.2 Similitudes y diferencias	5
1.3 Ventajas del modelo SaaS	7
1.3.1 Perspectiva del cliente	7
1.3.2 Perspectiva del proveedor de servicios	9
1.4 Desventajas del modelo SaaS	11
1.4.1 Perspectiva del cliente	11
1.5 Perspectiva del proveedor de servicios	12
1.6 El proveedor de servicios de aplicación	12
1.6.1 Tipos de proveedores de servicios de aplicación	13
1.6.2 Características de un buen proveedor	15
2. ASPECTOS TÉCNICOS DEL MODELO SAAS	19
2.1 Arquitectura física	19
2.1.1 Infraestructura que garantice la seguridad	19
2.1.2 Infraestructura que garantice la disponibilidad de la información	21
2.1.3 Infraestructura que garantice alto rendimiento	24
2.1.4 Infraestructura recomendada	24
2.2 Arquitectura de software	27

2.2.1	Capa de presentación	29
2.2.2	Capa de negocios	32
2.2.3	Capa de datos	33
3.	ANÁLISIS Y DISEÑO DE UN PROTOTIPO BASADO EN EL MODELO SaaS	35
3.1	Especificación de requerimientos del software	35
3.1.1	Introducción	35
3.1.1.1	Propósito	35
3.1.1.2	Audiencia del documento	36
3.1.1.3	Alcance del proyecto	36
3.1.2	Descripción general	36
3.1.2.1	Perspectiva del producto	36
3.1.2.2	Características del producto	37
3.1.2.3	Tipos de usuario	38
3.1.2.4	Ambiente operativo	39
3.1.2.5	Limitaciones de diseño e implementación	39
3.1.3	Características del sistema	40
3.1.3.1	Módulo de registro de usuarios (suscripciones)	40
3.1.3.1.1	Descripción y prioridad	40
3.1.3.1.2	Secuencias de estímulo / respuesta	41
3.1.3.1.3	Requerimientos funcionales	42
3.1.3.2	Módulo de administración de usuarios y cuentas	42
3.1.3.2.1	Descripción y prioridad	42
3.1.3.2.2	Secuencia de estímulos / respuestas	43
3.1.3.2.3	Requerimientos funcionales	45
3.1.3.3	Módulo de contabilidad	45
3.1.3.3.1	Descripción y prioridad	45
3.1.3.3.2	Secuencia de estímulos / respuestas	49
3.1.3.3.3	Requerimientos funcionales	50
3.1.4	Requerimientos de interfaz externa	51
3.1.4.1	Interfaces de usuario	52

3.1.4.1.1	Interfaz de consulta general	53
3.1.4.1.2	Interfaz de consulta específica	53
3.1.4.1.3	Interfaz de ingreso y modificación	54
3.1.5	Otros requerimientos no funcionales	55
3.1.5.1	Requerimientos de rendimiento	55
3.1.5.2	Requerimientos de seguridad	56
3.1.5.3	Requerimientos de calidad del software	56
3.2	Especificación de diseño del software	57
3.2.1	Introducción	57
3.2.1.1	Propósito	57
3.2.1.2	Audiencia del documento	57
3.2.1.3	Alcance	57
3.2.2	Descripción general del sistema	57
3.2.2.1	Módulos iniciales	57
3.2.3	Consideraciones de diseño	58
3.2.3.1	Dependencias y asunciones	58
3.2.3.2	Límites generales	59
3.2.3.3	Objetivos y lineamientos	60
3.2.3.4	Metodología de desarrollo	61
3.2.4	Estrategias de arquitectura	62
3.2.5	Arquitectura del sistema	63
3.2.5.1	Diseño de la base de datos	63
3.2.5.2	Módulos y clases:	67
3.2.5.2.1	Módulo de administración:	67
3.2.5.2.2	Módulo común:	70
3.2.5.2.3	Módulo de contabilidad:	74
3.2.5.2.4	Módulo de usuarios:	82
3.2.5.3	Interfaces de usuario	83
3.2.5.3.1	Páginas del módulo de suscripciones:	85
3.2.5.3.2	Páginas del módulo de administración de usuarios y cuentas	87

3.2.5.3.3	Páginas del módulo de contabilidad	88
3.2.6	Políticas y tácticas	92
3.3	Especificación de casos de prueba del software	93
3.3.1	Introducción	93
3.3.1.1	Interacción con el equipo	94
3.3.2	Objetivo de los casos de prueba	94
3.3.3	Visión general del proceso	94
3.3.4	Estrategia de pruebas	95
3.3.4.1	Pruebas de caja blanca	96
3.3.4.2	Pruebas de caja negra	96
3.3.5	Pruebas de sistema	96
4.	PROPUESTA DE NEGOCIOS BASADA EN EL PROTOTIPO	97
4.1	Resumen ejecutivo	97
4.2	Introducción a Creativasoft	99
4.2.1	Visión de la empresa	99
4.2.2	Misión de la empresa	100
4.3	Propuesta de productos y servicios	100
4.3.1	Creativa Contabilidad	100
4.4	Situación actual del proyecto	104
4.5	Inversión Inicial	106
4.6	Costos de mantenimiento mensual	106
4.7	Tarifas	107
4.8	Equipo humano	108
4.9	Planes de expansión	109
4.9.1	Mejoras que podrían ser incorporadas al producto	109
4.9.2	Nuevos módulos que podrían ser incorporados al producto	110
4.10	Mercado	111
4.10.1	Mercado objetivo	111
4.10.2	Entorno competitivo	111
4.11	Plan de Mercadotecnia	113

5. CASOS DE ÉXITO DEL MODELO SaaS EN LA ACTUALIDAD	115
5.1 Salesforce.com	115
5.2 Vocus	117
5.3 Workday inc.	119
5.4 Oco inc.	121
CONCLUSIONES	123
RECOMENDACIONES	125
BIBLIOGRAFÍAS	127

ÍNDICE DE ILUSTRACIONES

FIGURAS

1	Arquitectura de una red de clusters de alta disponibilidad	23
2	Arquitectura de una infraestructura recomendada para soportar aplicaciones SaaS	25
3	Arquitectura de software recomendada para una aplicación Web bajo el modelo SaaS	29
4	Diagrama de relaciones entre módulos de la suite financiera	37
5	Diagrama de estructura de un mensaje informativo o de error	52
6	Diagrama de estructura de interfaz de consulta general	53
7	Diagrama de estructura de interfaz de consulta específica	54
8	Diagrama de estructura de interfaz de ingreso y modificación	55
9	Logo de Creativa contabilidad	101
10	Modelo de uso e infraestructura	102
11	Estructura de contenidos	105
12	Módulos de creativa finanzas	110

TABLAS

I	Plantilla de caso de pruebas	95
II	Inversión inicial	106
III	Costos mensuales de mantenimiento	107

IV	Tarifas de creativa contabilidad	108
V	Competidores nacionales con aplicaciones de escritorio y venta directa	112
VI	Competidores nacionales con aplicaciones de escritorio y modelo de renta	112
VII	Competidores internacionales con aplicaciones de escritorio y venta directa	112
VIII	Competidores internacionales con aplicaciones de escritorio y modelo de renta	113

GLOSARIO

Certificados SSL	Archivos que contienen un código de encriptación que permite la autenticación y privacidad de la transmisión de información entre extremos a través de Internet.
Dataset	Componente que almacena información obtenida de una consulta a una base de datos en una aplicación de software.
DNS	Sistema de nombres de dominio (Domain Name System). Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.
Dropdown	Componente Web que cuenta con características muy similares a los ComboBox de las aplicaciones de escritorio. Este control se puede expandir para mostrar un listado de opciones de las cuales el usuario puede seleccionar una o varias dependiendo de como esté configurado.
Interoperabilidad	Es la capacidad de un sistema de comunicarse con otro

sistema distinto a él.

LOPD Ley Orgánica de Protección de Datos de Carácter Personal. Es una ley que tiene por objeto garantizar y proteger, en lo que concierne a los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Project Manager Desde el punto de vista de sistemas, un Project manager es un profesional que tiene la responsabilidad de planear, administrar, ejecutar y concluir un proyecto de software.

Protocolo HTTP Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol). Es un protocolo utilizado en cada transacción realizada en la Web. Define la sintaxis y la semántica que utilizan los elementos de software que funcionan en una plataforma Web. Este protocolo utiliza el esquema de petición-respuesta entre un cliente y un servidor. Al cliente que realiza la petición se le conoce como "user agent" o "agente usuario". A la información transmitida se le conoce como recurso y se identifica a través de un URL (Localizador Uniforme de Recursos).

Protocolo ICMP Protocolo de mensajes de control y error de Internet. La utilidad de este protocolo no está en el transporte de datos, sino en controlar el estado de los paquetes enviados a través de una red. Este protocolo verifica si un paquete no pudo alcanzar su destino, si su tiempo de vida ha expirado,

si el encabezado de un paquete lleva un valor no permitido, si es un paquete de eco o respuesta, etc.

Redundancia En ingeniería, es la replicación de componentes críticos de un sistema con la intención de aumentar la confiabilidad del mismo. Esto permite que el fallo de un componente pueda ser contrarrestado con la utilización de un componente idéntico listo para funcionar.

REST Transferencia de estado representacional (Representational State Transfer). Es un estilo de arquitectura de software para sistemas de hipertexto como el World Wide Web (www). Se refiere a un conjunto de principios de una arquitectura de red que especifica como los recursos deben ser definidos y disecionados.

Servicios Web Son sistemas diseñados para soportar la interoperabilidad entre sistemas a través de una red.

SAN Red de área de almacenamiento (Storage Area Network). Es una arquitectura para enlazar dispositivos de almacenamiento remoto con servidores, a tal punto que en los sistemas operativos de los servidores, los dispositivos de almacenamiento remoto aparecen como dispositivos locales.

Tester Profesional del área de informática cuya función es realizar pruebas sobre diversos componentes de un sistema de software, con el fin de asegurar su funcionalidad.

RESUMEN

La tecnología de la información avanza aceleradamente y junto con ella, la cantidad de proveedores de servicios de informática aumenta cada vez más, lo que ha provocado que se busquen nuevas estrategias para mantenerse competitivos en un mercado ya saturado.

El uso de Internet a nivel global ha aumentado exponencialmente y esto ha permitido que los paradigmas de desarrollo de software evolucionen del clásico enfoque de desarrollo de aplicaciones específicas a terceros y se desarrollen aplicaciones flexibles y versátiles, que le permitan a personas y empresas hacer uso de sistemas que satisfagan cualquier tipo de necesidad desde cualquier parte del mundo.

Los avances tecnológicos dan la oportunidad de que las aplicaciones ofrecidas a través de Internet puedan ser tan robustas como las aplicaciones instaladas en una máquina, por lo que las empresas de desarrollo de software han optado por ofrecer sus sistemas, a través de paquetes de servicios integrales, cambiando la estrategia de venta de licencias a través de un producto empaquetado, a una venta de licencias a través de usuarios de Internet que hacen uso de las aplicaciones a través de un explorador o browser, y cuentan con ventajas adicionales como la capacidad de hacer uso de las aplicaciones desde cualquier lugar con conexión de Internet, disponibilidad de los sistemas las 24 horas del día los 7 días de la semana, bajos precios en comparación a licencias de productos empaquetados, personalización

avanzada para cubrir necesidades específicas de cada negocio, mejoras periódicas de los sistemas, soporte técnico de alta calidad , entre otras.

Estas ventajas han provocado que la popularidad de este tipo de servicios aumente y que compañías de la talla de Walmart y Office Depot depositen su confianza en sistemas de informática que funcionan bajo este enfoque y cubran necesidades que antes no podían ser satisfechas debido a las limitaciones que existían.

OBJETIVOS

1. Presentar al lector todos los conceptos generales que dejen claro en que consiste el desarrollo de software, orientado a prestar servicios (modelo SaaS).
2. Presentar al lector los distintos aspectos técnicos necesarios para soportar aplicaciones desarrolladas bajo el modelo SaaS.
3. Recomendar al lector una arquitectura física que cubra de forma óptima los aspectos técnicos presentados.
4. Proponer estrategias de negocio que le permitan al lector conocer formas de obtener ganancias a partir del modelo SaaS.
5. Introducir al lector en el desarrollo de aplicaciones orientadas a prestación de servicios a través de la documentación de análisis, diseño e implementación de un prototipo que funcione bajo el modelo SaaS.
6. Presentar una propuesta de negocios basada en el prototipo implementado que le sirva de guía al lector para emprender utilizando el modelo de desarrollo de software orientado a la prestación de servicios.

INTRODUCCIÓN

En la actualidad el mercado tecnológico se encuentra saturado de empresas que proveen servicios de desarrollo de software. Esto ha provocado que se desarrollen nuevas estrategias que le permitan a las empresas mantener su competitividad.

Los avances tecnológicos y el aumento de proveedores de servicios de Internet han permitido que aplicaciones, que antes podían ser utilizadas únicamente a través de un servidor empresarial o instalándolas localmente en una computadora, puedan ahora ser publicadas a través de portales Web y ser utilizadas con ayuda de un explorador o browser.

Estas facilidades han evolucionado el enfoque clásico de venta de software en el que se ofrecen sistemas a través de productos empaquetados y venta de licencias personales o empresariales, a un enfoque en el cual los sistemas son ofrecidos al cliente en forma de servicios integrales con ventajas adicionales de valor a los clientes que hacen uso de ellas.

A través del presente trabajo se plantea el enfoque de desarrollo de software orientado a la prestación de servicios, que recibe el nombre de Modelo SaaS (nombre derivado de sus siglas en inglés que significan Software as a Service o en español, software como un servicio).

Se presentan los conceptos generales que explican en qué consiste este modelo de desarrollo de software, así como las diferencias de este modelo con

el enfoque clásico de desarrollo de software específico a terceros y venta de productos empaquetados. Además, se listan las ventajas y desventajas de ambos modelos.

Se describen los diferentes aspectos que envuelven este enfoque de desarrollo de software, que incluyen la arquitectura física y de software necesaria para soportar este tipo de aplicaciones.

Además, se presenta el análisis y diseño que involucra el desarrollo de un prototipo basado en este modelo y un plan de negocios que toma como base el prototipo desarrollado para proponer una estrategia de venta de estos sistemas.

Por último, se presentan algunos casos de éxito recientes que demuestran el nivel de avance que ha tenido este modelo a nivel mundial y su popularidad creciente.

1. CONCEPTOS GENERALES

1.1 ¿Qué es el modelo SaaS?

El concepto "software como un servicio" nació entre los años 2000 y 2001 junto a conceptos como "Negocios Remotos". El término en inglés "SaaS (Software as a Service)" fue usado ampliamente después de ser usado por primera vez en una conferencia relacionada con el tema en marzo del año 2005.

El modelo de desarrollo de software orientado a prestación de servicios, también conocido como modelo SaaS, es una forma relativamente nueva de vender y distribuir software. Le permite a una persona individual o a una pequeña o mediana empresa, hacer uso de aplicaciones de software y servicios de tecnología avanzados a cambio de una cuota periódica. La clave de este modelo es el enfoque que se le da a una aplicación de software, ya que es tratada como un servicio y no como un producto. Por lo tanto, los clientes no pagan por adquirir el software, sino por hacer uso de él. Cabe mencionar que las aplicaciones de software no dejan de ser productos en este modelo, ya que el proveedor de servicios es propietario de los productos de software pero alquila las licencias de sus productos en forma de servicios a sus clientes. Las aplicaciones de software ofrecidas bajo este modelo también reciben el nombre de "aplicaciones de software bajo demanda".

De forma general, este modelo consiste en brindar a usuarios, accesos a una aplicación de software (como una contabilidad) utilizando un protocolo estándar como el protocolo HTTP, con ayuda de un explorador o a través de

aplicaciones cliente con propósitos específicos, y brindar servicios a los usuarios a través de estos accesos dependiendo de un paquete específico o una cuota seleccionada. Este tipo de servicios también pueden integrarse a aplicaciones propias de una empresa para manejar procesos de negocio si esto fuera requerido, siendo esta integración generalmente, a través de servicios Web o Aplicaciones REST.

Algunas de las características asociadas a este modelo incluyen:

- El proveedor de servicio de aplicaciones es dueño y maneja por completo, de forma independiente o a través de otra empresa, las aplicaciones de software que ofrece.
- El proveedor es dueño de toda la infraestructura necesaria para soportar las aplicaciones de software.
- El proveedor pone la información y las aplicaciones de software a disponibilidad del cliente a través de un explorador e Internet o un "*thin client*" (aplicación cliente "delgada" que utiliza la información de un servidor al que se encuentra conectada para realizar sus operaciones).
- El proveedor cobra una cuota periódica por uso de la aplicación que puede ser mensual, semestral, anual, etc.

Las aplicaciones de software distribuidas bajo este modelo deben cumplir con ciertas características clave descritas a continuación:

- Las aplicaciones deben estar disponibles comercialmente y ser accesadas y administradas a través de una plataforma en red.
- Sus actividades o funciones deben ser administradas en locaciones centrales en vez de ser administradas desde las localizaciones de cada

cliente, permitiéndole a estos acceder a las aplicaciones remotamente a través de Internet.

- La arquitectura debe permitir que una instancia de la aplicación corra en un servidor y permita que varios clientes accedan a la misma concurrentemente, al contrario del software tradicional que debe correr un número de instancias igual al número de clientes a utilizarlo.
- Las aplicaciones deben permitir ser actualizadas centralizadamente lo que elimina la necesidad de bajar parches o actualizaciones posteriores.

Los precios de las aplicaciones ofrecidas en este modelo trabajan generalmente bajo una base por usuario, cobrando cuotas adicionales si se desea aumentar ancho de banda o espacio de almacenamiento en el servidor. Este modelo puede ser extremadamente atractivo a pequeñas y medianas empresas, ya que reduce drásticamente los costos del software y los servicios añadidos a éste.

En el modelo de desarrollo de software tradicional, las empresas proveedoras venden aplicaciones grandes y costosas con enfoque de productos, a corporaciones o empresas grandes, pero actualmente, ofrecen una opción en línea con enfoque de servicios con pago de cuotas periódicas para clientes que usualmente son pequeñas o medianas empresas.

El modelo de desarrollo de software orientado a prestación de servicios le permite a las empresas crear valor a través de la administración estratégica de la información y sus procesos de negocios. Esto es posible ya que las empresas pueden enfocarse únicamente en mejorar su interacción con los clientes ya que este modelo elimina las barreras tecnológicas mejorando los procesos de negocios que hagan uso de tecnología, al mismo tiempo que reduce costos administrativos a la empresa y les representa ahorro en gastos

por los bajos precios de los servicios ofrecidos. Actualmente la tecnología le sirve de palanca a las pequeñas y medianas empresas para poder competir con corporaciones o empresas grandes y mantenerse en el mercado.

El uso de este modelo permite enfocar los recursos de las empresas en sus estrategias de negocios en vez de enfocarlos en la tecnología necesaria para llevarlas a cabo.

1.2 Modelo tradicional vrs. Modelo SaaS

1.2.1 Conceptos generales del modelo tradicional

El modelo tradicional de desarrollo de software consiste en la adquisición de una aplicación de software por parte de un cliente a través de una licencia. Esta licencia le permite a un cliente hacer uso de funcionalidades específicas de la aplicación, dependiendo del tipo de licencia adquirido. En este modelo una aplicación de software es tomada como un producto y no como un servicio ya que ésta es vendida como un paquete con licencia por lo que el cliente es el dueño de la aplicación y puede hacer uso de ella en la forma que lo desee siempre y cuando se mantenga dentro de los límites definidos por la misma.

Dentro de este modelo es el cliente el que se encarga de mantener la infraestructura necesaria para soportar la aplicación. Este modelo es muy usado por empresas grandes, ya que las necesidades tecnológicas en sus procesos de negocio resultan ser más específicas que las de una pequeña o mediana empresa y son éstas quienes generalmente cuentan con el capital necesario para pagar una aplicación hecha a la medida y mantener la infraestructura que la soporte. Además, el hecho que el enfoque de desarrollo SaaS sea relativamente nuevo provoca que estas empresas no tengan la confianza suficiente para compartir su información con un proveedor de servicios de

software por ser considerada demasiado importante. Esta y otras desventajas serán vistas más adelante.

1.2.2 Similitudes y diferencias

Al momento de realizar una comparación entre ambos modelos de negocio, podemos darnos cuenta que son pocas las similitudes entre ellos, ya que la diferencia de enfoques aumenta la brecha en la forma de funcionar de cada uno. Entre las similitudes identificadas entre ambos modelos se pueden describir las siguientes:

- Ambos modelos hacen uso de licencias.
- Los modelos ofrecen a los clientes funcionalidades específicas de una aplicación de software de acuerdo a una presentación adquirida (Ya sea que la aplicación sea vendida en distintos paquetes o licencias.). Esto puede observarse claramente en ambos enfoques si tomamos como ejemplo de un modelo tradicional, las distintas ediciones que ofrece Microsoft para su Sistema Operativo Windows XP (Home, Professional), y como ejemplo de un modelo SaaS, las distintas versiones de la contabilidad en línea Servicont que ofrece el sitio www.servipyme.cl (Servicont Plus, Servicont Premium).
- El cliente es el dueño de la información manejada en las aplicaciones desarrolladas bajo ambos modelos. Actualmente existe la falsa creencia de que en el modelo SaaS es el proveedor de servicios de aplicación el propietario de la información, lo que ha generado desconfianza por parte de los clientes de utilizar este tipo de servicios. Cabe aclarar que el proveedor de servicios es únicamente propietario del espacio físico en el cual se almacena la información, más no de la información como tal, lo que debería quedar bien definido en los términos de contratación del servicio que el proveedor establezca con el cliente.

A pesar de que ambos modelos manejen licencias existe diferencia en la forma de ofrecer las mismas, ya que en el modelo tradicional, el cliente adquiere la licencia al momento de comprar el producto y se vuelve propietario de ella, mientras que en el modelo SaaS es la empresa proveedora del servicio la propietaria absoluta de las licencias, que son únicamente rentadas por el cliente durante el tiempo que éste pague la cuota periódica correspondiente. Al hablar de las diferencias que existen entre ambos modelos de desarrollo se pueden mencionar las siguientes:

- En el modelo tradicional el cliente realiza un solo pago para adquirir un producto de software mientras que en el modelo SaaS, el cliente paga una cuota periódica (mensual, semestral, anual, etc.) para hacer uso de los servicios que ofrece la aplicación proporcionada.
- Los costos de una aplicación desarrollada bajo el modelo tradicional son altos la mayoría de las veces, mientras que los costos de utilizar una aplicación en el modelo SaaS son mucho más accesibles.
- Las actualizaciones de aplicaciones desarrolladas bajo el modelo tradicional suelen tener costos muy elevados, mientras que para el cliente las actualizaciones de las aplicaciones desarrolladas bajo el modelo SaaS son transparentes en costos.
- En el modelo tradicional, el cliente es el dueño de la licencia del producto de software, mientras que en el modelo SaaS, el cliente alquila la licencia del producto de software, propiedad del proveedor, bajo la forma de un servicio.
- Las aplicaciones en el modelo SaaS son desarrolladas exclusivamente en plataforma Web, mientras que las aplicaciones en el modelo tradicional pueden ser desarrolladas en plataforma Web o escritorio. Esto se debe a que el acceso a las aplicaciones en el modelo SaaS se realiza

vía Internet a través de un explorador (Internet Explorer, Mozilla FireFox, etc.), mientras que las aplicaciones en el modelo tradicional pueden ser accesadas ya sea por un explorador si se encuentran desarrolladas en una plataforma Web y almacenadas en un servidor remoto, o localmente, a través de su instalación en el equipo a utilizar si fuesen desarrolladas en plataforma de Escritorio.

- Al momento de adquirir una aplicación desarrollada bajo el modelo tradicional, es el cliente el que debe poseer la infraestructura que soporte la misma, mientras que en el modelo SaaS, es el proveedor de servicios de aplicación el que posee la infraestructura necesaria, y el cliente solo hace uso de la aplicación a través de un explorador e Internet.

1.3 Ventajas del modelo SaaS

1.3.1 Perspectiva del cliente

Dentro de las ventajas que el cliente obtiene al utilizar los servicios de aplicaciones desarrolladas bajo el modelo SaaS, se pueden mencionar las siguientes:

- En empresas pequeñas, medianas y de recién comienzo, se reduce la necesidad de una inversión inicial fuerte en aplicaciones de software e infraestructura que soporte las mismas, permitiendo a las empresas utilizar herramientas avanzadas a precios moderados.
- El cliente no debe preocuparse por la instalación e integración de su equipo y las aplicaciones de software a utilizar.
- En empresas que cubren un área geográfica grande, se reducen drásticamente los costos de comunicación y se elimina la duplicidad de información, procesos de negocio y aplicaciones de software, entre sucursales y la agencia central.

- No se necesita poseer un departamento de informática con personal calificado que se encargue de dar mantenimiento a las aplicaciones.
- Las aplicaciones desarrolladas bajo el modelo SaaS poseen los mismos beneficios que un sistema desarrollado bajo el modelo tradicional pero a un menor costo.
- Se eliminan los intermediarios ya que casi siempre la relación de servicios es entre el cliente y el proveedor.
- Por una cuota periódica, los clientes pueden hacer uso de un servicio que incluya alojamiento de la aplicación de software e información manejada en ella, mantenimiento de la aplicación, soporte técnico, actualizaciones periódicas, utilización de servicios avanzados proporcionados por las aplicaciones, capacitación a usuarios, uso de licencias, infraestructura de soporte, seguridad y comunicación entre sucursales. Gracias a esto se reduce de manera importante la carga administrativa y de gestión de una empresa.
- Se pueden implementar nuevos modelos de trabajo, ya que las aplicaciones pueden ser utilizadas desde cualquier lugar. Por lo tanto, los trabajadores pueden acceder a las aplicaciones incluso desde su casa.
- El costo en tecnología es previsible y relativamente constante, por lo que no es necesario estar preparados para grandes cambios en los costos al introducir nuevas versiones de las aplicaciones utilizadas o dar mantenimiento a las mismas.
- Se reduce el costo de utilizar un producto no deseado, ya que el cliente puede probar una aplicación durante un tiempo y si la aplicación no satisface sus necesidades de negocio, el cliente puede simplemente cambiar a otro proveedor, a diferencia de los productos desarrollados bajo el modelo tradicional, en los que se paga el valor total del producto, antes de poder evaluarlo.

- La empresa puede trabajar con información en tiempo real desde cualquiera de sus sucursales.
- La información se mantiene más segura ya que las empresas proveedoras de servicios de aplicación suelen almacenar la información en centros de datos que poseen altos niveles de seguridad, que resultarían muy difíciles de costear para una pequeña o mediana empresa. Además, los centros de datos ya poseen políticas de recuperación a catástrofes, copias de seguridad, replicación de servidores, autenticación en accesos, certificados digitales y más aspectos que implicarían una mayor administración y uso de recursos.
- El servicio le permite a los clientes añadir sucursales y ampliar redes o estaciones de trabajo de forma sencilla ya que normalmente solo es necesaria la adición de usuarios de la misma forma que se haría en una red local.
- Le permite a las empresas enfocarse en su cadena de valor ya que no es necesario enfocar una gran cantidad de recursos (financieros, humanos, etc.) en el área tecnológica.
- Se facilita crear nuevas formas de integración y cooperación entre empresas.

1.3.2 Perspectiva del proveedor de servicios

También existen ventajas por parte del proveedor de servicios al utilizar el modelo de desarrollo SaaS como modelo de negocios. Entre éstas se pueden incluir:

- Se reducen los costos de distribución del software a los clientes ya que las aplicaciones se encuentran centralizadas en los servidores del proveedor y utilizadas únicamente a través del explorador.

- Se reducen los costos de mantenimiento, actualización y soporte ya que la empresa proveedora se asegura de que todos sus clientes estén utilizando la misma aplicación de software con las mismas actualizaciones, por lo que los servicios de soporte son especializados y se necesita menor cantidad de recursos para su administración.
- Se recibe un ingreso fijo de efectivo cada cierto período de tiempo que le permite tener cierta estabilidad a la empresa proveedora, contrario a los ingresos percibidos en el modelo tradicional que resultan variables y dependen de la cantidad de ventas.
- Mejora la atención al cliente ya que la resolución de errores es rápida debido a que se hace de forma centralizada sin necesidad de crear parches y aplicarlos en todos los clientes, pues todas las actualizaciones se hacen directamente sobre el servidor.
- Elimina el problema de que los clientes compartan ilegalmente el software con otros y la piratería, ya que si el cliente no paga su cuota queda sin servicio.
- El hecho que hayan muchos clientes utilizando la aplicación al mismo tiempo haciendo uso de diferentes funcionalidades permite que se puedan realizar estudios para optimizar las aplicaciones de una forma más sencilla.
- Abre las puertas a nuevas formas de negocio al consolidar una aplicación entre varios clientes. Un ejemplo de esto puede ser una aplicación de contabilidad en línea que permite añadir servicios extra como impresión de libros, declaración de impuestos en línea, pago de impuestos en línea, publicidad contable entre clientes, etc.

1.4 Desventajas del modelo SaaS

1.4.1 Perspectiva del cliente

El modelo SaaS puede presentar algunas desventajas para el cliente, las cuales serán descritas a continuación:

- Para hacer uso de aplicaciones de software desarrolladas bajo el modelo SaaS, es necesario poseer una conexión de Internet de buena calidad ya que la mayoría de transacciones realizadas en estas aplicaciones requieren de una aceptable cantidad de banda ancha para que la velocidad de respuesta en el lado del cliente pueda ser comparable con una aplicación de escritorio. Además, si la conexión de Internet falla, no se puede seguir haciendo uso de las aplicaciones, por lo que se depende completamente del proveedor de servicio de Internet.
- Temor de almacenar la información de la empresa en un lugar remoto fuera de la misma ya que esta información puede ser confidencial. Este factor se convierte en un problema importante al momento de comercializar una aplicación desarrollada bajo el modelo SaaS.
- Si la empresa proveedora o la empresa cliente no implementa políticas de seguridad adecuada, puede aumentar el riesgo de robo de la información al acceder a las aplicaciones desde cualquier lugar.
- El cliente se encuentra propenso a aumentos de cuotas por parte de la empresa proveedora de los servicios. Para evitar este tipo de situaciones debe pactarse una cuota a largo plazo a través de un contrato.
- La empresa cliente depende de la supervivencia de la empresa proveedora para poder seguir haciendo uso de las aplicaciones y mantener su información segura. El riesgo puede disminuir si se pacta con la empresa proveedora una política de envío periódico de backups

de la información para que esta pueda ser utilizada por otras aplicaciones ante cualquier eventualidad.

1.5 Perspectiva del proveedor de servicios

- Para prestar este tipo de servicios es necesario realizar inversiones fuertes en infraestructura que garantice calidad y seguridad de las aplicaciones ofrecidas.
- Los costos de call center, soporte técnico, personal administrativo, etc. Son altos ya que se requiere de una gran cantidad de empleados para poder atender una mayor cantidad de clientes.
- Aumenta la dificultad de crear tarifas que se adapten a la utilización de recursos de la mayoría de clientes.

1.6 El proveedor de servicios de aplicación

Un proveedor de servicios de aplicación es una empresa que ofrece a sus clientes el uso de múltiples aplicaciones de software a través de una conexión segura de Internet utilizando como modelo de pago una cuota periódica. El proveedor de servicios de aplicación es la base del modelo de desarrollo de software orientado a prestación de servicios (SaaS).

El proveedor de servicios de aplicación permite que sus clientes no necesiten adquirir servidores para instalar las aplicaciones en ellos, ya que él mismo actúa como "departamento de informática" contratado por el cliente, siendo el proveedor el que se encargue de mantener la infraestructura necesaria que permita que las aplicaciones de software funcionen correctamente. Los clientes pueden hacer uso de las aplicaciones del proveedor

sin necesidad de instalarlas en sus máquinas ya que accesan a ellas a través de un browser.

Un proveedor de servicios de aplicación se distingue por su capacidad de acelerar y poner en marcha sus aplicaciones de software y la posibilidad de transferir servicios y operaciones a terceros si esto es requerido.

El problema principal que afronta un proveedor de servicios de aplicación radica en convencer a sus clientes de que su información permanecerá segura en manos de terceros. Además, uno de los retos más grandes que debe afrontar un proveedor de servicios de aplicación es la forma de cobrarle a sus clientes las cuotas periódicas, ya que éstas deberán ser cobradas a clientes alrededor del mundo, por lo que los sistemas de cobro deben ser robustos y muy bien administrados para evitar el mal manejo de los ingresos de la empresa.

1.6.1 Tipos de proveedores de servicios de aplicación

En el transcurso de los últimos años se ha ido catalogando a los diferentes tipos de proveedores en cuatro categorías descritas a continuación:

- *Proveedor especialista*

Este proveedor ofrece un solo servicio a través de una infraestructura y aplicaciones de software enfocadas en él. Ejemplos de este tipo de proveedores son las empresas que se dedican al procesamiento a terceros de pagos por tarjeta de crédito como *2Checkout* (www.2checkout.com).

- *Proveedor por Mercado Vertical*

Ofrece una solución en la forma de un paquete completo para un tipo de cliente específico. Las aplicaciones de este tipo de proveedores y los servicios ofrecidos por ellos cubren muchas áreas y funciones relacionadas con el mismo tipo de cliente. Dentro de este tipo de proveedores se encuentran las empresas que ofrecen aplicaciones como CRM en línea y otros, por ejemplo *Zoho* (www.zoho.com).

- *Corporaciones*

Ofrecen una amplia gama de servicios y soluciones para todo tipo de clientes y sus necesidades, como es el caso de *Google*. Aunque la idea de pensar en *Google* como un proveedor de servicio de aplicaciones pueda sonar fuera de lugar, *Google* es un excelente ejemplo que describe de forma muy clara como funciona el modelo de desarrollo de software orientado a prestación de servicios y el alcance del mismo.

- *Proveedores Locales*

Estos ofrecen servicios a pequeñas o medianas empresas dentro de un área limitada.

Algunos especialistas definen un quinto tipo de proveedor de servicio de aplicaciones llamado "Proveedor por Volumen". Este proveedor es básicamente un especialista que ofrece un paquete completo a bajo costo a través de su sitio Web. Este fue el caso de *Paypal* cuyo volumen era reducir el costo por unidad de cada transacción.

Además de estos tipos, algunas empresas multi-línea utilizan conceptos del modelo de desarrollo de software orientado a prestación de servicios como un modelo de negocios en particular para ciertos clientes en específico. Este es el caso de *CPA* (www.cpaasp.net).

1.6.2 Características de un buen proveedor

Independientemente de las aplicaciones de software que un proveedor ofrezca hay varios aspectos que deben ser tomados en cuenta para considerar al proveedor de servicios como confiable y bueno.

El contar con sistemas y procedimientos que garanticen la seguridad de las aplicaciones hace más atractiva una aplicación SaaS. Es probable que empresas interesadas estén dispuestas a realizar auditorías de seguridad a las aplicaciones que ofrece un proveedor. Es por esto que resulta muy recomendable contar con auditorías externas realizadas previamente a los productos ofrecidos para aumentar la confianza de los clientes. Todos los integrantes de la empresa proveedora deben seguir las políticas de seguridad que deberían abarcar como mínimo el cumplimiento de las normas LOPD. Otra medida de seguridad es la creación de un plan de contingencia que describa los procedimientos a seguir ante la aparición de algún suceso significativo que pueda traer consecuencias graves a la empresa.

Un proveedor de servicios de aplicación debe contar con soporte 24/7 o en su caso, de un horario suficiente para atender a los usuarios de las aplicaciones. Además, sus aplicaciones deben contar con un buen sistema de ayuda en línea que pueda proporcionar asistencia no personalizada a los usuarios así como la automatización del sistema de información sobre actualizaciones y mejoras.

Es importante contar con un plan de formación presencial que permita dar a conocer los alcances de la aplicación así como las mejores prácticas de uso. Esto le permite a la empresa proveedora obtener una buena retroalimentación de las aplicaciones para que las mejoras faciliten la satisfacción de las necesidades de la mayoría. Además, se puede contar con sistemas de comentarios y sugerencias en línea para una mayor comunicación con el cliente. Debería ser algo habitual realizar encuestas a los clientes para conocer su nivel de satisfacción.

Los clientes necesitan que el proveedor de servicios de aplicación pueda ser capaz de adaptarse a sus necesidades futuras manteniendo o mejorando su calidad de servicio. Esto implica que el proveedor pueda ser capaz de ampliar su capacidad de almacenamiento de información, número de usuarios, cantidad de transacciones, rendimiento de la aplicación, métodos de interacción con otras aplicaciones, etc. Además, también es necesario que las aplicaciones cuenten con una alta adaptabilidad a las necesidades específicas de la mayoría de los clientes.

En cuanto a las mejoras, es importante que el proveedor pueda mostrar líneas de desarrollo futuro de sus aplicaciones, ya que es muy común que los clientes deseen conocer las mejoras y nuevas funcionalidades que se irán incorporando a las aplicaciones y su planificación cronológica aproximada. Es necesario que el proveedor conozca las necesidades de sus clientes para que las mejoras produzcan el mayor beneficio posible.

Un aspecto crítico es la estabilidad de la empresa. Para que los clientes utilicen con confianza una aplicación SaaS, es necesario que el proveedor muestre una imagen financiera estable y auto sostenible. Para ello es necesario planificar de una manera exhaustiva los costos de mantener las aplicaciones

funcionando con un alto nivel de calidad y acoplar las tarifas de las mismas para que puedan soportar el nivel de crecimiento de usuarios y servicios.

2. ASPECTOS TÉCNICOS DEL MODELO SAAS

2.1 Arquitectura física

La infraestructura utilizada para soportar aplicaciones desarrolladas bajo el modelo SaaS puede ser muy diversa, ya que existe un sinnúmero de herramientas que pueden ser de gran utilidad para mejorar varios aspectos de calidad de nuestras aplicaciones. Existen ciertos aspectos mínimos que deben ser asegurados para que una arquitectura física que soporte aplicaciones desarrolladas bajo el modelo SaaS garantice su calidad de funcionamiento. Estos aspectos son descritos a continuación.

2.1.1 Infraestructura que garantice la seguridad

Al pensar en la infraestructura de seguridad, existen dos tipos de infraestructura que se deben tomar en cuenta, ya que ambos son extremadamente importantes y la ausencia de uno de ellos puede traer consecuencias catastróficas para una empresa proveedora de servicios de aplicación. Por un lado se encuentra la infraestructura que debe garantizar la seguridad de la infraestructura en sí, y por el otro, se encuentra la infraestructura que debe garantizar la seguridad de la información y aplicaciones de software.

Dentro de la infraestructura con la que debemos contar para garantizar la seguridad de la infraestructura en sí, se tiene la siguiente:

- Un cuarto frío que permita que los servidores se encuentren a una temperatura adecuada. Las aplicaciones SaaS son utilizadas por muchos clientes al mismo tiempo, por lo que los servidores que las alojan deben realizar procesamientos continuos que provocan que el equipo se caliente demasiado. Si no se cuenta con equipo que mantenga una temperatura baja en el ambiente que almacena los servidores, éstos podrían sobrecalentarse y quemarse o dejar de funcionar ya que los procesadores tienden a congelarse y parar su funcionamiento si sobrepasan cierta temperatura.
- Se debe contar con accesos restringidos a los ambientes por lo que es necesario poseer algún tipo de identificador que no permita el ingreso de cualquier persona, como lectores de huella digital, teclados para claves, lectores de retina, etc. Este equipo garantiza tanto la seguridad de la infraestructura como de la información ya que evita que el equipo y/o la información puedan ser robados o destruidos.
- También se debe contar con equipo para prevención de desastres como extintores, instalaciones antisísmicas, personal de seguridad, etc.

En cuanto a la seguridad de la información y las aplicaciones, se debe tomar en cuenta lo siguiente:

- Se debe hacer uso de cortafuegos (firewalls) que bloqueen el ingreso de conexiones no autorizadas a los servidores.
- Se deben utilizar enrutadores (routers) para que garanticen el envío eficiente de la información a través de los servidores
- Se debe contar con balanceadores de carga que distribuyan eficientemente los accesos en los distintos servidores que alojan las aplicaciones para evitar la sobre utilización de algunos servidores y el desperdicio de recursos en otros.

- Deben manejarse protocolos de seguridad de redes como SSH para evitar la intervención de las comunicaciones.
- Se deben adquirir certificados de seguridad SSL que permitan que las comunicaciones entre el servidor y los clientes sean seguras.
- Utilizar Antivirus, anti spyware, anti troyanos, etc. Es recomendable para proteger los servidores de cualquier infección.
- Se puede hacer uso de sistemas capaces de detectar intrusos en los sistemas a través de la búsqueda de anomalías en las fuentes más comunes de ataques en aplicaciones Web como lo son, los correos electrónicos, DNS, ICMP, etc.

2.1.2 Infraestructura que garantice la disponibilidad de la información

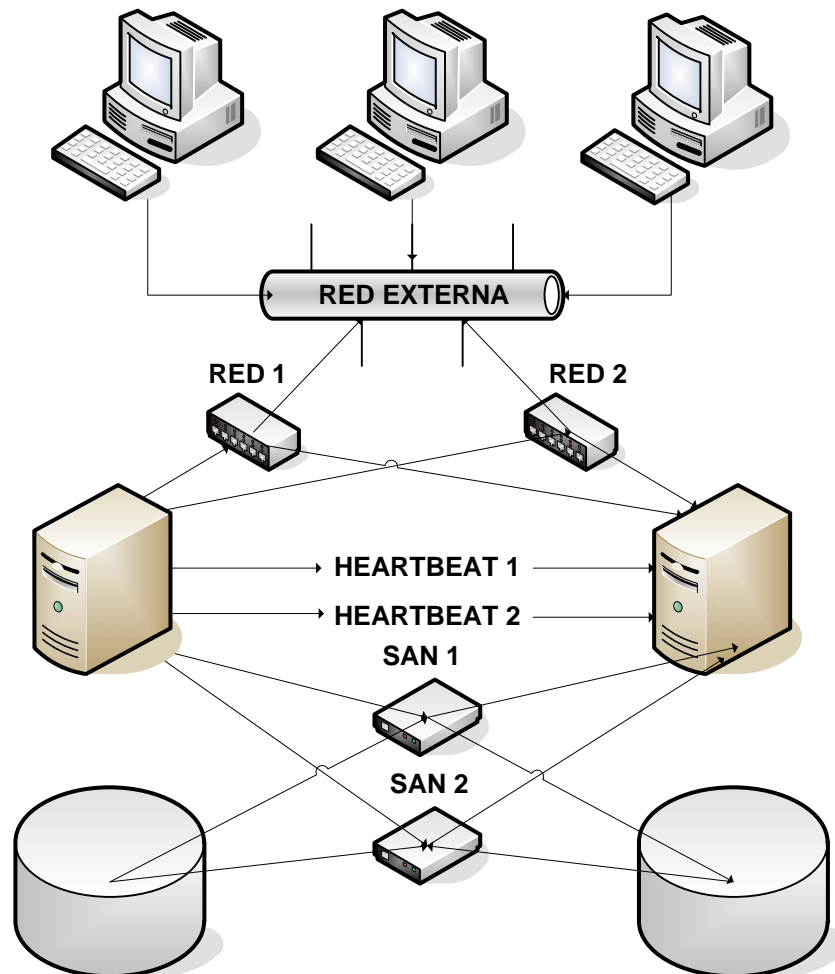
La demanda por sistemas de software y hardware que garanticen su funcionamiento durante casi la totalidad del tiempo ha crecido exponencialmente. Por su naturaleza, las aplicaciones de software desarrolladas bajo el modelo SaaS no pueden carecer de esta característica ya que su rango de tiempo de utilización es de casi 24 horas, debido a que su capacidad de ser utilizadas a través de la Web permite que sus clientes accedan a ellas desde cualquier parte del mundo, en horarios de trabajo distintos. Es por esto que la infraestructura que soporta este tipo de aplicaciones debe garantizar la alta disponibilidad de la información y los sistemas de software en todo momento. La arquitectura que posee este tipo de infraestructura también recibe el nombre de "Arquitectura de tolerancia a desastres". A continuación se describen los aspectos mínimos que debe tener una arquitectura que garantice la alta disponibilidad:

- El sistema de servidores que soportan las aplicaciones y las bases de datos con la información de los clientes debe estar configurado en

clusters. Este tipo de configuración opera con sistemas redundantes de servidores o nodos que son utilizados para proporcionar servicio cuando los servidores en uso fallan. Normalmente, si un servidor de aplicaciones falla, las aplicaciones permanecerán sin uso hasta que el servidor sea reparado. Para remediar esto, las configuraciones en cluster detectan fallos en el hardware o software y reinician la aplicación de forma inmediata en otro servidor sin necesitar intervención administrativa. Este proceso recibe el nombre de "Recuperación a fallos (Failover)". Como parte del proceso, el software de clustering puede configurar un servidor antes de iniciar las aplicaciones en él. Por ejemplo, podría ser necesario que ciertos archivos del sistema deban ser configurados, sean necesarias configuraciones en los dispositivos de red o algunas aplicaciones complementarias deban estar corriendo para que las aplicaciones bajo el modelo SaaS funcionen correctamente. Los servidores se encuentran enlazados a través de conexiones "heartbeat" que verifican el estado de los mismos, y a su vez se encuentran enlazados a las bases de datos a través de dispositivos SAN.

La configuración más común de un sistema de clusters incluye 2 nodos, ya que ésta es la cantidad de servidores mínima para proveer redundancia. El siguiente diagrama muestra la arquitectura básica de una configuración de servidores en cluster.

Figura 1. Arquitectura de una red de clusters de alta disponibilidad



- Es necesario contar con un sistema de backups que permita que la información no se pierda en caso de falla de alguno de los dispositivos de almacenamiento. Si se cuenta con los recursos, se puede contar con un sistema de base de datos en espera que garantice la disponibilidad de la información de los clientes e incluso permita que las consultas de información sean más rápidas, si se utiliza esta base de datos también como una base de datos de consulta.

- Ya que las aplicaciones desarrolladas bajo el modelo SaaS deben ser montadas en servidores Web que publiquen las mismas, se debe garantizar que la conexión de Internet de estos servidores se mantenga siempre funcionando. Para esto es necesario contar con una conexión principal y una de respaldo de diferente proveedor de servicio de Internet para garantizar que pueda utilizarse una segunda conexión si la conexión principal falla.

2.1.3 Infraestructura que garantice alto rendimiento

Es necesario que el equipo de hardware que soporta las aplicaciones desarrolladas bajo el modelo SaaS sea capaz de responder a múltiples peticiones simultáneas en el menor tiempo posible. Es por esto que no es recomendable utilizar computadoras personales como sustitutos de servidores formales. Un servidor de aplicaciones SaaS debería permitir la utilización de múltiples procesadores para trabajar en peticiones de forma simultánea, alta capacidad de expansión de memoria RAM, soporte para varios dispositivos de almacenamiento y múltiples entradas de red para evitar un cuello de botella en el que todas las peticiones pasen por la misma entrada.

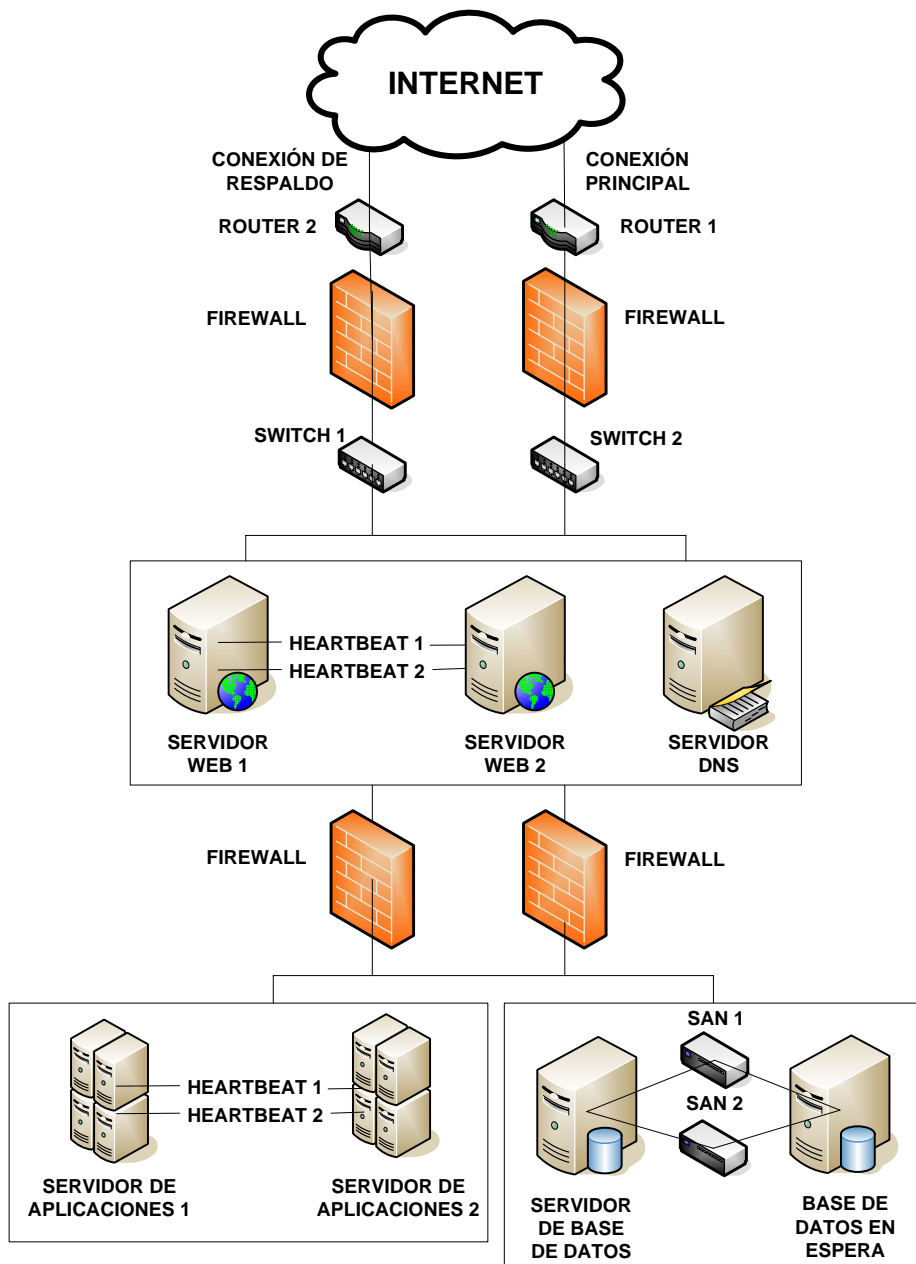
Además es necesario separar los distintos aspectos funcionales de la aplicación en servidores específicos que se enfoquen únicamente en porciones de la misma. Esta separación es posible cuando se diseña una correcta arquitectura de software de la que hablaremos adelante.

2.1.4 Infraestructura recomendada

Al unir todos los elementos mencionados con anterioridad, se puede decir que se cuenta con la infraestructura necesaria para soportar aplicaciones desarrolladas bajo el modelo SaaS y garantizar su buen funcionamiento y

calidad. A continuación se muestra un diagrama con una configuración adecuada que involucra todas las herramientas descritas anteriormente.

Figura 2. Arquitectura de una infraestructura recomendada para soportar aplicaciones SaaS



En la parte superior de la figura se cuenta con una conexión principal a Internet y una de respaldo proporcionada por un proveedor de servicios de Internet distinto, que permite mantener la conexión de la aplicación en caso que la conexión principal falle o deje de funcionar. Se puede observar que ambas conexiones se encuentran protegidas con cortafuegos (firewalls) que garantizan la seguridad de las conexiones entrantes.

En el siguiente bloque se tienen servidores de nombres que permiten colocarle un nombre a las aplicaciones para que puedan ser identificadas en Internet. Además se cuenta con servidores Web que tienen almacenados los Sitios Web de las aplicaciones. Los servidores Web cuentan con certificados de seguridad SSL que garantizan una transmisión de información segura entre el cliente y la aplicación.

Los bloques posteriores se encuentran conectados al bloque mencionado anteriormente, a través de una conexión segura protegida por cortafuegos (firewalls).

En el bloque inferior izquierdo se encuentran los servidores de aplicaciones que contienen todas las funciones y procedimientos de la capa de negocios de la que se hablará más adelante en la arquitectura de software.

En el bloque inferior derecho se encuentra configurado un sistema de base de datos en espera que garantiza la alta disponibilidad de la información de los clientes.

Como se puede apreciar en la figura anterior, todos los aspectos funcionales de la aplicación (sitios Web que sirven como interfaz de usuario, lógica del negocio, almacenamiento de la información del cliente) fueron separados en distintos bloques con servidores específicos (servidores Web, servidores de aplicaciones, servidores de bases de datos) que permiten maximizar el

rendimiento de la aplicación evitando la sobrecarga de los servidores disminuyendo la cantidad de funciones que deben ejecutar.

Es importante mencionar que la infraestructura física necesaria para soportar aplicaciones que funcionan bajo el modelo SaaS la proporciona la mayoría de servicios de hosting disponibles en la actualidad a precios bastante cómodos y accesibles, lo que permite que publicar una aplicación de este tipo en Internet no represente una inversión inicial muy grande.

2.2 Arquitectura de software

Habiendo detallado la arquitectura física necesaria para soportar aplicaciones desarrolladas bajo el modelo SaaS, es necesario también hablar acerca de la estructura de software que una aplicación de este tipo debe poseer.

La Web ha ido evolucionando a través de distintas etapas. La etapa cero, es la más primitiva de todas, en las que los sitios Web eran básicamente documentos de texto en esencia estáticos. Con la llegada de javascript y los applets, la Web dio un paso atrás ya que la mayoría de proyectos utilizaron estas herramientas únicamente para mejorar la apariencia de los sitios pero sin ninguna mejora relevante. Empresas mucho más inteligentes salieron de esta etapa para introducirse en la etapa dos, en la cual, el cliente y el servidor fueron hechos de una forma más inteligente. Esto significó evolucionar a mecanismos que permitieran mezclar la experiencia del usuario con bases de datos que se encontraban en la parte "de atrás", esencialmente permitiendo que la mayoría de las aplicaciones fueran accesibles a través de un explorador. Actualmente, nos encontramos en la etapa tres, en la que los sitios tienen una mejor semántica y hacen mejor uso del potencial de la computación distribuida.

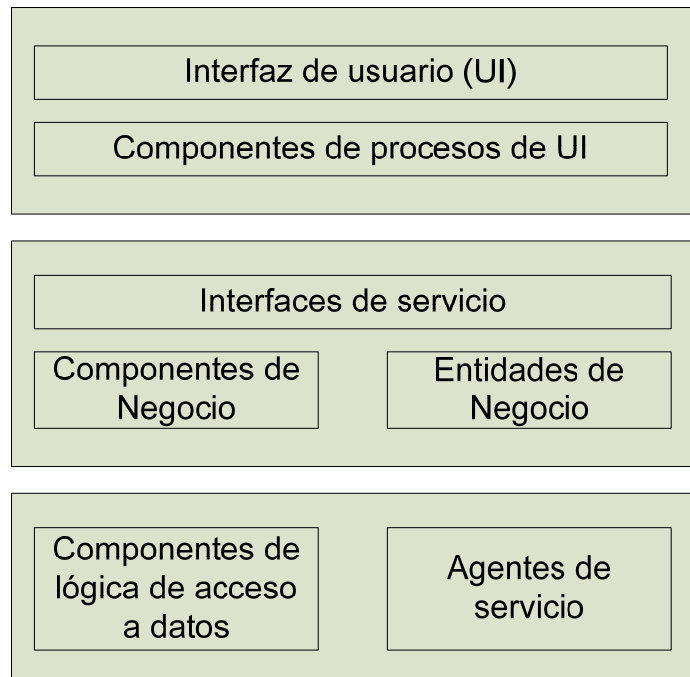
Durante la transición de la etapa dos a la tres, sitios exitosos que proveían valor real a un negocio fueron transformándose en verdaderas aplicaciones Web, aplicaciones que ahora son utilizadas como base del modelo SaaS. Una aplicación Web (utilizada en el modelo SaaS), es un sitio Web en el cual las entradas del usuario, incluyendo la navegación en el sitio y el ingreso de datos, afectan el estado de un negocio. En este tipo de aplicaciones existen tres factores críticos en los que se debe enfocar el desarrollo:

- Integración de aplicaciones de uso clásico o cotidiano (Legacy applications).
- Evolución continua.
- Arquitectura para tráfico pico.

Factores críticos secundarios incluyen seguridad y garantizar una experiencia cómoda y satisfactoria para el usuario. En estos sitios la velocidad es lo más importante, velocidad de desarrollo de las aplicaciones y velocidad en la experiencia de uso del usuario.

El análisis de la arquitectura de la mayoría de aplicaciones Web divididas en capas revela muchos tipos de componentes comunes. A través de la siguiente figura se presenta la arquitectura de capas recomendada para aplicaciones Web utilizadas en modelos SaaS:

Figura 3. Arquitectura de software recomendada para una aplicación Web bajo el modelo SaaS.



2.2.1 Capa de presentación

Esta capa está compuesta de todos los componentes que permiten la interacción de los usuarios con las aplicaciones. La capa se encuentra dividida en la interfaz de usuario (UI) y los componentes de procesos de interfaz de usuario.

Todas las aplicaciones Web desarrolladas bajo el modelo SaaS necesitan proveer una forma de interacción entre el usuario y la aplicación. Tomando como ejemplo una contabilidad en línea, el sitio debe proveer alguna forma que le permita al usuario ingresar cuentas contables, generar pólizas, ingresar rubros, etc. Las interfaces de usuario de las aplicaciones Web utilizadas en el modelo SaaS son presentadas como páginas Web desarrolladas principalmente

en lenguaje HTML, mostrando la información de los clientes a través de la generación de ciertas partes de las páginas utilizando lenguajes del lado del servidor como C# o Visual Basic si se utiliza ASP.NET, PHP, etc.

En muchos casos, la interacción del usuario con la aplicación sigue un proceso predecible. Por ejemplo, en la contabilidad en línea se puede implementar un proceso para ver el detalle de pólizas contables ingresadas anteriormente que obligue al usuario a seleccionar una póliza de una tabla y muestre la información de la póliza al hacer clic sobre una lupa que se encuentre a un costado de la póliza seleccionada. Para ayudar a la sincronización y composición de esta interacción del usuario con la aplicación, puede ser de mucha utilidad manejar estos procesos haciendo uso de componentes separados. De esta manera el flujo del proceso y la administración de los estados del mismo no se encuentran codificados de una forma rústica dentro de la interfaz de usuario, permitiendo que este motor de interacción pueda ser reutilizado en interfaces de usuario distintas.

Los componentes de interfaz de usuario administran su interacción con la aplicación. Ellos se encargan de desplegar información al usuario, recibir información de él, interpretar eventos que el usuario realiza al actuar con la información del negocio, cambiar el estado de la interfaz de usuario o ayudar al usuario a realizar cierta tarea.

Las interfaces de usuario de una aplicación SaaS usualmente están compuestas de elementos que despliegan información al usuario y reciben y validan información de él. Cuando un usuario realiza una acción dentro de la página, se levanta un evento que ejecuta cierto código incluido en funciones del controlador del evento. Estas funciones en su momento pueden llamar

componentes de negocios, de acceso a datos o de procesos de usuario, y actualizar la interfaz de usuario (el sitio Web) de acuerdo a la acción ejecutada.

Los elementos de una interfaz de usuario deben cubrir ciertos aspectos para asegurar que una aplicación Web del modelo SaaS funcione correctamente:

- Deben asistir al usuario en el ingreso de información proveyendo consejos visuales (tool tips), validaciones y los controles apropiados para cada tarea.
- Deben restringir los tipos de información que un usuario puede ingresar. Por ejemplo, un campo que reciba cantidades debe permitir únicamente el ingreso de números.
- Deben validar semánticamente la información que se ingresa. Por ejemplo, un campo que reciba edades no debe permitir que se ingresen edades ilógicas como números negativos o edades superiores a los 150 años.
- Deben realizar mapeos o transformaciones simples de valores ingresados por el usuario para proveer información necesaria a controles siguientes en el flujo de los procesos. Por ejemplo, un dropdown (seleccionador de opciones) puede mostrar el nombre de cada departamento de Guatemala, pero enviar el código del departamento seleccionado por el usuario al dropdown de municipios, para que muestre los municipios apropiados.
- Deben permitir dar formato a la información mostrada al usuario para que sea más legible.
- Deben proveer al usuario información del estado de una transacción. Por ejemplo mensajes de “en proceso...” cuando se esté ejecutando alguna acción en el sitio.

2.2.2 Capa de negocios

Esta capa es el núcleo de una aplicación. Está compuesta de todos los componentes que ejecutan la lógica de la aplicación. En esta capa se realizan los procesos y funcionalidades específicas de una aplicación, relacionados con el propósito real de la misma.

Después que la información requerida es recolectada por algún proceso de usuario, ésta puede ser utilizada para ejecutar algún proceso de negocio. Por ejemplo, después que los datos de una transacción contable han sido recolectados, el proceso de creación de una póliza contable puede iniciar. Este tipo de operaciones son realizadas con componentes de negocios. Estos componentes son necesarios para implementar reglas y tareas del negocio.

Para transformar la lógica de negocios como servicio, es necesario desarrollar interfaces de comunicación que puedan ser utilizadas desde diferentes aplicaciones o capas. Estas interfaces reciben el nombre de “Interfaces de Servicio”

Las aplicaciones Web del modelo SaaS usualmente necesitan la transferencia de información entre componentes. Por ejemplo, en la contabilidad en línea puede ser necesario traer de la capa de acceso de datos toda la información de una póliza contable para que sea desplegada al usuario a través del sitio. Este tipo de operaciones son realizadas a través de “Entidades de negocio”. Las entidades de negocio son básicamente estructuras de datos como Datasets, Estructuras XML e incluso clases orientadas a objetos personalizadas, que representan entidades del mundo real con las que debe trabajar la aplicación, como las pólizas contables.

Los componentes mencionados en los párrafos anteriores forman parte de la capa de negocios. La lógica de la aplicación puede ser implementada a través de estos componentes y ser utilizada directamente desde la capa de presentación, o encapsulada en servicios que puedan ser utilizados a través de una interfaz de servicios, normalmente Web Services a ser consumidos desde la capa de presentación.

Al momento de implementar procesos de negocio que estén compuestos de otros procesos de negocio, el código debe asegurar que al ejecutar el proceso de negocio principal, si existe alguna falla en algún proceso de negocio interno, la ejecución del proceso principal aborta y todo regrese al estado inicial antes de ejecutarse dicho proceso, lo que garantiza la ejecución atómica de todos los procesos de negocio involucrados.

2.2.3 Capa de datos

La capa de datos se encarga de la administración de los datos para manipulación y consulta.

La mayoría de aplicaciones SaaS necesitan acceder a datos almacenados, durante la ejecución de algún proceso de negocio. Es necesario abstraer la lógica para acceder a esta información en una capa separada de componentes. Los componentes de esta capa reciben el nombre de “Componentes de lógica de acceso a datos”. Estos componentes centralizan el acceso a los datos y facilitan su configuración y mantenimiento. Cada uno de estos componentes implementa métodos para realizar operaciones de creación, lectura, actualización y eliminación en entidades específicas del negocio.

Al momento de crear componentes de lógica de acceso de datos se deben tener en cuenta los siguientes aspectos:

- Los componentes deben contar con métodos para insertar, actualizar, eliminar y devolver datos. Esto incluye la creación de métodos de pagineo cuando las cantidades de datos son demasiado grandes.
- Se pueden crear clases auxiliares (“helper”) para centralizar la administración de las conexiones y el código que trabaje con alguna fuente de datos específica.
- Se deberían implementar las consultas y operaciones como procedimientos almacenados si la fuente de datos lo soporta, para asegurar un mejor rendimiento y un fácil mantenimiento.

Cuando las aplicaciones necesitan utilizar la funcionalidad de algún servicio externo, podría ser necesario desarrollar componentes que administren la comunicación con estos servicios. Por ejemplo, la contabilidad en línea podría proporcionar un servicio de pago de impuestos en línea con la Superintendencia de Administración Tributaria que necesite el desarrollo de alguno de los componentes descritos anteriormente. Estos componentes reciben el nombre de “Agentes de servicio”.

3. ANÁLISIS Y DISEÑO DE UN PROTOTIPO BASADO EN EL MODELO SAAS

Después de haber dejado claros los conceptos generales y los aspectos técnicos del modelo SaaS en capítulos anteriores, es necesario aplicar los mismos para presentar las características que una aplicación desarrollada bajo este modelo posee. Para esto se desarrollará un prototipo que consiste en un módulo de contabilidad en línea que formaría parte de una suite financiera.

A continuación se presentará la documentación de análisis y diseño para este prototipo.

3.1 Especificación de requerimientos del software

3.1.1 Introducción

3.1.1.1 Propósito

El propósito de este documento es describir los requerimientos funcionales y no funcionales que estarán involucrados en el desarrollo del módulo de contabilidad de la suite financiera. Además, también serán descritos los requerimientos para el módulo de administración de usuarios y el módulo de registro de usuarios nuevos, necesarios para poder hacer uso del módulo de contabilidad.

3.1.1.2 Audiencia del documento

Este documento (o partes de él dependiendo del grado de confidencialidad del mismo) está dirigido a project managers, arquitectos, desarrolladores y testers involucrados en este proyecto.

3.1.1.3 Alcance del proyecto

Este proyecto tiene como propósito el desarrollo del módulo de registro de usuarios nuevos que permitirá a un usuario ingresar a un sitio Web a crear una cuenta para utilizar el servicio contable. También se desarrollará el módulo de administración de cuentas que le permitirá a un usuario administrar los usuarios que harán uso del sistema y las funciones que podrán desempeñar dentro de él. Por último, se desarrollará el módulo contable que permitirá el ingreso de partidas contables, catálogo de cuentas y generación de los principales libros contables.

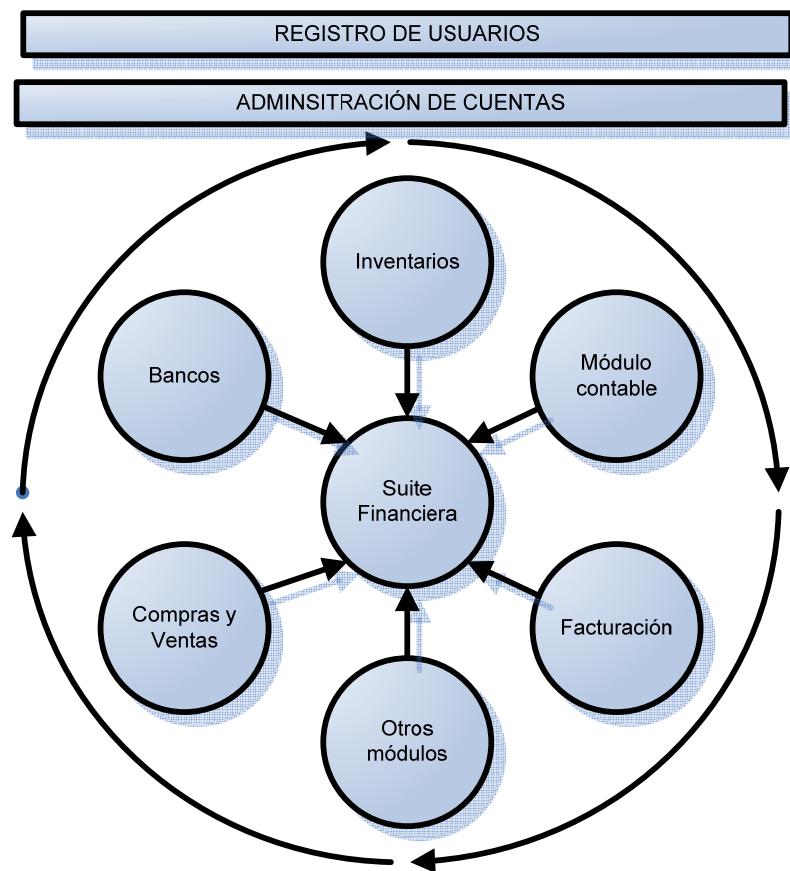
3.1.2 Descripción general

3.1.2.1 Perspectiva del producto

Este documento define los módulos de administración de cuentas, registro de usuarios y el módulo contable. Estos tres módulos formarían parte de una suite financiera que contará con otros módulos como bancos, inventarios, facturación, compras y ventas, entre otros. El módulo contable es esencial para los siguientes módulos ya que sentaría las bases que permitirían que los siguientes módulos guarden los movimientos contables correspondientes, respectivos a sus operaciones. El módulo de administración de cuentas y el módulo de registro de usuarios serán necesarios para poder hacer uso de la suite.

El siguiente diagrama muestra la relación que tendrían los distintos módulos de la suite financiera.

Figura 4. Diagrama de relaciones entre módulos de la suite financiera.



3.1.2.2 Características del producto

El módulo contable contará con las siguientes secciones: Administración de cuentas contables, administración de partidas contables, perfil de usuario, generación de libros contables.

El módulo de administración de cuentas contará con las siguientes secciones: Administración de usuarios, administración de empresas, administración de roles, administración de módulos.

El módulo de registro de usuarios permitirá el ingreso de nuevos usuarios y la adquisición inicial de los módulos deseados. En este caso el módulo contable.

3.1.2.3 Tipos de usuario

- **Administrador**

Este usuario podría acceder a todos los módulos de la suite financiera con todos los privilegios. Además tendría la capacidad de administrar las empresas bajo la cuenta principal, los módulos a utilizar por estas empresas y los usuarios que harán uso de esos módulos, además de los privilegios que cada usuario poseerá dentro de cada módulo.

Ya que este documento abarca hasta el módulo contable, se presentarán únicamente los tipos de usuario para este módulo:

- **Observador**

Este usuario podrá acceder a todas las secciones del módulo contable (administración de cuentas, administración de partidas contables, etc.) únicamente para revisar la información que cada sección contenga. Además, este usuario podrá generar libros contables.

- **Contador**

Este usuario tendrá acceso a todas las secciones del módulo contable y podrá realizar cualquier función dentro de ellas. Este usuario no podrá auditar partidas contables a menos que se le asigne ese rol adicionalmente.

- Auxiliar de contabilidad

Este usuario podrá acceder únicamente a la sección de administración de partidas contables para trabajar en ellas.

- Auditor

Este usuario únicamente podrá auditar partidas contables.

3.1.2.4 Ambiente operativo

- La aplicación funcionará bajo una plataforma Web.
- El sistema deberá ser instalado en un servidor con sistema operativo Windows Server 2003.
- El sistema deberá ser utilizado a través de exploradores de Internet por medio de un canal seguro proporcionado por un certificado SSL.

3.1.2.5 Limitaciones de diseño e implementación

- Deberá utilizarse como administrador de base de datos SQL Server 2005 Professional Edition.
- Deberá utilizarse como plataforma de desarrollo ASP.NET con lenguaje de programación C#.
- Deberá utilizarse el .NET Framework versión 2.0
- Cualquier reporte deberá ser desarrollado utilizando Microsoft Reporting Services.
- El sistema deberá soportar accesos concurrentes.
- La comunicación con los usuarios deberá realizarse a través de canales seguros utilizando protocolos seguros.
- El manual de estándares de programación y documentación del sistema se encuentra especificado en el Anexo 1.

3.1.3 Características del sistema

3.1.3.1 Módulo de registro de usuarios (suscripciones)

3.1.3.1.1 Descripción y prioridad

Este módulo permitirá el ingreso de usuarios nuevos al sistema para hacer uso del módulo contable de la suite financiera. La prioridad de este módulo es alta ya que el mismo representa el paso inicial para hacer uso del sistema. La información de usuario solicitada en este módulo sería la siguiente:

- Nombre completo
- Correo electrónico
- Contraseña con campo para confirmarla

Al iniciar una cuenta en el sitio, el usuario dueño de la cuenta deberá ingresar la información de la primera empresa con la que trabajará, que será tomada como la empresa principal. La información de la empresa a solicitar sería la siguiente:

- País (para este prototipo únicamente se permitiría seleccionar Guatemala).
- Departamento
- Municipio
- Dirección
- Tipo de constitución legal
- Moneda (para este prototipo únicamente se permitiría seleccionar el Quetzal)
- Identificación

Además se solicitará la siguiente información acerca del representante legal de la empresa:

- País (únicamente Guatemala)
- Departamento
- Municipio
- Dirección
- Identificación
- Nombre completo en campos separados
- Fecha de nacimiento

Al registrar al usuario principal de una cuenta se deberá solicitar de forma obligatoria que se seleccione el módulo contable y la tarifa que utilizará para la cuenta.

La página de registro deberá mostrar un convenio de confidencialidad que el usuario deberá aceptar para poder registrar una cuenta.

3.1.3.1.2 Secuencias de estímulo / respuesta

- El usuario ingresa a la página de registro.
- El usuario ingresa la información del usuario principal de la cuenta.
- El usuario ingresa la información de la empresa principal de la cuenta.
- El usuario ingresa la información del representante legal de la empresa principal.
- El usuario selecciona la tarifa del módulo contable que va a utilizar.
- El usuario presiona el botón de aceptar de la página de registro.
- El sistema almacena la información en una base de datos y envía un correo al usuario para activar su cuenta.

- El usuario revisa el correo enviado y hace clic en el link de activación.
- El sistema activa la cuenta del usuario para que pueda ser utilizada.

3.1.3.1.3 Requerimientos funcionales

REQ-A-1: Administración de cuentas.

REQ-A-1.1: Creación de cuentas.

REQ-A-1.2: Activación de cuentas.

REQ-B-1.3: Modificación de información comercial y personal de las cuentas.

REQ-C-1.4: Eliminación de cuentas.

3.1.3.2 Módulo de administración de usuarios y cuentas

3.1.3.2.1 Descripción y prioridad

Este módulo permitirá el manejo de usuarios y módulos para las empresas asociadas al usuario principal de la cuenta. A través de este módulo el dueño de la cuenta podrá seleccionar las funcionalidades que desea utilizar para determinadas empresas y especificar que usuarios pueden hacer uso de las mismas. La prioridad de este módulo es alta, ya que en él se designan las funcionalidades para hacer uso del sistema.

Desde este módulo el usuario principal de la cuenta tendrá la capacidad de crear empresas nuevas asociadas a su cuenta. La información solicitada para las empresas será la misma que la solicitada en el módulo de registro de usuarios. Además, el usuario principal de la cuenta podrá crear usuarios secundarios que harán uso del módulo contable para las distintas empresas asignadas a la cuenta.

La información solicitada para crear un usuario secundario será la siguiente:

- Nombre de usuario
- Correo electrónico

El sistema deberá enviar un correo electrónico que le informará a una persona que un usuario secundario ha sido creado para ella. El correo contendrá un link que dirigirá a la persona a una página en la que podrá activar el usuario y especificar una contraseña para el mismo.

El usuario principal de la cuenta tendrá la capacidad de asignar usuarios secundarios a una o múltiples empresas. Además podrá asignar roles a los usuarios secundarios dependiendo de la empresa a la que hayan sido asignados. Los roles se encuentran especificados en el apartado “Tipos de usuario” de la descripción general de este documento.

3.1.3.2.2 Secuencia de estímulos / respuestas

Secuencia 1

- El usuario principal de la cuenta ingresa a la página de administración de empresas.
- El usuario selecciona la opción de crear una nueva empresa.
- El usuario ingresa la información de una empresa.
- El sistema almacena la información en una base de datos.

Secuencia 2

- El usuario principal de la cuenta ingresa a la página de administración de empresas.

- El usuario selecciona la opción de modificar la información de una empresa.
- El usuario realiza una de las siguientes operaciones en la página de modificación de información de una empresa:
 - El usuario modifica la información de una empresa.
 - El usuario asigna usuarios secundarios a un módulo específico utilizado por una empresa.
 - El usuario asignaría a la empresa posibles módulos posteriores al módulo contable.

Secuencia 3

- El usuario principal de la cuenta ingresa a la página de administración de usuarios.
- El usuario ingresa la información de un usuario secundario.
- El sistema almacena la información en una base de datos y envía un correo electrónico de notificación al dueño del usuario.
- El dueño del usuario revisa el correo y presiona el link que se encuentra dentro del mensaje.
- El link redirige al dueño del usuario a una página de activación de usuarios secundarios.
- El usuario secundario especifica una contraseña.
- El sistema activa al usuario secundario y almacena la contraseña en una base de datos.

Secuencia 4

- El usuario principal de la cuenta ingresa a la página de administración de usuarios.
- El usuario selecciona la opción de modificar la información de un usuario secundario.

- El usuario selecciona una empresa que el usuario secundario utilizará.
- El usuario selecciona el módulo contable.
- El usuario selecciona uno de los roles disponibles para el módulo contable.
- El usuario asigna el rol al usuario secundario para la empresa seleccionada.

3.1.3.2.3 Requerimientos funcionales

REQ-B-1: Administración de empresas.

REQ-B-1.1: Ingreso de empresas nuevas.

REQ-B-1.2: Modificación de información de empresas existentes.

REQ-B-1.3: Eliminación de empresa.

REQ-B-2: Asignación de módulo a empresa.

REQ-B-3: Asignación de paquetes a módulos por empresa.

REQ-B-4: Administración de usuarios.

REQ-B-4.1: Ingreso de usuarios nuevos.

REQ-B-4.2: Notificación a usuarios nuevos.

REQ-B-4.2: Eliminación de usuarios.

REQ-B-5: Asignación de módulos a usuarios.

REQ-B-6: Asignación de roles a usuarios por módulo.

3.1.3.3 Módulo de contabilidad

3.1.3.3.1 Descripción y prioridad

En este módulo se registrarán todas las operaciones contables de una empresa y se generarán los libros contables con el formato legal de Guatemala. Por ser la base del prototipo, este módulo tiene una prioridad alta.

En este módulo se podrán realizar las siguientes operaciones:

- Mantenimiento general del catálogo de cuentas contables. La información mínima a almacenar para una cuenta contable es la siguiente:
 - Código de cuenta.
 - Cuenta contable padre (si la tuviese).
 - Nombre de la cuenta.
 - Tipo de cuenta.
 - Descripción.
 - Especificar si la cuenta es transaccional.
- Mantenimiento general de pólizas contables. La información mínima a almacenar para una póliza contable es la siguiente:
 - Fecha de transacción.
 - Estado de la póliza.
 - Descripción.
 - Detalle de la póliza que puede contener un número de registros indefinido. En cada registro se debe seleccionar una cuenta y especificar un monto en el debe y/o haber.
- Mantenimiento general de compras. La información mínima a almacenar para una compra es la siguiente:
 - Cuenta contable de desembolso.
 - Tipo de documento.
 - Número de documento.
 - Fecha del documento.
 - Fecha de vencimiento del documento.
 - Descripción.
 - Identificación del contribuyente.
 - Nombre del contribuyente.

- Dirección del contribuyente.
- Estado de la compra.
- Categoría de la compra.
- Detalle de la compra que puede contener un número de registros indefinido. En cada registro se debe seleccionar una cuenta y especificar el monto con IVA.
- Mantenimiento general de ventas. La información mínima a almacenar para una venta es la siguiente:
 - Cuenta contable que recibe la venta.
 - Tipo de documento.
 - Número de documento.
 - Fecha del documento.
 - Fecha de vencimiento del documento.
 - Descripción.
 - Identificación del contribuyente.
 - Nombre del contribuyente.
 - Dirección del contribuyente.
 - Estado de la venta.
 - Categoría de la venta.
 - Detalle de la venta que puede contener un número de registros indefinido. En cada registro se debe seleccionar una cuenta y especificar el monto con IVA.
- Generación de los siguientes libros contables:
 - Libro Diario
 - Libro Mayor
 - Balance de saldos
 - Balance general
 - Estado de resultados
 - Libro de compras

- Libro de ventas
- Mantenimiento de la configuración del módulo. Dentro de la sección de configuración del módulo se podrán agregar categorías de compras, categorías de ventas, rubros de compras y rubros de ventas. Además, se podrá modificar la configuración contable de la aplicación. Los campos mínimos que podrán ser modificados para la configuración contable son los siguientes:
 - Mes inicial del período contable.
 - Mes final del período contable.
 - Año del período contable.
 - Mes del período contable.
 - Estado por defecto de las pólizas contables.
 - Número de registro del contador.
 - País del contador.
 - Tipo de identificación del contador.
 - Número de identificación del contador.
 - Departamento del contador.
 - Municipio del contador.
 - Dirección del contador.
 - Nombre completo del contador en campos separados.
 - Fecha de nacimiento del contador.
- Mantenimiento del perfil de usuario. La información mínima a almacenar para el perfil de usuario es la siguiente:
 - País.
 - Tipo de identificación.
 - Identificación.
 - Departamento.
 - Municipio.
 - Dirección.

- Nombre completo en campos separados.
- Fecha de nacimiento.

3.1.3.3.2 Secuencia de estímulos / respuestas

Secuencia 1

- El contador ingresa al catálogo de cuentas contables.
- El contador ingresa o modifica los datos de una cuenta contable.
- El sistema actualiza el catálogo de cuentas contables en una base de datos.

Secuencia 2

- El contador o el auxiliar de contabilidad ingresa a la página de administración de partidas contables.
- El contador o el auxiliar de contabilidad ingresa la información de una partida nueva o modifica una partida existente si esta NO se encuentra auditada o es de un período contable que no se está trabajando.
- El sistema almacena la información de la partida contable en una base de datos.

Secuencia 3

- El contador o el auxiliar de contabilidad ingresa a la página de administración de compras y ventas.
- El contador o el auxiliar de contabilidad ingresa o modifica la información de una compra o una venta.
- El sistema almacena la venta o compra en una base de datos y le genera una partida contable.

Secuencia 4

- El auditor ingresa a la página de administración de partidas contables.

- El auditor audita las partidas contables.

Secuencia 5

- El contador o el administrador ingresa a la página de reportería.
- El contador o el administrador selecciona un libro a generar.
- El sistema despliega el reporte generado en la pantalla.
- El contador o el administrador exporta el reporte a un archivo Excel o PDF.

Secuencia 6

- El administrador ingresa a la sección de configuración contable.
- El administrador realiza una de las siguientes operaciones:
 - Modifica la información general de la aplicación.
 - Agrega o modifica las categorías de compras.
 - Agrega o modifica las categorías de ventas.
 - Agrega o modifica los rubros de compras.
 - Agrega o modifica los rubros de ventas.
- El sistema almacena los cambios realizados en una base de datos.

Secuencia 7

- El administrador ingresa al perfil de usuario.
- El administrador modifica la información del perfil de usuario.
- El sistema almacena la información en una base de datos.

3.1.3.3.3 Requerimientos funcionales

REQ-C-1: Administración de cuentas contables.

REQ-C-1.1: Ingreso de cuentas contables.

REQ-C-1.2: Modificación de cuentas contables.

REQ-C-1,3: Eliminación de cuentas contables.

REQ-C-2: Administración de partidas contables.

REQ-C-2.1: Ingreso de partidas contables.

REQ-C-2.2: Modificación de partidas contables sin auditar.

REQ-C-2.3: Anulación de partidas contables sin auditar.

REQ-C-2.4: Auditoria de partidas contables.

REQ-C-3: Administración de compras y ventas.

REQ-C-3.1: Ingreso de compras y ventas.

REQ-C-3.2: Modificación de compras y ventas cuya partida contable no se encuentre auditada.

REQ-C-3.3: Anulación de compras y ventas cuya partida contable no se encuentre auditada.

REQ-C-3.4: Generación de partidas contables por cada compra o venta.

REQ-C-4: Generación de libros contables a través de reportes.

REQ-C-4.1: Exportación de reportes a documentos PDF.

REQ-C-4.2: Exportación de reportes a documentos en Excel.

REQ-C-5: Configuración contable de la aplicación.

REQ-C-5.1: Modificación de la configuración general de la aplicación.

REQ-C-5.2: Mantenimiento de las categorías de compras.

REQ-C-5.3: Mantenimiento de las categorías de ventas.

REQ-C-5.4: Mantenimiento de los rubros de compras.

REQ-C-5.5: Mantenimiento de los rubros de ventas.

REQ-C-6: Mantenimiento del perfil de usuario.

3.1.4 Requerimientos de interfaz externa

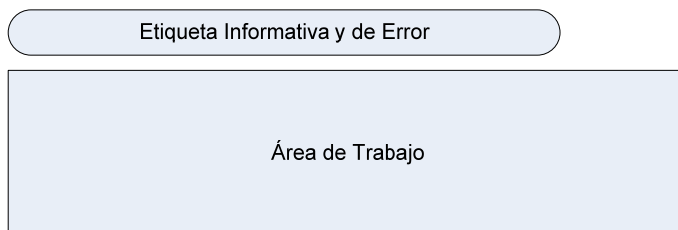
Las interfaces deberán ser diseñadas en forma de páginas Web que conformen el portal del prototipo.

3.1.4.1 Interfaces de usuario

A continuación se detallan las especificaciones generales de las interfaces:

- Todas las interfaces deberán mostrar la información general del usuario (nombre y empresa).
- Todas las interfaces deberán contar con el menú principal de la aplicación en la parte superior.
- Todas las interfaces deberán mostrar un link a la página de soporte.
- Todas las interfaces deberán mostrar la opción de salir de la aplicación.
- Los mensajes informativos y de error deberán ser mostrados en forma de etiquetas en la parte superior de cada área. Una etiqueta asignada a cada área.

Figura 5. Diagrama de estructura de un mensaje informativo o de error.

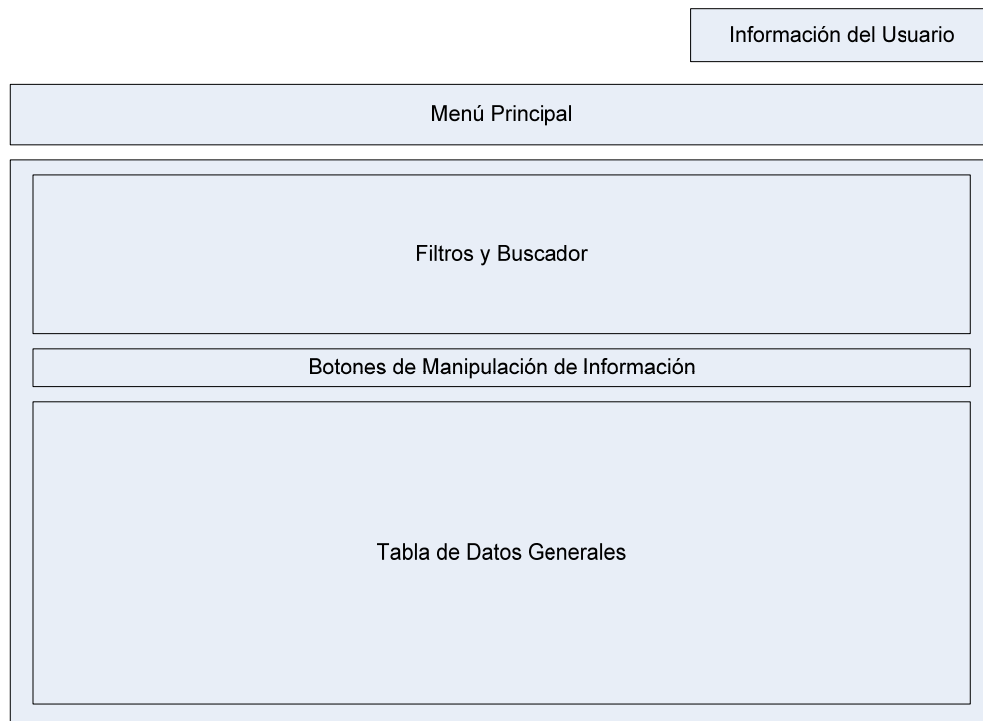


- Las interfaces deberán mostrar el período de tiempo (año y mes) en el que se está trabajando cuando sea pertinente.
- Todas las interfaces deberán contar con una opción de cancelar una operación, cuando se trate de una interfaz de manipulación de datos, y una opción de regresar a la pantalla anterior, cuando se trate de una interfaz de consulta de datos.
- Todas las interfaces deberán mostrar el logo de la aplicación en la parte superior izquierda.

3.1.4.1.1 Interfaz de consulta general

Esta interfaz será utilizada en cada sección que deba desplegar datos generales acerca de un área en específico como información de pólizas, compras, ventas, cuentas contables, etc. La estructura de dicha interfaz se muestra en el siguiente diagrama:

Figura 6. Diagrama de estructura de interfaz de consulta general.

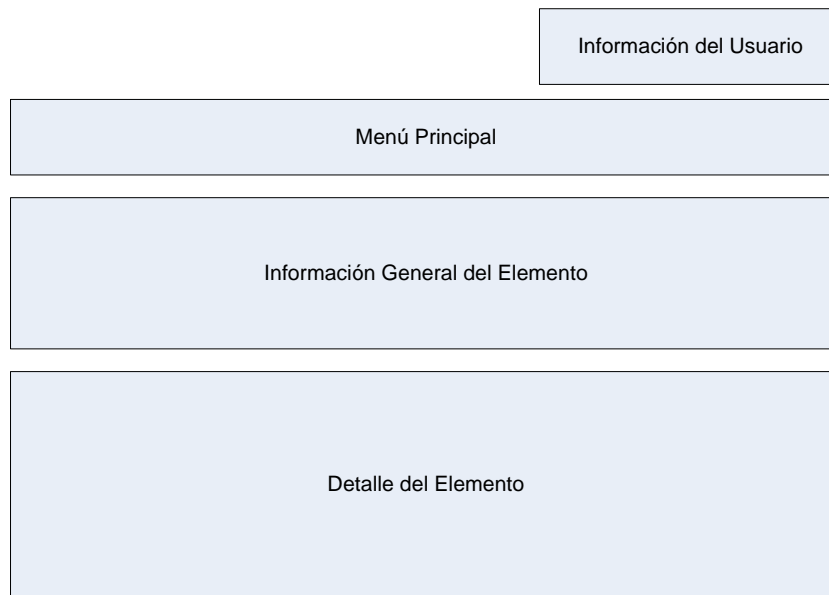


3.1.4.1.2 Interfaz de consulta específica

Esta interfaz será utilizada para desplegar información detallada acerca de un elemento relacionado con un área específica como una póliza contable, una

compra, una venta o una cuenta contable. El diagrama de la estructura de esta interfaz se presenta a continuación:

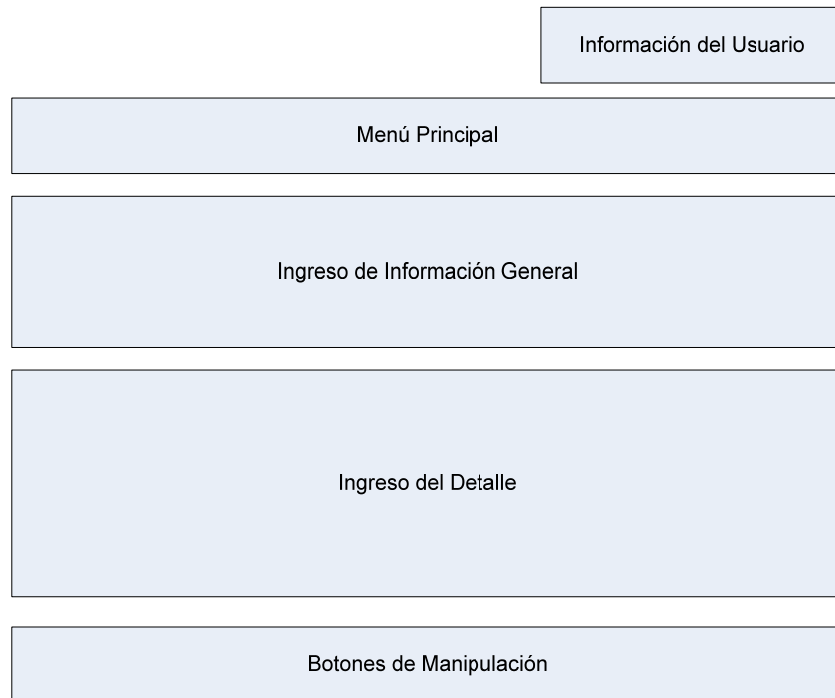
Figura 7. Diagrama de estructura de interfaz de consulta específica.



3.1.4.1.3 Interfaz de ingreso y modificación

Esta interfaz será utilizada para realizar el ingreso o la modificación de datos de un elemento de cualquier área en específico. El diagrama con la estructura de dicha interfaz se muestra a continuación:

Figura 8. Diagrama de estructura de interfaz de ingreso y modificación.



3.1.5 Otros requerimientos no funcionales

3.1.5.1 Requerimientos de rendimiento

- Cualquier operación en la aplicación que requiera interacción con la base de datos deberá tomar menos de 8 segundos.
- El sistema debe ser capaz de soportar accesos múltiples sin sacrificar la velocidad de respuesta de las operaciones.
- Deberá implementarse una arquitectura de hardware que permita que el sistema se mantenga en línea, al menos un 99% del tiempo.

3.1.5.2 Requerimientos de seguridad

- Cualquier operación dentro del sistema podrá ser realizada únicamente por usuarios registrados a través de un nombre y una contraseña.
- El nombre de un usuario deberá tener como mínimo 8 caracteres.
- La contraseña de un usuario deberá tener como mínimo 8 caracteres e incluir números y letras.
- Si un usuario registrado se equivoca más de tres veces al ingresar su contraseña, su cuenta quedará automáticamente bloqueada.
- Al momento de ingresar al sistema, se creará una sesión para el usuario en uso que no permitirá que el mismo ingrese desde otro explorador al mismo tiempo.
- La información común utilizada en distintas páginas de la aplicación deberá ser transferida a través de variables de sesión.
- La sesión de un usuario deberá expirar después de un tiempo sin actividad. Este tiempo podrá ser configurable.

3.1.5.3 Requerimientos de calidad del software

- Cada módulo del sistema deberá ser independiente para poder ser reutilizado si es requerido.
- El diseño visual de la aplicación Web deberá encontrarse separado en hojas de estilo CSS.
- Cualquier función Javascript utilizada en la aplicación deberá encontrarse separada de las páginas Web a través de librerías.
- El diseño visual del sitio debe ser atractivo y sencillo.
- Ninguna operación dentro del sitio debe tomar más de tres pasos para ser llevada a cabo.

3.2 Especificación de diseño del software

3.2.1 Introducción

3.2.1.1 Propósito

El propósito de este documento es proporcionar el diseño del sistema a desarrollar que permita conocer detalladamente el funcionamiento y la estructura del mismo así como el de cada uno de sus módulos.

3.2.1.2 Audiencia del documento

Este documento (o partes de él dependiendo del grado de confidencialidad del mismo) está dirigido a project managers, arquitectos, desarrolladores y testers involucrados en este proyecto.

3.2.1.3 Alcance

Este documento abarca el diseño de los módulos de administración de usuarios, registro de usuarios nuevos y contabilidad de la suite financiera.

3.2.2 Descripción general del sistema

El sistema a desarrollar consiste en un prototipo de una suite financiera en línea que estaría compuesta de módulos que serán utilizados de forma independiente por un usuario registrado dentro de la misma.

3.2.2.1 Módulos iniciales

Para el prototipo que será desarrollado como ejemplo de una aplicación que funciona bajo el modelo SaaS se implementarán los siguientes módulos:

- Módulo de registro de usuarios nuevos: Este es el módulo principal de la suite financiera. A través de él, un usuario registrado tendrá la capacidad de agregar empresas a su cuenta, crear usuarios y asignarlos junto con módulos a las mismas, y a través de roles, otorgarle los privilegios que un usuario tendrá sobre un módulo asignado a una empresa en específico.
- Módulo de suscripciones: Este módulo permitirá el registro de usuarios nuevos a la suite financiera en línea. A través de este módulo se le creará una cuenta principal a cada usuario registrado que le permitirá agregar empresas, módulos y usuarios secundarios de los módulos en una empresa en específico.
- Módulo de contabilidad: Este módulo le permitirá a un usuario registrado manejar las operaciones contables de una empresa. Las operaciones que permitirá realizar serán: Administración de pólizas contables, administración de cuentas contables, administración de compras, administración de ventas, generación de libros contables a través de reportes, administración de configuración contable y administración de perfil de usuario.

3.2.3 Consideraciones de diseño

3.2.3.1 Dependencias y asunciones

- El sistema deberá ser desarrollado utilizando Visual Studio 2008 con .Net Framework 2.0.
- La aplicación deberá tener una interfaz Web desarrollada con ASP.NET.
- Todas las capas de la aplicación deberán ser desarrolladas en lenguaje C#.

- El servidor Web a utilizar será un IIS versión 6.0.
- La aplicación deberá estar separada en capas que permitan que la base de datos, las operaciones en la base de datos, la lógica de negocios y la interfaz de usuario, trabajen de forma independiente.
- Deberá utilizarse como servidor de base de datos SQL Server 2005.
- Los módulos de la aplicación deberán ser desarrollados de tal forma que sean completamente independientes y reutilizables.

3.2.3.2 Límites generales

- El prototipo funcional a desarrollar únicamente contará con los módulos de administración de usuarios, suscripciones y contabilidad.
- Cada operación dentro de la aplicación deberá ser realizada en tres pasos como máximo.
- La aplicación debe estar separada como mínimo en 4 capas, capa de base de datos, capa de acceso a la base de datos, capa de lógica de negocios e interfaz Web de usuario.
- La aplicación debe ser desarrollada para plataforma Web 2.0.
- Los reportes deben ser desarrollados específicamente utilizando Reporting Services como herramienta de edición.
- Los reportes deben tener la capacidad de ser exportados a archivos Excel o PDF.
- Las tablas dentro de la base de datos deben estar separadas con esquemas que describan específicamente el módulo al que pertenecen.
- Los módulos de negocios (administración de usuarios y contabilidad) podrán ser utilizados únicamente por usuarios registrados con un nombre de usuario y una contraseña.
- Cuando un usuario ingrese al sistema, se deberá crear una sesión en la base de datos que asegure que el mismo usuario no pueda ingresar a la aplicación desde otro explorador al mismo tiempo.

- Si un usuario realiza tres intentos de ingreso fallidos su cuenta deberá ser bloqueada y el usuario deberá solicitar la reactivación de la misma.
- Si un usuario permanece un período de tiempo sin actividad, la sesión debe ser terminada automáticamente por el servidor. Este tiempo debe ser configurable.
- La información común utilizada en múltiples páginas de la aplicación deberá ser transferida a través de variables de sesión específicas para el usuario registrado.
- Las operaciones dentro de la aplicación que interactúen con la base de datos deberán procesarse en un máximo de 8 segundos.
- El pagineo de las tablas generales que muestren pólizas contables, cuentas contables, compras y ventas, deberá ser controlado desde las consultas a la base de datos y no desde la aplicación. Es decir, si una tabla muestra como máximo 10 registros, únicamente esta cantidad deberá ser solicitada directamente a la base de datos en lugar de solicitar una cantidad mayor y limitar el número de registros desde la aplicación.
- La aplicación deberá ser desarrollada en lenguaje C# con .Net Framework 2.0.

3.2.3.3 Objetivos y lineamientos

- El módulo de suscripciones debe permitir ingresar la información del usuario principal de la cuenta, que actuará como usuario administrador, y de la empresa inicial bajo la cual se utilizarán los servicios contables.
- El módulo de administración de usuarios y cuentas deberá permitir realizar las siguientes operaciones:
 - Ingreso de empresas nuevas a la cuenta principal.
 - Creación de solicitudes para agregar usuarios a la cuenta principal.

- Adición de módulos a una empresa. Para el prototipo, el único módulo disponible será el módulo de contabilidad.
- Adición de usuarios secundarios a un módulo específico de una empresa para hacer uso de él, y la selección de un rol que le permitirá tener ciertos privilegios en dicho módulo.
- El módulo de contabilidad deberá permitir realizar las siguientes operaciones:
 - Mantenimiento del catálogo de cuentas contables.
 - Mantenimiento de pólizas contables.
 - Mantenimiento de compras.
 - Mantenimiento de ventas.
 - Generación de libros contables para ser observados en el sitio o exportados a archivos PDF o Excel.
 - Modificación de la configuración de la aplicación.
 - Actualización del perfil del usuario.
- Las páginas generales de cada mantenimiento deberán mantener los siguientes criterios:
 - Utilizar buscadores con filtros a los campos más relevantes.
 - Se debe mostrar la información en forma de tablas.
 - Las tablas deben tener opciones de pagineo con tamaño de página configurable.
 - Se debe mostrar el total de registros que una tabla posee.
 - Cada registro debe contar con íconos de modificación y verificación.

3.2.3.4 Metodología de desarrollo

Para el desarrollo de esta aplicación se trabajará utilizando la metodología de desarrollo de software en cascada.

3.2.4 Estrategias de arquitectura

- La aplicación deberá ser desarrollada en plataforma Web utilizando ASP.NET 2.0 con lenguaje C#.
- El motor de base de datos a utilizar será SQL Server 2005 edición estándar.
- El diseño de la base de datos deberá separar en esquemas las tablas a utilizar en un módulo específico. Deberá existir un esquema común para las tablas que serán utilizadas por múltiples módulos del prototipo.
- La capa de conexión a la base de datos deberá ser creada utilizando el generador de código de cuarta generación LLBL Gen Pro versión 2.0, en modalidad “adapter”.
- Se deberán crear clases intermedia que contendrá la lógica de negocios de la aplicación. Cada módulo de la aplicación (contabilidad, suscripciones y administración de usuarios) tendrá su propia clase intermedia que contendrá la lógica de negocios específica del módulo.
- Existirá una clase intermedia común que contendrá toda la lógica de negocios a utilizar en todos o varios módulos de la aplicación.
- Se deberá utilizar el diseñador de reportes de Visual Studio .NET y reporting services para la generación de los libros contables.
- Las páginas Web en las que se ingrese información deberán contar con filtros de validación de información para cada control utilizado del lado de la página Web. No se deberá realizar una solicitud al servidor para validar la información.
- Los errores dentro de la aplicación deberán ser capturados en las clases intermedias que manejen la lógica de negocios y traducidas a mensajes entendibles por el usuario.
- El estilo de las páginas deberá ser manejado a través de archivos CSS.

- El sistema deberá permitir el acceso concurrente de múltiples usuarios y manejar sesiones separadas para cada uno de ellos.

3.2.5 Arquitectura del sistema

3.2.5.1 Diseño de la base de datos

La base de datos contará con los siguientes esquemas:

- Esquema administrativo: Este esquema contendrá las tablas que almacenen información específica del módulo de suscripciones. Las tablas a incluir en este esquema son las siguientes:
 - Estado_suscriptor: Almacenará los diferentes estados que puede tener una suscripción dentro del sistema.
 - Estado_suscriptor_empresa: Almacenará el estado de una empresa dentro de una suscripción.
 - Estado_suscriptor_empresa_modulo: Almacenará el estado de un módulo asignado a una empresa dentro de una suscripción.
 - Estado_suscriptor_empresa_modulo_usuario: Almacenará el estado de un usuario relacionado a un módulo asignado a una empresa dentro de una suscripción.
 - Modulo: Almacenará los distintos módulos que contendrá el prototipo.
 - Modulo_permiso: Almacenará las diferentes acciones que podrán ser realizadas por un usuario dentro de un módulo.
 - Modulo_rol: Almacenará los distintos roles o tipos de usuario que existirán para un módulo específico.

- Modulo_rol_permiso: Almacenará las distintas acciones que podrán ser realizadas por un rol o tipo de usuario en específico dentro de un módulo.
 - Modulo_tarifa: Almacenará la información general de las distintas tarifas disponibles para el prototipo.
 - Modulo_tarifa_contabilidad: Almacenará las características de las distintas tarifas disponibles para el módulo de contabilidad.
 - Modulo_tarifa_rol: Almacenará la cantidad de usuarios disponibles en un módulo para cada rol o tipo de usuario.
 - Suscriptor: Almacenará la información general de una suscripción en el sistema.
 - Suscriptor_empresa: Almacenará las diferentes empresas asociadas a una suscripción.
 - Suscriptor_empresa_modulo: Almacenará los módulos asignados a una empresa asociada a una suscripción.
 - Suscriptor_empresa_modulo_usuario: Almacenará los usuarios que podrán hacer uso de un módulo asignado a una empresa asociada a una suscripción.
 - Suscriptor_empresa_modulo_usuario_rol: Almacenará el rol de un usuario que podrá hacer uso de un módulo asignado a una empresa asociada a una suscripción.
 - Suscriptor_usuario: Almacenará la asociación de los usuarios registrados a una suscripción.
- Esquema contable: Este esquema estará formado por todas las tablas que almacenen información específica del módulo contable. Las tablas que contendrá son listadas a continuación:
 - Categoría_compra: Almacenará las categorías existentes para una compra.

- Categoría_venta: Almacenará las categorías existentes para una venta.
- Compra: Almacenará la información de cada compra ingresada.
- Compra_impuesto_aplicado: Almacena los impuestos que deben ser aplicados a una compra.
- Contador: Almacenará la información específica de los contadores.
- Cuenta_contable: Almacenará las cuentas del catálogo contable.
- Empresa_configuracion_contable: Almacenará toda la información acerca de la configuración contable de una empresa.
- Empresa_configuracion_contable_impuesto: Almacenará las cuentas contables que serán utilizadas para representar los impuestos en las partidas contables.
- Estado_cuenta_contable: Almacenará los diferentes estados que pueden tener las cuentas contables.
- Estado_poliza_contable: Almacenará los diferentes estados que pueden tener las pólizas contables.
- Impuesto: Almacenará las características de los impuestos.
- Impuesto_tipo_documento: Almacenará la relación entre los impuestos y los documentos en los que aplican.
- Poliza_contable: Almacenará la información de las pólizas contables.
- Poliza_contable_detalle: Almacenará el detalle de cada póliza contable.
- Rubro_contable_compra: Almacenará las cuentas contables que serán utilizadas para representar una compra en las partidas contables.

- Rubro_contable_venta: Almacenará las cuentas contables que serán utilizadas para representar una venta en las partidas contables.
 - Tipo_cuenta_contable: Almacenará los tipos de cuenta contable.
 - Venta: Almacenará la información de cada venta ingresada.
 - Venta_impuesto_aplicado: Almacenará los impuestos que deben ser aplicados a una venta.
- Esquema de usuarios: Este esquema estará formado por todas las tablas que almacenen información específica para la administración de usuarios. Las tablas que contendrá son listadas a continuación:
 - Estado_usuario: Almacenará los diferentes estados que puede tener un usuario.
 - Usuario: Almacenará la información específica de los usuarios.
- Esquema común: Este esquema estará formado por todas las tablas que almacenen información común utilizada por múltiples módulos. Las tablas que contendrá son las listadas a continuación:
 - Contribuyente: Almacenará la información específica de los contribuyentes.
 - Departamento: Almacenará los departamentos del país de Guatemala.
 - Empresa: Almacenará la información específica de las empresas suscritas.
 - Moneda: Almacenará el catálogo de monedas.
 - Municipio: Almacenará los municipios de cada departamento del país de Guatemala.

- País: Almacenará los países que podría utilizar la aplicación. Para el prototipo únicamente existirá Guatemala.
- Persona: Almacenará la información específica a cada persona suscrita.
- Tipo_constitucion_legal: Almacenará el tipo de constitución legal de las empresas suscritas.
- Tipo_documento: Almacenará los diferentes tipos de documento utilizados en las compras y ventas.
- Tipo_documento_pais: Almacenará los tipos de documentos utilizados en cada país para compras y ventas.
- Tipo_identificacion: Almacenará los distintos tipos de documento de identificación a utilizar en el prototipo.
- Tipo_identificacion_empresa: Almacenará los tipos de documento de identificación a utilizar para las empresas.
- Tipo_identificacion_persona: Almacenará los tipos de documento de identificación a utilizar para las personas.

3.2.5.2 Módulos y clases:

La aplicación estará dividida en múltiples módulos que se encargarán de funcionalidades específicas de la aplicación y permitirán su independencia y reutilización.

3.2.5.2.1 Módulo de administración:

Manejará todas las operaciones realizadas en la sección de suscripciones y asignación de módulos a empresas. También manejará los índices que serán utilizados para revisar los permisos que tiene un usuario dentro de la aplicación. Contará con dos clases principales:

- **ModuloManager:** Contendrá los procedimientos necesarios para obtener los distintos módulos disponibles a ser utilizados por empresas suscritas a la aplicación. Para el prototipo únicamente se encontrará disponible el módulo de contabilidad.

Las funciones incluidas en esta clase serán las siguientes:

- **ObtenerTodos:** Esta función devolverá todos los módulos disponibles para ser utilizados en el sistema. Para el prototipo únicamente devolverá el módulo de contabilidad que se encuentra almacenado en la base de datos.
 - **Obtener:** Esta función devolverá un módulo en específico en base a su ID en la base de datos.
- **SuscripcionManager:** Contendrá todas las funciones y procedimientos involucrados en la administración de suscripciones, como la inscripción de usuarios a una cuenta, la asignación de roles y módulos, suscripción de cuentas nuevas, etc.

Las funciones incluidas en esta clase serán las siguientes:

- **Suscribir:** Esta función permitirá ingresar a un nuevo suscriptor.
- **Activar:** Este procedimiento se encargará de cambiar el estado de una suscripción a activa.
- **Desuscribir:** Ese procedimiento se encargará de cambiar el estado de una suscripción a inactiva.
- **BloquearSuscriptor:** Este procedimiento se encargará de cambiar el estado de una suscripción a bloqueada.

- Accesar: Esta función indicará si un id de suscriptor y su contraseña son correctos para permitirle el ingreso al módulo de administración de usuarios y cuentas.
- SuscribirseEmpresa: Este procedimiento se encargará de suscribir a una empresa bajo una cuenta.
- DesuscribirseEmpresa: Este procedimiento se encargará de cambiar el estado de una empresa bajo una cuenta a inactivo.
- SuscribirseModulo: Este procedimiento se encargará de asignar un módulo a una empresa bajo una cuenta.
- DesuscribirseModulo: Este procedimiento se encargará de cambiar el estado de un módulo asignado a una empresa a inactivo.
- CambiarTarifaModulo: Este procedimiento se encargará de cambiar la tarifa de un módulo asignado a una empresa.
- SuscribirseUsuario: Este procedimiento se encargará de ingresar un nuevo usuario bajo una suscripción.
- SuscribirseUsuarioModulo: Este procedimiento se encargará de asignar un usuario a un módulo específico de una empresa.
- DesuscribirseUsuario: Este procedimiento se encargará de cambiar el estado de un usuario asignado a una suscripción a inactivo.
- DesuscribirseUsuarioModulo: Este procedimiento se encargará de cambiar el estado de un usuario asignado a un módulo a inactivo.
- SuscribirseRol: Este procedimiento se encargará de asignar un rol específico a un usuario bajo un módulo.
- ObtenerEmpresasPorUsuario: Esta función devolverá las empresas a las que se encuentra asignado un usuario.
- ObtenerModulosPorUsuario: Esta función devolverá los módulos que puede utilizar un usuario bajo una empresa.
- ObtenerEmpresasPorSuscriptor: Esta función devolverá las empresas asociadas a una suscripción.

- ObtenerModulosPorEmpresa: Esta función devolverá los módulos que están asignados a una empresa bajo una suscripción.
- ObtenerUsuariosPorSuscripcion: Esta función devolverá los usuarios inscritos a una suscripción.
- ObtenerUsuariosPorModulo: Esta función devolverá los usuarios asignados a un módulo bajo una empresa en específico.
- ObtenerRolesPorUsuario: Esta función devolverá los roles asignados a un usuario bajo un módulo.
- ObtenerPermiso: Esta función determinará si un usuario tiene acceso a una operación específica dentro de un módulo.
- ObtenerLimiteDetallePoliza: Esta función devolverá el número máximo de filas permitido en el detalle de una póliza contable de acuerdo a una tarifa específica.
- EsPermitidoIngresarTransaccion: Esta función determinará si una transacción puede ser ingresada al sistema de acuerdo a una tarifa específica.
- EsPermitidoAgregarUsuarioRol: Esta función determinará si un usuario puede ser asignado a un rol de acuerdo a una tarifa específica.

3.2.5.2.2 Módulo común:

Manejará todas las operaciones realizadas en múltiples secciones de la aplicación. Contará con las siguientes clases principales:

- ContribuyenteManager: Contendrá las funciones y procedimientos necesarios para obtener, ingresar y actualizar la información de los contribuyentes.

Las funciones incluidas en esta clase serán las siguientes:

- Obtener: Esta función devolverá la información de un contribuyente en base a su id.
 - ObtenerTodosPorIdentificacion: Esta función devolverá todos los contribuyentes que coincidan con una identificación específica.
 - Guardar: Esta función ingresará a un contribuyente nuevo o modificará la información de un contribuyente existente.
 - Eliminar: Esta función eliminará la información de un contribuyente de la base de datos.
- DepartamentoManager: Contendrá las funciones necesarias para obtener los departamentos del país de Guatemala.
Las funciones incluidas en esta clase serán las siguientes:
 - ObtenerTodos: Esta función devolverá todos los departamentos de un país específico.
 - ObtenerTodosConMunicipios: Esta función devolverá todos los departamentos con sus municipios de un país específico
- EmpresaManager: Contendrá las funciones y procedimientos necesarios para obtener, ingresar y actualizar la información de las empresas suscritas a la aplicación.
Las funciones incluidas en esta clase serán las siguientes:
 - Obtener: Esta función devolverá la información de una empresa.
 - Guardar: Esta función ingresará a una empresa nueva o modificará la información de una empresa existente.
- FinanzasHelper: Contendrá procedimientos y funciones genéricos que serán utilizados para obtener información de múltiples tablas de la base

de datos o devolver información necesaria para operaciones especiales dentro de la aplicación, como el número total de registros en una tabla, generación de códigos únicos de activación de suscripciones, etc.

Las funciones incluidas en esta clase serán las siguientes:

- GenerarCodigoActivacionUnico: Esta función generará un código genérico de activación para las suscripciones y los usuarios.
- MonedaManager: Contendrá las funciones necesarias para obtener las monedas utilizadas en la aplicación.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodas: Esta función devolverá todas las monedas disponibles en el sistema.
- MunicipioManager: Contendrá las funciones necesarias para obtener los municipios de cada departamento del país de Guatemala.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los municipios de un departamento.
- PaisManager: Contendrá las funciones necesarias para obtener los países para los cuales la aplicación se encontrará disponible. Para el prototipo únicamente se encontrará Guatemala.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los países disponibles para el sistema.

- ObtenerTodosConDepartamentos: Esta función devolverá todos los países disponibles para el sistema con sus departamentos o equivalentes.
 - ObtenerTodosConDepartamentosYMunicipios: Esta función devolverá todos los países disponibles para el sistema con sus departamentos y municipios o equivalentes.
- PersonaManager: Contendrá las funciones y procedimientos necesarios para obtener, ingresar y actualizar la información de las personas que utilicen la aplicación.

Las funciones incluidas en esta clase serán las siguientes:

- Obtener: Esta función devolverá la información de una persona.
 - Guardar: Esta función ingresará una persona nueva o modificará la información de una persona existente.
- TipoConstitucionLegalManager: Contendrá las funciones necesarias para obtener los tipos de constitución legal que pueden tener las empresas suscritas a la aplicación.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los tipos de constitución legal disponibles para el sistema.
 - Obtener: Esta función devolverá la información de un tipo de constitución legal.
- TipoDocumentoManager: Contendrá las funciones necesarias para obtener los tipos de documento.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los tipos de documento para un país específico.
 - Obtener: Esta función devolverá la información de un tipo de documento.
- TipoidentificacionManager: Contendrá las funciones necesaria para obtener los tipos de identificación.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los tipos de identificación disponibles en el sistema para un país específico.
- ObtenerTiposDeldentificacionDeEmpresas: Esta función devolverá todos los tipos de identificación válidos para las empresas de un país específico.
- ObtenerTiposDeldentificacionDePersonas: Esta función devolverá los tipos de identificación válidos para las personas de un país específico.
- Obtener: Esta función devolverá la información de un tipo de identificación.

3.2.5.2.3 Módulo de contabilidad:

Manejará todas las operaciones realizadas en la sección de contabilidad de la aplicación. Contará con las siguientes clases principales:

- BalanceGeneralManager: Contendrá la función que generará el libro de balance general.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerBalanceGeneral: Esta función devolverá la información necesaria para generar el libro de balance general.
- BalanceSalDOSManager: Contendrá la función que generará el libro de balance de saldos.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerBalanceSalDOS: Esta función devolverá la información necesaria para generar el libro de balance de saldos.
- CatalogoContableManager: Contendrá las funciones y procedimientos necesarios para administrar el catálogo contable.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerCuentas: Esta función devolverá las cuentas contables de una empresa específica.
- ObtenerArbolCuentas: Esta función devolverá todas las cuentas contables hijas de una cuenta padre para una empresa específica.
- ObtenerArbolCuentasActiva: Esta función devolverá todas las cuentas contables hijas con estado activo de una cuenta padre para una empresa específica.
- ObtenerCuenta: Esta función devolverá la información de una cuenta contable.
- Guardar: Esta función ingresará una cuenta contable nueva o modificará la información de una existente para una empresa específica.
- Eliminar: Esta función eliminará una cuenta contable de la base de datos para una empresa específica.

- **CategoriaCompraManager:** Contendrá las funciones y procedimientos necesarios para administrar el catálogo de categorías de compras.

Las funciones incluidas en esta clase serán las siguientes:

- **ObtenerTodas:** Esta función devolverá todas las categorías creadas bajo una empresa específica.
 - **Obtener:** Esta función devolverá la información de una categoría de compra.
 - **Guardar:** Esta función ingresará una categoría de compra nueva o modificará una ya existente.
 - **Eliminar:** Esta función eliminará una categoría de compra de la base de datos.
-
- **CategoriaVentaManager:** Contendrá las funciones y procedimientos necesarios para administrar el catálogo de categorías de ventas.
- Las funciones incluidas en esta clase serán las siguientes:
- **ObtenerTodas:** Esta función devolverá todas las categorías creadas bajo una empresa específica.
 - **Obtener:** Esta función devolverá la información de una categoría de venta.
 - **Guardar:** Esta función ingresará una categoría de venta nueva o modificará una ya existente.
 - **Eliminar:** Esta función eliminará una categoría de venta de la base de datos.
-
- **ConfiguracionContableManager:** Contendrá las funciones y procedimientos necesarios para obtener, ingresar y actualizar la configuración contable.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerConfiguracionContable: Esta función devolverá la configuración contable de una empresa específica.
 - Guardar: Esta función almacenará los cambios en la configuración contable de una empresa específica.
 - CambiarPeriodoContable: Esta función cambiará el año contable sobre el que se trabajará para una empresa específica.
 - CerrarPeriodoContableMes: Esta función cambiará el mes contable sobre el que se trabajará para una empresa específica.
- ContadorManager: Contendrá las funciones y procedimientos necesarios para obtener, ingresar y actualizar la información de los contadores.

Las funciones incluidas en esta clase serán las siguientes:

- Guardar: Esta función modificará la información del contador de una empresa específica.
 - Obtener: Esta función devolverá la información del contador de una empresa específica.
- EstadoCuentaContableManager: Contendrá las funciones necesarias para obtener los estados de una cuenta contable.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los estados que puede tener una cuenta contable dentro del sistema.
- EstadoPolizaContableManager: Contendrá las funciones necesarias para obtener los estados de una póliza contable.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los estados que puede tener una póliza contable dentro del sistema.
- EstadoResultadosManager: Contendrá la función que generará el libro de estado de resultados.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerEstadoResultados: Esta función devolverá la información necesaria para generar el libro de estado de resultados.
- ImpuestoManager: Contendrá las funciones necesarias para obtener los impuestos a utilizar en la aplicación.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los impuestos aplicados en un país específico.
- ObtenerVigentes: Esta función devolverá todos los impuestos vigentes en un país específico.
- ObtenerIVAAplicado: Esta función devolverá el valor por defecto del IVA. Para el prototipo únicamente devolverá el IVA de Guatemala.
- LibroComprasManager: Contendrá la función que generará el libro de compras.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerLibroCompras: Esta función devolverá la información necesaria para generar el libro de compras.
 - ObtenerCompras: Esta función devolverá todas las compras de una empresa para un período contable específico.
 - ObtenerPeriodosAnioDisponibles: Esta función devolverá todos los años en los cuales se han ingresado pólizas contables para una empresa específica.
 - Guardar: Esta función ingresará una compra para una empresa específica.
 - Auditar: Esta función cambiará el estado de una póliza contable de una compra a auditada.
 - Anular: Esta función cambiará el estado de una póliza contable de una compra a anulada.
 - ObtenerIVAAplicado: Esta función devolverá el IVA aplicado a una compra. Para el prototipo únicamente devolverá el IVA aplicado a una compra en Guatemala.
- LibroDiarioManager: Contendrá la función que generará el libro diario.
Las funciones incluidas en esta clase serán las siguientes:
 - ObtenerLibroDiario: Esta función devolverá la información necesaria para generar el libro diario.
- LibroMayorManager: Contendrá la función que generará el libro mayor.
Las funciones incluidas en esta clase serán las siguientes:
 - ObtenerLibroMayor: Esta función devolverá la información necesaria para generar el libro mayor.

- LibroVentasManager: Contendrá la función que generará el libro de ventas.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerLibroVentas: Esta función devolverá la información necesaria para generar el libro de ventas.
 - ObtenerVentas: Esta función devolverá todas las ventas de una empresa para un período contable específico.
 - ObtenerPeriodosAnioDisponibles: Esta función devolverá todos los años en los cuales se han ingresado pólizas contables para una empresa específica.
 - Guardar: Esta función ingresará una venta para una empresa específica.
 - Auditar: Esta función cambiará el estado de una póliza contable de una venta a auditada.
 - Anular: Esta función cambiará el estado de una póliza contable de una venta a anulada.
 - ObtenerIVAAplicado: Esta función devolverá el IVA aplicado a una venta. Para el prototipo únicamente devolverá el IVA aplicado a una venta en Guatemala.
- PolizaContableManager: Contendrá las funciones y procedimientos necesarios para administrar las pólizas contables.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerPolizas: Esta función devolverá las pólizas contables de un período indicado para una empresa específica.
- ObtenerPoliza: Esta función devolverá la información de una póliza contable.

- ObtenerPeriodosAnioDisponibles: Esta función devolverá todos los años en los que se haya ingresado pólizas contables para una empresa específica.
 - Guardar: Esta función ingresará una póliza contable nueva o modificará la información de una existente si ésta no se encuentra anulada o auditada.
 - Auditar: Esta función modificará el estado de una póliza contable a auditado.
 - Anular: Esta función modificará el estado de una póliza contable a anulado.
- RubroContableCompraManager: Contendrá las funciones y procedimientos necesarios para administrar los rubros contables para las compras.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerCuentas: Esta función devolverá las cuentas contables activas que hayan sido escogidas como rubros contables para el libro de compras.
 - ObtenerRubros: Esta función devolverá todos los rubros contables de una empresa específica.
 - Agregar: Esta función agregará una cuenta contable a los rubros disponibles para compras.
 - Eliminar: Esta función eliminará un rubro contable de la base de datos.
- RubroContableVentaManager: Contendrá las funciones y procedimientos necesarios para administrar los rubros contables para las ventas.
- Las funciones incluidas en esta clase serán las siguientes:

- ObtenerCuentas: Esta función devolverá las cuentas contables activas que hayan sido escogidas como rubros contables para el libro de ventas.
 - ObtenerRubros: Esta función devolverá todos los rubros contables de una empresa específica.
 - Agregar: Esta función agregará una cuenta contable a los rubros disponibles para ventas.
 - Eliminar: Esta función eliminará un rubro contable de la base de datos.
- TipoCuentaContableManager: Contendrá las funciones necesarias para obtener los tipos de cuenta contable.

Las funciones incluidas en esta clase serán las siguientes:

- ObtenerTodos: Esta función devolverá todos los tipos de cuenta contable dentro del sistema.

3.2.5.2.4 Módulo de usuarios:

Manejará todas las operaciones realizadas en la sección de administración de usuarios. Contará con una clase principal:

- UsuarioManager: Contendrá las funciones y procedimientos necesarios para administrar a los usuarios que harán uso de la aplicación.

Las funciones incluidas en esta clase serán las siguientes:

- Accesar: Esta función determinará si un usuario tiene acceso a un módulo en específico.

- Salir: Esta función liberará a un usuario que se encuentre utilizando un módulo.
- Guardar: Esta función ingresará a un usuario nuevo o modificará la información de un usuario existente.
- Obtener: Esta función devolverá la información de un usuario en base a su id.
- ActivarCuenta: Este procedimiento se encargará de colocar el estado de un usuario a activo.
- BloquearCuenta: Este procedimiento se encargará de colocar el estado de un usuario a bloqueado.

3.2.5.3 Interfaces de usuario

Por ser una aplicación desarrollada para ser utilizada a través de un explorador de Internet, todas las interfaces de la aplicación deberán ser desarrolladas como páginas Web.

Inicialmente la aplicación tendrá una serie de páginas que compondrán el sitio Web general al que podrá ingresar cualquier persona que desee obtener información del sistema o suscribirse a algún servicio. Este sitio podrá ser accedido a través de una dirección de Internet con dominio .com (por ejemplo: www.contabilidad.com). Las páginas de uso público a desarrollar para ese prototipo son las siguientes:

- Página de inicio: Esta página contendrá los links a las distintas secciones del sitio. También contendrá frases resumidas que explicarán el propósito de la aplicación y llamarán la atención del cliente para que éste se anime a probarla.

La página deberá incluir el logo principal de la aplicación y dos logos llamativos que envíen al cliente a una página que explique como será garantizada la seguridad de la información de las personas que hagan uso del sistema, y una página que muestre el convenio de confidencialidad del sitio.

Los links del sitio deberán ser mostrados a través de un menú general que será incluido en todas las páginas públicas del sitio. Los links a incluir en el menú general serán los siguientes:

- Home: Envió al cliente a la página de inicio.
 - ¿En que consiste el sistema?: Envió al cliente a la página de descripción del sistema.
 - Demo: Envió al cliente al módulo de contabilidad para hacer uso de él a través de usuarios de prueba. Existirá un usuario de prueba para cada rol configurado en el módulo que le permitirá al cliente utilizar el sistema desde las diferentes perspectivas ofrecidas.
 - Tarifas: Envió al cliente a la página que detalla las tarifas de uso del sistema.
 - Foro: Envió al cliente a la página principal del foro del sistema. El foro no será incluido en el prototipo.
 - Preguntas frecuentes: Envió al cliente a la página de preguntas frecuentes.
 - Contáctenos: Envió al cliente a la página de contáctenos que contendrá información a cerca de cómo comunicarse con los dueños del sistema.
- Página ¿En qué consiste el sistema?: Esta página describirá en detalle el sistema y sus módulos. Responderá como mínimo a las siguientes preguntas:
 - ¿En qué consiste el sistema?

- ¿Qué se necesita para utilizar el sistema?
- ¿Qué beneficios se obtienen al suscribirse al sistema?
- ¿Qué ventajas se obtienen sobre cualquier otro sistema o herramienta?
- Página Demo: Para el prototipo, esta página contendrá botones que le permitirán a un cliente ingresar al módulo de contabilidad con usuarios que tendrán los privilegios de alguno de los roles que se encuentren configurados para el módulo. Los roles que deberán estar disponibles en esta página son los siguientes:
 - Administrador
 - Contador
 - Auxiliar de contabilidad
 - Auditor
 - Monitor
- Página de tarifas: Esta página deberá incluir una tabla con el detalle de las distintas tarifas ofrecidas para el sistema. En la parte superior de la tabla se deberá incluir una imagen o un botón por tarifa, que envíe al cliente a la página de suscripción al sistema. Cada botón deberá enviarle a la página de suscripción la tarifa que fue seleccionada por el cliente.
- Foro: Para el prototipo, el foro no será implementado ya que el sistema no será publicado en una dirección de Internet real para su venta.
- Página de preguntas frecuentes: Esta página contendrá un listado de preguntas frecuentes que podrían surgirle a los clientes.
- Página Contáctenos: Esta página presentará la información que le servirá a los usuarios para comunicarse con los dueños del sistema.

3.2.5.3.1 Páginas del módulo de suscripciones:

El módulo de suscripciones estará compuesto únicamente por dos páginas.

La primera página contendrá un formulario que solicitará la información del cliente que quiere hacer uso del sistema. La información a solicitar en el formulario es la siguiente:

- Datos de la suscripción: Incluirá el nombre del suscriptor, la contraseña para ingresar al módulo de administración de usuarios y su correo electrónico.
- Datos de la empresa: Incluirá los campos mínimos listados en la sección de especificación de requerimientos de software. La información solicitada será específica de la empresa principal sobre la cual la cuenta será registrada.
- Datos del representante legal de la empresa: Incluirá los campos mínimos listados en la sección de especificación de requerimientos de software. La información solicitada será específica del representante legal de la empresa principal sobre la cual la cuenta será registrada.
- Datos del usuario: Se solicitará el nombre y el correo electrónico del primer usuario que hará uso de los módulos del sistema. Para el prototipo, se tratará de un usuario administrador que podrá hacer uso del módulo de contabilidad con todos los privilegios existentes para el mismo. La persona que haga uso de este usuario puede ser distinta al cliente que se esté suscribiendo al sistema. El correo electrónico servirá para enviar un link de activación del usuario administrador.
- Datos del servicio: En esta sección, el cliente podrá seleccionar el módulo inicial que será utilizado por la empresa principal de la cuenta y la tarifa con la que hará uso de los servicios.

La página también deberá desplegar el convenio de confidencialidad que el usuario deberá aceptar para hacer uso de los servicios del sistema. Al momento

de que el cliente termine su suscripción al sistema le será enviado un correo con un link para activar su cuenta.

La segunda página se encargará de activar la cuenta del cliente suscrito y le mostrará un cuadro de ingreso de suscriptores al módulo de administración de usuarios y cuentas.

3.2.5.3.2 Páginas del módulo de administración de usuarios y cuentas

Las páginas incluidas en este módulo le permitirán a un suscriptor realizar operaciones de manejo de cuenta como creación de usuarios nuevos, creación de empresas nuevas, asignación de usuarios a empresas bajo diferentes roles, asignación de módulos a empresas, etc. Estas páginas podrán ser utilizadas únicamente por suscriptores con cuentas activas.

Las páginas a incluir en este módulo serán las siguientes:

- **Página de ingreso:** Esta página incluirá un formulario de ingresos que le solicitarán a un cliente su ID de suscriptor y su contraseña para ingresar a las funcionalidades del módulo.
- **Página principal:** Esta página mostrará en dos tablas separadas, las empresas y los usuarios inscritos bajo la suscripción. La tabla de empresas incluirá un botón que dirigirá al suscriptor a la página de ingreso de empresas nuevas, y botones en cada empresa para editar su información.

La tabla de usuarios incluirá un botón que dirigirá al suscriptor a la página de ingreso de usuarios nuevos, un botón para eliminar usuarios seleccionados y botones en cada usuario para editar su información.

- Página de ingreso de empresas nuevas: Esta página incluirá un formulario de ingreso de información de la empresa a inscribir. Dentro de este formulario se solicitará información específica de la empresa, de su representante legal y del módulo principal y tarifa a utilizar por esta empresa dentro del sistema. Para el prototipo únicamente se podrá seleccionar el módulo de contabilidad.
- Página de ingreso de usuarios nuevos: Esta página incluirá un formulario en el que únicamente se solicitará el nombre del nuevo usuario y su correo electrónico. La página enviará un correo electrónico a la dirección ingresada con un link para que el usuario sea activado.
- Página de edición de información de empresas registradas: Esta página incluirá un formulario con la información de la empresa seleccionada, que le permitirá al suscriptor la edición de la misma. Además, la página incluirá una tabla con todos los usuarios asignados a la empresa y le permitirá al suscriptor, la asignación de usuarios existentes a un módulo nuevo dentro de la empresa. Para el prototipo, el suscriptor únicamente podrá asignar usuarios existentes al módulo de contabilidad de la empresa.
- Página de edición de información de usuarios existentes: Esta página mostrará la información del usuario que únicamente podrá ser modificada por el mismo. Además, contendrá una tabla de roles por módulo y empresa y le permitirá al suscriptor, asignar distintos roles en varios módulos de diferentes empresas al mismo usuario.

3.2.5.3.3 Páginas del módulo de contabilidad

Las páginas incluidas en este módulo les permitirán a los distintos usuarios asignados a una empresa en específico, realizar las diferentes operaciones contables de acuerdo a los privilegios asignados a sus roles.

Las páginas a incluir en este módulo serán las siguientes:

- Página de ingreso: Esta página contendrá un formulario de ingreso para usuarios del módulo. En el formulario se solicitará el nombre del usuario y la contraseña.
- Página de selección de empresa: Esta página le permitirá a un usuario, seleccionar la empresa en la que desea trabajar, en el caso de que se encuentre asignado a múltiples empresas. La página mostrará una tabla con las empresas a las que el usuario se encuentra asignado y cada empresa tendrá un botón de selección que enviará al usuario al módulo contable de la empresa en la que se trabajará.
- Página de observación de pólizas contables: Esta página contendrá una tabla que mostrará la información general de las pólizas contables ordenadas por fecha, desde la ingresada más recientemente. La tabla podrá mostrar un número de pólizas configurable y tendrá botones para cambiar de página, agregar pólizas contables nuevas y auditar o anular pólizas contables activas.
Además, contará con un cuadro de filtros que permitirán buscar pólizas contables de acuerdo a parámetros específicos como el estado de la póliza, la fecha de la transacción, la fecha de ingreso, etc.
- Página de ingreso y edición de pólizas contables: Esta página estará formada por dos formularios. Un formulario solicitará la información general de la póliza. Otro formulario servirá para especificar el detalle de la póliza y tendrá la capacidad de agregar un número de filas dinámico, limitado únicamente por la tarifa.
- Página de verificación de pólizas contables: Esta página contará con un cuadro que mostrará la información general de la póliza y una tabla que mostrará el detalle de la misma.

- **Página de observación de cuentas contables:** Esta página contendrá una tabla que mostrará la información general del catálogo de cuentas contables, ordenadas por código de cuenta. La tabla podrá mostrar un número de cuentas contables configurable y tendrá botones para cambiar de página, agregar cuentas contables nuevas y eliminar, inhabilitar o habilitar cuentas contables existentes.
Además, contará con un cuadro de filtros que permitirán buscar cuentas contables de acuerdo a parámetros específicos como el estado de la cuenta, el tipo, etc.
- **Página de ingreso y edición de cuentas contables:** Esta página estará formada por un formulario que solicitará la información de la cuenta contable.
- **Página de verificación de cuentas contables:** Esta página contará con un cuadro que mostrará la información de la cuenta contable.
- **Página de observación de compras:** Esta página contendrá una tabla que mostrará la información general de las compras, ordenadas por fecha, desde la ingresada más recientemente. La tabla podrá mostrar un número de compras configurable y tendrá botones para cambiar de página, agregar compras nuevas y auditar o anular compras activas.
Además, contará con un cuadro de filtros que permitirán buscar compras de acuerdo a parámetros específicos como el estado de la compra, la fecha del documento, la fecha de ingreso, etc.
- **Página de ingreso y edición de compras:** Esta página estará formada por dos formularios. Un formulario solicitará la información general de la compra. Otro formulario servirá para especificar el detalle de la compra y tendrá la capacidad de agregar un número de filas dinámico, limitado únicamente por la tarifa. Cada compra tendrá una póliza contable asociada.

- Página de verificación de compras: Esta página contará con un cuadro que mostrará la información general de la compra y una tabla que mostrará el detalle de la misma.
- Página de observación de ventas: Esta página contendrá una tabla que mostrará la información general de las ventas, ordenadas por fecha, desde la ingresada más recientemente. La tabla podrá mostrar un número de ventas configurable y tendrá botones para cambiar de página, agregar ventas nuevas y auditar o anular ventas activas.
Además, contará con un cuadro de filtros que permitirán buscar ventas de acuerdo a parámetros específicos como el estado de la venta, la fecha del documento, la fecha de ingreso, etc.
- Página de ingreso y edición de ventas: Esta página estará formada por dos formularios. Un formulario solicitará la información general de la venta. Otro formulario servirá para especificar el detalle de la venta y tendrá la capacidad de agregar un número de filas dinámico, limitado únicamente por la tarifa. Cada venta tendrá una póliza contable asociada.
- Página de verificación de ventas: Esta página contará con un cuadro que mostrará la información general de la venta y una tabla que mostrará el detalle de la misma.
- Página de reportes: Esta página contará con un formulario para especificar el rango de fechas para el cual se generará el reporte y la selección del libro contable a generar. Al momento de presionar el botón de generar libro, el mismo deberá ser desplegado en esta página y el mismo deberá presentar la opción de ser exportado a archivo Excel o PDF.
- Página de configuración contable: Esta página estará formada por secciones o subpáginas que servirán para configurar diferentes aspectos

del módulo de contabilidad. Las secciones que contendrá serán descritas a continuación:

- Sección general: Esta sección permitirá editar los parámetros con los que funciona el módulo de contabilidad, como el período fiscal que se está trabajando y los datos del contador.
- Sección de categorías de compras: Esta sección permitirá el ingreso y eliminación de categorías de compras.
- Sección de categorías de ventas: Esta sección permitirá el ingreso y eliminación de categorías de ventas.
- Sección de rubros de compras: Esta sección permitirá la selección de cuentas contables para ser asignadas como rubros de compras.
- Sección de rubros de ventas: Esta sección permitirá la selección de cuentas contables para ser asignadas como rubros de ventas.
- Página de perfil de usuario: Esta página permitirá la modificación de la información personal del usuario que esté haciendo uso del módulo de contabilidad.

3.2.6 Políticas y tácticas

Como políticas generales podemos listar las siguientes:

- La aplicación será desarrollada en ASP.NET con versión de framework 2.0 en lenguaje C#.
- La capa de conexión a la base de datos será generada con el software de cuarta generación llamada LLBL Gen Pro versión 2.0 en modalidad "Adapter".

- La aplicación será desarrollada bajo el modelo de capas, con los módulos y clases descritos anteriormente.
- Cada módulo deberá ser desarrollado como una clase de .NET independiente que será referenciada en el proyecto del sitio Web a través de archivos DLL compilados.
- Se creará un proyecto de .NET separado que contendrá todos los reportes a ser utilizados para la generación de los libros contables.
- Las interfaces o páginas del sistema deberán estar separadas en aplicaciones Web independientes que harán uso de los módulos compilados. Las aplicaciones Web en las que se deberá dividir el sistema son las siguientes:
 - Administración: Esta aplicación Web contendrá todas las páginas con las que un suscriptor administre su cuenta.
 - Suscripciones: Esta aplicación Web contendrá todas las páginas con las que un cliente podrá suscribirse al sistema.
 - Contabilidad: Esta aplicación Web contendrá todas las páginas que manejarán el módulo de contabilidad del sistema.
 - General: Esta aplicación Web contendrá todas las páginas que serán de acceso público a cualquier persona que visite el sitio Web.

3.3 Especificación de casos de prueba del software

3.3.1 Introducción

Esta especificación provee una visión general para definir la estrategia de pruebas del prototipo y hace referencia a los diferentes estándares que serán aplicados al realizar las mismas. Se utilizarán criterios de prueba bajo los paradigmas de caja blanca y caja negra.

3.3.1.1 Interacción con el equipo

El prototipo será desarrollado y probado por la misma persona por lo que no será necesario definir la interacción de un equipo de pruebas.

3.3.2 Objetivo de los casos de prueba

El objetivo de los casos de prueba es encontrar y reportar la mayor cantidad de errores para mejorar la integridad del prototipo. A pesar de que no se puede realizar un trabajo de pruebas exhaustivo, se realizará la mayor cantidad de pruebas posible para lograr esta meta. Además, se desea lograr que todas las características presentadas en la especificación de diseño del software funcionen adecuadamente.

3.3.3 Visión general del proceso

El flujo general del proceso de pruebas será presentado a continuación:

- Identificación de los requerimientos a ser probados.
- Identificación de los casos de prueba particulares que serán utilizados para probar cada módulo.
- Revisión de los casos de prueba y resultados para asegurar el funcionamiento adecuado de la unidad.
- Identificación de los resultados de cada prueba.
- Realización de las pruebas.
- El éxito de las pruebas será necesario para que cada componente sea integrado al prototipo final.

- Cada caso de pruebas que no resulte exitoso será verificado y corregido en el prototipo y se realizarán los casos de prueba nuevamente.

3.3.4 Estrategia de pruebas

Se hará uso de una plantilla para reportar cada caso de pruebas del prototipo. La forma de la plantilla será presentada a continuación:

Tabla I. Plantilla de caso de pruebas.

Tipo de prueba		
Número		
Nombre		
Descripción		
Componentes a ser probados		
1		
2		
Especificaciones		
Entradas		Salidas / Resultados esperados
Pasos de la prueba		
1		
2		
3		
4		

3.3.4.1 Pruebas de caja blanca

Estas pruebas serán realizadas a nivel de código fuente para identificar errores específicos del lenguaje de programación como mala sintaxis, errores de lógica o para verificar el funcionamiento de funciones y módulos específicos. Los casos de prueba de unidad serán desarrollados para validar la funcionalidad del código del prototipo. Todas las funciones descritas en la especificación de diseño del software serán probadas y validadas.

3.3.4.2 Pruebas de caja negra

Estas pruebas serán realizadas a nivel de interfaz. Servirán para verificar que todas las entradas en el sistema produzcan las salidas deseadas. Se deberán probar todas las páginas Web del prototipo para los distintos módulos de los que está formado. Entre las pruebas a incluir se encontrará el ingreso de valores inválidos en los formularios de ingreso y filtros de búsqueda, ingreso de valores válidos para verificar el funcionamiento de

3.3.5 Pruebas de sistema

Se realizará este tipo de pruebas para detectar errores que únicamente podrían ser expuestos al utilizar el sistema completamente integrado. Para esto, deberán acoplarse los distintos módulos del prototipo en un sitio Web completo, y cada módulo deberá ser accesible dentro del mismo.

Deberán también realizarse pruebas para garantizar el buen rendimiento y tiempo de respuesta de la aplicación y su facilidad de uso.

4. PROPUESTA DE NEGOCIOS BASADA EN EL PROTOTIPO

En el capítulo anterior se presentó un análisis y diseño completo para desarrollar una aplicación utilizando el modelo SaaS. Este capítulo permitió observar los aspectos técnicos a nivel de ingeniería de software para desarrollar este tipo de aplicaciones pero es necesario también conocer los aspectos que nos permitirían posicionar en el mercado a una empresa que utilice la venta de los servicios que estas aplicaciones prestan como su forma de ingresos principal. Para esto se presenta el ejemplo de un plan de negocios de una empresa ficticia llamada “Creativosoft” que tiene como producto principal la aplicación llamada “Creativa Finanzas” prestando servicios contables en línea a través de su módulo “Creativa Contabilidad”.

4.1 Resumen ejecutivo

- Creativosoft será una empresa que se dedicará al desarrollo de software orientado a la prestación de servicios (SaaS). Como actividad secundaria se dedicará al desarrollo de software a encargo, consultorías y renta de infraestructura para soportar aplicaciones.
- La empresa será fundada en Diciembre del año 2009 por Víctor Álvarez.
- La empresa surge por la necesidad del fundador de aumentar las opciones para adquirir tecnología y maximizar el número de personas que deseen hacer uso de este recurso, movidas por el deseo de aumentar su competitividad, profesionalismo, calidad y control de procesos de negocio.

Es por esto que Creativosoft se dará a la tarea de desarrollar aplicaciones propias de la empresa, con estándares y técnicas de desarrollo que garanticen una alta calidad comparable a cualquier producto existente en el mercado, y a través de un modelo de renta de software, proveer a sus clientes de poderosas herramientas que satisfagan sus necesidades empresariales o personales por una cuota mensual cómoda.

- El modelo de renta de software evitará que nuestros clientes tengan que realizar inversiones en infraestructura que soporte sus aplicaciones, seguridad, asistencia técnica, y la inversión de la aplicación como tal, ya que Creativosoft, a través de sus aplicaciones en línea, proveerá de todo esto como parte de sus servicios, añadiendo poderosas ventajas que solo poseen las aplicaciones en línea, como el uso de las mismas en cualquier parte del mundo a través de una computadora con conexión a Internet únicamente, actualizaciones y mejoras constantes en las aplicaciones, servicios administrativos realizados por el equipo de Creativosoft y muchas más, sin costo alguno.
- El mercado actual en Guatemala no ha expandido sus horizontes hacia la renta de software en línea, lo que nos colocaría como una empresa pionera con ventaja competitiva sobre cualquier otra empresa de desarrollo guatemalteca.
- Este tipo de prestación de servicios es cada vez más utilizado a nivel mundial ya que se provee una aplicación integral, bajo una infraestructura física y de servicios al cliente, sólida, que le permite a cualquier persona desarrollar competitivamente su empresa o negocio.

4.2 Introducción a Creativosoft

La función principal de la empresa será desarrollar aplicaciones de software propias, con el objetivo de prestar servicios a través de ellas. Como actividades secundarias, la empresa se dedicará a desarrollar aplicaciones específicas a encargo, realizar consultorías en términos de adquisición de infraestructura y tecnología, y renta infraestructura para soportar aplicaciones de otras empresas, además de brindar soporte técnico a las mismas.

La empresa será creada debido a que su fundador Víctor Álvarez, ha observado un aumento importante en el número de personas, negocios, empresas y organizaciones que muestran una necesidad común: "Hacer uso de recursos tecnológicos que les permitan desarrollarse competitivamente, pero a través de mecanismos que les garanticen calidad, eficiencia, seguridad, rendimiento y atención al cliente, sin necesidad de realizar fuertes inversiones ni poseer un gran conocimiento técnico".

4.2.1 Visión de la empresa

- Dominar tecnológicamente Guatemala a través de nuestro enfoque innovador y creativo de desarrollo de software.
- Estandarizar la forma en la que las personas realizan sus actividades cotidianas a través de herramientas creadas para satisfacer sus necesidades, con la capacidad de acoplarse a cualquier sistema de uso actual importante, privado o del estado.

4.2.2 Misión de la empresa

Proveer tecnología de alta calidad a la mayoría de la población, a través de metodologías de pago cómodas que les garanticen aplicaciones confiables, de alta calidad, seguras y eficientes sin necesidad de hacer inversiones extras.

Actualmente la futura empresa se encuentra desarrollando proyectos que permitan suplir las necesidades financieras del mercado, entre los cuales destaca *Creativa Finanzas* y su primer módulo *Creativa Contabilidad*.

El equipo utiliza las herramientas que más se adecuen al proyecto en desarrollo para garantizar su calidad y alto rendimiento. Entre las herramientas utilizadas están, PHP, ASP, ASP.NET, Java, MySQL, PostgreSQL, SQL Server, Oracle.

4.3 Propuesta de productos y servicios

4.3.1 Creativa Contabilidad

Este proyecto es parte de *Creativa Finanzas*, un sistema financiero en línea que permitirá a nuestros clientes hacer uso de poderosas herramientas para satisfacer sus necesidades dentro de ese ámbito.

Creativa Finanzas se encontraría dividido en módulos que representan distintas áreas dentro de un software financiero y podrían ser adquiridos por los clientes según sus necesidades.

Cada módulo contaría con un conjunto de tarifas que dependerían de la magnitud de operaciones realizadas en el mismo. En general se pretende crear

una tarifa gratuita que llene las necesidades de la mayoría de empresas del mercado objetivo.

La aplicación abarcaría desde módulos básicos como el de contabilidad (que actualmente recibe el nombre de *Creativa Contabilidad*), hasta módulos más complejos como facturación, bancos, inventarios, facturación, cuentas por cobrar y nóminas.

Figura 9. Logo de Creativa Contabilidad.



Para poder hacer uso de la aplicación se necesitaría ingresar al sitio web **www.creativafinanzas.com** y crear una suscripción. Esta suscripción le permitiría a un cliente llevar el control de múltiples empresas y múltiples usuarios. No existiría límite para el número de empresas inscritas bajo una suscripción ni para el número de usuarios. Las limitaciones se encontrarían en las tarifas de cada módulo.

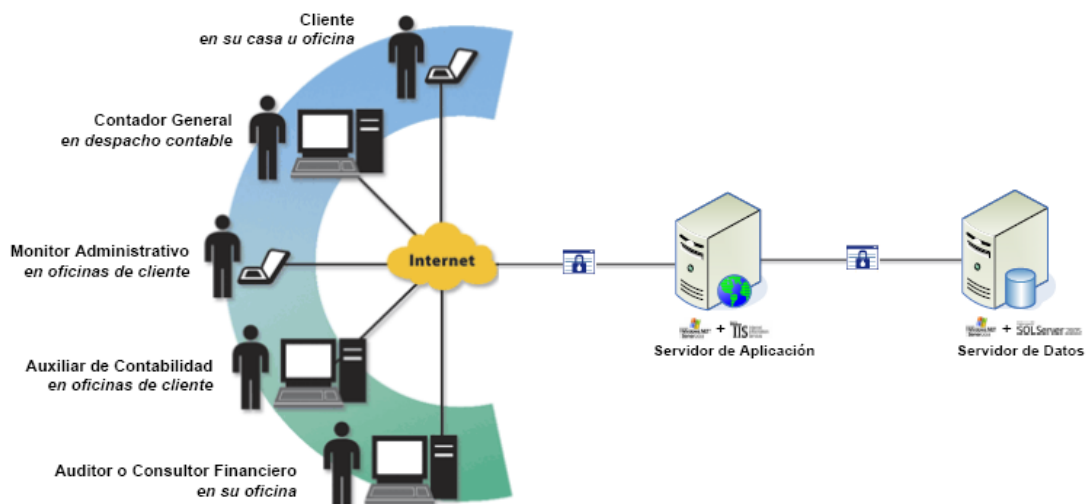
El sistema de suscripciones está desarrollado para que un suscriptor pueda asignarle a una empresa múltiples usuarios en múltiples módulos bajo distintos roles, por ejemplo, un suscriptor podría inscribir una empresa y solicitar el módulo contable para la misma, bajo una tarifa. De acuerdo a esta tarifa, el suscriptor puede asignarle a esta empresa un grupo de usuarios que utilicen el módulo adquirido y que tengan múltiples funciones, como por ejemplo asignar

un usuario Contador, un Auxiliar de Contabilidad, un Auditor, etc. Cada uno de estos usuarios contará con permisos para realizar operaciones específicas dentro de la aplicación. De esta manera el suscriptor podrá administrar a su personal y asignarle funciones de acuerdo a su nivel en la empresa. Además, cada usuario podría ser utilizado en distintos módulos dependiendo del papel que juegue en cada área que los módulos representen, así como también ser asignado en múltiples empresas.

Cada módulo tendría una tarifa por empresa, de esta manera, si se tiene un módulo en múltiples empresas, se debería realizar un pago por cada empresa que haga uso de él (si la tarifa requiere pago).

El sistema está desarrollado para que un suscriptor pueda hacer uso de módulos nuevos que la empresa vaya proveyendo y asignándolos a empresas que los necesiten. Así pues, no existen límites en el desarrollo de la aplicación financiera y se puede volver tan compleja y completa como sea necesario.

Figura 10. Modelo de uso e infraestructura.



Las ventajas del sistema son las siguientes:

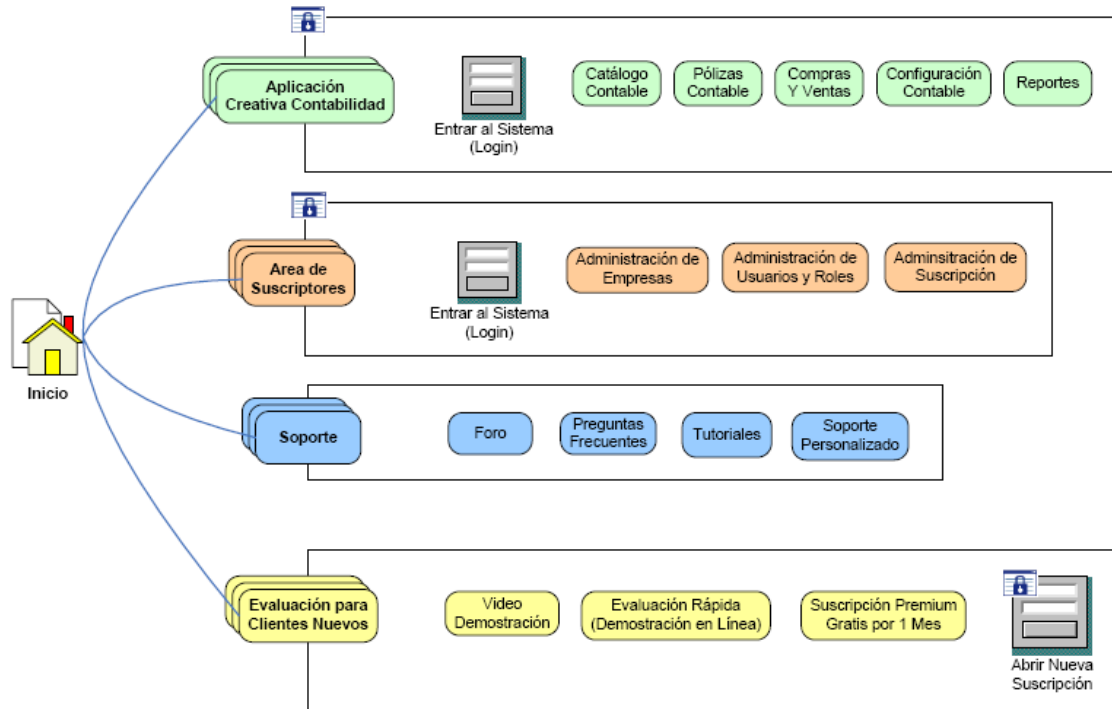
- Para hacer uso de la herramienta únicamente se necesita de una computadora con conexión a Internet (*Ver Figura 5*).
- El sistema es transportable, ya que puede ser utilizado desde cualquier lugar que posea una conexión a Internet.
- El sistema es utilizado bajo la modalidad de prestación de servicios de software, por lo que el cliente no necesita realizar una inversión fuerte para adquirirlo. En su lugar, el cliente realiza un pago mensual cómodo. De hecho en la mayoría de casos de nuestros clientes, sus necesidades se apegarán a un plan gratuito.
- El cliente no necesitará realizar fuertes gastos invirtiendo en infraestructura (servidores, licencias de software, redes, acondicionamiento de ambientes, etc.) para soportar el sistema. Tampoco necesitará un departamento de Informática para realizar tareas como configuración, copias de seguridad y mantenimiento. Por ser un sistema en línea, nuestra empresa contaría con la infraestructura necesaria para soportarlo y con el equipo necesario para realizar cualquier operación administrativa solicitada por el cliente.
- El sistema sería actualizado y mejorado constantemente y esto no representaría un costo adicional para el cliente. Además, no sería necesario que el cliente deba dejar de utilizar el sistema para realizar las actualizaciones y mejoras, pues se cuenta con las ventajas de cualquier sistema en línea en el que todas estas operaciones son realizadas una única vez en el servidor.

- El sistema se apegaría a las leyes actuales de la república, y cualquier cambio en ellas, sería reflejado en el sistema de manera sencilla, por lo que los usuarios no deben realizar inversiones adicionales para que esto sea llevado a cabo.
- El soporte técnico es parte integral del producto y el cliente podría hacer uso de él en un horario establecido.
- Nuestra empresa se encargaría de proveer las condiciones necesarias para garantizar que el sistema sea seguro, eficiente y confiable.

4.4 Situación actual del proyecto

Actualmente *Creativa Finanzas* se encuentra en etapa de prototipo con su primer módulo llamado *Creativa Contabilidad*, módulo contable que permite llevar el control de los libros Diario, Mayor, Balance de Saldos, Balance General, Estado de Resultados, Compras y Ventas.

Figura 11. Estructura de contenidos.



Las páginas o grupos de páginas que tiene impreso un candado tendrían la característica de que la información que envían y reciben viajaría a través de canales seguros de Internet. Esto quiere decir que la información viajaría encriptada y no puede ser descifrada por un tercero que intente interceptarla. Este estatus de seguridad sería certificado por una empresa en el campo, internacionalmente conocida y prestigiosa.

La base de datos se encontraría en servidores en otro país, en un lugar apropiadamente acondicionado para que los equipos rindan el 100% y no estén expuestos a problemas de seguridad. Además, se prepararía una planificación de copias de seguridad de la información, para respaldar la disponibilidad de los datos del cliente.

4.5 Inversión Inicial

- Se debería contratar un servicio de hosting (alojamiento de sitios web) que soporte aplicaciones desarrolladas en Microsoft .NET con versión de framework 2.0. y brinde el uso de bases de datos Microsoft SQL Server 2005.
- Se debería adquirir un certificado de seguridad SSL para realizar operaciones a través de un canal encriptado.
- Se debería subcontratar a una persona que se encargue de prestar soporte técnico a los primeros clientes de la empresa. Esta persona trabajaría desde su casa y se encargaría de contestar llamadas de soporte y responder foros.
- Se debería contar con un fondo inicial para pagar servicios básicos como Internet, electricidad, etc. durante al menos 6 meses.
- Se debería adquirir el servicio de oficina virtual que permitiría contar con una secretaria de tiempo completo que transferiría las llamadas de soporte técnico a la persona contratada.

La tabla que describe la inversión inicial será presentada a continuación:

Tabla II. Inversión inicial

Descripción	Cantidad	Precio Unitario	Precio Total
Servicio de hosting para ASP.NET 2.0 y SQL Server 2005	6 meses	Q131.66	Q790.00
Certificado de seguridad SSL	1 año	Q1,640.00	Q1,640.00
Personal de soporte técnico	1 persona / 6 meses	Q2,000.00	Q12,000.00
Fondo para servicios básicos	6 meses	Q500.00	Q3,000.00
Servicio de oficina virtual	6 meses	Q400.00	Q2,400.00
Total			Q19,830.00

*Los precios fueron obtenidos del mercado tecnológico actual. Algunos precios obtenidos en dólares fueron convertidos a quetzales utilizando una tasa de cambio de Q8.20

4.6 Costos de mantenimiento mensual

- Tomando en cuenta que una persona podría tomar un promedio de 30 llamadas por día laboral (4 llamadas por hora), se debería subcontratar a

una persona de soporte técnico por cada 50 clientes que se suscriban al sistema contable, asumiendo que un cliente realizaría un promedio de 10 llamadas al mes. Esto permitiría mantener un buen nivel de soporte técnico ya que una persona capacitada sería capaz de atender a un máximo de 90 clientes por mes.

- Se debería cubrir la cuota mensual de servicio de hosting.
- Se debería cubrir la cuota mensual de servicio de oficina virtual.
- Se debería acumular una cuota mensual para la compra del siguiente año del certificado de seguridad SSL.
- Se debería cubrir una cuota mensual para el pago de servicios básicos de funcionamiento de la empresa como servicio de Internet, electricidad, etc.

La tabla que describe los costos de mantenimiento mensual del servicio se presenta a continuación:

Tabla III. Costos mensuales de mantenimiento

Descripción	Precio
Personal de soporte técnico	Q2,000.00
Servicio de hosting	Q150.00
Servicio de oficina virtual	Q400.00
Cuota acumulada para compra anual de certificado SSL	Q150.00
Servicios básicos	Q500.00
Total	Q3,200.00

*Los costos mensuales se encuentran calculados para una cantidad máxima inicial de 50 clientes.

4.7 Tarifas

Creativa Contabilidad se dividirá en tarifas para poder satisfacer las necesidades de distintos tipos de empresa. En la tabla siguiente se detalla la configuración y precio de cada tarifa.

Tabla IV. Tarifas de Creativa contabilidad

Tarifa	Básica	Intermedia	Profesional
Transacciones	200	1500	Ilimitadas
No. De cuentas por póliza	10	25	Ilimitadas
Usuarios Administrador	1	1	1
Usuarios Contador	1	3	5
Usuarios Auxiliar de contabilidad	0	3	5
Usuarios Auditor	0	1	3
Usuarios Monitor	0	1	2
Tarifa Mensual	Gratis	Q249.00	Q399.00

Los precios de las tarifas serían aplicados a cada empresa suscrita bajo una cuenta, lo que significa que un mismo suscriptor que tenga múltiples empresas bajo su cuenta debería cancelar el precio de la tarifa por cada una de las empresas que posea. La tarifa gratuita sería mantenida a través de publicidad aplicada a los sitios utilizados por los usuarios suscritos bajo la misma.

La meta a corto plazo de la empresa consistiría en suscribir a 50 empresas bajo la tarifa Intermedia que representarían un ingreso mensual de Q 12,450.00. Dados los costos de mantenimiento mensual, el nivel de ganancias permitiría la expansión del producto para que se incorporen otros módulos a la suite financiera.

4.8 Equipo humano

- *Equipo Directivo:* Encargado de tomar las decisiones en cuanto a políticas, administración, nuevas funcionalidades, cronogramas de desarrollo, métodos de soporte, etc.

- *Dirección Administrativa:* Encargada de garantizar el funcionamiento del área administrativa, que incluye secretaría, mensajería, servicio al cliente y mercadeo.
- *Dirección de Contenidos:* Encargada de proveer información de mercado y de otras herramientas, para promover las mejoras funcionales de la aplicación. El ejecutor es el equipo de Desarrollo de Software y el de Mantenimiento y Soporte.
- *Dirección Técnica:* Encargada de informarse en el mercado, en las pruebas de rendimiento y en los planes de infraestructura para promover mejoras en calidad y rendimiento de la aplicación. Esto incluye métodos para mejorar la administración del proyecto. Los ejecutores son el equipo de Desarrollo de Software y el de Mantenimiento y Soporte.

4.9 Planes de expansión

4.9.1 Mejoras que podrían ser incorporadas al producto

- Inclusión de Planilla del IVA y su integración con los libros de Compras y Ventas.
- Integración de Compras y Ventas con el sistema de formularios fiscales de SAT.
- Integración de Compras y Ventas y los formularios fiscales con la Banca Electrónica.
- Mejoras en usabilidad.

- Adición de reportes financieros especializados, como Flujos de Caja, Gráficas de Tendencia, y Presupuesto.
- Realizar una Edición Educativa inducible en instituciones educativas para el aprendizaje de la contabilidad.

4.9.2 Nuevos módulos que podrían ser incorporados al producto

Creativa Finanzas es una suite financiera. Por este motivo, Creativa Contabilidad es solo la punta del iceberg de lo que podría llegar a ser una gama integrable y versátil de módulos. El plan de expansión podría incluir el desarrollo, pruebas e implementación de estrategias para la inducción de los otros cuatro módulos mostrados en la siguiente figura.

Figura 12. Módulos de creativa finanzas.



Entre los objetivos de expansión a mediano plazo podría incluirse la internacionalización a Centro América.

4.10 Mercado

4.10.1 Mercado objetivo

Cualquier empresa pequeña o mediana, o un representante financiero de ésta que requiera administrar la contabilidad de una o varias empresas. En este perfil caben:

- Contadores independientes
- Despachos contables
- Profesionales independientes
- Empresas pequeñas y medianas

4.10.2 Entorno competitivo

A pesar de que hay muchas aplicaciones de software contable, la combinación de valor *Web + Renta + Enfoque Nacional y/o Latinoamericano* es reducida. La brecha entre las propuestas típicas y la de *Creativa Finanzas* es básicamente esa combinación conveniente.

Dentro de la competencia directa existente en Guatemala se encuentran empresas que ingresaron al país a través de franquicias, sucursales de operación o desarrollo nacional. A continuación se presentan tablas con los competidores a nivel nacional.

Tabla V. Competidores nacionales con aplicaciones de escritorio y venta directa

Producto	Proveedor	País Sede	Sitio Web	Propuesta de Valor	Precio Lista Aprox.
Memory Conty	Memory Computación	Uruguay	http://www.memorycomputacion.com/conty.php	Sistema contable altamente reconocido por varias entidades y extendido a latinoamérica por el sistema de franquicias. Famoso por su integración con productos de Microsoft Office.	US\$ 462.33
SAE-COI	ASPEL	México	http://www.aspel.com.mx/mx	Sistema contable utilizado por muchos despachos contables y firmas auditoras. Famoso por sus usabilidad y por poseer todas las características de un software de ese nivel. No requiere licencias de otros productos para su instalación.	US\$ 366.03

Tabla VI. Competidores nacionales con aplicaciones de escritorio y modelo de renta

Producto	Proveedor	País Sede	Sitio Web	Propuesta de Valor	Precio Lista Aprox.
SAE-COI	ASPEL	México	http://www.aspel.com.mx/mx	Sistema contable utilizado por muchos despachos contables y firmas auditoras. Famoso por sus usabilidad y por poseer todas las características de un software de ese nivel. No requiere licencias de otros productos para su instalación.	US\$ 58.00/mes

También existen empresas que no poseen una entidad representativa en el país, sin embargo sus sistemas pueden ser adquiridos y utilizados sin restricción de fronteras. En las siguientes tablas se muestran a los distintos competidores potenciales sin presencia en el territorio nacional.

Tabla VII. Competidores internacionales con aplicaciones de escritorio y venta directa.

Producto	Proveedor	País Sede	Sitio Web	Propuesta de Valor	Precio Lista Aprox.
QuickBooks: Premier Accountant Edition 2007	Intuit	USA	http://quickbooks.intuit.com/product/accounting_software/premier_accounting_solutions_for_accountants.html	Sistema dirigido al mercado anglosajón, intuitivo, amigable y con posibilidades de integración. El software contable más usado en USA.	US\$ 1,199.95
Microsoft Accounting	Microsoft	USA	http://office.microsoft.com/en-us/accounting	Sistema para gestión gerencial express que incluye manejo de finanzas (contabilidad). Relativamente nuevo pero poderoso, integrado con otros productos MS.	US\$ 500.00

Tabla VIII. Competidores internacionales con aplicaciones de escritorio y modelo de renta.

Producto	Proveedor	País Sede	Sitio Web	Propuesta de Valor	Precio Lista Aprox.
QuickBooks, Online Edition	Intuit	USA	http://oe.quickbooks.com/rr/product_info.cfm?sc=QBC-V51-HME-HMEPGE-TSTA-060213	Sistema dirigido al mercado anglosajón, intuitivo, amigable y con posibilidades de integración. El software contable más usado en USA.	US\$ 20.00/mes
ServiCont	Servipymes	Chile	http://www.servipyme.cl/servi_cont.htm	Sistema dirigido a la pequeña y mediana empresa de Chile. Funcionalidades básicas de contabilidad.	US\$ 62.54/mes

4.11 Plan de Mercadotecnia

Se planea subcontratar el servicio de consultoría de mercado para definir objetivos claros y estrategias que garanticen dichos objetivos.

A grandes rasgos, las acciones que se deberían incluir en la estrategia mercadológica, son:

- Crear y fortalecer alianzas con sectores claves en la creación de confianza y valor, tales como colegios profesionales de contabilidad y auditoría, instituciones gubernamentales encargadas de fiscalización, instituciones educativas de contabilidad, firmas financieras, etc.
- Utilizar medios noticiosos gratuitos que se interesen en publicar el proyecto. Esto puede incluir: periódicos, boletines de instituciones financieras y educativas, sitios Web.
- Introducir la marca en actividades relacionadas al contexto del servicio, tales como seminarios financieros, reuniones de colegio de profesionales, capacitaciones de instituciones fiscalizadoras, etc.
- Uso de medios masivos de comunicación para la expansión y cobertura.

Los costos de las estrategias son inciertos al momento, ya que no se dispone de un plan concreto de mercado ni de cotizaciones realistas de estas acciones.

Sin embargo, con los puntos anteriormente dichos, se pretende iniciar las actividades en ese orden (de bajo costo a alto costo) para capitalizar a la empresa con asociados y patrocinadores que financien las subsecuentes actividades.

5. CASOS DE ÉXITO DEL MODELO SAAS EN LA ACTUALIDAD

Habiendo cubierto los distintos aspectos del desarrollo de aplicaciones de software basadas en el modelo SaaS, desde la teoría y los detalles técnicos, hasta la creación de un modelo de negocios basado en un prototipo desarrollado bajo este modelo, nos queda únicamente listar alguno de los casos de éxito de empresas que han logrado posicionarse en el mercado internacional ofreciendo productos bajo el modelo de prestación de servicios.

5.1 Salesforce.com

Salesforce.com es un vendedor de sistemas de gestión de relaciones con clientes (CRM) que trabaja para empresas a través de Internet utilizando el modelo de software como un servicio.

La empresa fue fundada por el ex ejecutivo de Oracle Marc Benioff en 1999. En junio del 2004 la empresa se hizo pública en el mercado de valores de Nueva York bajo el símbolo CRM.

La sede central de la empresa se encuentra ubicada en San Francisco, California, con sedes regionales en Dublín (para cubrir Europa, Este medio y África), Singapur (para cubrir Asia menos Japón) y Tokio (para cubrir Japón). Otras oficinas importantes se encuentran ubicadas en Toronto, Nueva York, Londres, Sydney y San Manteo California. Los servicios de salesforce.com han sido traducidos a 16 idiomas y actualmente son utilizados por más de 55,400 clientes y mas de 1,500,000 suscriptores.

La solución CRM de Salesforce.com se encuentra dividida en varios módulos: Ventas, Servicio y soporte, Gestión de relaciones con socios, Mercadeo, Contenido, Ideas y Estadísticas. El producto recibe el nombre de plataforma Force.com y permite que programadores externos puedan crear aplicaciones adicionales que se integren a la aplicación principal de la plataforma y sean alojadas en la infraestructura de Salesforce.com. Las aplicaciones son desarrolladas en APEX (un lenguaje de programación propietario de Salesforce.com parecido a Java) y VisualForce (una sintaxis parecida al XML para construir interfaces HTML, AJAX y Flex).

Los usuarios de Salesforce.com pueden personalizar su aplicación CRM. Dentro del sistema hay secciones como Contactos, Reportes y Cuentas. Cada sección contiene campos asociados a la misma. Por ejemplo, en la sección de Contactos se encuentra información de correos electrónicos, números de teléfono, etc. Cada sección puede ser personalizada agregando campos definidos por el usuario. Además, la personalización puede ser efectuada a nivel de plataforma completa, agregando aplicaciones adicionales a la aplicación principal de Salesforce.com. Esto se lleva acabo agregando secciones definidas por el usuario para características específicas como Finanzas, Recursos Humanos, etc.

Además de la interfaz Web, Salesforce.com provee un conjunto de servicios Web (Web services) que permite la integración con otros sistemas. En abril del año 2009, la empresa presentó una versión ligera de su producto que puede ser utilizada por dispositivos móviles como Blackberry, Iphone y dispositivos basados en Windows.

5.2 Vocus

Vocus es una empresa fundada por Rick Rudman y Bob Lentz en 1992, que provee sistemas para relaciones públicas bajo el modelo de software como un servicio. Sus sistemas proveen monitoreo de noticias, distribución de comunicados de prensa, listas de medios y estadísticas que ayudan a profesionales de relaciones públicas a comunicarse con los medios y el público. En la actualidad, estos sistemas son utilizados por más de 3,500 organizaciones y se encuentran disponibles en 7 idiomas. Su sede central se encuentra en Lanham Maryland pero cuentan con oficinas a través de Europa y Asia.

Vocus vende versiones de su software bajo demanda para relaciones públicas y gubernamentales con el objetivo de ayudar a empresas de todos tamaños, locales e internacionales, a comunicarse y relacionarse con periodistas, analistas, oficiales públicos y otras audiencias clave. Sus sistemas se enfocan en tres áreas principales: Relaciones con los medios, distribución de noticias y monitoreo y análisis de noticias. El software de relaciones públicas consiste en módulos individuales como noticias en tiempo real y bajo demanda, base de datos de contactos de medios, estadísticas y manejo de proyectos, que se encuentran relacionados entre sí y son vendidos en distintas versiones o paquetes.

Su edición empresarial está diseñada para empresas grandes y les permite administrar actividades de relaciones públicas, además de medir los resultados en las inversiones realizadas en esa área a través de distribución y monitoreo de noticias.

Su edición profesional está diseñada para empresas medianas y les permite crear, llevar control y analizar publicidad.

Su edición pequeña está diseñada para empresas pequeñas y les provee herramientas para generar cobertura de publicidad y manejo de ventas.

La empresa cuenta con alianzas estratégicas para la distribución de sus comunicados de prensa. Entre las compañías con las que se encuentra aliada se pueden listar a Google News (noticias de Google), Yahoo! News (noticias de Yahoo!), Lycos News (noticias de Lycos), Topix.net, Excite News (noticias de Excite) y eMediaWire. Además, la empresa cuenta con sitios Web internacionales para la distribución de sus comunicados en el mercado español, francés y chino.

Vocus maneja una gama completa de sitios Web para todos sus mercados, entre los cuales se pueden mencionar:

- PRWeb.com: Este es el sitio Web principal de la empresa. A través de él los usuarios tienen la capacidad de publicar y distribuir sus comunicados de prensa en línea.
- PRWebUK.com: Este sitio proporciona una distribución optimizada para llegar específicamente a periodistas, escritores de blogs, consumidores y otros, ubicados en el Reino Unido.
- PRWeb Photowire: Este sitio proporciona un servicio de fotografías digitales.
- PRWeb Podcast: Este sitio presenta entrevistas grabadas acerca de los comunicados de prensa.
- RSS Pad: Permite a los usuarios crear alimentadores RSS.

- NewsPad: Este portal proporciona un buscador general para todos los sitios Web de la empresa.
- 301url: Este sitio permite la asistencia a usuarios para la creación de direcciones Web (urls) más pequeñas.

Actualmente la empresa se encuentra desarrollando estrategias para cubrir más mercado a través de aplicaciones de uso popular, como la reciente capacidad de los usuarios de los sitios Web, de compartir sus noticias a través del sitio Web Twitter.com al mismo tiempo que las suben a los sitios de la empresa.

5.3 Workday inc.

Workday es una empresa fundada por David Duffield, el fundador y anterior CEO (Gerente General) de la empresa PeopleSoft, y Aneel Bhusri, el anterior Gerente Estratégico de la misma empresa.

La empresa se dedica a la venta de soluciones de software para la administración financiera y de recursos humanos bajo el modelo de software como servicio.

Los módulos ofrecidos por la empresa se listan a continuación:

- Administración de capital humano: Permite el manejo de beneficios, compensaciones, control de asistencia, desarrollo, rendimiento y reclutamiento de personal nuevo.
- Planillas: Permite el agrupamiento de empleados, manejo de reglas para el cálculo de sueldos y pago de empleados de acuerdo a las necesidades organizacionales y política de las empresas. Además

permite la configuración completa de sueldos y deducciones, balances y prestaciones acumuladas, pagos grupales, procesamiento de pagos, reportes y actualización de impuestos.

- Administración financiera: Proporciona herramientas para el manejo de la contabilidad, cuentas de clientes, cuentas de proveedores, manejo de efectivo, manejo de recursos de negocio, gastos y contrataciones.
- Administración de costos de trabajadores: Combina las características de las herramientas de manejo de gastos, contrataciones y administración de recursos humanos en una solución que extiende las herramientas de administración de capital humano y le permite a las compañías entender y manejar los costos totales de su fuerza laboral.
- Red de beneficios: Le provee a las compañías que manejan recursos humanos un catálogo de integraciones prefabricadas conectadas a proveedores de beneficios, dando a estas compañías la capacidad de evaluar, seleccionar y ofrecer los planes de beneficios más apropiados para su fuerza de trabajo.
- Solución base: Provee herramientas para el manejo de inteligencia de negocios, uso amigable de los módulos, integración bajo demanda con aplicaciones de terceros, seguridad, desarrollo y procesos de negocio.

La empresa proporciona ayuda a compañías para que puedan construir estrategias que les permitan volverse auto-sostenibles, a través de servicios de evaluación, desarrollo y ciclo de vida.

Recientemente la empresa logró un acuerdo con Flextronics, una compañía global de manufactura de electrónicos con un valor de 34 billones de dólares. El acuerdo le proporciona a Workday una entrada a sus sistemas de 200,000 usuarios nuevos.

5.4 Oco inc.

Oco es una empresa que provee herramientas de inteligencia de negocios bajo el modelo de software como servicio. El objetivo principal de la empresa es brindar una visión esencial de negocios que permita el crecimiento de sus clientes a través de la generación de información útil para las empresas, como sus productos, ventas y comportamiento de sus clientes.

Oco provee soluciones enfocadas a problemas de negocio específicos en industrias determinadas. Entre los módulos que la empresa provee se pueden listar:

- Fuerza de clientes y ventas: Provee herramientas que le permiten a las compañías conocer el nivel de ganancia que reciben de sus clientes y su nivel de satisfacción. Además proveen fuentes de información sobre las ventas de las compañías.
- Cadena de suministros: Provee una solución que permite la colaboración con proveedores, clientes y distribuidores. A través de esta solución las compañías tienen la capacidad de desarrollar procesos de planificación de ventas y operaciones, manejo de inventarios con clientes y proveedores, y manejo de costos de transporte de suministros.
- Servicios de rendimiento: Provee estadísticas y análisis que permiten capturar e integrar métricas de rendimiento y costos de funcionamiento.
- Servicios para empresas de manufactura.
- Servicios para empresas de ventas al por menor.

Entre los clientes que hacen uso de los servicios de Oco se encuentran BlueSky Brands, Dunkin' Donuts, Casual Male, Cradles to crayons, LogmeIn, entre otros.

Recientemente, la empresa llegó a un acuerdo con Office Depot para brindarles servicios de reportería de ventas al por menor y mejorar su entendimiento de las tendencias del mercado.

CONCLUSIONES

1. El modelo SaaS proporciona un enfoque de desarrollo de software en el cual los sistemas son ofrecidos en forma de paquetes de servicios completos que se acomodan a las necesidades de los clientes a los que se encuentran orientados, contrario al enfoque tradicional en el que los sistemas son vendidos a través de productos empaquetados. El uso de estos servicios generalmente se adquiere a través de una cuota periódica que puede ser mensual, semestral o anual.
2. Las principales ventajas de hacer uso de este tipo de aplicaciones, se encuentra la capacidad de acceder a las mismas desde cualquier lugar con conexión a Internet, reducción de costos por la adquisición de licencias, soporte técnico de alta calidad, disponibilidad 24/7 de las aplicaciones, mejoras periódicas y un rendimiento adecuado de los sistemas que depende únicamente de la velocidad de conexión a Internet con la que se cuenta.
3. Las aplicaciones desarrolladas bajo el modelo SaaS se encuentran alojadas en servidores propios de la empresa que ofrece los servicios de software y es ésta quien se encarga de asegurar la infraestructura física y tecnológica que asegure el funcionamiento adecuado de las mismas. Actualmente empresas pequeñas o medianas que ofrecen sus productos a través de este modelo, hacen uso de servicios de hosting, con compañías que ya cuentan con todas las características necesarias para soportar este tipo de aplicaciones.

4. Dentro de las estrategias utilizadas para obtener ganancias a través de aplicaciones ofrecidas bajo el modelo SaaS, se encuentra la venta de paquetes de servicios que dependen del tamaño de las necesidades de los usuarios a utilizarlos, venta de publicidad en paquetes de servicios gratuitos utilizados por clientes pequeños, servicios añadidos propios del mercado al que se encuentren enfocadas las aplicaciones y venta de herramientas de desarrollo que permitan el acoplamiento de estas aplicaciones a sistemas propietarios de los clientes.
5. A nivel mundial, el aumento del uso de Internet y los avances tecnológicos han permitido que las aplicaciones desarrolladas bajo el modelo SaaS cuenten con características que les permitan ser igual o más robustas que las aplicaciones desarrolladas bajo el modelo clásico, lo que ha permitido que empresas grandes como Walmart aprovechen las ventajas que estos modelos ofrecen para enfocarse en su línea de negocios, a fin de reducir costos en servicios de informática especializados, ofrecidos por empresas que trabajan bajo este enfoque.
6. A nivel nacional, a pesar que el uso del Internet ha aumentado, la cultura de desconfianza que todavía existe acerca de la confidencialidad de información transmitida por la Web, la inestabilidad de los servicios de Internet y su costo, han provocado que las aplicaciones que funcionan bajo el modelo SaaS no florezcan y aún no se encuentre un caso concreto de éxito, de empresas que hayan crecido haciendo uso de este modelo de desarrollo de software.

RECOMENDACIONES

1. El éxito de una aplicación desarrollada bajo el modelo SaaS depende del grado de especialización en el mercado al que se encuentra orientada y el nivel de personalización que le permitirá acomodarse a las necesidades específicas de cada cliente, además de la calidad del soporte técnico ofrecido.
2. Al desarrollar aplicaciones que funcionan bajo el modelo SaaS se debe tomar en cuenta que el rendimiento de las mismas dependerán de la velocidad de conexión del cliente que este haciendo uso de ellas, por lo que el diseño deberá hacer énfasis en la velocidad de respuesta y la facilidad de uso de las aplicaciones.
3. Unos de los aspectos más importantes para el cliente en este tipo de aplicaciones es la confidencialidad de la información y la seguridad de las transacciones realizadas en ellas, por lo que es necesario desarrollar estrategias que garanticen estos aspectos como la creación de acuerdos de confidencialidad, tecnologías de comunicación segura, etc.

BIBLIOGRAFÍAS

1. Conceptos generales.

<http://msdn.microsoft.com/en-us/library/aa905332.aspx> (01/10/2006)

http://www.cio.com/article/109706/The_Truth_About_Software_as_a_Service_SaaS_ (21/05/2007)

<http://www.saas-attack.com/SaaSModel/tabid/187/Default.aspx>
(01/01/2008)

<http://www.mailxmail.com/curso-software-servicio-saas/unico-capitulo-software-as-service-saas> (13/01/2009)

http://searchcloudcomputing.techtarget.com/sDefinition/0,,sid201_gci1170781,00.html (15/05/2009)

http://www.sappiens.com/CASTELLANO/articulos.nsf/Gesti%C3%B3n_Tecnol%C3%B3gica/Software_as_a_Service:_Ventajas_y_Retos_del_software_como_servicio/F12508B79F12D7F9C125750500584569!opendocument (07/11/2008)

2. Arquitectura física de soporte de aplicaciones SaaS.

<http://www.softics.ru/docs/oracle10r2/server.101/b10726/architectures.htm>
(01/03/2005)

<http://www.moratalaz.jazztel.es/pdfs/disponibilidad.pdf> (01/06/2006)

http://download.microsoft.com/download/4/4/c/44c6ac38-fc73-4181-a1c4-9d27f435f546/Windows_Server_TechDay_-_Alta_disponibilidad_en_Infraestructuras_Web.ppt (01/01/2007)

http://searchoracle.techtarget.com/tip/0,289483,sid41_gci1244412,00.html (20/02/2007)

http://www.puntogmx.net/wordpress/wp-content/uploads/2007/04/cluster_ha.pdf (01/04/2007)

3. Arquitectura de software de aplicaciones SaaS.

http://www.ibm.com/developerworks/ibm/library/it-ooch_web/
(01/06/2001)

<http://www.idi.ntnu.no/emner/dif8914/essays/Ziemer-essay2002.pdf>
(28/11/2002)

<http://msdn.microsoft.com/en-us/library/ms954595.aspx> (01/12/2002)

http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseno_de_sistemas_web_modernos.pdf (12/01/2004)

4. Estructuración de un plan de negocios.

http://www.sba.gov/espanol/Biblioteca_en_Linea/plandenegocios.html
(20/02/2008)

<http://www.infomipyme.com/Docs/GT/Offline/inicioempresa/plandenegocio.htm> (01/03/2007)

<http://www.learnthat.com/business/learn/1411/How-to-Create-a-Great-Business/> (07/05/2008)

<http://www.myownbusiness.org/s2/> (01/06/2008)

<http://www.entrepreneur.com/businessplan/> (01/01/2009)

5. Casos de éxito.

http://www.cio.com/article/181450/PeopleSoft_Vets_Born_Again_Can_Two_Legacy_ERP_Guys_Get_IT_Executives_to_Buy_into_On_Demand_Applications_ (08/02/2008)

http://www.cio.com/article/206551/Business_Intelligence_and_On_Demand_The_Perfect_Marriage_ (27/03/2008)

<http://www.oco-inc.com/> (01/01/2009)

<http://www.workday.com/> (01/01/2009)

<http://www.salesforce.com> (01/01/2009)

<http://www.vocus.com/content/index.asp> (01/01/2009)

http://www.cio.com/article/474127/Starbucks_Next_Generation_CIO_Young_Fast_and_In_Control (07/01/2009)

http://www.cio.com/article/477499/GE_CIO_Gets_His_Head_in_the_Cloud_for_New_SaaS_Supply_Chain_App (22/01/2009)

http://advice.cio.com/thomas_wailgum/saas_success_four_recent_deals_show_its_working_for_davids_and_goliaths?page=0%2C0 (23/01/2009)

<http://en.wikipedia.org/wiki/PeopleSoft> (01/05/2009)