Benemérita Universidad Autónoma de Puebla

Una solución basada en SOA (Arquitectura Orientada a Servicios) para un sistema de información altamente escalable

Tesis Profesional

Para obtener el título de:
Ingeniero en Ciencias de la Computación

Presenta
Fausto Leyva Cepeda

Asesor
Dr. Abraham Sánchez López

Puebla, Pue.
Otoño 2016

Esta página fue dejada en blanco intencionalmente

DEDICATORIAS

Dedico este trabajo de tesis a mis padres, que han sido una gran inspiración y fortaleza para seguir adelante en cada paso de mi vida.

AGRADECIMIENTOS

Agradezco a mis padres Fausto y Aurora, que me han sabido guiar y aconsejar en los eventos más cruciales de mi vida, agradezco por brindarme un apoyo incondicional y una comprensión infinita. De corazón, Gracias.

Agradezco a mis hermanas Aurora y Melissa que siempre han sabido como levantarme el ánimo, que siempre han visto por mí, y siempre han tenido tiempo para una broma o una película cuando más la necesito.

A mis vecinos que más que compañeros de piso han sido unos excelentes amigos, con apoyo incondicional para escuchar problemas, además de siempre tener una ingeniosa idea para pasar de un estrés total a un momento memorable.

A mi asesor de tesis el Doc. Abraham, que gracias a su ayuda he podido ver un lado de la computación más profunda y con significado. Gracias

Y por último pero no menos importante a mi novia María José que gracias a ella he podido aventurarme a lugares que no me creía capaz de llegar, ha sido una fuente de inspiración y un apoyo en todo momento. Gracias.

TABLA DE CONTENIDO

CAPÍTULO I

Introducción

"Vivimos en tiempos difíciles. La economía del mercado local está siendo reemplazada por la economía del mercado global, y la gente de marketing manda. Como consecuencia, se tiene que ser rápido y flexible para sobrevivir. Es el renacimiento del Darwinismo: No es la especie más fuerte, o la más inteligente la que sobrevive, es la que mejor se adapta al cambio."

(Josuttis, 2007)

Uno de los trabajos más difíciles de los "Software Manager" de las empresas es mantenerse al día con los rápidos e interminables cambios tecnológicos. Y no sólo el cambio constante de la tecnología hace que sea un trabajo complicado, si no el cambio de la empresa misma puede hacer que un sistema quede obsoleto fácilmente o que no se adapte a todas las necesidades de una empresa sin generar grandes costos de implementación o monetarios, por lo que la forma de desarrollo de software tradicional muchas veces queda fuera de combate.

La clave para una adaptación correcta de una empresa al cambio en un mundo donde los sistemas de información controlan la economía global es precisamente la flexibilidad y facilidad que tenga su sistema de información para crecer y volverse más complejo según las demandas del mercado. Hablamos de sistemas altamente escalables.

Un sistema altamente escalable es un sistema que puede responder fácil y rápidamente a los cambios que se requieran, como en este caso el de una empresa en crecimiento, pero que a su vez no genere altos costos, tanto financieros, como de tiempo de implementación entre otros. Por lo que las formas tradicionales de resolver este tipo de problemas se vuelven obsoletas y/o demasiado costosas.

Las soluciones de sistemas de información basados en servicios web han sido creadas para satisfacer precisamente este tipo de problemáticas, ya que este tipo de tecnología cambiará fundamentalmente la forma en la que se hacen los sistemas internos y concretamente un sistema de información basado en una Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture) es exactamente lo que se necesita para una solución escalable, los beneficios son; expandir las opciones de tecnología a trabajar, hacer el sistema de información más flexible y responsivo, reducir el tiempo de implementación y reducir el costo de mantenimiento.

Y si bien un sistema basado en SOA es una excelente opción para una empresa en crecimiento, esta no sólo puede simplemente comprar un sistema SOA, se tiene que entender y vivir. SOA es un paradigma, otra forma de pensar, es un valioso sistema de arquitectura y diseño. Es básicamente el negocio de la organización. Si bien a lo largo de toda la historia se han hecho promesas que los sistemas de información serán de mayor capacidad con menor tiempo de implementación, con un sistema basado en SOA esto se puede volver una realidad.

I.I Planteamiento del problema.

Los sistemas de información son una herramienta fundamental para los negocios hoy en día, sin embargo, no todos los sistemas están diseñados para todos los posibles cambios que sufren las empresas a lo largo de toda su vida, ya que es difícil predecir los cambios que este necesitará, y la actualización de un sistema puede resultar costoso si no está diseñado adecuadamente, o simplemente la tecnología supera en sí mismo al sistema. Además, por si esto fuera poco, la migración de un sistema de información a otro puede resultar aún más costoso para una empresa, ya que a pesar de lidiar con los problemas diarios, un cambio de sistema puede generar un retraso en el crecimiento.

¿Cómo una empresa podría lidiar con un sistema para que pueda crecer fácilmente, que se pueda adaptar a los cambios y que además pueda generar bajos costos para una empresa?

Durante este proyecto se plantea una forma de pensar diferente a la programación tradicional, donde los sistemas de información tienden a la globalización y adaptación de un mundo de negocios diferente. Un mundo donde "Cloud-Computing" ya es una realidad y que es una tendencia fuerte en la programación.

I.II Sujeto de estudio.

La empresa objetivo tiene como rubro la venta, renta y reparación de equipo de cómputo. La empresa cuenta actualmente con una casa matriz que consta de 3 departamentos; tienda, almacén y administración. Esta empresa al mismo tiempo cuenta con dos sucursales que se administran por si solas y cuentan con un sistema propio; las cuales cuentan solamente con el departamento de tienda. Las sucursales se encuentran en ciudades diferentes, y es imposible conocer la existencia de mercancía o estado de alguna reparación sólo con el sistema.

La empresa actualmente tiene como medio de control un sistema de información para punto de venta local a cada sucursal que no controla todos los departamentos, sólo el departamento de tienda, y por lo que imposibilita la comunicación con otras sucursales y departamentos. Este sistema no cuenta con un estándar en cuanto a la estructura de la información y no puede interactuar con otros sistemas, como lo es por ejemplo el sistema de facturación electrónica. Al ser un sistema local, este no se puede administrar remotamente ni tampoco se puede monitorear, lo cual imposibilita una supervisión eficiente del negocio.

I.III Objetivos generales

El objetivo primordial para este trabajo de tesis es el desarrollo de un sistema de información altamente escalable, que pueda responder fácilmente a los cambios del negocio, (negocio en crecimiento) y que además pueda generar bajos costos de implementación. Además que este trabajo se convierta en una solución real para la empresa.

I.IV Objetivos específicos

Específicamente se pretende desarrollar un sistema basado en una arquitectura SOA, la cual es ideal para sistemas distribuidos y altamente escalables. Este sistema estará destinado a la administración de la empresa, por lo que pretende resolver los problemas de comunicación de la misma, y al mismo tiempo agilizar los procesos de negocios.

I.V Justificación

Este proyecto de investigación ayudara a sentar bases para una plataforma de servicios capaz de generar amistades empresariales por medio de gestión de Servicios Web, lo cual ayudaría al crecimiento y desarrollo de la economía del país. Al mismo tiempo ofrecerá una nueva ventana de negocio para el desarrollo de software que no siempre ha sido propio del mercado mexicano o de sus consumidores, ya que abrirá oportunidades no tan solo para empresas dedicadas a este tipo de desarrollo, sino también a desarrolladores *FreeLance* mexicanos.

I.VI Resumen

Un sistema de información altamente escalable es un sistema flexible que puede rápidamente adaptarse a los cambios requeridos por un mundo donde los sistemas de informática mandan.

Una Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture) es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos.

En este trabajo de tesis se propone la creación de un sistema de información basado en servicios web bajo la Arquitectura Orientada a Servicios (SOA) para una empresa de venta de equipo de cómputo, la cual se encuentra en crecimiento con servicios diversificados y con una dinámica de movimiento que puede generar una interoperabilidad de los componentes del sistema. SOA permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

I.VII Resumen por capítulos

Capítulo I. Descripción general del trabajo de tesis. En este capítulo se hace una introducción al trabajo de tesis, también se exponen las razones por las cuales se desarrolla y como se empezará a trabajar este tema.

Capítulo II. Marco teórico. Este capítulo está dedicado al trabajo de investigación realizado previo al desarrollo del tema. Engloba lo que se conoce del tema, las ventajas y todos los conceptos relacionados a este trabajo.

Capítulo III. Metodología. Durante este capítulo se desarrolla el proceso de realización del proyecto, partiendo de la recopilación de información de la empresa, hasta el desarrollo del Bus de Servicio Empresarial (ESB, siglas del inglés Enterprise Service Bus) para la gestión de Servicios Web de procesos empresariales.

Capítulo IV. Navegación de la solución. Aquí se muestra cómo funciona la solución usando como ejemplo una función de la misma.

Capítulo V. Conclusiones y trabajo futuro. Aquí se encuentran las conclusiones del proyecto y la experiencia obtenida, así como el trabajo futuro planeado para el desarrollo de este.

CAPITULO II

Marco teórico

Dado que la mira central de este análisis estará puesta en una solución con SOA (de las siglas en inglés "Service Oriented Architecture") será necesario plantear algunos parámetros que sirvan de ejes conceptuales sobre los que apoyar la lectura interpretativa de SOA.

II.I SOA

Una Arquitectura Orientada a Servicios es un concepto difícil de definir. Autores a lo largo de toda la historia de SOA no se han podido poner de acuerdo en una definición concreta. Sin embargo, se puede entender como una visión general de lo que el concepto representa.

II.I.I Historia de SOA

El concepto de SOA no es nuevo, ya que este se concibió por primera vez en los últimos años de los 90s (Josuttis, 2007), y a pesar que aún no se acuñaba el término como tal, ya existían acercamientos a los principios básicos (los cuales no han cambiado mucho desde entonces) de lo que hoy conocemos como una arquitectura orientada a servicios.

Decir que SOA fue inventada por alguien es como decir que el Internet tiene un único dueño. Inclusive el termino SOA no fue acuñado por una sola persona, ya que este surgió a principios de los años 2000 como "una nueva idea de hacer software" (Townsend, 2008).

Para comprender mejor el concepto debemos cruzar por las diferentes generaciones que han tenido las aplicaciones distribuidas y ver como SOA entra en juego.

II.I.I.I Primera generación de sistemas distribuidos

La primera generación de aplicaciones distribuidas fueron en su mayoría representados por soluciones centralizadas y basadas en arquitecturas cliente-servidor, o aplicaciones distribuidas punto-a-punto (Figura 2.1). Un ejemplo de estas aplicaciones son aplicaciones de transferencia de archivos, navegador web, audio y video, video conferencias, etc. (Exposito & Codé, 2014)

Alguna de estas aplicaciones implementada a grandes sistemas distribuidos se convierte en una tarea interminable, ya que este tipo de sistemas cuentan con características que hacen que el sistema se convierta en un desastre de proporciones titánicas (no solamente refiriéndose a la tarea en sí, sino al costo que implica).

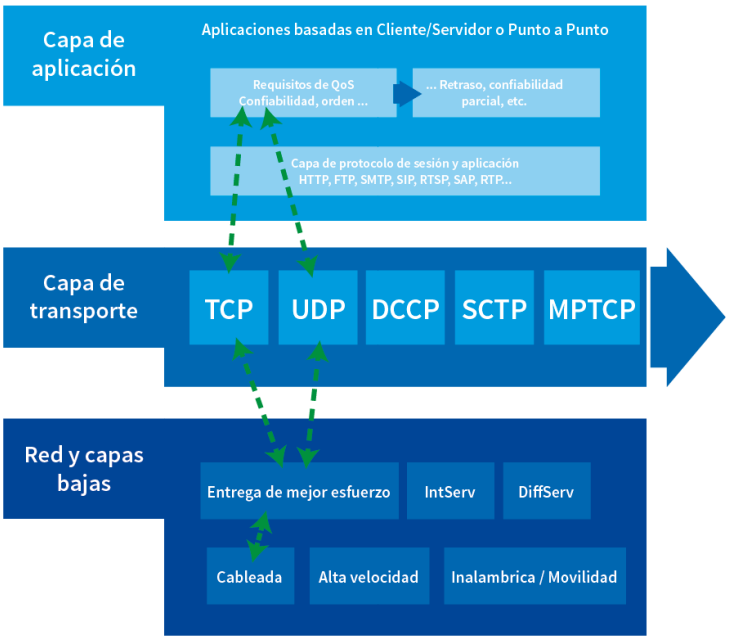


Figura 2.1 Primera generación de sistemas distribuidos. (Exposito & Codé, 2014)

La principal de estas características es que los grandes sistemas distribuidos son por naturaleza sistemas heterogéneos, lo que implica además de muchas cosas que la comunicación entre el sistema se vuelve demasiado complejo. Ilustrando esto con un ejemplo, supongamos que tenemos una empresa que tiene un sistema distribuido, este necesita intercambiar información por tres departamentos, los cuales necesitan consumir información de varios sistemas externos, los cuales trabajan sobre protocolos de comunicación diferentes, por lo tanto, debe existir un medio de comunicación independiente por cada par de proveedores de información y el sistema consumidor de la empresa. Haciendo unos cuantos cálculos podemos entonces decir que la cantidad de canales de comunicación que deben existir para que la empresa tenga una comunicación total seria de la siguiente fórmula matemática: (Exposito & Codé, 2014). El problema está representado en la figura 2.2.

$$Enlaces = n * \frac{(n-1)}{2}$$

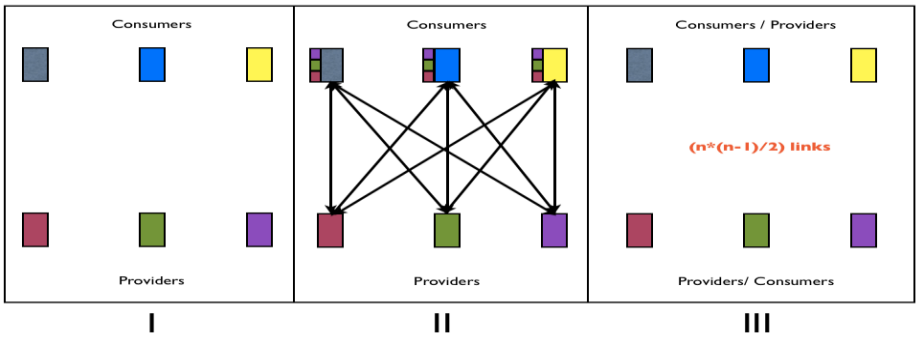


Figura 2.2 Ejemplo de comunicación de un sistema distribuido de la primera generación. (Exposito & Codé, 2014)

Como podemos ver una solución de este estilo es demasiado costosa de implementar tanto en tiempo como en mantenimiento, por lo tanto este sistema imposibilita a la empresa a crecer porque la implementación de más departamentos (consumidores) involucra la creación de nuevos canales de comunicación que son demasiados costosos de manejar, o en el peor de los casos, donde un proveedor se necesita reemplazar por otro, involucra cambiar una gran parte del sistema. El mantenimiento de una solución así es demasiado agotadora y muy propensa a fallos.

Y por si esto fuera poco, hacer una solución de este estilo implica el conocimiento sobre cada una de las plataformas a operar, ya que debido a que cada es una conexión única es necesario conocer todos los ambientes para hacer una solución más eficiente.

De nuevo se ilustrara esta complejidad con un ejemplo: el desarrollo de una aplicación móvil basada en la web. Si el desarrollador tiene que acceder directamente a la API de la capa de transporte, será necesario incluir dentro de la lógica de la aplicación, la selección del protocolo de transporte adecuado para ser utilizado no sólo cuando se inicia la aplicación, sino también en virtud de las diferentes situaciones de contexto red. Esto significa que el desarrollador debe tener en cuenta no sólo las situaciones estáticas, donde la aplicación web puede funcionar conectados a través de Wi-Fi o redes celulares, sino también poner en práctica la lógica adecuada para hacer frente a la entrega al pasar de una red a otra, o incluso cuando el terminal se desconecta y vuelto a conectar.

Por último, para grandes sistemas distribuidos, los cuellos de botella son el suicidio. (Josuttis, 2007)

II.I.I.II Segunda generación de sistemas distribuidos

Con la gran problemática que se encuentran para los grandes sistemas de información distribuidos, surge la siguiente generación, la cual lleva consigo el término middleware como emblema.

"Middleware o lógica de intercambio de información entre aplicaciones es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones y sincronizaciones que son necesarias en los sistemas distribuidos. De esta forma, se provee una solución que mejora la calidad de servicio, así como la seguridad, el envío de mensajes, la actualización del directorio de servicio, etc. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El middleware abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipos de servicios de middleware. Por lo general el middleware del lado cliente está implementado por el Sistema Operativo, el cual posee las bibliotecas que ejecutan todas las funcionalidades para la comunicación a través de la red."

(Bishop & Karne, 2003)

Middleware es un intermediario, un software que ayudara al programador para no lidiar con las muchas conexiones punto a punto para sistemas distribuidos heterogéneos.

"[…] con el fin de hacer frente a esta complejidad, se han invertido importantes esfuerzos para facilitar la implementación en funcionalidades de red que permita a los desarrolladores de aplicaciones distribuidas concentrarse en la lógica de la aplicación. Esto significa, por ejemplo, que los desarrolladores no deberían tener que hacer frente a la selección y configuración del sistema de comunicación o cómo hacer frente a los errores, el retrasos o cualquier otro evento inesperado durante la transmisión de datos entre los componentes distribuidos de la aplicación […]."

(Exposito & Codé, 2014)

Este software, tal como lo dice en el extracto anterior, hace que las ventajas que ofrece para los grandes sistemas distribuidos crezcan de tal forma que la programación ya no sea demasiado compleja o costosa.

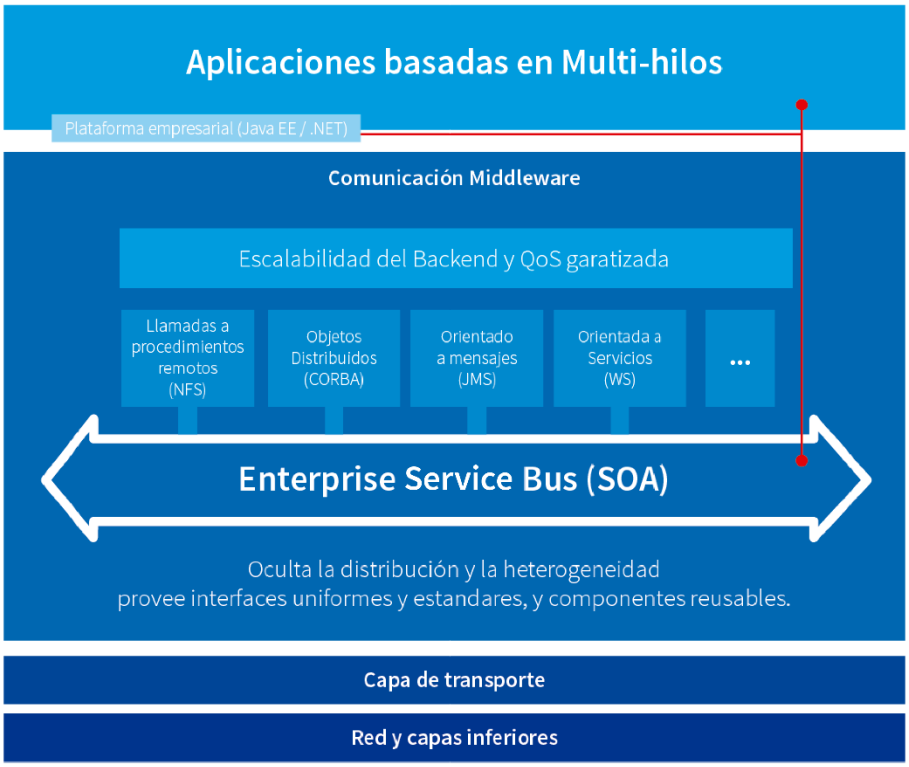


Figura 2.3 Segunda generación de sistemas distribuidos. (Exposito & Codé, 2014)

En la figura 2.3 podemos ver que la segunda generación de sistemas distribuidos se basa en la comprensión de un mediador el cual es llamado Bus de Servicio Empresarial (ESB, siglas del inglés Enterprise Service Bus) el cual es precisamente el middleware que se utiliza para la comunicación entre todas las partes del sistema distribuido, inclusive haciendo uso de tecnologías de comunicación diferentes y paradigmas diferentes (comunicación síncrona y asíncrona).

El uso de un ESB incrementa la disponibilidad, el performance, la escalabilidad, facilita el mantenimiento (por ejemplo, incluyendo mejores servicios) y la evolución de los sistemas distribuidos (por ejemplo, modificando la orquestación de servicios) (Exposito & Codé, 2014). De esta forma nace lo que conocemos como SOA.

II.I.I.III Tercera generación de sistemas distribuidos

Para la tercera generación los grandes sistemas distribuidos que ya cuentan con un conocimiento y utilización de middleware para una efectiva comunicación sobre todas las partes de un sistema heterogéneo presentan otras problemáticas en el diseño de aplicaciones cada vez más grandes.

"En los años 90, Middleware abordaba la necesidad de integración empresarial y funcionalidad web. Hoy en día, ayuda a los negocios a unirse a la economía de las API y aprovechar el ecosistema de socios para crear nuevas oportunidades empresariales."

IBM

Así que para la tercera generación surge un diseño un poco diferente para la comunicación de los grandes sistemas distribuidos. Es la evolución de la plataforma de una Arquitectura Orientada a Servicios presentada por La Nube y el Paradigma de Computo Autónomo.

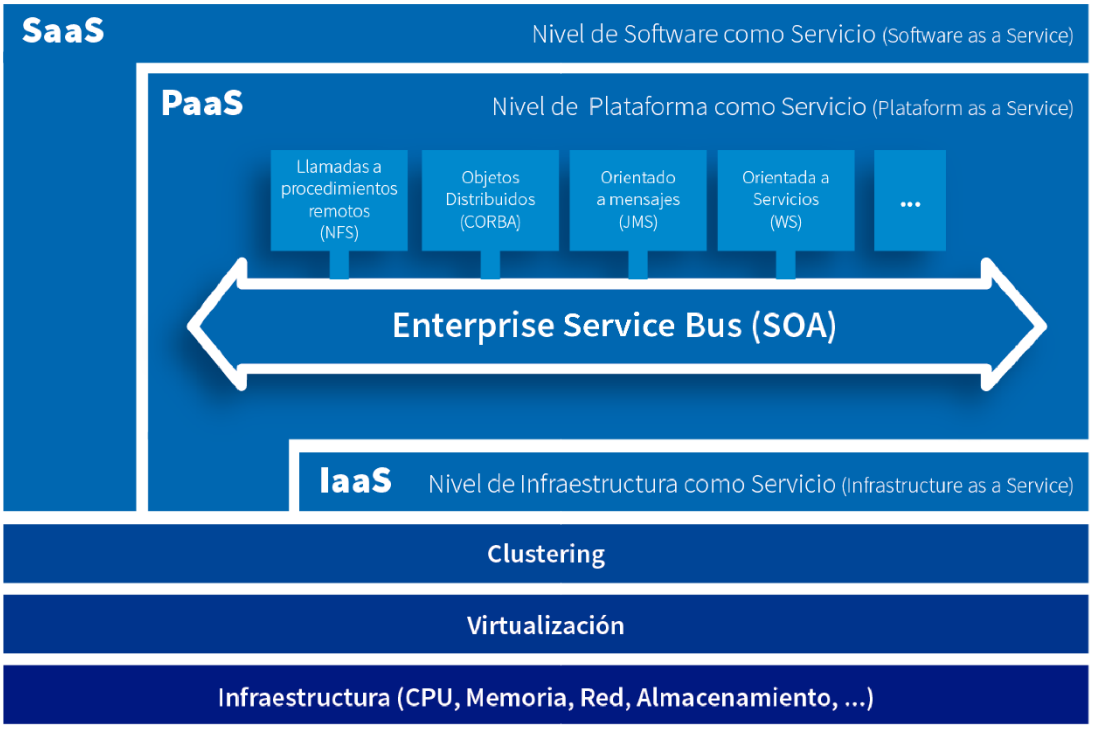


Figura 2.4 Tercera generación de sistemas distribuidos. (Exposito & Codé, 2014)

Como se puede ver en la figura 2.4 una arquitectura SOA se convierte en solo una capa para lo que hoy conocemos como computo en la nube y que nos sirve para una mejor administración tanto de recursos físicos (IaaS – Infraestructura como Servicio) hasta como un mejor manejo de las aplicaciones como servicios (SaaS – Software como Servicio).

II.I.II ¿Qué es SOA?

Entonces, ¿Qué es SOA? Bueno, una definición más simple de SOA podría ser una arquitectura de acoplamiento flexible, diseñado para satisfacer las necesidades del negocio en organización.

A esta condensada definición le pueden faltar algunos elementos importantes como lo son la misma definición de WebService, WSDL, SOAP, entre otros. Sin embargo, muchos analistas y expertos de la industria han confundido el concepto de arquitectura orientada a servicios con implementaciones orientadas a servicios. Esto sólo se ha sumado a la confusión asociada con SOA y sus conceptos relacionados. Esto puede llevar a resultados desastrosos. (Josuttis, 2007)

SOA, a través de la tecnología de servicios Web (WS), ofrece tecnologías middleware que permiten la interoperabilidad y permiten integrar aplicaciones heterogéneas utilizando interfaces basadas en estándares, servicios y un esquema de comunicación flexible. Las aplicaciones integradas que utilizan SOA están unidas ligeramente, pueden intercambiar datos fácilmente y pueden estar involucrados con los procesos de negocio, pero que a su vez no generen dependencia, por lo que permite la reutilización.

Sin embargo, cabe aclarar que el término SOA tampoco está necesariamente ligado Servicios Web, ya que por sí solos no generan una Arquitectura SOA:

"Aunque los Servicios Web no se traducen necesariamente en SOA, y no todos SOA se basa en Servicios Web, la relación entre las dos direcciones de la tecnología es importante y que son mutuamente influyentes […]"

[Erl, Thomas. 2005. Service-Oriented Architecture: Concepts, Technology, and Design. Upper Saddle River, NJ: Prentice Hall.]

II.II ESB

ESB (Enterpise Service Bus, en español Bus de Servicios Empresariales) se han propuesto como la solución de middleware que permite la integración entre los componentes heterogéneos dentro de arquitecturas orientadas a servicios. ESB representan una consolidación de su ancestro EAI(Enterprise Application Integration). Las principales funciones de un producto ESB son los mensajes de enrutamiento entre las aplicaciones, la transformación de mensajes, los mecanismos de entrega de mensajes fiables, mediación y composición entre aplicaciones.

Dentro del paradigma SOA, un bus de servicios empresariales (ESB) actúa como un mediador para facilitar el suministro y el consumo de los servicios. En contraste con las soluciones EAI centralizados, el uso de un ESB aumenta la disponibilidad, fiabilidad, rendimiento y escalabilidad y facilita el mantenimiento (por ejemplo, incluyendo un mejor rendimiento o servicios más adaptados) y evolución (por ejemplo, mediante la inclusión de nuevos servicios o modificar la lógica de orquestación). Con el fin de hacer frente a los inconvenientes de la EAI, se propuso el (ESB) paradigma Enterprise Service Bus. ESB puede agruparse o federada con el fin de hacer frente a la escalabilidad y tolerancia a fallos limitaciones de la EAI. Del mismo modo, los ESB se basan en un enfoque normalizado de mensajería común, por lo tanto hacer frente a las limitaciones de interoperabilidad

del EAI. La mediación entre consumidores y proveedores de servicios se consigue mediante las funcionalidades de enrutamiento de mensajes proporcionados por los ESB. La siguiente figura (Figura 2.5) ilustra el papel desempeñado por los ESB dentro de las arquitecturas distribuidas y orientadas a servicios.

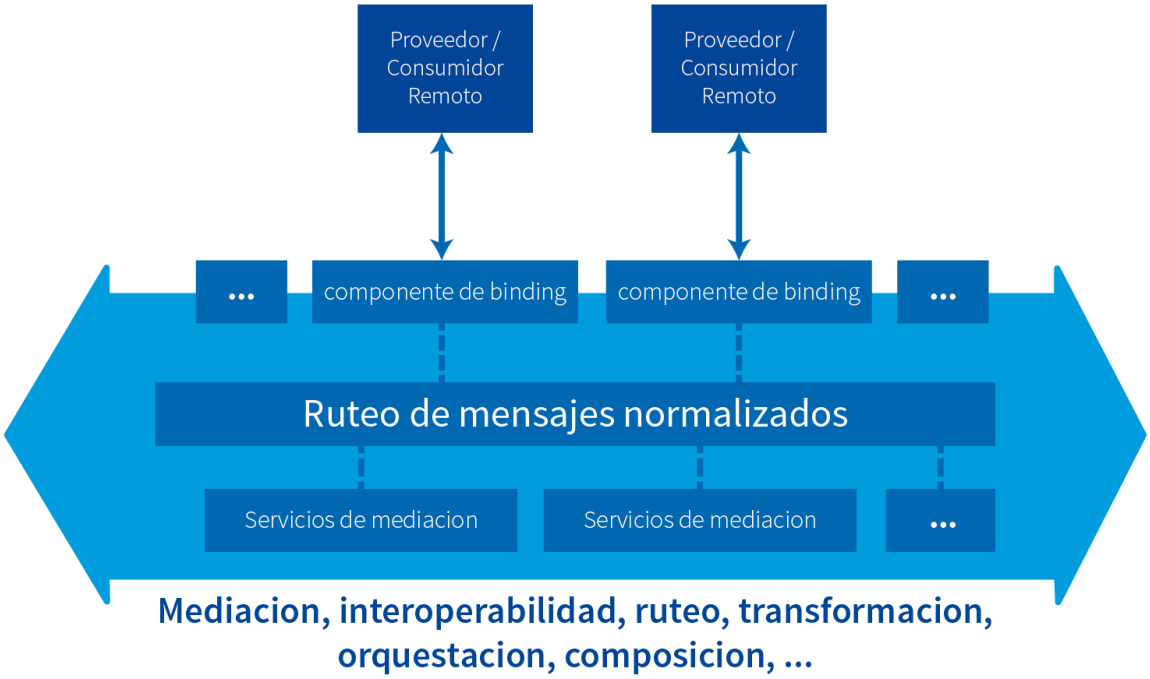


Figura 2.5 Enterprise Service Bus. (Exposito & Codé, 2014)

El ESB es la parte práctica de lo que conocemos como orquestación de servicios. El enfoque que se utilizará para diseñar el bus de servicios es conocido como MIDDLE-OUT, ya que este permite especificar cada uno de los procesos sin necesidad de perder de vista el BACKEND y el FRONTEND.

A su vez, la orquestación de servicios es la organización de los servicios proporcionados por el BACKEND del sistema para dar forma a los procesos empresariales por medio de operaciones de comunicación.

II.II.I Orquestación de servicios contra Coreografía de servicios.

Existe más de una forma de organizar los servicios para que puedan funcionar adecuadamente. Hasta ahora sólo se mencionó la orquestación de servicios como parte primordial del proceso de negocios y de ESB, sin embargo, existe también la coreografía de servicios que consta de otra forma de organizar la comunicación de servicios. A continuación se hacen unas analogías al respecto.

- Empezando por el nombre, una orquestación de servicios hace una analogía a una orquesta, donde los participantes (músicos en escena) reaccionan a un control centralizado (director).
- A diferencia de esto, una coreografía de servicios hace una analogía de una coreografía expuesta por un grupo de bailarines, que no reaccionan a un control centralizado, si no que estos se desenvuelven en escena mediante reglas establecidas (coreografía) y a reacciones externas como la música, el compás, etc.

Por lo tanto, una coreografía de servicios propone una forma más general de ver una propuesta de solución, que consta básicamente en organizar los servicios de tal forma que puedan seguir uno después de otro (mensajes de una sola dirección) sin la necesidad de conocer la estructura de todo el proceso o una forma centralizada, teniendo una idea más "pura" de un sistema distribuido.

Sin embargo, una solución basada en coreografía de servicios puede resultar contraproducente, ya que debido a la forma distribuida en la que los servicios trabajan resulta demasiado difícil poder tener el estado de algún proceso de negocio, al punto que nadie sepa cómo están sucediendo los proceso (lo cual es una pesadilla para las empresas con procesos de negocios largos).

Una solución basada en orquestación de servicios es una forma más organizada de llevar a cabo los procesos de negocios, ya que a pesar de ser una solución con un punto de centralización de servicios para un sistema distribuido, se pueden tener un mejor control y seguimiento a los procesos de negocio.

II.III Servicios Web

Un Servicio Web es un componente software que puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. Permiten exponer y hacer disponibles funcionalidades (servicios) de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos WEB estándar. Cada Servicio Web se responsabiliza de realizar un conjunto de funciones concretas y bien definidas. Servicios Web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos

"Un Servicio Web (Web Service [WS]) es una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre "agentes" software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet."

World Wide Web Consortium [W3C]

Puntos clave:

- Interoperabilidad
- Uso de estándares abiertos
- Mínimo acoplamiento

Interoperabilidad: distintas aplicaciones, en lenguajes de programación diferentes, ejecutadas sobre cualquier plataforma, pueden utilizar los Servicios Web para intercambiar datos.

- La interoperabilidad se consigue mediante el uso de estándares abiertos.
- Servicios Web se asientan sobre protocolos y estándares ya existentes y muy difundidos (HTTP, XML, etc.)
- Uso de protocolos específicos extensibles ⇒ no imponen restricciones sobre las aplicaciones a las que dan acceso ni sobre las tecnologías que las implementan (independencia de lenguaje y de plataforma)

- OASIS y W3C: organizaciones responsables de definir la arquitectura y estándares para los Servicios Web – FJRP, FMBR 2008/09 ccia SCS – 1 Pueden verse como una evolución de los mecanismos RPC
- Uso de protocolos estándar de internet (HTTP, SMTP) como mecanismo para el transporte de los mensajes (invocación, respuesta,...)
- Mensajes intercambiados se encapsulan dentro de mensajes HTTP (oSMTP)
- Evitan problemas con firewalls y filtrado de puertos no privilegiados
- Para la red el tráfico de Servicios Web es trafico HTTP (o SMTP) normal
- Uso de lenguajes basados en XML
- Los mensajes intercambiados son representados en documentos XML

II.IV SOAP

Básicamente SOAP (Simple Object Access Protocol) es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de servicios web", ofreciendo un framework de mensajería básica en el cual los servicios web se pueden construir. Este protocolo está basado en XML y se conforma de tres partes:

- Sobre (envelope): el cual define qué hay en el mensaje y cómo procesarlo
- Conjunto de reglas de codificación para expresar instancias de tipos de datos
- La Convención para representar llamadas a procedimientos y respuestas.
- El protocolo SOAP tiene tres características principales:
- Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
- Neutralidad (SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS).
- Independencia (SOAP permite cualquier modelo de programación).

Como ejemplo de cómo el modelo SOAP pueda ser utilizado, consideraremos un mensaje SOAP que podría ser enviado a un servicio web para realizar la búsqueda de algún precio en una base de datos, indicando para ello los parámetros necesitados en la consulta. El servicio podría retornar un documento en formato XML con el resultado, un ejemplo, precios, localización o características. Teniendo los datos de respuesta en un formato estandarizado procesable (en inglés "parsable"), éste puede ser integrado directamente en un sitio Web o aplicación externa.

La arquitectura del protocolo SOAP está formada por varias capas de especificación: MEP (Message Exchange Patterns) para el formato del mensaje, enlaces subyacentes del protocolo de transporte, el modelo de procesamiento de mensajes, y la capa de extensibilidad del protocolo. SOAP es el sucesor de XML-RPC, a pesar de que toma el transporte y la neutralidad de la interacción, así como el envelope / header / body, de otros modelos (probablemente de WDDX).

Algunas de las Ventajas de SOAP son:

- No está asociado con ningún lenguaje: los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.
- No se encuentra fuertemente asociado a ningún protocolo de transporte: La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- No está atado a ninguna infraestructura de objeto distribuido. La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.
- Aprovecha los estándares existentes en la industria: Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de trasporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.
- Permite la interoperabilidad entre múltiples entornos: SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del BACKEND ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

II.V WSDL

WSDL son las siglas de Web Services Description Language, un formato XML (Xtensible Markup Language) que se utiliza para describir servicios Web. WSDL fue desarrollado principalmente por Microsoft e IBM, y se presentó al W3C por 25 empresas. WSDL describe la interfaz pública a los servicios Web. WSDL se usa en combinación con SOAP y XML Schema. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar. WSDL está en el centro del marco de servicios web, proporcionando una forma común en el que para representar los tipos de datos que se pasa en los mensajes, las operaciones que se realizarán en los mensajes, y el mapeo de los mensajes en la red transporta. (Newcomer, 2002)

II.V XML

XML, siglas en inglés de Xtensible Markup Language ("lenguaje de marcas Extensible"), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. XML es un lenguaje de marcado para documentos que contengan información estructurada.

La información estructurada contiene tanto el contenido (palabras, imágenes, etc.) y alguna indicación de cuál es el papel que desempeña el contenido (por ejemplo, el contenido en un encabezado de sección tiene un significado diferente del contenido de una nota al pie, que significa algo diferente de lo contenido en una figura, título o contenido en una tabla de base de datos, etc.). Casi todos los documentos tienen cierta estructura. (Newcomer, 2002)

XML es un mecanismo para identificar las estructuras en un documento. La especificación XML define una forma estándar de añadir marcas a los documentos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Se dice que un documento XML está bien formado (Well-Formed Document) cuando no tiene errores de sintaxis. Esto incluye los siguientes aspectos:

- Los nombres de los elementos y sus atributos deben estar escritos correctamente.
- Los valores de los atributos deben estar escritos entre comillas dobles o simples.
- Los atributos de un elemento deben separarse con espacios en blanco.
- Se tienen que utilizar referencias a entidades donde sea necesario.
- Tiene que existir un único elemento raíz.
- Todo elemento debe tener un elemento padre, excepto el elemento raíz.
- Todos los elementos deben tener una etiqueta de apertura y otra de cierre.
- Las etiquetas deben estar correctamente anidadas.
- Las instrucciones de proceso deben estar escritas de forma correcta.
- La declaración XML debe estar en la primera línea escrita correctamente.

Por otro lado, se dice que un documento XML es válido (Valid) cuando, además de no tener errores de sintaxis, no incumple ninguna de las normas establecidas en su estructura. Dicha estructura se puede definir utilizando distintos métodos, tales como:

- DTD (Document Type Definition, Definición de Tipo de Documento).
- XML Schema.
- RELAX NG (REgular LAnguage for XML Next Generation).

(Newcomer, 2002)

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Entre las tecnologías XML disponibles se pueden destacar:

XSL : Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo móvil, o un conjunto de páginas de un catálogo, informe o libro.

XPath: Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML.

XLink: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML. Funciona de forma similar a un enlace en una página Web, es decir, funciona como lo haría <a href="">, sólo que a href es un enlace unidireccional. Sin embargo, XLink permite crear vínculos bidireccionales, lo que implica la posibilidad de moverse en dos direcciones. Esto facilita la obtención de información remota como recursos en lugar de simplemente como páginas Web.

XPointer: Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido. XPointer funciona como una sintaxis que apunta a ciertas partes de un documento XML, es como una extensión de XPath. Se utiliza para llegar a ciertas partes de un documento XML. Primero, XLink permite establece el enlace con el recurso XML y luego es XPointer el que va a un punto específico del documento. Su funcionamiento es muy similar al de los identificadores de fragmentos en un documento HTML ya que se añade al final de una URI y después lo que hace es encontrar el lugar especificado en el documento XML. Al ser XPointer una extensión de XPath, XPointer tiene todas las ventajas de XPath y además permite establecer un rango en un documento XML, es decir, con XPointer es posible establecer un punto final y un punto de inicio, lo que incluye todos los elementos XML dentro de esos dos puntos.

XQL: Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web. Finalmente, XQL, lenguaje de consultas, se basa en operadores de búsqueda de un modelo de datos para documentos XML que puede realizar consultas en infinidad de tipos de documentos como son documentos estructurados, colecciones de documentos, bases de datos, estructuras DOM, catálogos, etc. (XML oreilly)

II.V.I XML Schema Definition Language

XML Schema es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el World Wide Web Consortium (W3C). XML Schema es un lenguaje de esquema escrito en XML, basado en la gramática y pensado para proporcionar una

mayor potencia expresiva que la Definición de Tipo de Documento (DTD), menos capaces al describir los documentos a nivel formal.

Los documentos esquema (usualmente con extensión .xsd de XML Schema Definition (XSD)) se concibieron como una alternativa a las DTD, más complejas, intentando superar sus puntos débiles y buscar nuevas capacidades a la hora de definir estructuras para documentos XML. El principal aporte de XML Schema es el gran número de tipos de datos que incorpora. De esta manera, XML Schema aumenta las posibilidades y funcionalidades de aplicaciones de procesado de datos, incluyendo tipos de datos complejos como fechas, números y cadenas.

Un ejemplo de la estructura de un documento esquema vacío sería el siguiente:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="0.1" xml:lang="es">

 </xsd:schema>
```

II.VI Modelado SOA

SOA es un paradigma, por lo cual no incluye una arquitectura concreta. Sin embargo una parte fundamental de SOA es la infraestructura que aloja los servicios y que forma parte de la vista general, esta es llamada "Bus de Servicios Empresariales" (ESB, de la siglas en ingles de Enterprise Service Bus").

Ahora la cuestión recae en cómo identificar los servicios en estos procesos de negocios. Para lo cual se toma un enfoque ágil, el cual pueden lidiar tanto con el desarrollo de los servicios como con la integración de estos con el backend, dando como resultado el diseño del Bus de Servicios (ESB).

II.VI.I Enfoque ágil para el diseño de composición de servicios

El enfoque ágil o también llamado enfoque "middle-out" es para el diseño de servicios engloba una mezcla de dos enfoques diferentes para el diseño:

- Top-down: Se descompone el proceso de negocio, hasta las partes más fundamentales del proceso, hasta llegar al nivel básico de servicios.
- Botton-up: Se parte desde los servicios básicos diseñando una estructura creciente, centrándose en el backend hasta llegar al proceso de negocio.

(Josuttis, 2007)

Este tipo de enfoque hace que el diseño para SOA sea mucho más fácil, ya que hace que la orquestación de servicios se conforme de una forma más natural. A continuación e muestra un ejemplo de este enfoque (Figura 2.6).

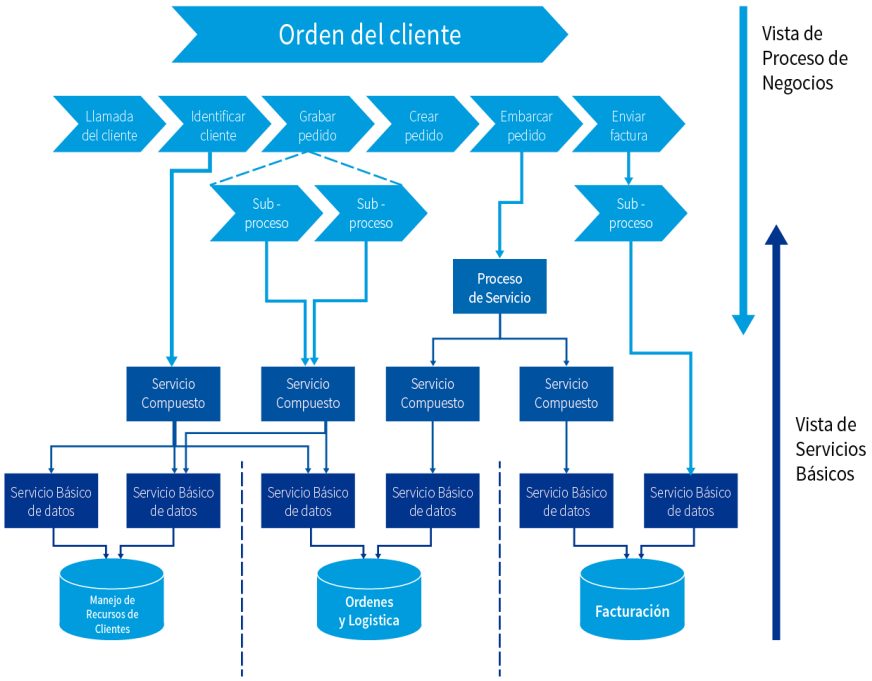


Figura 2.6 Ejemplo de orquestación servicios con enfoque ágil

II.VII Modelado de negocios (UML)

UML es una notación estándar para el modelado visual de los sistemas de software y se utiliza ampliamente en el desarrollo de sistemas orientados a objetos en todo el mundo. Una de las razones del éxito generalizado de UML es que es independiente y metodología se puede utilizar para expresar los resultados de los proyectos de diferente tamaño y propósito. En particular, UML se puede utilizar para el modelado de negocios y la modelización de otros sistemas que no son de software. De hecho, el modelado de negocios con UML puede ser considerado una extensión de la disciplina de modelado basado en UML, relacionados con el modelado del sistema utilizando la misma notación. Como se ha subrayado por muchos autores y profesionales, utilizando el mismo lenguaje visual a lo largo de todo el ciclo de vida del desarrollo del proyecto proporciona una gran ventaja para todos los interesados en el proyecto haciendo que el proceso de desarrollo sea más eficiente. Al tener los analistas de negocios y desarrolladores de sistemas que utilizan los mismos conceptos de modelado, el riesgo de costosos errores relacionados con diferente comprensión de los conceptos de metodología se mitiga significativamente. Sin embargo, como se ha señalado por varios especialistas en metodología, "El uso de modelos similares construye para varios propósitos pueden llevar a la confusión" (Anonimo, 2007).

Formalmente: "El constructor del caso de uso se utiliza para definir el comportamiento de un sistema u otra entidad semántica sin revelar la estructura interna de la entidad. Cada caso de uso especifica una secuencia de acciones, incluyendo variantes, que la entidad puede llevar a cabo, interactuando con los actores de la entidad". (Anonimo, 2007)

La notación UML del actor y casos de uso asociado se muestra en la siguiente figura (Figura 2.7). Para diferenciar los símbolos utilizados para sistemas o modelos de negocios UML utiliza extensiones llamados estereotipos que suelen definirse por el modelador: Actor, estereotipado como <<actor de negocio>> puede convertirse en un actor de negocios y casos de uso, estereotipado como << caso de uso de negocio> > puede convertirse en un caso de uso de negocio, como se muestra en la misma figura 2.7.

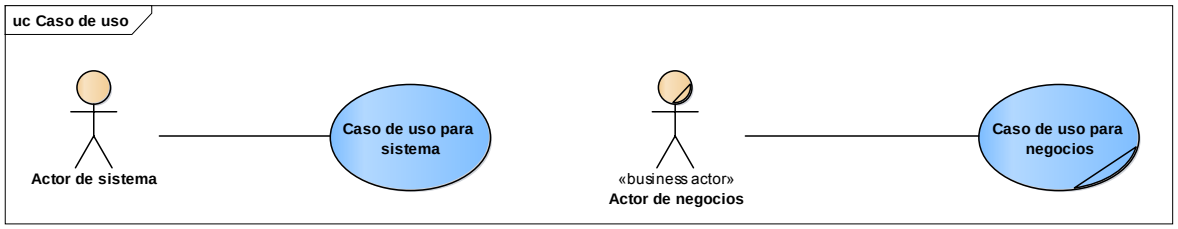


Figura 2.7 Modelado de negocios con UML

II.VIII Modelado de negocios (BPM)

Los servicios son parte del proceso de negocios, por esta razón se debe pensar en los procesos de negocio para poder traer servicios al juego. (Josuttis, 2007)

Varios métodos y varias herramientas han sido desarrolladas para describir los procesos de negocio y flujos de trabajo. Estos métodos y herramientas difieren en sus construcciones, notación, facilidad de uso y otros aspectos. A menudo se emplean diferentes métodos en diferentes etapas del proceso de desarrollo.

La integración de BPM (de las siglas en ingles de "Business Process Model") sirve principalmente para modelar todos los rubros del negocio sin la necesidad de especificar las estructuras de datos, y que a su vez resulta más fácil para la comprensión del negocio en sí. Con esto no solo se puede simplificar el negocio, sino que así también es más fácil la integración de servicios como parte del proceso de negocios. (Van der Aalst, Desel, & Oberweis, 2000). Un ejemplo de este modelado está descrito en al figura 2.7

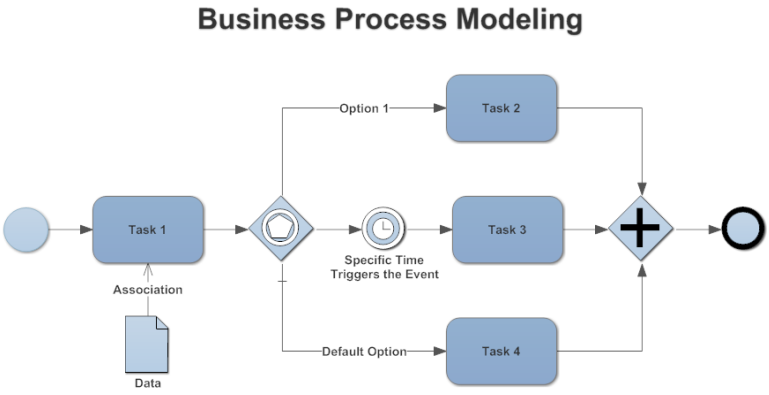


Figura 2.7 Ejemplo de modelado de negocios con BPM

CAPÍTULO III

Metodología

Durante este capítulo se explica el proceso de desarrollo de la solución propuesta, partiendo de la recopilación de información de la empresa, hasta el desarrollo de la solución misma.

III.I Recolección de datos

La empresa cuenta con diferentes departamentos, estos tienen necesidades específicas, los cuales se deben comunicar para realizar tareas en conjunto.

El área de tienda es la parte más importante y más extensa de toda la empresa, en esta se llevan a cabo la mayoría de las actividades relacionadas con el cliente, y es la que controla el flujo de ingresos. En este departamento se encuentran toda la atención a clientes y es donde se debe tener un especial cuidado en el control de todos los activos. Se necesita de un monitoreo constante y de completa disponibilidad. Figura 3.1

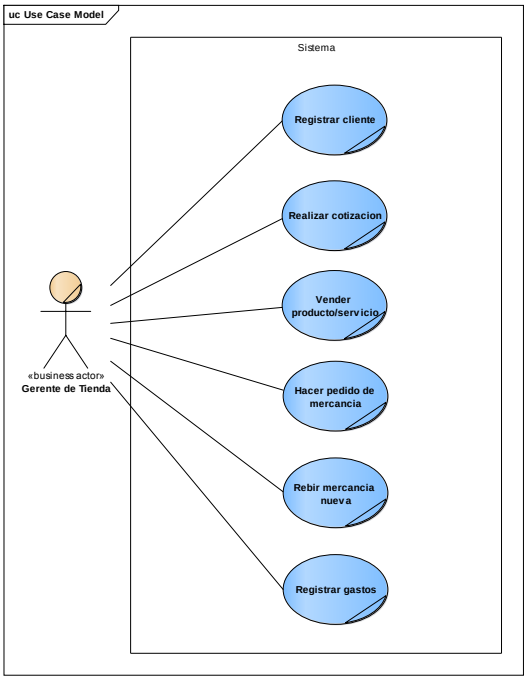


Figura 3.1 Diagrama de casos de usos para el departamento de tienda

El departamento de almacén esencialmente se encarga de repartir mercancía a las tiendas, incluyendo la misma de la casa matriz, sin embargo también se encarga que el catálogo de productos este actualizado y completo; características de los productos, precios actualizados, descripción correcta, fotografías, fichas técnicas, etc.

Este departamento inicia el ciclo de vida de la mercancía, aquí se captura e ingresa a la tienda por medio de facturas de compra. Por lo tanto las actividades que se realizan en este departamento se muestran en la Figura 3.2

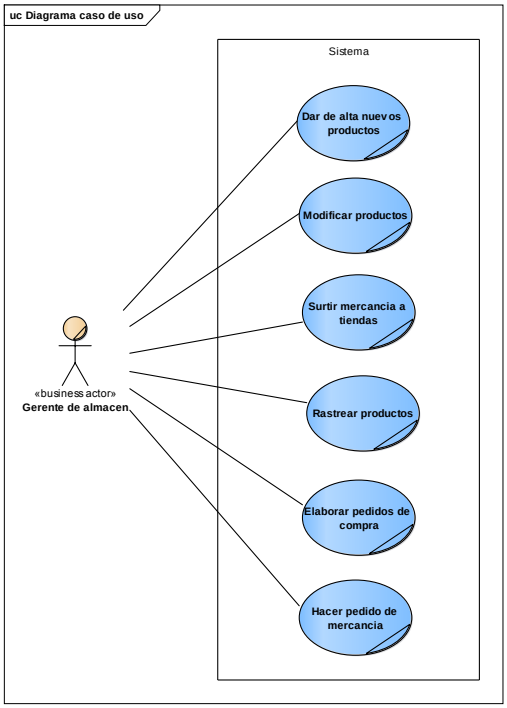


Figura 3.2 Diagrama de casos de usos para el departamento de almacén

Por último, el departamento de administración es un departamento más versátil, y no está activo todo el tiempo. Incluyen diferentes actividades, como lo es la compra de mercancía nueva, el manejo de gastos, cotizaciones importantes, supervisiones a sucursales, entre otros. Figura 3.3

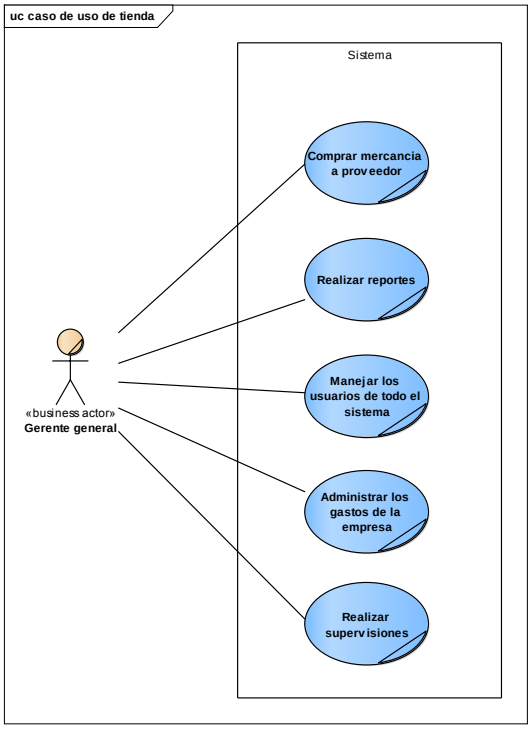


Figura 3.3 Diagrama de casos de usos para el departamento de administración

III.II Desarrollo de casos de uso

Los casos de uso expuestos fueron extraídos de entrevistas realizadas en la empresa, a base de observaciones realizadas durante el proceso empresarial y basadas en el documento empresarial "Manual de políticas y procedimientos" (véase en Bibliografía).

III.II.I Registro de clientes

El usuario podrá registrar un nuevo cliente, el cual quedará registrado en una base de datos global, existen diversos tipos de clientes, por default se genera un cliente básico, para escalar se necesitan requisitos especiales.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Debe contar con los datos personales del cliente a ingresar, dependiendo del tipo de cliente son los datos necesarios.

Flujo de eventos principal

1. El usuario deberá seleccionar el tipo de cliente a registrar, básico por default.
2. Una vez seleccionado, se debe capturar los datos solicitados por el sistema, los cuales pueden variar según el cliente.
3. El sistema valida los datos ingresados por el usuario.
4. Cuando los datos ya estén registrados, se necesita confirmar los datos del cliente.
5. Una vez realizada la confirmación finaliza el flujo de eventos para el caso de uso.

Flujo de eventos alternativos

1. En el paso 3, en caso que la confirmación sea negativa, se podrá regresar a la edición del cliente o se podrá cancelar toda la operación.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en los campos correspondientes al cliente, inténtelo de nuevo.

III.II.II Cotizaciones.

El usuario podrá registrar productos que, no importando su existencia, puedan ser puestos en un formato de cotización. Este formato de cotización contendrá el precio de los productos solicitados, un cliente (puede ser cliente de mostrador), vigencia, folio y la persona que lo expide.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación.

Flujo de eventos principal

1. El usuario deberá seleccionar un cliente para realizar una cotización.
2. Una vez seleccionado un cliente el usuario podrá seleccionar algún(os) producto(s) o servicio(s), los cuales los podrá buscar con o sin existencia.
3. Una vez seleccionados los productos, el usuario elegirá el medio de envío al cliente, el cual puede ser electrónico o impreso.
4. Una vez realizada la acción de envío finaliza el flujo de eventos para el caso de uso.

Flujo de eventos alternativos

1. En el paso 1, en caso que el cliente no se encuentre registrado en la base de datos de la empresa, éste se podrá registrar. Véase el caso de uso de "registrar nuevo cliente".
2. En el paso 2, si el nivel de usuario es adecuado, se podrá registrar un nuevo producto, o ingresar la información del producto de forma manual y temporal.
3. En el paso 3, en caso de generar algún error en alguna opción de las opciones de envío, regresara a la opción de cambiar la forma de envío, o cancelar la operación.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en la impresión, se solicita al usuario volver a intentar la impresión, o elegir imprimir luego.
- Error en el envío de información en el correo electrónico, se solicita al usuario intentar más tarde o cambiar de opción de envió.

III.II.III Venta de productos/servicios a clientes.

En este caso de uso el usuario registrará una venta, esta puede ser tanto de mercancía como de servicios. El precio de cada uno de los rubros de la venta puede variar dependiendo del tipo de cliente al que se le cargue la venta.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Deberá contar con los datos correspondientes del producto, servicio y cliente.

Flujo de eventos principal

1. El usuario deberá seleccionar el cliente para la correspondiente venta, el cliente por default es el "cliente mostrador".
2. Una vez seleccionado el cliente, el usuario registrará uno por uno los rubros de la venta, en este caso el sistema si validará que sea un rubro válido.

3. Cuando se capturen todos los productos, el usuario registrará el vendedor que efectuó la venta.
4. Después se registrará la venta, esto sólo se logrará después de ingresada la forma de pago del cliente.
5. Una vez realizado el cobro de la venta, termina el flujo de eventos para este caso de uso.

Flujo de eventos alternativos

1. En el paso 1, si el cliente al que se necesita hacer una venta no está disponible aun en la base de datos del sistema, se podrá registrar. Véase el caso de uso "registrar nuevo cliente".
2. En el paso 2, si el producto que desea comprar el cliente no se encuentra disponible en la sucursal donde se efectúa la compra, el usuario podrá verificar la existencia de otras sucursales y hacer un pedido de este producto.
3. En el paso 4, si la forma de pago es cheque, la mercancía no podrá salir de la sucursal hasta que sea salvo buen cobro, la cual se tiene que confirmar con el departamento de contabilidad, por lo que la mercancía no se podrá dar de baja del inventario hasta que se confirme el cobro.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en la impresión, se solicita al usuario volver a intentar la impresión, o elegir "imprimir luego".
- Producto no existe en el catálogo, intente con otro código.
- Producto no encontrado en existencia, intente en otra sucursal.

III.II.IV Pedidos de mercancía.

El usuario podrá registrar productos que no se encuentren en existencia o que nunca existieran en la sucursal.

Pre-condiciones

El usuario deberá contar la información necesaria para ingresar en el sistema.

Flujo de eventos principal

1. El usuario seleccionará el cliente que solicitó la mercancía; cliente mostrador por default.
2. Una vez seleccionado el cliente, se debe capturar el producto que solicita, éste se puede seleccionar del catálogo de productos o se puede capturar de forma manual.
3. El usuario puede elegir ser notificado o no cuando llegue el producto deseado.
4. Cuando los datos ya estén registrados, se necesita confirmar los datos del cliente.
5. Una vez realizada la confirmación finaliza el flujo de eventos para el caso de uso.

Flujo de eventos alternativos

1. En el paso 4, en caso que la confirmación sea negativa, se podrá regresar a la edición del cliente.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.II.V Recepción de envíos de almacén.

En este caso de uso el usuario registrará la entrada de mercancía a la sucursal, cotejando la mercancía recibida contra la indicada en un formato de entrega de mercancía, qué afectará al inventario de la sucursal correspondiente.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Se deberá contar con el formato de recepción de mercancía. La mercancía deberá ya estar cotejada con el formato.

Flujo de eventos principal

1. El usuario buscará el número de pedido correspondiente al formato de despacho de mercancía.
2. El usuario confirmará en el sistema que recibió el pedido, cambiando su estatus de "en trasporte" a "entregado" por medio de una clave de seguridad incluida en el formato de entrega de mercancía.
3. Una vez confirmado el pedido, concluirá el flujo de eventos para este caso de uso.

Flujo de eventos alternativos

1. En el paso 2, en caso que la confirmación sea negativa, se cancelará toda la operación.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en la confirmación del pedido, se solicita al usuario volver a intentar.

III.II.VI Alta de gastos.

En este caso de uso el usuario registrará la salida de efectivo de la sucursal.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Se deberá contar la información necesaria del gasto realizado.

Flujo de eventos principal

1. El usuario registrará los datos del gasto, tomando una especial consideración al monto del gasto, el concepto y si cuenta con una factura de compra o un vale foliado.
2. Una vez capturado los datos se registrará en el sistema.
3. Una vez confirmado el gasto, concluirá el flujo de eventos para este caso de uso.

Flujo de eventos alternativos

1. En el paso 2, si la información no es suficiente no se podrá registrar el gasto y se cancelará la operación.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en el llenado de la información del gasto, se solicita al usuario volver a intentar.

III.II.VII Corte de caja

En este caso de uso el sistema brindará al usuario un resumen de todas las actividades realizadas durante el día. Éste puede ser un resumen de día donde sólo se muestran los totales o uno que sea detallado mostrando todas las actividades.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación.

Flujo de eventos principal

1. El usuario deberá seleccionar el tipo de resumen del día que quiera mostrar.
2. Se finaliza el flujo de eventos con la impresión del resumen.

Flujo de eventos alternativos

- Ninguno

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en la impresión, se solicita al usuario volver a intentar la impresión, o elegir “imprimir luego”.

III.II.VIII Alta de productos al sistema por medio de una factura.

En este caso de uso, el usuario deberá de dar de alta un nuevo producto enviado por algún proveedor en la base de datos de la empresa, éste se dará de alta por medio de una factura proporcionada por el proveedor. Este proceso dará inicio al ciclo de vida de un producto.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Debe contar con la información necesaria de la factura para ingresarla en el sistema.

Flujo de eventos principal

1. El usuario registrará registrara el número de factura y el proveedor.
2. El usuario registrará registrara todo los productos correspondientes a dicha factura, una a la vez.
3. Mientras ingresa los productos estos se le deben indicar un número de serie a cada producto en caso de que el mismo producto lo tenga.
4. Una vez ingresados todos los productos se le asignará un precio de venta.
5. Cuando el usuario finalice, deberá confirmar que los datos sean correctos.
6. Después de confirmar satisfactoriamente el ingreso de los datos, finaliza el flujo de eventos para registrar una factura de compra de mercancía.

Flujo de eventos alternativos

1. En el paso 2, en caso que el producto que se intenta registrar no está dado de alta éste se podrá registrar antes de seguir con la operación; revisar el caso de uso “Alta/modificación de descripción de cada producto”
2. En el paso 3, si el producto no cuenta con un número de serie este se podrá ignorar y seguir con la operación.
3. En el paso 4, puede no asignarse un precio de salida, ya que éste se puede modificar antes de salir de almacén.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Producto no existe en el catálogo, intente con otro código.
- El número de serie ya existe en el catálogo, intente con otro código.

III.II.IX Alta/Modificación de descripción de cada producto.

En este caso de uso el usuario deberá de dar de alta un nuevo producto en la base de datos de la empresa, este producto solo contara con las características de este, no incluirá existencia ni número de serie, pero si contará con un campo de “descontinuado” y el precio.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Además deberá contar con la información necesaria para registrar un producto en especial, como lo es una imagen o un archivo de ficha técnica (opcional).

Flujo de eventos principal

1. El usuario registrará todos los campos que sean necesarios para la correcta alta de mercancía.
2. Una vez ingresados los datos a mano deberá subir los archivos de imágenes y ficha técnica (opcional).
3. El sistema valida los datos ingresados por el usuario.
4. Cuando el usuario finalice, deberá confirmar que los datos sean correctos.
5. Después de confirmar satisfactoriamente el producto nuevo, finaliza el flujo de eventos para registrar un nuevo producto.

Flujo de eventos alternativos

1. En el paso 2, en caso de no contar con una ficha técnica, podrá registrar una ficha técnica a mano.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en el tipo de archivo subido, no corresponde a un archivo PDF.
- Error en el tipo de archivo subido, no corresponde a un archivo de imagen.
- Error en los campos correspondientes al producto, inténtelo de nuevo.

III.II.X Envío de mercancía a tiendas.

En este caso de uso el usuario deberá seleccionar mercancía del inventario de almacén y enviarla hacia otra sucursal. El sistema entregara un formato que contendrá un resumen de la operación y una clave llave para ingresar la mercancía en la sucursal correspondiente.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación.

Flujo de eventos principal

1. El usuario seleccionará los productos del inventario de almacén y los registrará para su traslado.
2. Cada producto deberá ser registrado con una opción para actualización de precio
3. Una vez finalizada la selección de productos, el usuario confirmará esta distribución.

4. Cuando se confirme el sistema mandará a imprimir un formato de entrega de mercancía, el cual se utilizará para confirmar las entregas de mercancía al transportista y a la sucursal correspondiente.

Flujo de eventos alternativos

1. En el paso 2, en caso que la confirmación sea negativa, se podrá regresar a la edición de la distribución de la mercancía o cancelar toda la operación.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en la impresión, se solicita al usuario volver a intentar la impresión, o elegir imprimir luego.

III.II.XI Rastreo/historial de productos.

Para este caso de uso, mientras exista algún producto con número de serie en el que se desconoce el historial que éste tiene, éste se puede rastrear para poder localizar su procedencia y su supuesta localización, así como si se ha vendido, si ha entrado a garantía o reparación.

Pre-condiciones

El usuario deberá autentificarse en el sistema con el nivel de seguridad correspondiente para poder realizar esta operación. Deberá contar con los datos del producto a rastrear.

Flujo de eventos principal

1. El usuario deberá ingresar el número de serie del producto a rastrear.
2. El sistema arrojará en pantalla el historial del producto, aparecerá en forma de lista según su historial en el tiempo
3. El usuario podrá elegir imprimir la información o enviarla por correo electrónico.
4. Fin del flujo de datos

Flujo de eventos alternativos

1. En el paso 1 el usuario podrá elegir el número de serie de una lista de todos los productos del sistema
2. En el paso 2 se podrá filtrar por el tipo de información que se desea desplegar, como compra, venta y reparación/garantía.
3. En el paso 3 el usuario podrá ignorar este paso, o hacer ambas acciones.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en la impresión, se solicita al usuario volver a intentar la impresión, o elegir imprimir luego.

- Error en el envío de información en el correo electrónico, se solicita al usuario intentar más tarde.
- Producto no existe en el catálogo, intente con otro código.

III.II.XII Revisión de pedidos hechos por sucursal, incluyendo la mercancía pedida por clientes.

En este caso de uso el usuario del departamento de compras obtiene del sistema una lista de productos que se necesitan comprar, esta lista en sí misma es una sugerencia de compra que se basa en información de otras partes del mismo sistema. Con esta información el usuario del departamento de compras podrá generar una orden de compra.

Pre-condiciones

El usuario debe autentificarse en el sistema como parte del departamento de compras.

Flujo de eventos principal

1. El usuario hace una consulta la lista de productos pendientes por comprar.
2. El sistema busca la lista de mercancía vendida, ordenándola por prioridad de venta, por precio y existencia, dándole un peso a cada producto ayudando a priorizar la compra.
3. El sistema busca la lista de productos solicitados por los clientes y que la tienda nunca ha tenido, agregándolos a una lista de sugerencia de compra.
4. El sistema busca la lista de productos solicitados por los clientes y que la tienda no tiene en existencia, por lo que agrega aún más peso a estos productos.
5. El sistema busca la lista de productos que ya están pedidos en el sistema, por lo que a estos productos se le agrega la prioridad máxima.
6. El sistema devuelve al usuario de compras una lista de productos pendientes de compra.
7. Finaliza el flujo de consultar lista de compras.

Flujo de eventos alternativos

1. El usuario puede elegir sólo consultar la lista o agregar algunos productos a una orden de compra.
2. El usuario podrá visualizar la cantidad de productos actuales que existen en cada sucursal antes de agregarlos a una orden de compra.
3. El usuario podrá marcar algunos productos como descontinuados, haciendo que ya no aparezcan en la siguiente sugerencia de compra.
4. El usuario podrá ver la lista de proveedores de donde se obtuvo ese producto.
5. El usuario podrá ver la información detallada del producto.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.II.XIII Elaboración de pedidos de compra.

En este caso de uso el usuario del departamento de compras genera una orden de compra que será autorizada o modificada por el gerente de compras.

Pre-condiciones

El usuario debe autentificarse en el sistema como usuario del departamento de compras.

Flujo de eventos principal

1. El usuario deberá seleccionar que proveedor deberá surtir esa orden de compra
2. El usuario deberá agregar productos, ya sea de la lista de sugerencia de compra proporcionada por el mismo sistema o de forma manual.
3. El usuario manda la orden de compra al gerente de compras y este la autoriza.
4. Termina el flujo de generar orden de compra y continuara con el proceso de compra.

Flujo de eventos alternativos

1. El usuario podrá visualizar la cantidad de productos actuales que existen en cada sucursal antes de agregarlos a una orden de compra.
2. El usuario podrá marcar algunos productos como descontinuados, haciendo que ya no aparezcan en la siguiente sugerencia de compra.
3. El usuario podrá ver la lista de proveedores de donde se obtuvo algún producto en especial.
4. El usuario podrá ver la información detallada del producto.
5. El gerente rechazo la orden de compra, se debe modificar.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.II.XIV Compra de mercancía a proveedores.

En este caso de uso el usuario deberá de dar de alta una nueva factura correspondiente a alguna compra realizada, puede corresponder a una orden de compra o no. El actor que realice esta operación puede ser del departamento de compras, contabilidad o almacén.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Deberá contar con dos documentos, un archivo XML que corresponda al archivo fiscal de la factura y un documento PDF que represente la factura impresa.

Flujo de eventos principal

1. El usuario deberá subir los dos archivos requeridos al sistema seleccionándolos en la PC.

2. El usuario deberá esperar a que se suban los archivos correspondientes, y confirmar que sean los correctos.
3. Para confirmar que sean los correctos se mostrara una vista previa de la factura y una vista decodificada del archivo XML.
4. Deberá seleccionar de forma manual el proveedor de la factura, y que tipo de factura es, esta puede ser de mercancía o para uso de la empresa.
5. Fin del flujo de eventos para el caso de uso.

Flujo de eventos alternativos

1. En el paso 2, si la factura no es la correspondiente, se podrá volver a seleccionar el archivo correcto o cancelar la operación.
2. En el paso 4, si el proveedor no se encuentra registrado en el sistema, se deberá registrar y seleccionarlo después.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
- Error en el tipo de archivo subido, no corresponde a un archivo XML.
- Error en el tipo de archivo subido, no corresponde a un archivo PDF.

III.II.XV Reportes de ventas, de compras, gastos, etc.

En este caso de uso el usuario visualizará un reporte del estado actual o pasado del rubro que se necesite, el cual puede ser de ventas, compras, gastos, cumplimiento de meta, balance general y reparaciones y servicios. Estos mismos rubros podrán ser visualizados por sucursal o sucursales, y además variable en el tiempo.

Pre-condiciones

El usuario deberá tener el nivel de seguridad correspondiente para poder acceder a este módulo del sistema.

Flujo de eventos principal

1. El usuario deberá elegir la forma de visualizar la información. Por predeterminado se mostrará el balance general de la empresa en el mes actual.
2. El usuario elegirá si desea sólo una visualización de la información o una impresión de la misma.

Flujo de eventos alternativos

1. En el punto 2. Si la impresora no está disponible en ese momento, el usuario podrá imprimir este reporte cuando la impresora esté disponible.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.II.XVI Control de usuarios del sistema.

En este caso de uso el usuario podrá hacer uso del manejo de los usuarios que tiene acceso al sistema, lo cual incluye el dar de alta nuevos usuarios, modificar la información de alguno, cambiar permisos y habilitar o deshabilitar el acceso de estos.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Deberá contar con la información necesaria para poder manejar a cada uno de los usuarios.

Flujo de eventos principal

1. El usuario deberá seleccionar la acción que desee para cada usuario.
2. Una vez introducida la información necesaria para cada una de las opciones se verificará la información y verificar los cambios.
3. Fin del flujo de eventos para el caso de uso.

Flujo de eventos alternativos

1. En el paso 1, si la opción requerida es alta de un usuario se deberá capturar toda la información necesaria y habilitar un usuario y contraseña.
2. En el paso 1, si la opción requerida es modificar la información de usuario se deberá capturar toda la información necesaria.
3. En el paso 1, si la opción requerida es el cambio en el nivel de seguridad un usuario se deberá seleccionar el nivel de seguridad correspondiente al usuario.
4. En el paso 1, si la opción requerida es la inhabilitación del acceso de un usuario al sistema se deberá seleccionar al usuario y confirmar la acción.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.II.XVII Supervisión y calificaciones.

En este caso de uso el usuario podrá realizar el proceso de supervisión a cada una de las sucursales y departamentos por separado, y además poder consultar la calificación de cada una de las supervisiones hechas anteriormente.

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación.

Flujo de eventos principal

1. El usuario seleccionará a que sucursal y departamento se hará la supervisión.

2. El usuario podrá evaluar dicho departamento con un formato predefinido.
3. Una vez terminado de evaluar la supervisión el usuario podrá ver la calificación actual que arroja el sistema.
4. Fin del flujo de eventos para el caso de uso.

Flujo de eventos alternativos

1. En el paso 2, si por alguna razón no se termina la captura de un formato se podrá guardar un parcial de dicho formato y recuperar antes de dos días para seguir evaluando.
2. En el paso 1, se puede ver la calificación de alguna supervisión pasada sin necesidad de hacer una evaluación en ese momento.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.II.XVIII Administración de gastos.

En este caso de uso el usuario podrá manejar de forma fácil el alta de gastos generados por la empresa

Pre-condiciones

El usuario debe autentificarse en el sistema con el nivel de seguridad necesario para realizar esta operación. Deberá contar con la información necesaria para el manejo de gastos.

Flujo de eventos principal

1. El usuario seleccionara un nuevo gasto.
2. Introducirá la información necesaria para poder dar de alta un nuevo gasto.
3. El usuario confirmará la información y finalizara el flujo de eventos.

Flujo de eventos alternativos

1. No aplica.

Encadenamiento de errores

- Error al conectarse al servidor, se solicita al usuario volver a intentarlo.

III.III Modelado de los procesos de negocios

Para continuar con el proceso de modelado debemos definir los procesos empresariales que definen a todo el negocio, los cuales se basan prácticamente en un juego de venta y compra de mercancía.

III.III.I Proceso de compra y distribución de mercancía

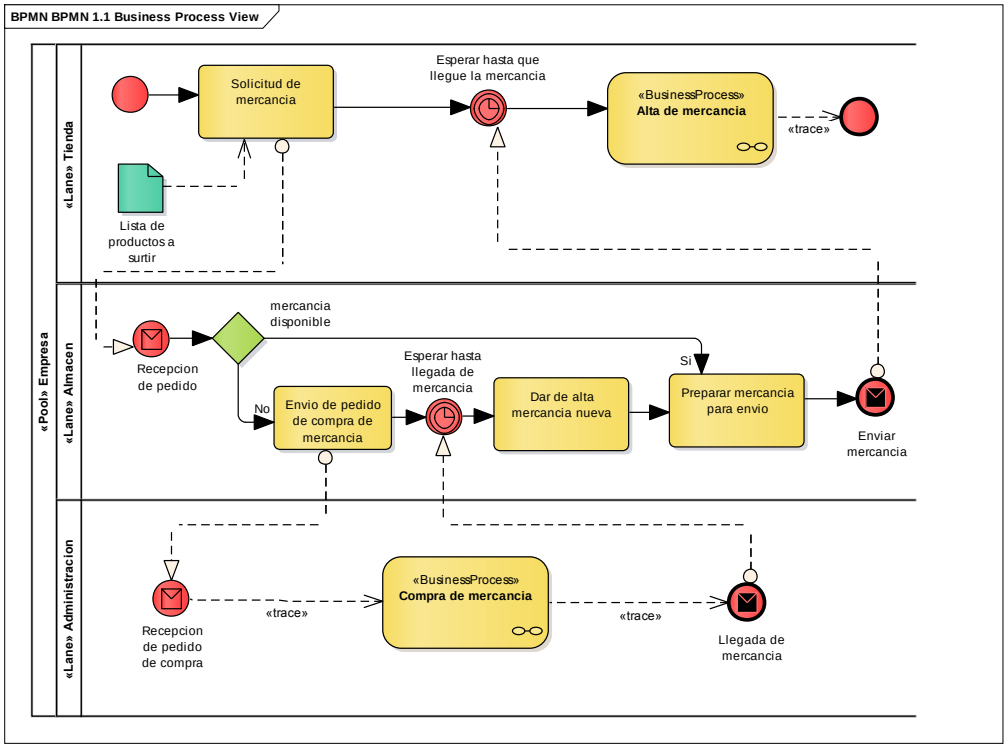


Figura 3.4 Proceso de compra y distribución de mercancía

El proceso de compra y distribución de la mercancía es la primera parte de las actividades de compra y venta la empresa. En el diagrama de la figura 3.4 se pueden ver las diferentes interacciones que tienen todos los departamentos, y que además puede verse como el proceso que necesita una mejor estrategia para poder dar mejores resultados.

III.III.II Proceso de venta de mercancía

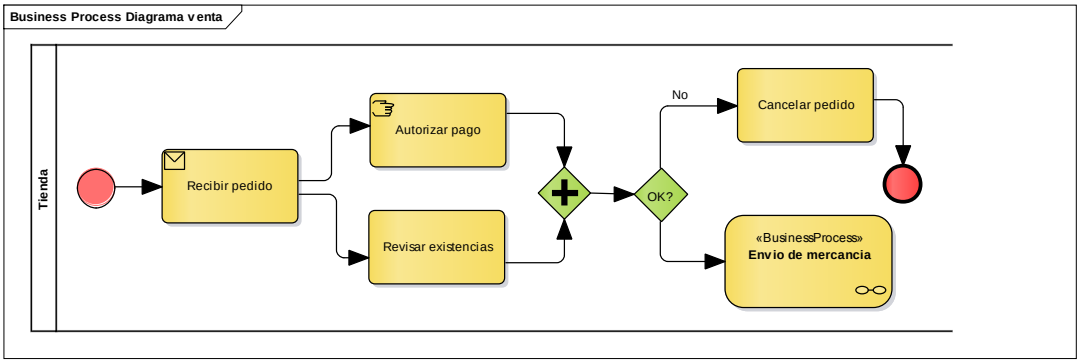


Figura 3.5 Proceso de venta de mercancía

El proceso de venta de la mercancía en esencia es simple, sin embargo, es el proceso que se debe poner más atención en cuestión de agilidad, ya que de esto depende la ganancia de la empresa, y refleja en sí mismo el servicio que se brinda. (Figura 3.5)

Si bien existen otros procesos que la empresa realiza, como lo son el proceso de supervisión, no es necesario su modelado, ya que están claramente especificados como procesos en el desarrollo de los casos de uso.

III.IV Análisis de los procesos de negocios

Después de analizar los proceso de negocio y analizando la circunstancia actual del negocio, podemos hacer una simplificación del problema, donde el almacén y el departamento de administración podrían trabajar en un solo proceso, haciendo que el proceso de compra se deje al departamento de almacén y el departamento de administración simplemente apruebe la compra hecha algún proveedor.

De la misma forma, el departamento de tienda pretende aumentar la agilidad del negocio haciendo que las fichas técnicas y precios estén a mayor disponibilidad de las personas que atienden a los clientes, así mismo, el agilizar el proceso de cotización de un producto puede ser más eficiente.

De esta forma se propone una solución para la empresa (Tabla 3.1).

Aplicación móvil para tienda	Aplicación de escritorio para tienda	Escritorio de servicios para administración y almacén
Clientes	Clientes	Compra de mercancía
Cotizaciones	Ventas	Reportes
Consulta de productos	Cotizaciones	Usuarios
	Ventas	Supervisión
	Cortes de caja	Manejo de productos
	Mercancía	Envíos
	Envíos	
	Pedidos de mercancía	

Tabla 3.1 Propuesta de los elementos dela solución empresarial

Como se puede observar, esta propuesta abarca todos los módulos que se proponen en los requisitos, así como agiliza el proceso de compra con la aplicación móvil, además que engloba mejor el proceso de compra de los productos. Esta es la propuesta del FRONTEND del sistema, que adelante se analizara a detalle.

De la misma forma en la que se analizó la situación de la empresa y a simplificación del problema, también se puede observar cuales pueden ser los objetos de negocios dentro del sistema, especificados a continuación en un modelado con UML.

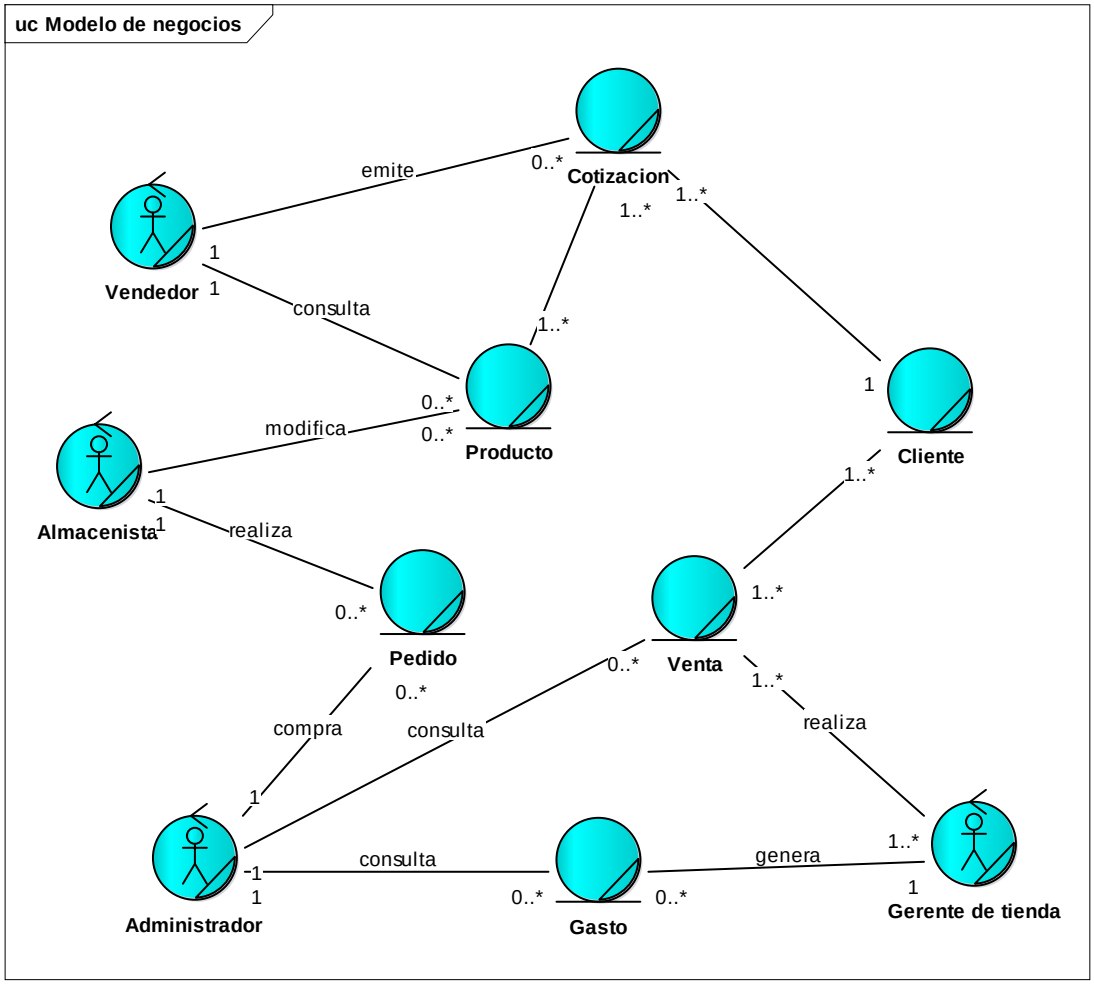


Figura 3.6 Diagrama de entidades de negocio de la empresa estudiada

Este es el diagrama de entidades de negocios (Figura 3.6), con esto podemos llegar más fácil a las clases y al diseño de la base de datos.

III.V FRONTEND

Como vimos, el análisis de los procesos de negocios desemboca en la parte de vista que tendrá el usuario del sistema. El diseño de éste es una propuesta se desarrolla a continuación

III.V.I Aplicación móvil para usuarios de tienda

Lo que hará la aplicación móvil será agilizar el proceso de atención al cliente, por lo que se necesita de una completa disponibilidad, y además de una fácil interacción con el usuario y que además tenga una muy práctica implementación.

Por lo tanto, se propone una implementación sobre el sistema operativo Android. Este sistema involucra muchas ventajas ideales para ser implementadas en esta solución, una de las más importantes es la de tiempo de implementación, y la otra es de bajo costo de uso.

Para esta aplicación es necesario un protocolo de comunicación, el cual será SOAP, ya que esta es la que soporte una comunicación (XML) para el Bus Empresarial de Servicios.

A continuación se muestra el diseño que tendrá la aplicación móvil (Figura 3.7).

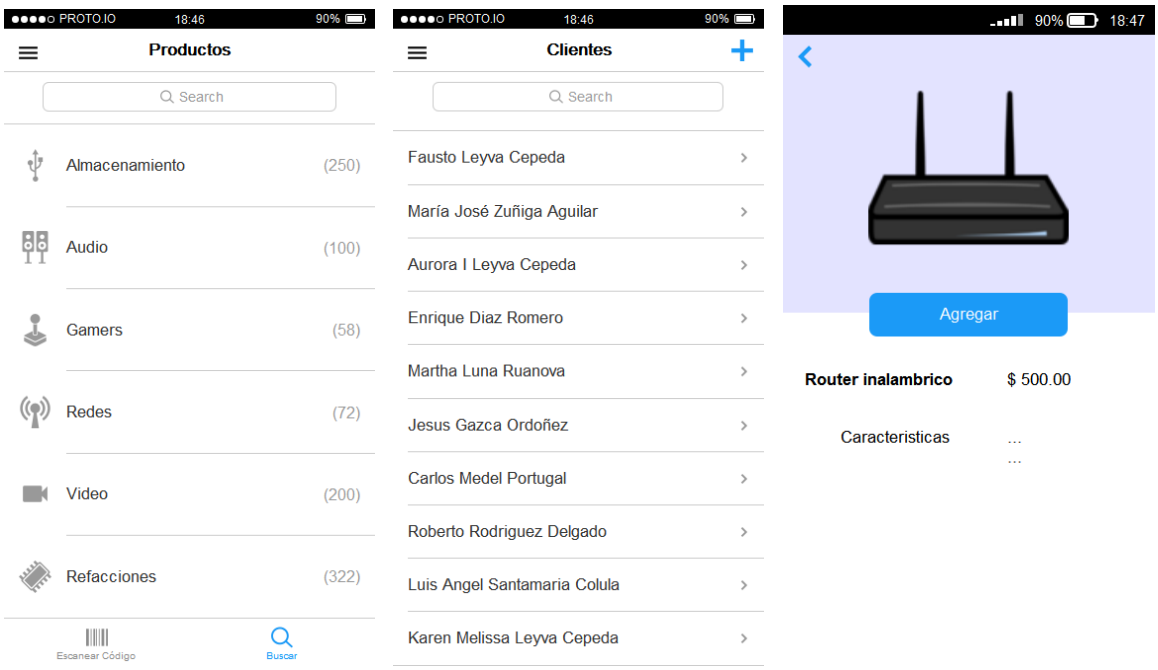


Figura 3.7 Capturas de pantalla de la aplicación móvil.

III.V.II Aplicación de escritorio para usuarios de tienda

Como ya se mencionó, SOA tiene la ventaja de trabajar con sistemas heterogéneos, por lo que la aplicación de escritorio usada actualmente en la empresa es una excelente opción para seguir trabajando. Esta aplicación fue desarrollada por la misma empresa, por lo que la modificación de la misma es más sencilla, sin embargo la aplicación fue modificada por la empresa y sólo se otorgaron los medios de comunicación con el servidor.

A continuación se muestra la aplicación de escritorio (Figura 3.8).

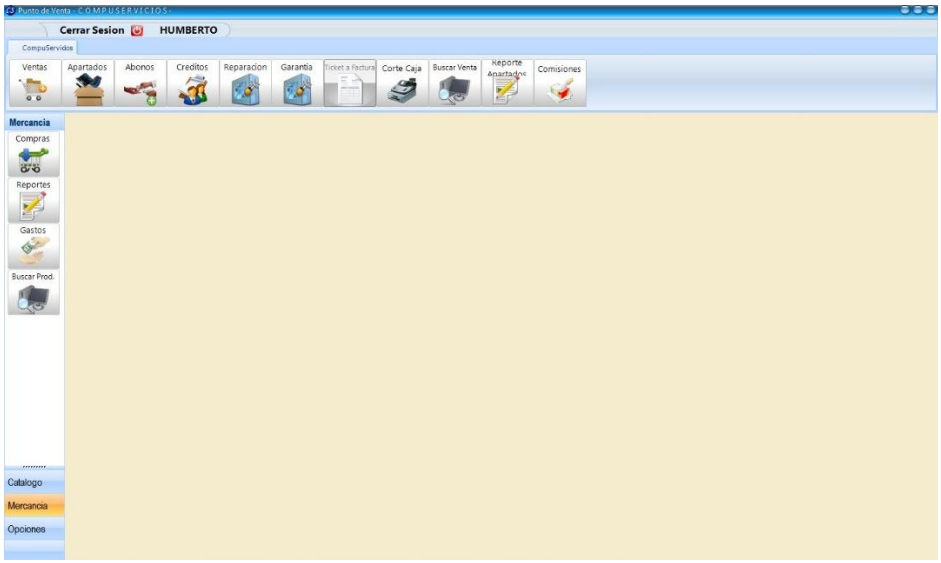


Figura 3.8 Captura de pantalla de la aplicación de escritorio

III.V.III Aplicación web para usuarios de almacén y administrativos.

Esta aplicación web será compartida para el departamento de administración y almacén, ya que alojará una forma más amigable de uso y comunicación entre departamentos más eficaz. Además que en esta plataforma se integrarán los servicios externos, como lo es el servicio de correo institucional.

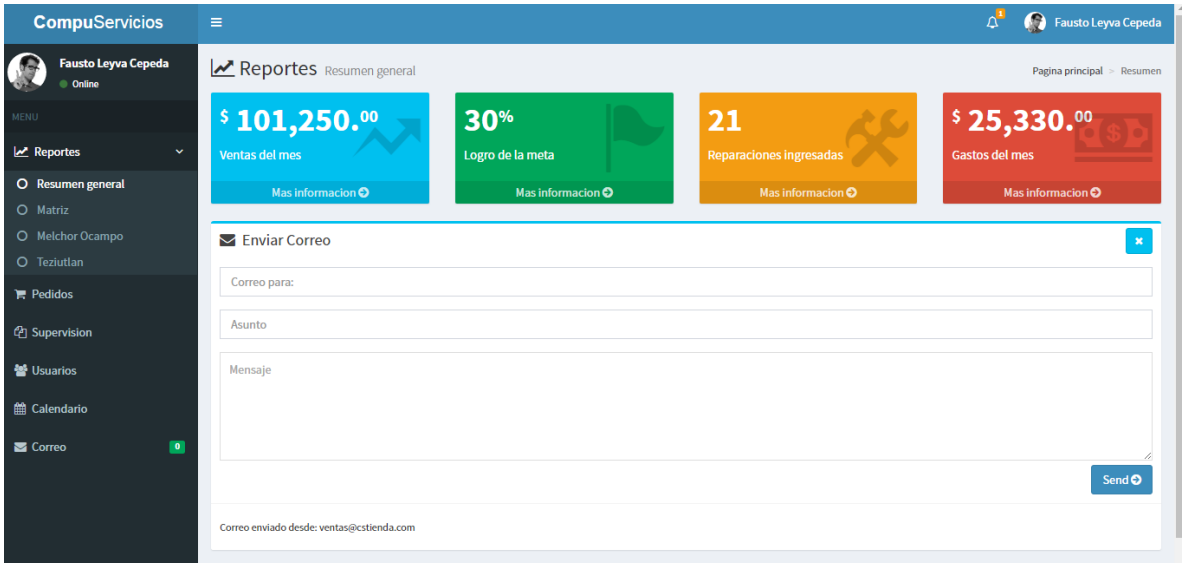


Figura 3.9 Captura de pantalla de la aplicación web

Para la implementación se utilizó un servicio de hosting alojado en un host contratado por la empresa a un tercero. La administración de éste es remoto junto con el servicio de correo electrónico.

III.VI BACKEND

Después de analizar la vista que va a tener el usuario (FRONTEND) ahora analizaremos la parte que el usuario no va a percibir. En esta se encuentra el diseño de la base de datos, los servicios de terceros que vamos a ocupar y los sistemas externos.

III.VI.I Diseño de la base de datos

Este es el diseño relacional de la base de datos a ocupar (Figura 3.10). Este diseño incluye el inventario de todas las tiendas y tiene que estar disponible todo el tiempo.

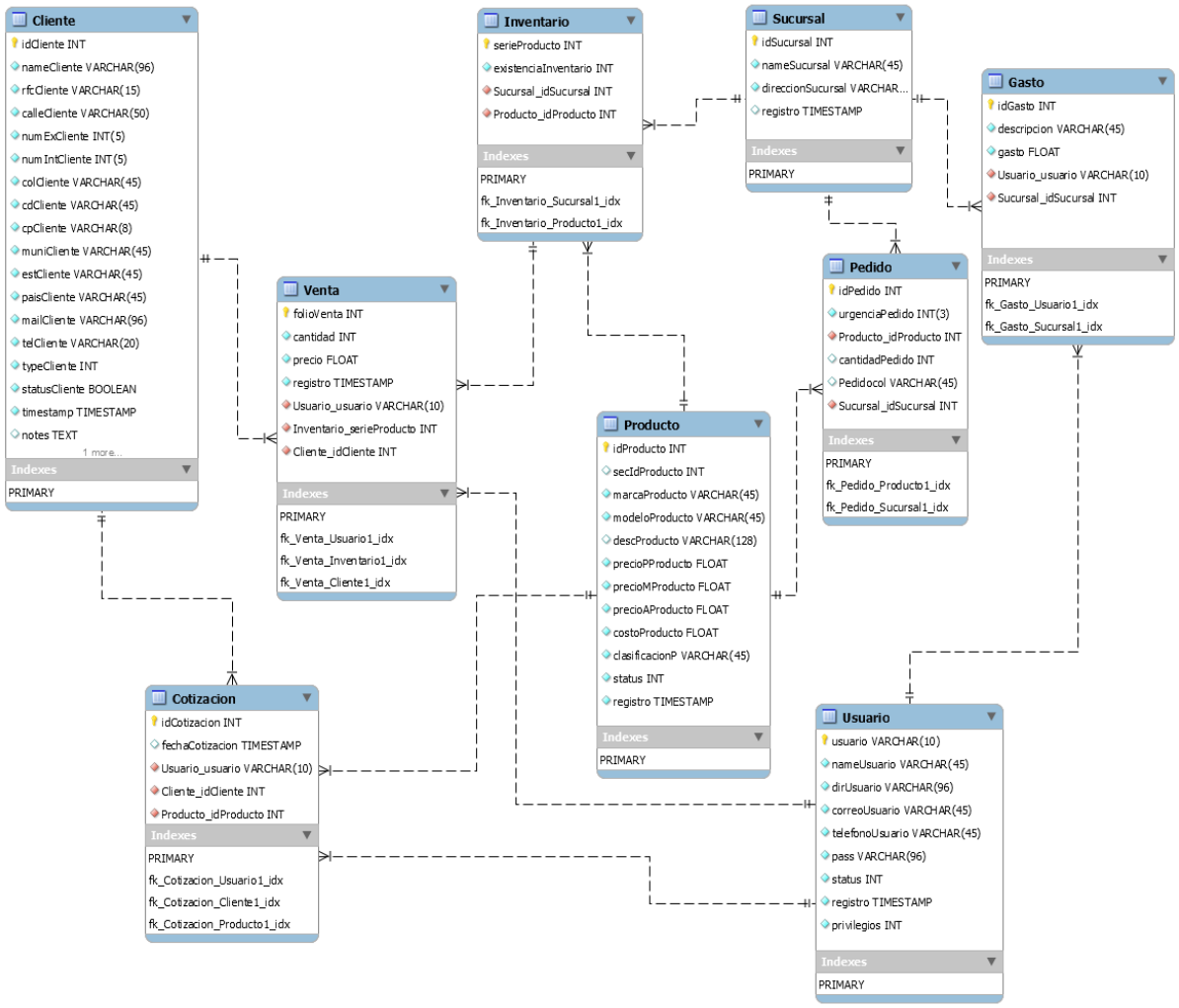


Figura 3.10 Diagrama relacional de la base de datos

La base de datos tiene una estrecha relación con las entidades de negocios por lo que se puede percibir una similitud ente estos diagramas.

Este diseño relacional abarca la mayoría de los requisitos presentados en el documento. Algunos procesos como los de supervisión estarán ligados a la plataforma correspondiente en donde se desarrolla, de esta forma tenemos un sistema altamente escalable.

Para la implementación de la base de datos se trabajó con MySQL en un servidor remoto.

II.VI.II Servidor de correo

El servicio de correo electrónico institucional se tiene alojado en un servicio de host ya integrado por la empresa hace tiempo, ahora este servicio se podrá integrar sin la necesidad de modificar la plataforma de uso convencional, por lo que la comunicación dentro y fuera de la empresa será mucho más eficaz.

Para la implementación del servidor de correo se utilizó un servicio de correo de terceros, el cual la empresa contrato junto con un servicio de hosting y un nombre de dominio.

III.VII Enterprise Service Bus (ESB)

El bus de servicios empresariales es la parte más importante de una Arquitectura Orientada a Servicios, por lo que el diseño de este se vuelve fundamental.

Como sabemos, el ESB es un middleware que sirve para comunicar FRONTEND con BACKEND de tal forma que puedan trabajar con sistemas heterogéneos hechos con cualquier lenguaje de programación (java, .NET, C++, etc.) y que además puede ser "plug and play", por lo que es fácil implementar soluciones para necesidades escalables. Es justo en el ESB donde los procesos empresariales toman forma de software. El siguiente esquema ejemplifica la visión general de toda la solución (figura 3.11).

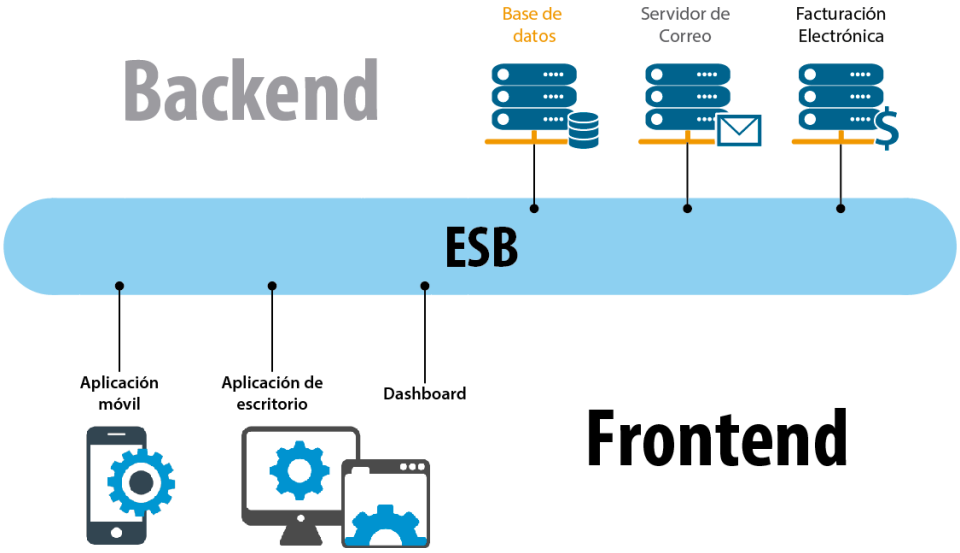


Figura 3.11 Diagrama general de la solución

El enfoque "middle-out" es el que se propone para esta solución, ya que partiendo que tenemos las necesidades en el BACKEND (diseño de una base de datos por ejemplo) y de los procesos de negocios modelados, podemos hacer una aproximación entre estos para poder brindar una solución más realista para la empresa. De esta forma el diseño del bus de servicios queda modelado por los siguientes diagramas (Figura 3.12 y 3.13).

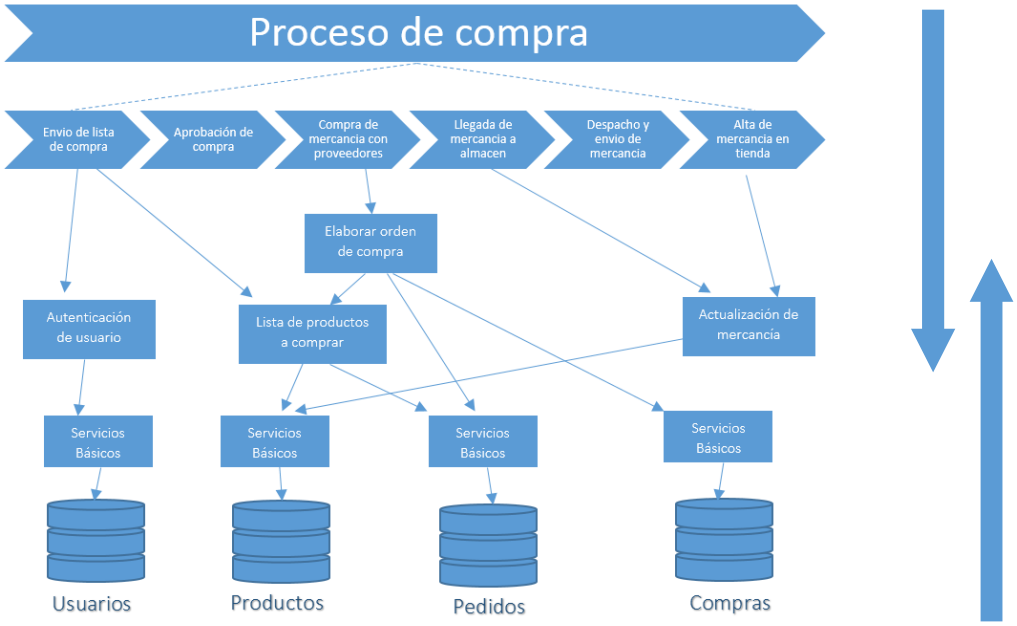


Figura 3.12 Diagrama del diseño de servicios para el proceso de compra

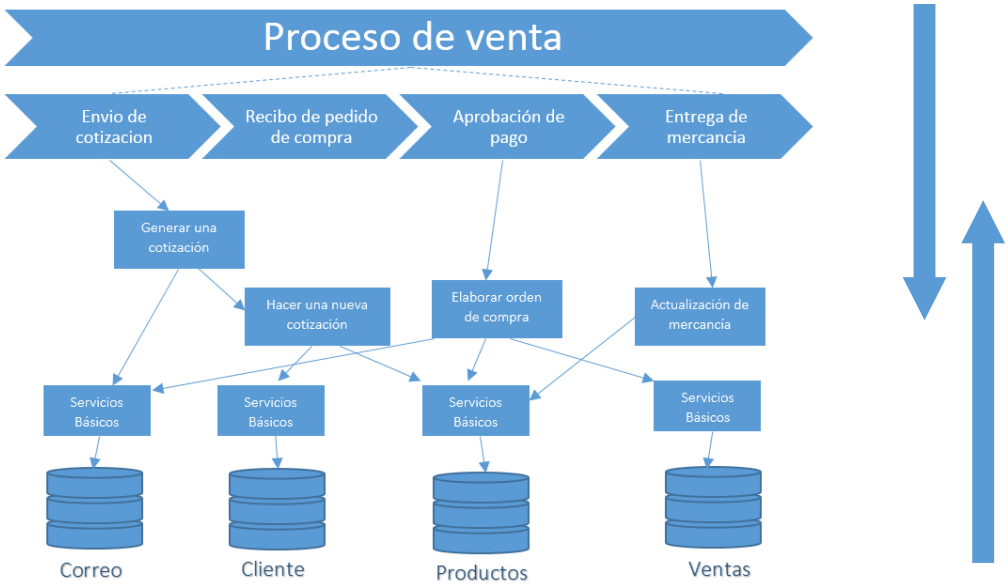


Figura 3.13 Diagrama del diseño de servicios para el proceso de venta

De esta forma, los módulos que se representan entre los servicios web básicos y los subprocesos son precisamente lo que contendrá el ESB.

Para plasmar la orquestación de servicios se ocupa un lenguaje llamado BPEL (de las siglas en inglés "Business Process Exectuion Language"). Es un lenguaje basado en XML y es ideal para la describir el flujo que llevaran los procesos de negocios y la realización de operaciones dentro del bus de servicios.

Un proceso definido en BPEL es de "login" el cual se muestra en la imagen siguiente.

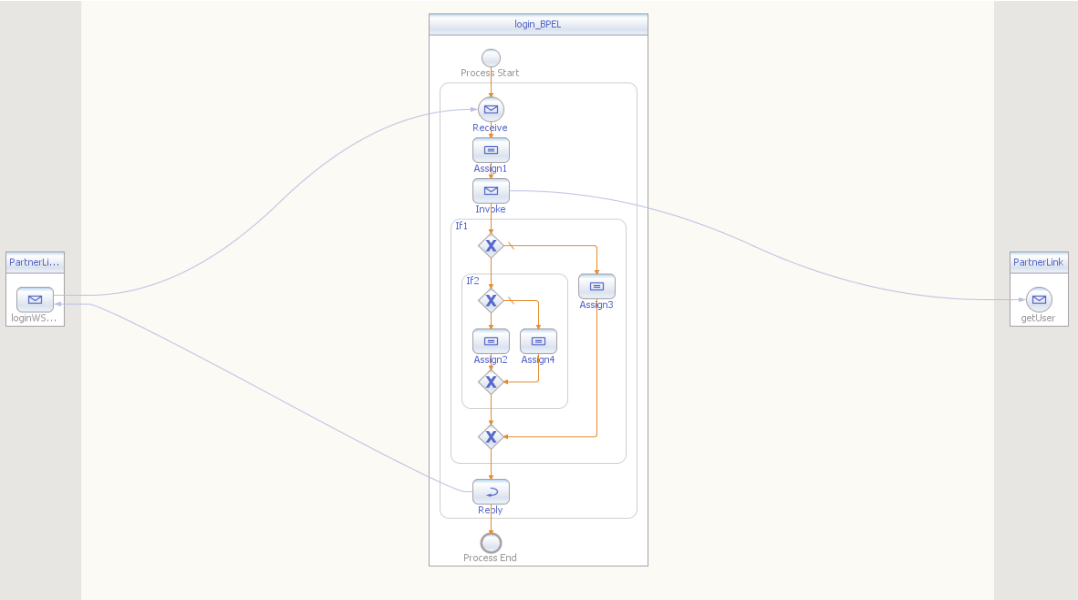


Figura 3.14 Ejemplo de la implementación de la orquestación de servicios con BPEL

Este diseño muestra del lado izquierdo el método de invocación del servicio orquestado, y del lado derecho los servicios que se están orquestando. Para este caso es únicamente un servicio proporcionado por la base de datos (consulta). Durante la parte del servicio podemos hacer un flujo de información (con diseño tipo scratch) para manejar la información y hacer operaciones.

Las herramientas utilizadas para la implementación fueron algo variadas. Se utilizó para la parte de BPEL un servidor con GlassFish en su versión 2.1 con soporte para SOA y JavaEE (como servidor de aplicaciones). Este servidor se implementó de manera local, sin embargo este servidor se recomienda montarlo sobre una plataforma de desarrollo en la nube (IaaS) donde las características del servidor se pueden modificar en tiempo real dándole una flexibilidad y escalabilidad necesaria para cada evento (inclusive apagarse en periodos innecesarios). El IDE (Integrated Development Environment, en español "Entorno de desarrollo integrado") ocupado fue OpenESB, el cual es una extensión de NetBeans para el desarrollo de SOA. El lenguaje ocupado para la orquestación de servicios es precisamente BPEL.

De esta forma la implementación final de este Bus de servicios se ve como la siguiente (Figura 3.15)

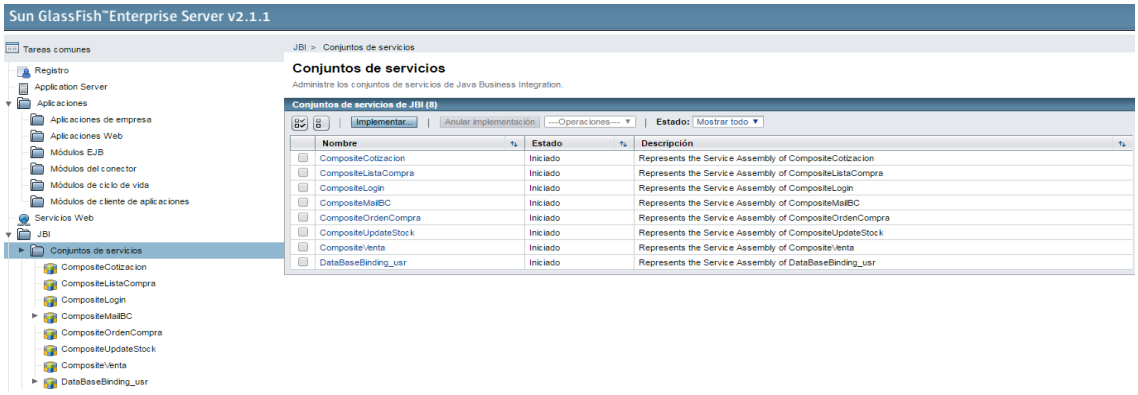


Figura 3.15 Implementación de la orquestación de servicios en el servidor GlassFish para SOA

CAPITULO IV

Navegación de la solución

A continuación se muestra un esquema general de la solución completa (Figura 4.1). En este esquema se muestran las diferentes partes de la solución y cómo interactúan entre sí.

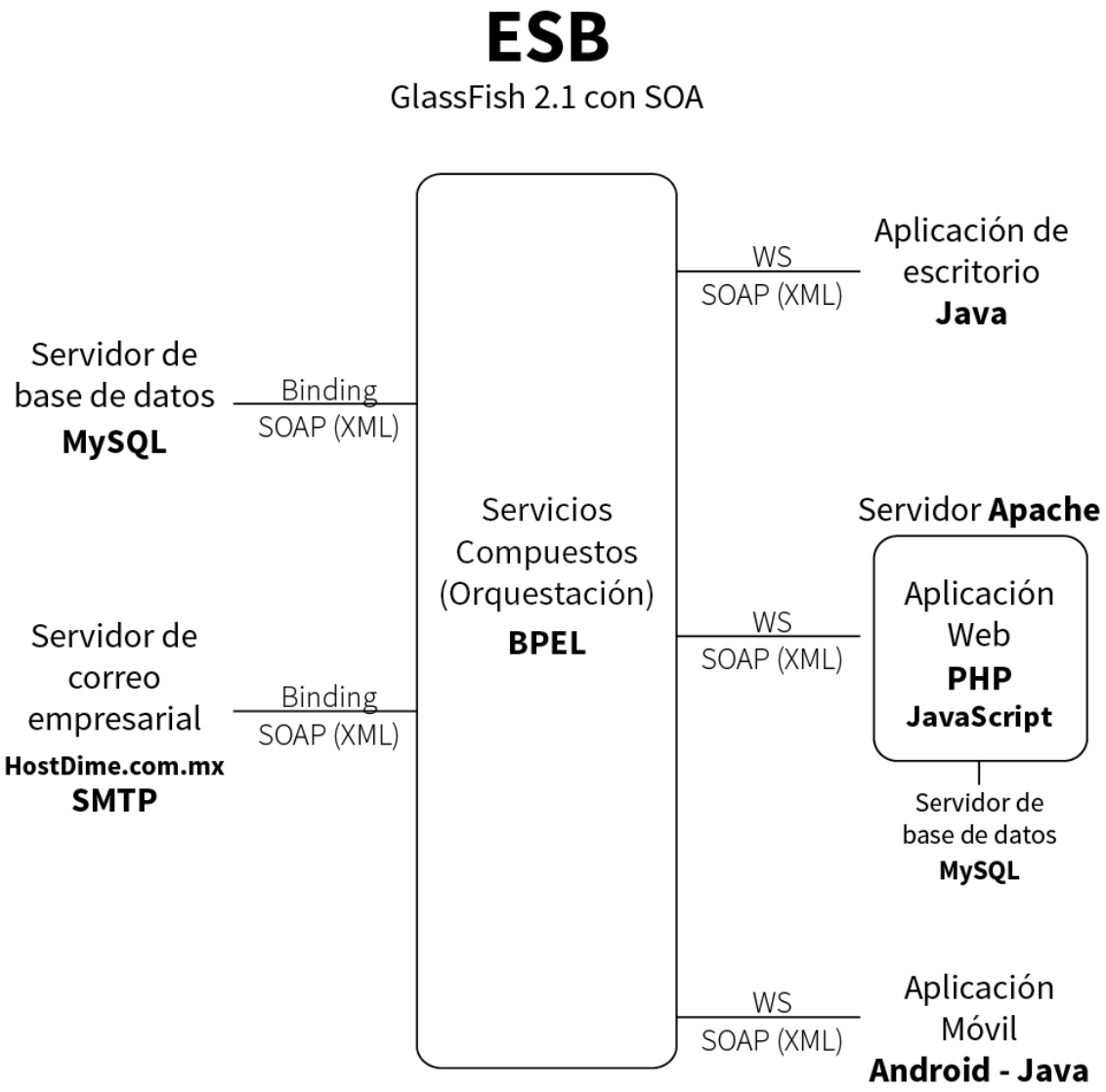
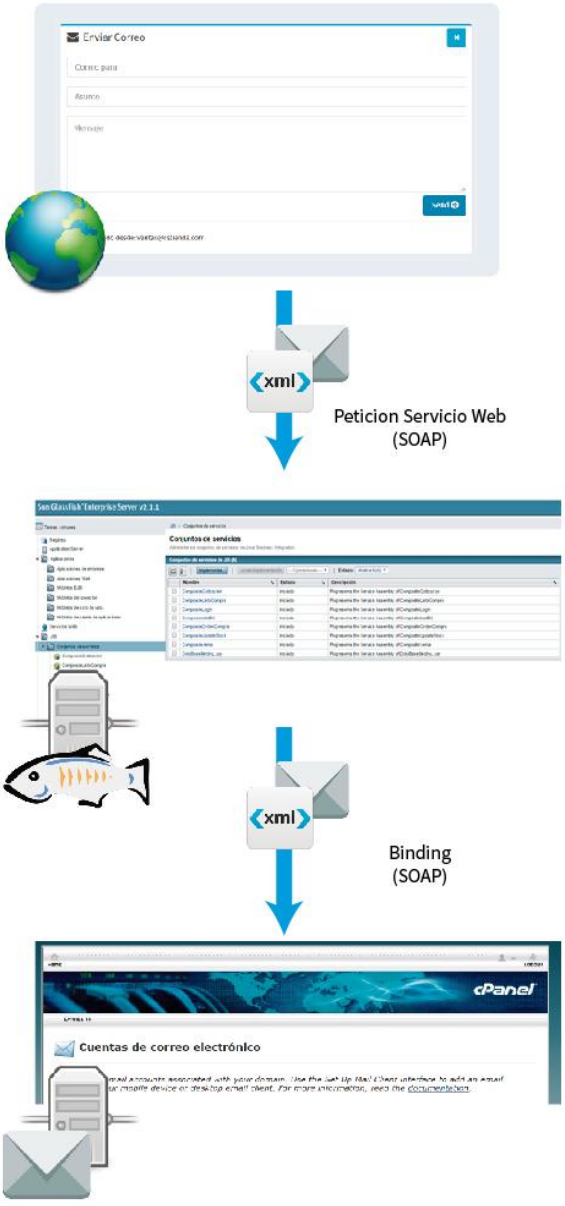


Figura 4.1 Esquema de implementación de la solución propuesta

El ESB es la parte más importante de la solución, este está implementado en un servidor GlassFish en su versión 2.1, ya que esta versión es la que soporta SOA. Este servidor se encuentran los procesos de negocios traducidos a servicios web orquestados con un lenguaje llamado BPEL. Este lenguaje basado en XML organiza los servicios encapsulados (binding) de nuestro BACKEND formado por un servidor de base de datos MySQL y un servidor de correo que usa SMTP de tal forma que las otras

aplicaciones puedan acceder a estos sin la necesidad de hacer una conexión individual para cada una de las aplicaciones del FRONTEND. La comunicación que se hace entre el BACKEND y el ESB es basada en SOAP. De la misma forma SOAP se utiliza para la comunicación del ESB con el FRONTEND. Este más específicamente es mediante Servicios Web ofrecidos por el servidor GlassFish el cual tienen los procesos de negocios listos para usar.

Cada una de las aplicaciones expuestas en el FRONTEND fue desarrollada en diferentes lenguajes, los cuales tienen formas diferentes de invocar servicios web. Sin embargo, gracias a que se usa SOAP, no hay necesidad de hacer servicios específicos para cada aplicación, ya que los métodos expuestos presentan una homogeneidad en la vista de resultados.



Se abre la aplicación web y se envia un correo electronico a un cliente con la información correspondiente a un producto (cotización de un producto).

En el servidor GlassFish (ESB) se hace una orquestación de servicios, haciendo una busqueda de la información del producto en la base de datos y enviado esta información al servidor de correos.

En el servidor de correos y en la base de datos se realizan las operaciones solicitadas por el usuario y finaliza el proceso.

Figura 4.2 Ejemplo del funcionamiento de la solución usando el proceso de envío de correo a un cliente.

CAPITULO V

V.I Conclusiones

Trabajar con SOA es la mejor experiencia que he tenido para trabajar sobre cualquier sistema o forma de programación, además que es la tendencia sobre la cual programar. Actualmente éste tema debería formar parte de cualquier plan de estudios para tener más gente preparada y lista para salir a la vida laboral.

SOA efectivamente tiene demasiadas ventajas, pero por si sola no representa una visión general de negocios. Para esta tesis hicieron falta conocimientos de sistemas de información un poco más especializados como lo son sistemas ERP (Enterprise Resource Planning) y basados un poco en estos se puede generar una solución más organizada y capaz de resolver un problema real de una empresa.

Otro punto importante y que cabe resaltar es que SOA no es para todos. Así como existen diferentes niveles para programación (alto y bajo nivel), SOA forma parte de un enfoque para soluciones empresariales que lo requieran, como sistemas distribuidos. No siempre necesitas de una plataforma orientada a servicios o soluciones en la nube para todo. Si no lo necesitas no lo uses.

Este trabajo de tesis fue diseñado para implementar una solución sobre una empresa que necesita una solución SOA, sin embargo, como lo dije anteriormente, no sólo se necesita un conocimientos de SOA para la implementación de éste, es más bien el idear una solución que abarca prácticamente todos los elementos vistos durante toda la carrera (como lo son diseño de aplicaciones móviles, web, arquitectura de software, lógica de algoritmos, redes, etc.), por lo que la culminación de esta con un proyecto de este estilo hace que todo lo estudiado cobre un sentido de solución diferente al sentido teórico al que estamos acostumbrados.

V.II Trabajo futuro

En este proyecto se abarca solamente la punta del iceberg de todo lo que podría ser un proyecto de crecimiento no sólo para una empresa si no para la implementación de muchas empresas.

Esta solución da pauta para una convivencia empresarial que busca una forma de crecimiento con bajos costos. Por lo que el nicho de mercado potencial se convierte en una oportunidad para negocios nacionales que busque el crecimiento.

Trabajar sobre la misma plataforma ya diseñada es la razón de la tesis; una solución altamente escalable.

Como trabajo posterior estaríamos preparando a las empresas a lo que hoy conocemos como "el internet de las cosas" y hacer una convivencia empresarial más armónica.

BIBLIOGRAFÍA

Marks, E. A., & Bell, M. (2006). *Executive's Guide to Service-Oriented Architecture.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Anonimo. (2007). *CireSeerx.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.133.4413

Bishop, T. A., & Karne, R. K. (2003). A SURVEY of MIDDLEWARE. *Computers and Their Applications*, 5.

Douglas K., B. (2003). *Web Services and Service-Oriented Architectures, The savvy manager's guide.* San Francisco, CA: Morgan Kaufmann Publishers.

Exposito, E., & Codé, D. (2014). *Smart SOA Plataforms in Cloud Computing Architectures.* Wiley-ISTE.

Josuttis, N. M. (2007). *SOA in Practice, The Art of Distributed System Design.* O'reilly.

Milan Zdravković*, M. T. (2007). SOA-Based Approach to the Enterprise Resourse Planning Implementation in Small Enterprises. *Mechanical Engineering Vol. 5, No 1*, 97 - 104.

Newcomer, E. (2002). *Understanding Web Services XML, WSDL, SOAP, and UDDI.* Pearson Education, Inc.

Townsend, E. (2008, Julio 10). *The 25-Year History of Service-Oriented Architecture.* Retrieved from Erik Townsend: www.eriktownsend.com

Van der Aalst, W., Desel, J., & Oberweis, A. (2000). *Business Process Management.* Springer.

Wassem, R. (2009). *SOA-Based Enterprise Integration.* The McGraw-Hill Companies, Inc.

CompuServicios (2011). *Manual de políticas y procedimientos*, 67.

Christian Vanessa Flores Lárate. "Desarrollo del sistema de información MIES (Ministerio Estudiantil) web utilizando servicios web", Tesis para obtener el título de Ingeniero en Ciencias de la Computación, FCC--BUAP, Verano 2012.

Manuel Maldonado Mendoza. "Desarrollo de un sistema de información basado en SOA (Arquitectura Orientada a Servicios)", Tesis para obtener el título de Ingeniero en Ciencias de la Computacion, FCC--BUAP, Primavera 2012.

Microsoft Developer Network / Library. "Service-Oriented Architecture(SOA)", Electronic document, Chapter 1: Service Oriented Architecture (SOA), https://msdn.microsoft.com/en-us/library/bb833022.aspx.

"Enabling "Real World" SOA through the Microsoft Platform", A Microsoft White Paper, December 2006. Available at http://www.microsoft.com/biztalk/solutions/soa/whitepaper.mspx

Yanet Espinol Martín, "Arquitectura de software. Arquitectura Orientada a Servicios" http://www.ilustrados.com/tema/12463/Arquitectura-software-Arquitectura-orientada-servicios.html

XML oreilly www.xml.com