



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA  
COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO DE SISTEMAS E INFORMÁTICA**

**TEMA: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL  
SISTEMA DE GESTIÓN DE INVENTARIOS Y VALIDACIÓN DE  
CONTRATOS DE SOPORTE DE LA EMPRESA SINETCOM  
S.A.**

**AUTOR: FLORES GRANJA, DIEGO ANTONIO  
BOZZANO LARREA, JOSÉ ANDRÉS**

**DIRECTOR: ING. CAMPAÑA, MAURICIO  
CODIRECTOR: ING. ANDRÉS DE LA TORRE**



**SANGOLQUÍ**

**2015**

# CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. DIEGO ANTONIO FLORES GRANJA y el Sr. JOSÉ ANDRÉS BOZZANO LARREA como requerimiento parcial a la obtención del título de INGENIEROS EN SISTEMAS E INFORMÁTICA.

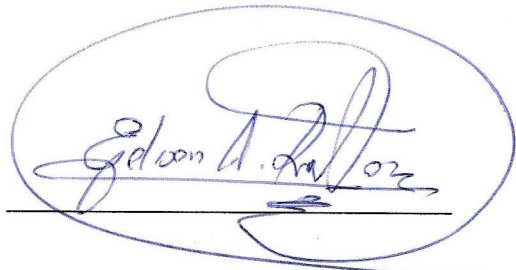
Sangolquí, Abril del 2015



A handwritten signature in blue ink is written over a circular official seal. The seal contains text around its perimeter, including 'UNIVERSIDAD TECNOLÓGICA NACIONAL' and 'FACULTAD DE INGENIERÍA'. The signature is highly stylized and overlaps the seal.

Ing. Mauricio Campana

Director de Tesis



A handwritten signature in blue ink is written over a horizontal line. The signature is in a cursive style and is enclosed within a large, hand-drawn oval shape.

Ing. Andrés de la Torre

Codirector de Tesis

## AUTORÍA DE RESPONSABILIDAD

Nosotros, Diego Antonio Flores Granja y José Andrés Bozzano Larrea declaramos que el presente trabajo es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación personal y que hemos consultado las referencias bibliográficas que se incluyen en el documento.

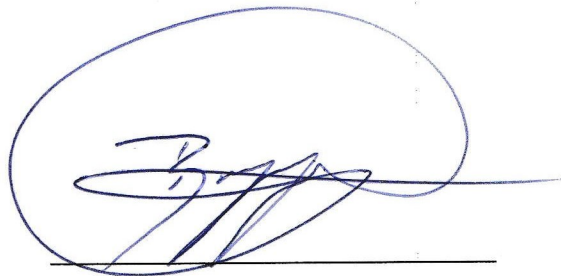
La Universidad de las Fuerzas Armadas ESPE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual por su reglamento y por la normativa institucional vigente.

Sangolquí, Abril del 2015



---

Diego Antonio Flores Granja



---

José Andrés Bozzano Larrea

## AUTORIZACIÓN

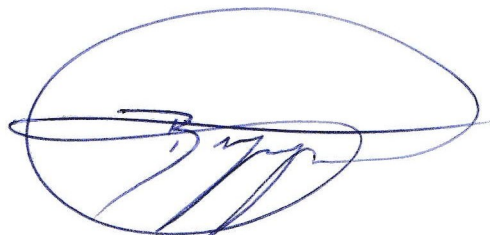
Nosotros, **DIEGO ANTONIO FLORES GRANJA Y JOSÉ ANDRÉS BOZZANO LARREA**, autorizamos a la Universidad de las Fuerzas Armadas ESPE la publicación, en la biblioteca virtual de la Institución, el trabajo "**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE INVENTARIOS Y VALIDACIÓN DE CONTRATOS DE SOPORTE DE LA EMPRESA SINETCOM S.A.**", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, Abril del 2015



---

Diego Antonio Flores Granja



---

José Andrés Bozzano Larrea

## **DEDICATORIA**

A mis amados padres y hermana que han sido un apoyo incondicional y un ejemplo de vida, enseñándome que el trabajo en equipo y sobre todo el amor y la unión familiar son fundamentales para lograr el éxito y la felicidad en el hogar.

A mí amada esposa que se ha convertido en un apoyo incondicional para mí en todo momento y la que me ha mostrado y ha enseñado todos los buenos aspectos de la vida.

Diego Flores Granja

## **DEDICATORIA**

Esta tesis representa uno de mis mayores logros, la culminación de mi carrera, esto no solo significa el fin de mi etapa universitaria, significa el inicio de mi vida como un profesional calificado y preparado para todo reto. Por todo lo que esto significa para mí, este proyecto se lo dedico a mis padres, no por el hecho de que sean mis padres, sino por lo que realmente representan, ellos son todo lo que yo admiro y respeto en esta vida. Gracias a ellos yo eh llegado a la posición que tengo hoy en día. Les dedico esto a ustedes, Armando y Margarita, por ser siempre los que han creído en mí.

José Andrés Bozzano Larrea

## **AGRADECIMIENTO**

A mi familia por darme un hogar ejemplar en el que he podido criarme y crecer con valores y principios correctos, por todo el apoyo incondicional que he recibido de ellos en todas las metas y proyectos que he deseado emprender.

A mi amada esposa que es mi mejor amiga y la persona que me impulsa a ser mejor cada día, la que me ha mostrado todo lo que puedo lograr si persigo mis sueños y me esfuerzo poniendo mi confianza en aquel Dios que nos dio la vida.

A mis compañeros y amigos de universidad con los que hemos avanzado en este corto trayecto y con los que hemos compartido muchos gratos y lindos recuerdos, con los que nos hemos apoyado dándonos una mano para poder finalizar una etapa más de nuestras vidas.

A mis profesores que han sido buenos ejemplos y que me han ayudado para crecer profesionalmente y me han dado las bases necesarias para seguir especializándome en el ámbito de mi carrera.

A mis jefes y estimados amigos el Sr. Cap. Jorge Yáñez y la Sra. Ms. Ma. Augusta Solano de Sala quienes me han dado la oportunidad de hacer este proyecto de tesis en su empresa y me han brindado todo el apoyo y colaboración para que este proyecto se lleve a cabo.

Y sobre todo a Dios, al que le debo todo lo que tengo y todo lo que soy, quien ha llenado mi vida de alegrías y bendiciones.

Diego Flores Granja

## **AGRADECIMIENTO**

Cada logro en la vida de una persona es el logro de todos aquellos que están apoyando y alentando cada meta cumplida. Es por ello que agradezco a cada persona que se involucró en el desarrollo de este proyecto.

Agradezco a Dios por darme la sabiduría necesaria para continuar por más largo que sea el camino. Por poner en mi vida a las personas indicadas que han aportado sobre mis acciones y decisiones. Te agradezco Dios por nunca abandonarme y por guiarme siempre para conseguir mis más profundos anhelos.

A mi mamá y mi papá, les agradezco por guiarme siempre por el camino correcto, por inculcarme que todos los esfuerzos valen la pena. Simplemente les agradezco por ser quien soy y por ayudarme a llegar a este punto de mi vida.

A mi hermana le agradezco por apoyarme en cada paso dado, por aconsejarme en cada etapa de mi vida y por siempre estar allí, cuando la he necesitado y cuando no.

Agradezco a todas las personas que de una u otra manera han dejado huella en mí, para que me convirtiera en lo que hoy en día soy.

José Andrés Bozzano Larrea



## ÍNDICE DE CONTENIDOS

<b>CERTIFICADO .....</b>	<b>II</b>
<b>AUTORÍA DE RESPONSABILIDAD .....</b>	<b>III</b>
<b>AUTORIZACIÓN .....</b>	<b>IV</b>
<b>DEDICATORIA.....</b>	<b>V</b>
<b>DEDICATORIA.....</b>	<b>VI</b>
<b>AGRADECIMIENTO.....</b>	<b>VII</b>
<b>AGRADECIMIENTO.....</b>	<b>VIII</b>
<b>ÍNDICE DE CONTENIDOS.....</b>	<b>IX</b>
<b>LISTADO DE FIGURAS.....</b>	<b>XII</b>
<b>RESUMEN.....</b>	<b>XV</b>
<b>ABSTRACT.....</b>	<b>XVI</b>
<b>CAPÍTULO 1: GENERALIDADES .....</b>	<b>1</b>
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES .....	2
1.3. DESCRIPCIÓN DEL PROBLEMA .....	3
1.4. JUSTIFICACIÓN E IMPORTANCIA.....	3
1.5. OBJETIVOS .....	4
1.5.1. <i>Objetivo General</i> .....	4
1.5.2. <i>Objetivos Específicos</i> .....	5
1.6. ALCANCE.....	5
1.7. METODOLOGÍA.....	6
1.8. HERRAMIENTAS Y SOFTWARE.....	9
<b>CAPÍTULO 2: MARCO TEÓRICO.....</b>	<b>11</b>
2.1. APLICACIÓN WEB .....	11
2.1.1. <i>Arquitectura Cliente – Servidor</i> .....	11
2.2. METODOLOGÍA.....	13

2.2.1.	<i>Proceso de diagramación</i>	14
2.3.	HERRAMIENTAS Y BASES DE DATOS	14
2.3.1.	<i>Plataforma Java</i>	14
2.3.2.	<i>Netbeans IDE</i>	17
2.3.3.	<i>Componente Visual Primefaces</i>	18
2.3.4.	<i>JEE 6</i>	20
2.3.5.	<i>JSF – Java Server Faces</i>	21
2.3.6.	<i>MySQL Server – Servidor de base de datos</i>	23
2.3.7.	<i>Glassfish AS. – Servidor de Aplicaciones</i>	23
2.3.8.	<i>StarUML</i>	24
2.3.9.	<i>ArgoUML</i>	25
2.3.10.	<i>MagicDraw</i>	25
2.4.	MESA DE AYUDA	26
2.5.	INGENIERÍA DE REQUERIMIENTOS IEEE 830	26
2.6.	PRUEBAS	27
2.6.1.	<i>Pruebas Caja Negra</i>	28
2.6.2.	<i>Pruebas de caja blanca</i>	28
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO</b>		<b>29</b>
3.1.	HISTORIAL DE REUNIONES	29
3.2.	ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE	31
3.3.	DIAGRAMAS UWE	33
3.3.1.	<i>Diagrama de casos de uso</i>	33
3.3.2.	<i>Diagrama de Clases</i>	34
3.3.3.	<i>Diagramas de Secuencia</i>	34
3.3.4.	<i>Diagrama de Despliegue y Componentes</i>	37
3.3.5.	<i>Diagrama Conceptual</i>	37
3.3.6.	<i>Diagrama de Navegación</i>	38
3.3.7.	<i>Diagrama de Presentación</i>	38
3.3.8.	<i>Modelo Físico de Base de Datos</i>	38
<b>CAPÍTULO 4: IMPLEMENTACIÓN</b>		<b>39</b>
4.1.	CONSTRUCCIÓN BASE DE DATOS	39
4.2.	ARQUITECTURA DE LA APLICACIÓN	39

4.3.	CODIFICACIÓN .....	41
4.3.1.	<i>Instalación de la Aplicación</i> .....	42
4.3.1.1.	<i>Hosting</i> .....	45
4.4.	PRUEBAS .....	45
4.4.1.	<i>Caja Blanca</i> .....	45
4.4.2.	<i>Caja Negra</i> .....	48
4.5.	SEGURIDAD DE LA APLICACIÓN .....	53
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....		<b>57</b>
5.1.	CONCLUSIONES .....	57
5.2.	RECOMENDACIONES.....	58
<b>BIBLIOGRAFÍA</b> .....		<b>60</b>
<b>ANEXOS</b> .....		<b>65</b>

## LISTADO DE FIGURAS

Figura 1: Arquitectura Cliente – Servidor .....	12
Figura 2: Arquitectura de las aplicaciones WEB - Un solo servidor .....	12
Figura 3: Compilación de clases JAVA .....	15
Figura 4: Estadísticas del uso de Java en diferentes entornos. ....	17
Figura 5: Interés en el tiempo de los componentes visuales de JSF.....	19
Figura 6: Modelo JEE .....	21
Figura 7: Renderizar la respuesta JSF.....	22
Figura 8: Arquitectura de Glassfish AS. ....	24
Figura 9: Concepto de Prueba de Caja Negra. ....	28
Figura 13: Diagrama de Caso de Uso - Módulo de Administración. ....	33
Figura 15: Diagrama de Secuencia - Crear.....	35
Figura 16: Diagrama de Secuencia - Ingreso Componente Electrónico. ....	36
Figura 17: Diagrama de Despliegue y Componentes.....	37
Figura 22: Arquitectura Sinetcom App. ....	39
Figura 23: Implementación de EJB. ....	41
Figura 24: Creación de Base de Datos. ....	43
Figura 25: Configuración de MySQL en Glassfish.....	44
Figura 26: Despliegue de aplicación .ear en Glassfish.....	44
Figura 27: Pruebas de Caja Blanca - Verificar Usuario. ....	46
Figura 28: Pruebas de Caja Blanca - Insertar Contrato.....	47

Figura 29: Resultado de consulta de Contratos después de Prueba.....	47
Figura 30: Pruebas de Caja Blanca - Buscar Contrato.....	48
Figura 31: Configuración de Listener SSL Glassfish. ....	54
Figura 32: Protocolo HTTPS activo.....	55
Figura 33: Utilidad de encriptación.....	56

## LISTADO DE TABLAS

Tabla 1: Historial de Reuniones .....	29
Tabla 2: Prueba de Caja Blanca - Verificar la existencia de un usuario .....	46
Tabla 3: Resultados de Casos de Prueba - Contratos .....	48
Tabla 4: Pruebas de Caja Negra.....	49
Tabla 5: Caso de Prueba RF07 .....	52
Tabla 6: Caso de Prueba RF08 .....	53

## **RESUMEN**

SINETCOM S.A. es una empresa que brinda servicios tecnológicos en el ámbito del almacenamiento de datos, la cual ha ido creciendo desde su fundación, generando así varios inconvenientes en el proceso de atención a los mismos, al crecer su demanda de servicios, se incrementó así paulatinamente el stock de equipos, partes y piezas que deben ser administrados de una u otra forma, y que lo han estado realizando de forma manual. Por tal motivo el presente trabajo tiene como objetivo solventar esta necesidad mediante el desarrollo de un sistema de administración de Contratos en la parte de atención a clientes, y administración de Inventarios. Para llevarlo a cabo se aplicó la metodología de desarrollo ágil UWE (UML-Based Web Engineering), la cual nos permite desarrollar un sistema multiplataforma y orientado a la web. La plataforma JEE para el desarrollo del sistema empresarial con estándares internacionales de desarrollo y los frameworks JSF y Primefaces para el manejo de las interfaces visuales, sobre una base de datos MySql de código abierto.

### **PALABRAS CLAVE:**

- **INVENTARIO**
- **UWE**
- **JEE**
- **PRIMEFACES**
- **MYSQL**

## **ABSTRACT**

SINETCOM S.A. it is an Enterprise that give technological services in the storage equipment area, this enterprise was growing up since their foundation but over the time it get some problems with this kind of services, like the increasing quantity of equipment, parts and pieces that needs to be managed with an automatic process. For that reason this Project is going to give a solution for that problem, with a development of a contract administrator system, with an inventory administrator module. This develop was made with UWE (UML-Based Web Engineering) that give us the possibility to have a multiplatform system and web oriented application. It has JEE platform for enterprise systems with international standards for develop, and JSF framework with Primefaces for the visual interfaces. All the application data is saved in MySQL database that is opensource.



## **CAPÍTULO 1: GENERALIDADES**

### **1.1. Introducción**

Las empresas nacionales que prestan servicios de soporte técnico en cualquier ámbito tecnológico, están expuestas a fracasar en sus intentos por llevar un proceso sistemático, organizado y de calidad al usuario final sin un control adecuado del mismo.

Cuando de gestión de servicios se trata, una empresa de soporte y consultoría técnica debe poseer una herramienta que le ayude a llevar su proceso de una forma ágil y precisa, que le ayude a dar solución a los incidentes que se presentan en el día a día en este tipo de empresas. (Mora Pérez, 2012)

Una mesa de ayuda (Service Desk) es un software que interactúa directamente con el cliente y el personal de servicio o soporte técnico, todos los actores involucrados con este tipo de software obtienen siempre grandes beneficios si hacen uso de este tipo herramienta, uno de ellos, es que permite dar seguimiento al estado de un caso o ticket hasta que este es resuelto completamente.

Una herramienta así ayudaría a verificar si se están cumpliendo los plazos en los SLA que se da a los servicios de la mesa de ayuda y daría la oportunidad de llevar un control completo y sistémico en el soporte de un cliente final, logrando de esta manera mayor satisfacción en los clientes/usuarios que forman parte de la mesa de ayuda. (Mora Pérez, 2012)

## 1.2. Antecedentes

La empresa Sinetcom S.A. enfrenta varios problemas al tratar de brindar un servicio de calidad a sus clientes desde ya algunos años, al ser esta una empresa que ofrece soluciones tecnológicas empresariales para centros de datos de cualquier magnitud, su abanico de clientes es muy amplio, por tanto se desea llevar un control sistematizado sobre las solicitudes de servicio y soporte que son requeridas por los clientes.

Actualmente no poseen una herramienta que permita gestionar su almacén tecnológico de piezas y partes; estas últimas son utilizadas para realizar reemplazos en caso de daños irreversibles, cabe recalcar que la empresa requiere también una herramienta que permita validar los contratos de soporte y las garantías a la fecha.

Adicionalmente la empresa no posee un software de Mesa de Ayuda que ayude a mantener orden y control sobre los casos de soporte que son requeridos por los clientes.

Sin una solución integral, que agrupe todos estos componentes, las tareas que conforman un soporte de calidad se vuelven complicadas y todo el proceso en sí se vuelve desordenado. Por último el descontento de los clientes es notorio al no cumplirse los SLAs especificados en los contratos de soporte. La empresa desea que sus clientes estén satisfechos con el servicio prestado, por lo tanto han decidido tomar acciones para que este proceso descontrolado e ineficiente tome ahora un giro que le permita a la empresa crear una relación más sólida con el cliente y cumplir con sus expectativas.

### **1.3. Descripción del Problema**

La empresa Sinetcom S.A. brinda servicios de soporte, sobre su base instalada, de productos tecnológicos alrededor del país. La empresa no posee un sistema informático que le permita mantener un repositorio común de clientes, contratos, inventarios y casos de soporte. Todos los documentos contractuales se guardan en un archivador desde hace varios años, si se desea hacer alguna consulta sobre un contrato específico se debe ir al archivador y buscar la carpeta con el contrato deseado, esto hace que el personal pierda demasiado tiempo y esfuerzo en tareas manuales que pueden llegar a ser automatizadas.

Los inventarios tecnológicos usados para la ejecución de garantías se llevan en simples hojas de cálculo que contienen cientos de registros, suelen encontrarse errores y falta de información en varios de estos registros, también se encuentran muchos registros duplicados. Una solución informática ayudaría a mitigar estos problemas.

Por último las constantes llamadas de soporte recibidas a diario por parte de los clientes buscando ayuda, no se logran controlar ni se lleva un registro formal y ordenado de todas las atenciones realizadas, tampoco de los tiempos de respuesta en el cumplimiento de SLAs, esta falta de calidad en el servicio puede costarle varias multas a la empresa por no cumplir con los acuerdos de nivel de servicio.

### **1.4. Justificación e Importancia**

Actualmente la empresa Sinetcom S.A. no posee una solución de software que permita optimizar el proceso de mesa de servicio, siendo este uno de los procesos más importantes de su negocio.

El sistema de gestión de inventarios y validación de contratos de soporte brindará a la empresa la capacidad de mantener control sobre el almacén de partes y piezas de varios fabricantes con los cuales trabaja. Esto le servirá a la empresa para mantener orden e integridad de la información que se maneja en los contratos de soporte y en las garantías de los equipos comercializados.

El módulo de validación de contratos permitirá que la empresa conozca sin error la fecha de vencimiento de cada uno de los contratos de soporte y la efectividad de las garantías de los equipos comercializados a la fecha; todo esto ayudará a que no se cometan errores al validar la caducidad de los contratos y al prestar servicio a clientes, fuera de las fechas límites contractuales.

Este sistema también permitirá realizar una validación previa de las garantías y de los contratos de soporte previo a la creación de tickets por incidentes en los equipos o cualquiera de sus partes.

La empresa no posee una herramienta de mesa de ayuda que le ayude en la automatización del proceso de soporte técnico, por tanto la herramienta Open Source OTRS Help Desk, en integración con el sistema de inventarios y validación de contratos, proporcionará a la empresa la funcionalidad que necesita para el seguimiento de tickets de soporte.

## **1.5. Objetivos**

### **1.5.1. Objetivo General**

Desarrollar el sistema de control de inventarios y validación de contratos de soporte para la empresa Sinetcom S.A., aplicando la metodología UWE, para mejorar y optimizar el proceso de mesa de ayuda en la empresa.

### 1.5.2. Objetivos Específicos

- Capturar los requerimientos de software necesarios para el desarrollo efectivo del sistema de inventarios y validación de contratos de soporte.
- Construir el software usando las buenas prácticas propuestas por la metodología UWE.
- Realizar las pruebas de caja negra, caja blanca y de integración de los diferentes módulos del sistema desarrollado previo paso a producción.
- Desplegar el sistema propuesto en un ambiente de producción siguiendo la arquitectura de software propuesta, con la completa aprobación de los interesados.

### 1.6. Alcance

El sistema propuesto contiene los siguientes módulos:

- **Módulo: Inventario de Repuestos**
  - Gestión de Inventarios. (Creación, actualización de partes y piezas de hardware de los proveedores Fujitsu y EMC).
  - Gestión de envíos por garantía. (Creación y actualización de garantías efectuadas sobre partes y piezas reemplazadas en el sitio de los clientes).
  - Reportes y consultas. (Reportes generales del inventario).
- **Módulo: Contratos de soporte**
  - Gestión de contratos de soporte. (Creación y actualización de contratos de soporte de la empresa Sinetcom S.A. con terceros a favor de

equipos de cómputo, almacenamiento y networking que se hayan adquirido).

- Anclado de partes y equipos a contratos de soporte y ejecución de garantías.
- Reportes y Consultas. (Reportes y consultas generales de contratos).
- Notificaciones sobre fechas de vencimiento de garantías técnicas, plazos de pago y demás fechas importantes en el ciclo de vida del contrato.

- **Módulo: Apertura de Tickets de clientes.**

- Apertura de tickets de soporte de clientes que tiene contrato de soporte sobre los equipos adquiridos en los contratos existentes, tanto vía telefónica como por una interfaz web para apertura de tickets directa por parte de los clientes.
- Administración de casos de soporte, SLAs, notificaciones y control total del ciclo de vida de un ticket de soporte hasta su resolución y cierre.
- Reportes y consultas (Generales).
- Interfaz de validación e ingreso de información para la creación de tickets.
- La validación se hará contra el sistema desarrollado, se validará el serial del equipo con contrato de soporte y garantía.

## **1.7. Metodología**

Un desarrollo de software orientado a la web posee características únicas comparadas a otros tipos de desarrollo de sistemas, como lo es la gran adaptabilidad visual requerida por los usuarios así como los cambios propios de las tecnologías web.

Para el tipo de sistema a construirse se desea que el ciclo de desarrollo de software sea rápido y eficaz. También es necesario que el modelado del sistema sea comprensible para cualquier tipo de persona sin conocimientos previos en el desarrollo de software, de esa forma se obtiene una mejor comprensión del sistema por parte del usuario. El analista y programador también deben beneficiarse de esa comprensión, solicitar retroalimentación del usuario final y trabajar de la mano con él. Tomando en cuenta estos puntos se ha decidido usar una metodología optimizada para desarrollos ágiles de aplicaciones web.

La metodología escogida para el desarrollo de este Sistema será la Metodología de Ingeniería Web Basada en UML (UWE), la cual está enfocada en el diseño sistemático, la personalización y la generación semiautomática de escenarios los cuales guiarán el proceso de desarrollo de la aplicación web.

Una de las técnicas que utiliza esta metodología es una variación del Lenguaje Unificado de Modelado UML, basándose en las técnicas de la misma, como son la notación UML y los mecanismos de extensión UML.

UWE está basado en el proceso unificado de Desarrollo (PUDES) y UML, pero orientados a la generación de aplicaciones web. Para este fin UWE utiliza varios modelos como por ejemplo: el modelo Lógico-Conceptual, el Modelo Navegacional, el modelo de visualización de Escenarios, etc. Y como diagramas puede utilizar el de estados, secuencia, colaboración y actividad.

UWE está especializada en especificar aplicaciones altamente adaptativas, lo que quiere decir que se preocupará con mucho énfasis en las características de personalización.

La fase de Captura, Análisis, y especificación de requisitos es en la cual el equipo de desarrollo básicamente se encarga de reunir, buscar, y

especificar las características que va a tener el aplicativo, clasificándolos en requisitos funcionales y no funcionales.

En esta fase se añade también las especificaciones de las necesidades de información, las necesidades de navegación, la adaptación que va a poseer y por ultimo las necesidades visuales o de interfaz de usuario, centrándose en el estudio de los casos de Uso, generando glosarios de términos y realizando los prototipos de las interfaces de usuario.

En la fase del Diseño del Sistema se la realizará basándose en la especificación de requerimientos obtenida de la primera fase, se procede a detallar como los requisitos van a funcionar y la estructura que tendrá el aplicativo web.

En la fase de codificación del software, se realizará en si el desarrollo de la aplicación web, materializando todo el diseño mediante la programación del aplicativo.

En la fase de pruebas se verificará que todos los requisitos estén implantados en el aplicativo y también el buen funcionamiento de la aplicación para descartar posibles errores en el código.

En la fase de instalación se procederá a instalar la aplicación en el espacio donde va a funcionar de manera apropiada, para así garantizar su funcionamiento al momento de brindar los respectivos servicios al usuario final.

En esta fase está incluida también la puesta en marcha de la arquitectura que el aplicativo utilizará, su o sus repositorios donde estará almacenada la información, en si la integración de todas la tecnologías necesarias para el funcionamiento del aplicativo.



En el mantenimiento se procederá a controlar, mejorar y optimizar el aplicativo ya desarrollado e instalado, para su posible depuración y corrección de errores y defectos que se hubieren filtrado en la fase de Pruebas.

Para este desarrollo se utilizará la metodología UWE, cuyo marco teórico será realizado utilizando las técnicas de investigación Bibliográfica de fuentes de información y la consulta web. Las tareas a realizar para este desarrollo son las dictadas por la Metodología:

- Captura.
- Diseño
- Codificación
- Pruebas
- Instalación.

El desarrollo utilizará exclusivamente la Plataforma JEE, tecnología JSF y aplicando UWE con sus técnicas UML para la fase de Diseño de Modelos.

## **1.8. Herramientas y Software**

La empresa Sinetcom S.A. mantiene la postura de usar herramientas de código abierto para el desarrollo e implantación del sistema propuesto.

Para la implementación del sistema se usarán las siguientes herramientas:

- IDE Netbeans: Plataforma de desarrollo.
- Glassfish Application Server: Servidor de aplicaciones WEB.
- MySQL Workbench: Modelado de Base de Datos y consola SQL.
- StarUML, ArgoUML y MagicDraw: Herramientas de modelado.

- Google SVN: Repositorio de control de cambios sobre código fuente.

El software sobre el cual se ejecutará el aplicativo y sus componentes a continuación:

- Ubuntu Server 12.04 LTS: Sistema Operativo Base del servidor de aplicaciones y motor de base de datos.
- Glassfish Application Server v3.1.2: Servidor de Aplicaciones de producción.
- MySQL Server v5.6: Servidor de Base de datos.
- Zimbra: Servidor SMTP para el envío de correos electrónicos.
- JRE/JDK v7: Plataforma de ejecución de código JAVA.

## **CAPÍTULO 2: MARCO TEÓRICO**

En el presente capítulo se describen los fundamentos teóricos que se han utilizado como parte del desarrollo del presente proyecto, entre los que se detallan la arquitectura Cliente – Servidor, la metodología de desarrollo UWE, las herramientas y software utilizados en las diferentes etapas del desarrollo de software.

### **2.1. Aplicación WEB**

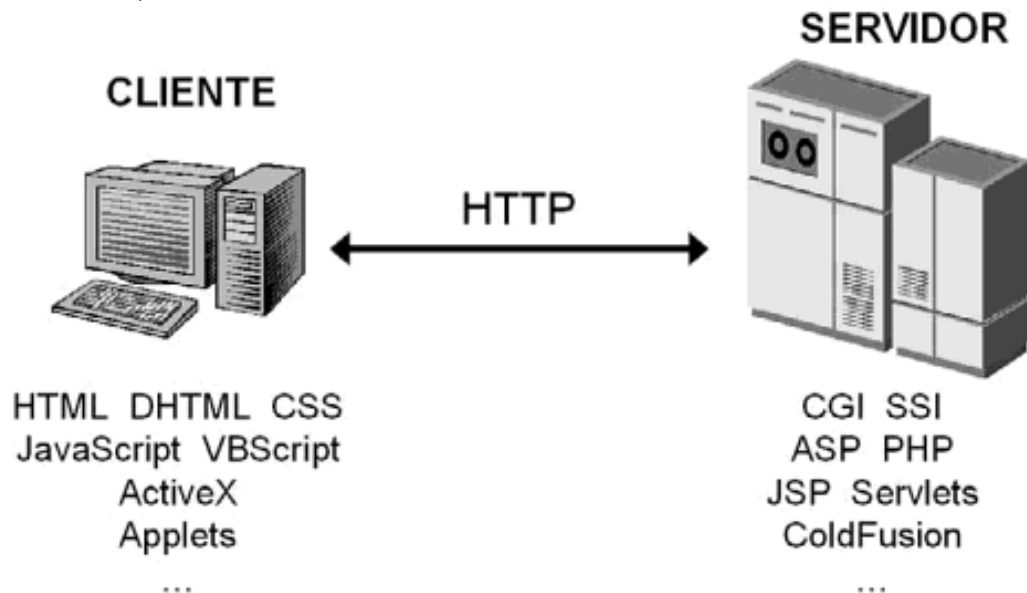
En un principio las aplicaciones empresariales se manejaban a través de aplicaciones de escritorio; sin embargo con el paso del tiempo y con el auge del Internet así como la aceptación global de sus estándares y tecnologías, las posibilidades de accesibilidad se hicieron sumamente grandes, permitiendo que los empleados de una empresa accedieran a sus aplicaciones empresariales no solamente desde sus puntos de trabajo, también desde cualquier parte del mundo. (Luján Mora, 2002) (Caivano & Villoria, 2009).

La Web 1.0 era de solo lectura, se encontraban solamente páginas estáticas con las cuales el usuario no podía interactuar. Con el paso del tiempo y con la introducción de la Web 2.0 esto cambió, las páginas se volvieron dinámicas, la información presentada se extraía ya desde repositorios como bases de datos y el usuario podía interactuar por completo con dichos contenidos. (Ramos, 2011) (Matango & Arevalo, 2014)

#### **2.1.1. Arquitectura Cliente – Servidor**

En una arquitectura cliente – servidor, como se observa en la Figura 1, existe una comunicación bidireccional entre un solicitante (cliente) y un proveedor de recursos o servicios (servidor) a través del protocolo http, ambos

se reparten tareas para llevar a cabo la comunicación. En simples palabras el cliente solicita recursos o servicios y el servidor atiende dicha petición. (Luján Mora, 2002)

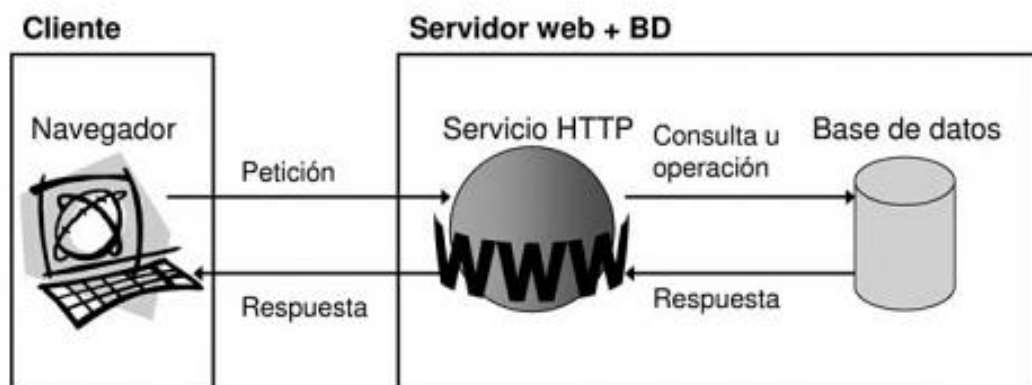


**Figura 1: Arquitectura Cliente – Servidor**

**Fuente:** (Luján Mora, 2002)

Existen dos tipos de servidores: aquellos que no mantienen el estado entre la comunicación cliente – servidor, que es propio del protocolo http, y aquellos que si mantienen el estado, lo que permite que las aplicaciones sean más interactivas con el cliente. (Luján Mora, 2002) (Wikipedia, 2013)

En la Figura 2, se puede apreciar cómo trabaja una aplicación web con la arquitectura Cliente – Servidor a través del protocolo HTTP.



**Figura 2: Arquitectura de las aplicaciones WEB - Un solo servidor**

**Fuente:** (Luján Mora, 2002)

## 2.2. Metodología

UWE fue creado por la investigadora Nora Koch al final del año 1990 con la intención de encontrar un estándar para el análisis y diseño de modelos de aplicaciones basadas en Web a partir de los métodos existentes en esa época, como lo eran OOHDM, RMM y WSDM. UWE se adaptó al estándar UML con el propósito de mantener la notación UML agregándole mecanismos de extensión que definen otros modelos propuestos por UWE. (KOCH & KRAUS, 2002) (Dept. Ciencia de la Computación e IA, 2014).

UWE provee una notación de dominio específico, un proceso de desarrollo de software basado en modelos, y da soporte a lo que se refiere a diseño e implementación de aplicaciones web. UWE reúne el poder del estándar UML en el modelado de software, juntamente con otros estándares OMG como MDA, MOF, OCL y XMI. (KOCH & KRAUS, 2002) (Dept. Ciencia de la Computación e IA, 2014).

Las características principales de UWE son:

- Separación de aspectos: Ayuda a representar diferentes puntos de vista de una aplicación web a partir de diferentes etapas en el ciclo de desarrollo de software.
- Desarrollo enfocado en modelos: El proceso de desarrollo está basado en modelos que ayudan a ver diferentes aspectos de la funcionalidad y presentación del producto software, también UWE enfatiza la transformación en todas las fases del desarrollo de software, esto permite la implementación de un sistema automatizado en múltiples pasos desde varios modelos y niveles de abstracción. (KOCH & KRAUS, 2002)

### 2.2.1. Proceso de diagramación

UWE contiene algunos modelos que son utilizados a lo largo del proceso de desarrollo de software. Estos se los describe a continuación:

- Modelo de casos de uso: Permite visualizar las funcionalidades del software junto a los actores del sistema.
- Modelo de Contenido y Contexto: El objetivo de este modelo es brindar una especificación visual de la información más importante del dominio del sistema. Se modela a través de un diagrama de clases.
- Modelo de navegación: Este modelo se utiliza para representar la estructura navegacional del sistema, incluyendo componentes como menús, links, estructuras de hipertexto, etc.
- Modelo de Proceso: Cada clase de proceso incluido en el modelo navegacional es redefinido en un modelo de proceso que consiste en un flujo de trabajo y opcionalmente en un modelo de estructura de proceso. Este modelo se lo representa en forma de un modelo de actividades de UML.
- Modelo de Presentación: Provee una visión abstracta de la interfaz gráfica del usuario (GUI) de una aplicación web, se basa en el modelo navegacional y abstrae aspectos concretos del UI, como son colores, tipos de letra y ubicación de los componentes gráficos en la página web.

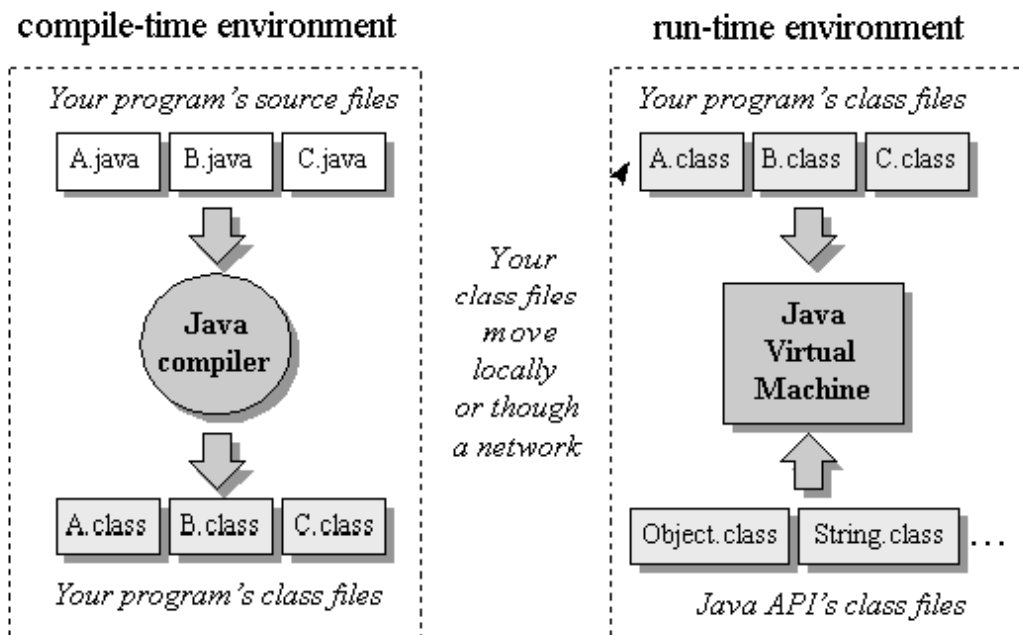
## 2.3. Herramientas y Bases de Datos

### 2.3.1. Plataforma Java

El lenguaje de programación Java fue creado en un inicio por el conocido científico - informático James Gosling como parte fundamental de la plataforma Java de Sun Microsystems empresa donde trabajaba Gosling y que hace pocos años fue adquirida por Oracle. (Wikipedia, 2014)

La sintaxis de este lenguaje de programación es muy similar a la sintaxis de C/C++, pero contiene menos herramientas y utilidades de bajo nivel.

Bill, V. explica que las aplicaciones desarrolladas en Java se ejecutan sobre el JVM (Java Virtual Machine), que es una máquina virtual de proceso nativo que es capaz de interpretar el bytecode (código compilado) de Java en cualquier tipo de plataforma, sin importar el sistema operativo subyacente y la arquitectura del procesador. A continuación se ilustra la compilación de código Java y la función del JVM.



**Figura 3: Compilación de clases JAVA**

**Fuente: Bill, V. (2000) Inside Java Virtual Machine**

Java es un lenguaje de programación orientado a objetos y basado en clases, toda su estructura y set de componentes de software están desarrollados bajo este paradigma de programación.

Entre las características más importantes de Java se encuentran:

- Simple
- Distribuido
- Sólido
- Seguro
- Portable
- Alto desempeño
- Dinámico
- Multi-hilo

Entre los componentes más importantes de JAVA están:

- JRE (Java Runtime Environment), es el entorno sobre el cual se ejecutan las aplicaciones desarrolladas en la plataforma Java, contiene los componentes básicos para la ejecución de aplicaciones Java.
- JDK (Java Development Kit), es un set de herramientas de desarrollo Java que es utilizado por los programadores para poder depurar y documentar sus aplicaciones, adicionalmente este tiene embebido el JRE.
- JVM (Java Virtual Machine), es un contenedor donde se interpreta el código bytecode de Java en cualquier plataforma de hardware y software, contiene adicionalmente un compilador JIT optimizado para la ejecución de las instrucciones de los programas Java.
- APIs, son un conjunto de librerías que contienen un grupo de paquetes que implementan las funcionalidades de Java, entre cada versión de Java se implementan varias mejoras y funcionalidades adicionales.
- Componentes y Librerías, son el conjunto de APIs que conforman la plataforma Java. Dentro de los componentes tenemos las librerías o bibliotecas centrales, bibliotecas de integración, bibliotecas para la interfaz de usuario, plugins, documentación y licencias.



- Extensiones, son otros componentes o librerías que no forman parte del JRE ni JDK pero que extienden la funcionalidad de Java en nuevas tecnologías desarrollados a lo largo del tiempo.

Gracias a todas estas ventajas y por la gran acogida entre la comunidad de programadores, se ha seleccionado JAVA como plataforma de desarrollo.

En la Figura 4, se observa la fama y apertura que posee Java como plataforma de desarrollo y lenguaje de programación alrededor del mundo.



*Figura 4: Estadísticas del uso de Java en diferentes entornos.*

*Fuente: Oracle (2013). Recuperado de <https://www.java.com>*

### 2.3.2. Netbeans IDE

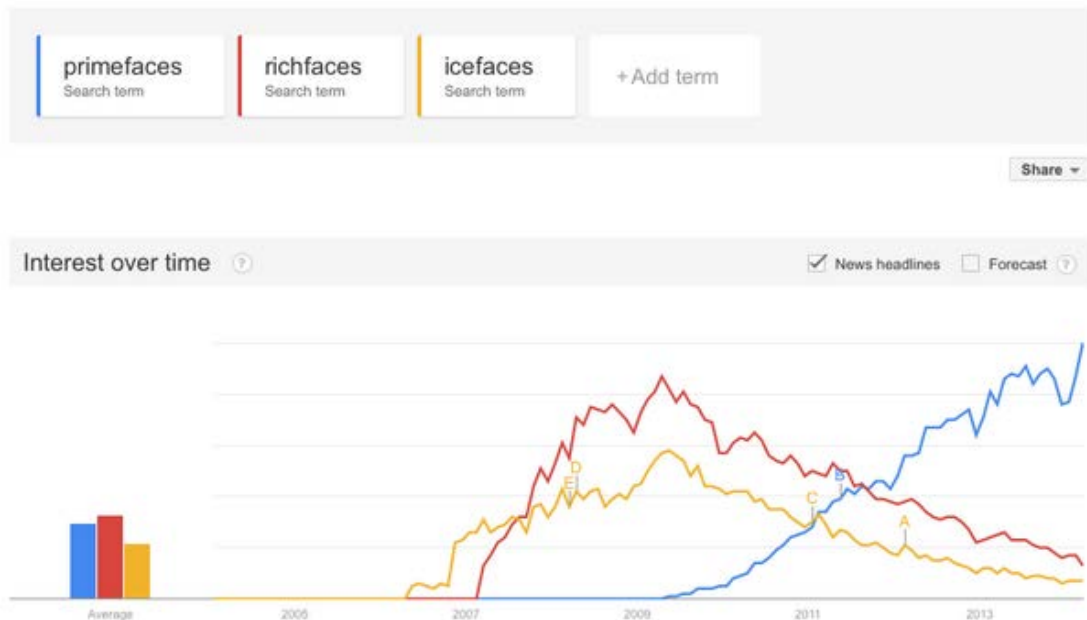
Es un entorno de desarrollo sofisticado que ayuda a los desarrolladores a construir cualquier tipo de aplicaciones ya sean estas aplicaciones de escritorio, empresariales, aplicaciones web y móviles. Este IDE provee a los desarrolladores una variedad de herramientas que facilitan el proceso de diseño e implementación de aplicaciones basadas en la plataforma JAVA y HTML5, también brinda soporte para el desarrollo de aplicaciones como PHP y C/C++. (Böck, 2011) (Corporation, 2014) (Leahy, 2012).

Netbeans nació como una iniciativa emprendida por estudiantes de Checoslovaquia en el año de 1996, fue desarrollado en su pureza con JAVA además de ser el primer IDE en serlo. Posteriormente la empresa Sun Microsystems, viendo el éxito de Netbeans entre la comunidad de programadores JAVA y teniendo el deseo de mejorar sus propias herramientas de desarrollo, apostó por Netbeans convirtiéndolo en un proyecto de código abierto, dándole el apoyo y financiamiento requerido. Más adelante con la compra de la empresa Sun Microsystems por parte de Oracle Corporation, este IDE ha sido sostenido y mejorado en muchos aspectos de funcionalidad y facilidad de uso, convirtiéndose en una de las Plataformas de desarrollo más robustas y confiables. (Böck, 2011) (Corporation, 2014)

### **2.3.3. Componente Visual Primefaces**

PrimeFaces es una librería de componentes visuales libres para JSF que brinda mayor poder a la interfaz gráfica de una aplicación web, ya que los componentes visuales nativos de Java Server Faces son muy limitados al momento de dar variedad y personalización al UI de una aplicación web. La versión más actual de PrimeFaces ofrece un poco más de 100 componentes individuales cubriendo una gran variedad de elementos visuales de interfaz web.

Ha medida que ha pasado el tiempo este componente ha tenido gran acogida entre la comunidad de programadores web que usan la tecnología JSF, esto se puede apreciar en la siguiente gráfica.



**Figura 5: Interés en el tiempo de los componentes visuales de JSF.**

**Fuente: PrimeTek (2014) Primefaces.org**

Entre sus principales características se encuentran:

- **Fácil Configuración:** PrimeFaces se puede integrar con varios IDEs de desarrollo como Eclipse y Netbeans, también se los puede encontrar en el conocido repositorio Maven como una dependencia que puede ser incluida en el proyecto, como es el caso en este proyecto.
- **Mayor cantidad de componentes:** PrimeFaces comparado con sus competidores como RichFaces y IceFaces, posee una gama de componentes más amplia y potente. PrimeFaces está basado en JQuery por lo que sus componentes visuales son muy atractivos y muy amigables para el usuario final.
- **Mayor documentación y ayudas en línea:** PrimeFaces posee documentación de fácil acceso, además se puede encontrar ejemplos en la página oficial que son muy útiles al momento de codificar con esta librería.

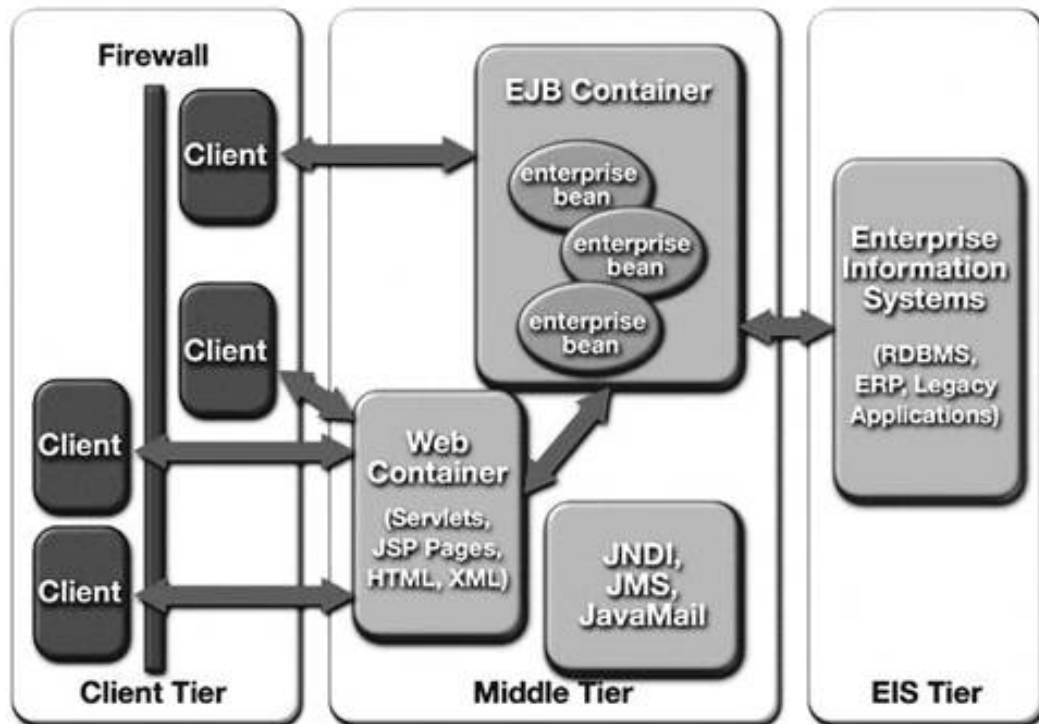
#### 2.3.4. JEE 6

La especificación Java EE (Enterprise Edition) fue desarrollada e impulsada por SunMicrosystems (ahora Oracle) hasta el día de hoy bajo el Java Community Process con la colaboración de expertos de la industria informática, organizaciones de código abierto así como comerciales, equipos de desarrollo de Java y personas alrededor del mundo. (Oracle Corporation, 2001)

La especificación J2EE la versión 5 cumple los requerimientos del JSR 244 y su lanzamiento final fue en mayo del 2006. Y por último la versión Java EE 6 se lanzó en el 2009. (Wikipedia.org, 2014) .

Entre las características más importantes de JEE se encuentran:

- Es una plataforma de programación JAVA con la diferencia que está utilizada para el desarrollo de aplicaciones empresariales por su capacidad de escalabilidad y la posibilidad de programación de aplicaciones distribuidas.
- A esta versión Enterprise se la considera un estándar por la JCP o Java Community Process.
- Está completamente orientado a desarrollar aplicaciones empresariales.
- En la siguiente Figura 6, se observa de forma global como se encuentra estructurado el estándar JEE.



*Figura 6: Modelo JEE*

*Fuente disponible en:*

*<http://www.sicuma.uma.es/es/formacion/doctoradoargentina/independientes/argentina08/Badaracco/j2ee.htm>*

Por sus grandes bondades, la plataforma Java Enterprise es la más indicada para el sistema a ser desarrollado ya que aparte de ser totalmente compatible con el servidor de aplicaciones escogido, Glassfish; y la base de datos, MySQL. Permitirá el desarrollo en capas con los estándares de calidad de una aplicación empresarial.

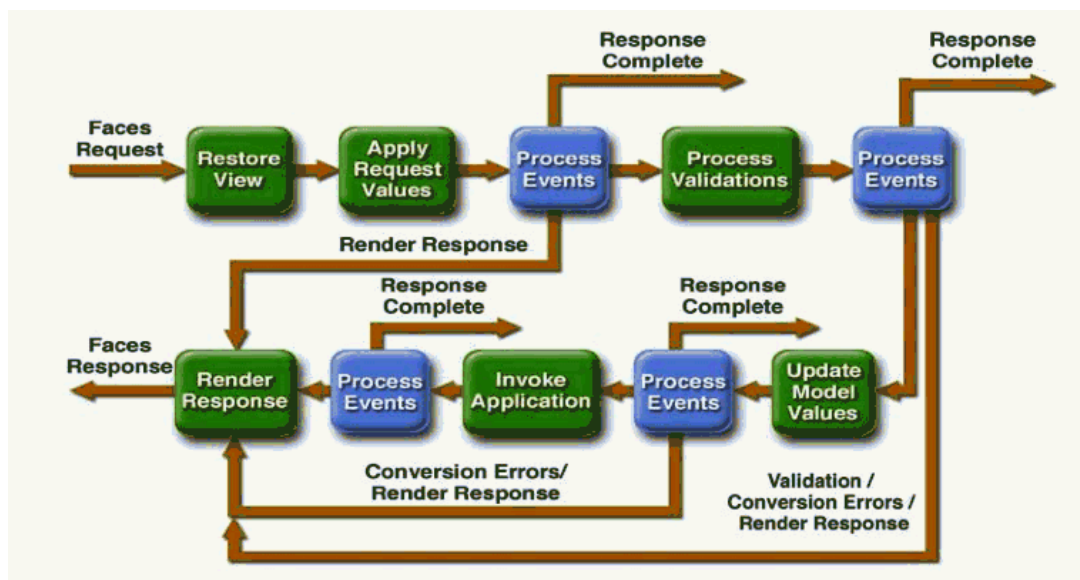
### **2.3.5. JSF – Java Server Faces**

JSF o Java Server Faces es un framework el cual utiliza una arquitectura basada en componentes para el desarrollo reutilizando componentes de interfaz de usuario, permitiendo así ocultar la programación compleja de la vista en los sistemas basados en la arquitectura MVC. Este framework no solo

se limita al desarrollo de interfaces de usuario sino también soporta características avanzadas tales como el Manejo de Eventos, y la validación de datos al momento que el usuario los ingresa. (Srinivasan, 2007).

Los principales elementos de JSF son los componentes de Interfaz de Usuario, Los ManagedBeans, Validadores, Conversores, Eventos, La navegación de la Página y los Renderers. (Srinivasan, 2007).

Una aplicación JSF posee varias fases durante la interacción del usuario con la aplicación web, en la Figura 7, se observa como es el flujo del proceso o ciclo de vida de una petición JSF.



**Figura 7: Renderizar la respuesta JSF.**

. Fuente: (Departamento Ciencia de la Computación e IA., s.f.)

La Tecnología JSF permite crear interfaces empresariales de una manera sencilla y sobre todo segura, ya que al ser un framework permite tener la tranquilidad de que su desarrollo cumple ciertos requerimientos de seguridad. Esto permite una mayor concentración en el diseño y deja de lado la seguridad de la aplicación la cual ya se encuentra implementada en esta. Otra ventaja es que permite validar la información al ser ingresada por el usuario para evitar

problemas futuros con el procesamiento de la información. (Departamento Ciencia de la Computación e IA., n.d.) (Loor, 2014) (Burns, Schalk, & Griffin, 2010).

### **2.3.6. MySQL Server – Servidor de base de datos**

MySQL, es un motor de bases de datos, sus creadores son MySQL AB de procedencia Sueca los cuales disponen del copyright de su código fuente y de su marca, en 2008 fue adquirida por Sun Microsystems y subsiguientemente en 2009 por Oracle Corporation. (Wikipedia.org, 2013).

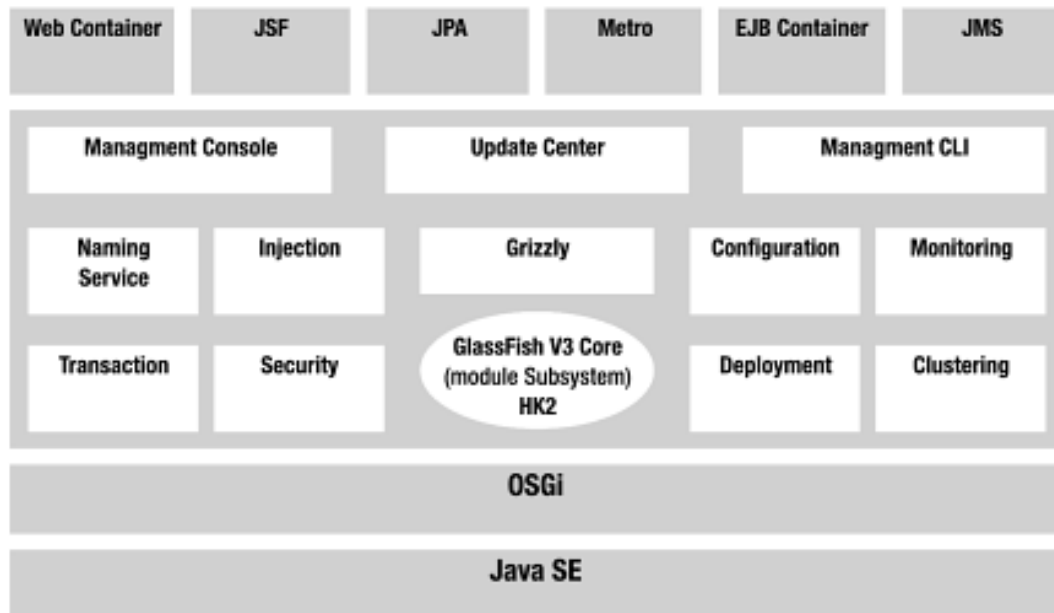
MySQL utiliza una licencia gratuita, se encuentra licenciado bajo GLP de la GNU. MySQL es de uso gratuito sin embargo también tiene una versión comercial, con la única diferencia que es el soporte que esta versión dispone y la integración de esta en software propietario. (Enríquez Toledo, Maldonado Ayala, Nakamura Ortega, & Nogueron Toledo).

Entre las principales características podemos enlistar que es un servidor veloz y robusto, soporta gran cantidad de datos, es multiplataforma, y aprovecha muy bien la potencia del host sobre el que se encuentra instalado. (Enríquez Toledo, Maldonado Ayala, Nakamura Ortega, & Nogueron Toledo).

### **2.3.7. Glassfish AS. – Servidor de Aplicaciones**

Es un servidor de aplicaciones web desarrollado para la plataforma JEE, en un inicio fue un proyecto impulsado por Sun Microsystems y posee varias características heredadas del servidor Tomcat. Actualmente existen dos tipos de versiones soportadas por Oracle; la versión “Community” o más conocida como “Open Source Edition” y la versión “Oracle” que posee asistencia técnica y requiere licenciamiento especial. (Oracle Corporation, 2014) (Goncalves, 2010).

Glassfish posee soporte para la tecnología JSF y posee una compatibilidad completa con el IDE de desarrollo Netbeans tanto en las implementaciones de JEE 6 y JEE7. Glassfish posee una consola de administración muy intuitiva y de fácil uso, su arquitectura se presenta de forma global en la Figura 8. (Goncalves, 2010).



**Figura 8: Arquitectura de Glassfish AS.**

*Fuente: (Goncalves, 2010)*

### 2.3.8. StarUML

Es una herramienta de modelado de software, es de código abierto que soporta UML (Unified Modeling Language). Está basada en UML 1.4 pero soporta también la notación UML 2.0, en sus nuevas versiones lanzadas al mercado contiene ya la compatibilidad con los estándares de UML 2.x. Soporta un total de 11 diagramas UML y permite la generación de código a varios lenguajes de programación a través del concepto de perfil UML. (Martinig, 2011) (StarUML Official Site, 2014).



Es una herramienta que ha sido usada ampliamente por empresas e individuos en el modelamiento de aplicaciones informáticas gracias a su simplicidad y facilidad de uso.

### **2.3.9. ArgoUML**

Es una herramienta de modelado de software altamente usada por los equipos de desarrollo de software, brinda soporte a todo el estándar de diagramas UML 1.4 y cuenta con varias extensiones de software que le dan mayor funcionalidad. (Tigris.org, 2001).

En este proyecto se ha utilizado ArgoUML juntamente con el plugin ArgoUWE disponible en la web para realizar el modelado de algunos de los diagramas propios de la metodología UWE.

### **2.3.10. MagicDraw**

Es una herramienta CASE visual muy potente que requiere de un tipo de licenciamiento de software propietario para su correcto funcionamiento y posee un equipo de soporte técnico para aquellos que adquieran el software. Está basado en Java de forma que se puede ejecutar en prácticamente cualquier sistema operativo que tenga instalado el JVM (Java Virtual Machine). Con esta herramienta se puede generar código fuente a partir de modelos y diagramas UML. (The University of Texas At Austin, 2001).

Posee una gran variedad de plugins disponibles en internet que extienden su funcionalidad. Uno de ellos es MagicUWE que fue utilizado en cierta parte del presente proyecto.

## **2.4. Mesa de Ayuda**

La mesa de ayuda es uno de los varios servicios de TI que implementa la disciplina de administración de servicios de TI (ITSM) como se define en el marco de trabajo ITIL. Según ITIL la intención o misión de la mesa de ayuda es el poder ser un punto central de contacto entre el usuario y el proveedor de servicios de TI para poder cumplir los objetivos del negocio y brindar satisfacción al cliente. (ITIL Resources, 2008) .

Básicamente lo que persigue la mesa de ayuda es poder dar resolución a las incidencias o problemas que ha tenido el cliente con determinada infraestructura de TI. Al observar el manejo de soporte técnico dentro de los grandes fabricantes de TI, se ve claramente que todos han adoptado este esquema de trabajo para poder dar seguimiento y resolución a casos de soporte de sus clientes finales, cumpliendo de esta forma con los acuerdos de nivel de servicio contratados.

## **2.5. Ingeniería de Requerimientos IEEE 830**

Es un estándar impulsado por la IEEE con el fin mantener un orden y un conocimiento pleno de los requerimientos funcionales y no funcionales de software que un cliente desea obtener a través de un desarrollo de software.

Básicamente la Especificación de Requerimientos de Software ayuda a los clientes a describir claramente lo que desea obtener con un software, sirve además como punto de partida para el diseño y la construcción del software y si se lo realiza de forma correcta desde las primeras fases del ciclo de desarrollo de software, permite la reducción de costos, esfuerzo y tiempo dentro de un proyecto. (Vidal, 2007).

Según (Matango & Arevalo, 2014) el proceso de ingeniería de requerimientos se divide por etapas, las cuales son:

- **Identificación de Requerimientos:** Es una de las etapas más importantes y complejas dentro de la ingeniería de requerimientos, ya que es donde se extraen todos los requerimientos de software que desean obtener los usuarios de un sistema. Se usan varias técnicas como entrevistas o lluvias de ideas para esta extracción de requerimientos.
- **Análisis y Negociación:** En esta etapa se identifican los requerimientos de software y se agrupan por complejidad y criticidad. Se interactúa con el usuario para negociar y definir la estructura final de los requerimientos.
- **Especificación de requerimientos:** Se emite un documento formal donde se especifican los detalles de la funcionalidad del sistema donde se incluyen funciones, limitaciones y características que han de gobernar el sistema.
- **Validación de Requerimientos:** Se deben especificar los requerimientos con consistencia y completitud, no deben existir ambigüedades ni malas interpretaciones de los requerimientos. Cualquier falla detectada debe ser corregida.

En el presente proyecto se realizó un documento completo como especificación de requerimientos de software (ERS) la cual está disponible en el Anexo A.

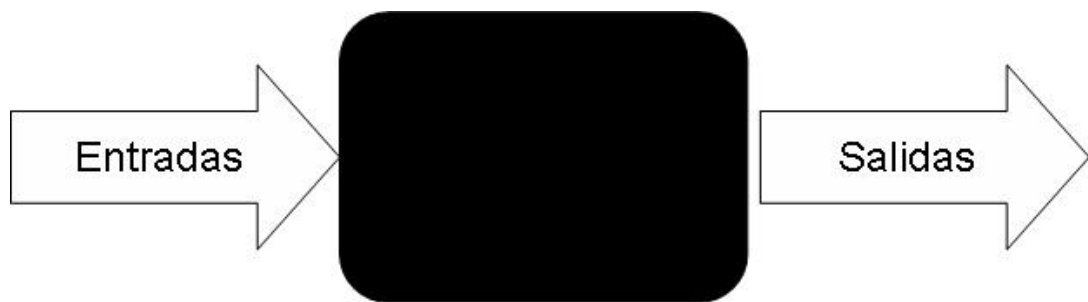
## **2.6. Pruebas**

En el presente proyecto se hará el uso de pruebas de caja negra las cuales permiten probar la funcionalidad del sistema juntamente con el usuario final. Por otro lado se hará el uso de pruebas de caja blanca para verificar que el procesamiento de datos dentro del código fuente vaya de acuerdo a las salidas esperadas.

### 2.6.1. Pruebas Caja Negra

Consiste en que un “tester” haga el ingreso de datos y se limite a ver los resultados obtenidos, sin necesidad de observar que es lo que sucede en el interior de la aplicación, este tipo de pruebas son también conocidas como pruebas de caja “opaca” o pruebas de funcionalidad. (Gómez Adán, 2012)

En la Figura 9, se observa cómo se da este comportamiento.



*Figura 9: Concepto de Prueba de Caja Negra.*

*Fuente: (Gómez Adán, 2012)*

En el presente proyecto se observarán algunos casos de prueba de caja negra que se realizarán con el usuario final del sistema.

### 2.6.2. Pruebas de caja blanca

También se las denomina pruebas estructurales, utilizan el código fuente del sistema y especialmente su estructura de control, para seleccionar sus casos de prueba.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO

En este capítulo se explica cómo se ejecutó los procesos de captura de requerimiento de software y diagramación del sistema.

### 3.1. Historial de Reuniones

En la siguiente tabla están detalladas las reuniones realizadas con la empresa Sinetcom S.A. con el objeto de capturar los requerimientos de software del sistema desarrollado.

**Tabla 1: Historial de Reuniones**

Número de Reunión	Participantes	Fecha	Inicio	Fin	Descripción
1	Stakeholders: <ul style="list-style-type: none"> <li>• Presidente de la compañía</li> <li>• Gerente General</li> <li>• Ayudante de Gerencia</li> <li>• Agente de Importaciones</li> <li>• Gerente Técnico</li> <li>• Analistas / Implementadores</li> </ul>	3/02/2014	11:00	14:00	Objetivos <ul style="list-style-type: none"> <li>• Lluvia de ideas</li> <li>• Análisis de los verdaderos desafíos de la empresa.</li> <li>• Expectativas del software a desarrollarse</li> <li>• Análisis del soporte técnico de la Empresa.</li> </ul> Resultados <ul style="list-style-type: none"> <li>• Se logró identificar las necesidades de la empresa respecto a los procesos que manejan en la parte de soporte técnico, manejo de refacciones y contratos de soporte.</li> <li>• Se identificó las actividades manuales de cada proceso y se propuso soluciones a través del sistema a desarrollarse.</li> </ul>

Continúa 

2	<p>Stakeholders:</p> <ul style="list-style-type: none"> <li>• Presidente de la compañía</li> <li>• Ayudante de Gerencia</li> <li>• Agente de Importaciones</li> <li>• Analistas / Implementadores</li> </ul>	10/02/2014	10:30	14:30	<p>Objetivos:</p> <ul style="list-style-type: none"> <li>• Conocer correctamente el proceso de soporte técnico y el manejo de garantías de partes y piezas de equipos de cómputo.</li> <li>• Recopilar toda la información necesaria de soporte técnico para proponer una solución de software a dicha área de la empresa.</li> </ul> <p>Resultados:</p> <ul style="list-style-type: none"> <li>• Se logró conocer completamente el proceso de soporte técnico.</li> </ul>
3	<p>Stakeholders:</p> <ul style="list-style-type: none"> <li>• Ayudante de Gerencia</li> <li>• Agente de Importaciones</li> <li>• Analistas / Implementadores</li> </ul>	03/03/2014	10:00	12:00	<p>Objetivos:</p> <ul style="list-style-type: none"> <li>• Conocer bien el proceso de importaciones y sobre todo el manejo de los inventarios.</li> </ul> <p>Resultados:</p> <ul style="list-style-type: none"> <li>• Se identificaron las actividades que se podrían automatizar en la búsqueda inventarios y generación de documentos de control de cambios de partes y piezas sobre clientes.</li> </ul>
4	<p>Stakeholders:</p> <ul style="list-style-type: none"> <li>• Agente de Importaciones</li> <li>• Analistas / Implementadores</li> </ul>	19/03/2014	09:00	10:30	<p>Objetivos:</p> <ul style="list-style-type: none"> <li>• Concretar las funcionalidades que tendrá el software respecto al control de inventarios tecnológicos.</li> </ul>


 Continúa

					<b>Resultados:</b> <ul style="list-style-type: none"> <li>Se llegó a un acuerdo con el cliente sobre las actividades que se van a poder automatizar y el tiempo que puede ahorrar al usar la funcionalidad en el nuevo sistema. No quedaron pendientes respecto a esta parte del sistema.</li> </ul>
5	<b>Stakeholders:</b> <ul style="list-style-type: none"> <li>Ayudante de Gerencia</li> <li>Analistas / Implementadores</li> </ul>	21/03/2014	09:30	11:00	<b>Objetivos:</b> <ul style="list-style-type: none"> <li>Concretar las funcionalidades que tendrá el software respecto al control y validación de contratos que maneja la compañía con sus clientes.</li> </ul> <b>Resultados:</b> <ul style="list-style-type: none"> <li>Se llegó a un acuerdo con el cliente sobre las funcionalidades que haría el sistema y como ayudaría en la automatización de actividades muy rutinarias y demorasas como lo es la consulta de cláusulas de un contrato.</li> </ul>

### 3.2. Especificación de Requerimientos de Software

La ERS fueron realizadas con el estándar de la IEEE 830 y se lo tiene disponible en el Anexo A.

La ERS contiene la siguiente estructura:

- Introducción
  - Propósito
  - Alcance
  - Definición de acrónimos y abreviaturas.
- Descripción General
  - Perspectivas del Producto
    - Interfaces de Usuario
    - Interfaces de Hardware
    - Interfaces de Software
    - Interfaces de Comunicación
    - Restricciones de Memoria
  - Funciones del Producto
  - Características de los Usuarios
  - Restricciones de diseño
  - Supuestos y dependencias
- Requerimientos Específicos
  - Requerimientos Funcionales
  - Requerimientos no funcionales
- Firmas de validación

Si se desea observar detalladamente este documento dirigirse al ANEXO

A.

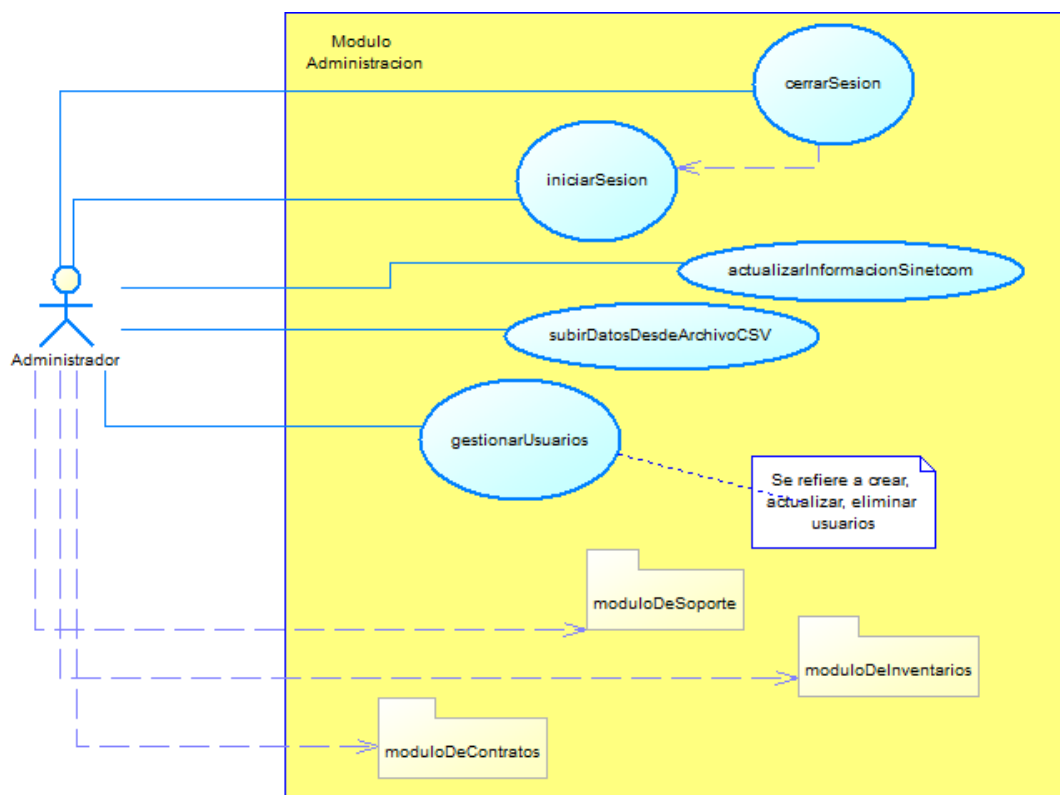


### 3.3. Diagramas UWE

#### 3.3.1. Diagrama de casos de uso

Los diagramas de casos de uso muestran de forma clara y sencilla la funcionalidad que tiene un software, es muy útil cuando se quiere identificar todas las funciones que hará un actor o usuario del sistema, además es fácil de entender. Todas las personas sin importar su conocimiento en modelado de sistemas puede comprender uno de estos diagramas UML.

En la (*Figura 10, Figura 11, Figura 12*) disponible en el ANEXO B y la Figura 13, se observa cada uno de los diagramas de casos de uso, según los módulos existentes en el sistema.



**Figura 13: Diagrama de Caso de Uso - Módulo de Administración.**

### **3.3.2. Diagrama de Clases**

El diagrama de clases es un diagrama que define el modelo de los objetos que se usarán para identificar las entidades de la Base de Datos, en este desarrollo se utilizó ORM para la generación de las clases. Esto se lo puede apreciar en la *(Figura 14: Diagrama de Clases)* del Anexo B.

### **3.3.3. Diagramas de Secuencia**

A continuación se muestran algunos de los diagramas de secuencia más representativos del sistema. Estos diagramas permiten ver la interacción del usuario en una operación con las diferentes capas del sistema.

## Crear Ticket

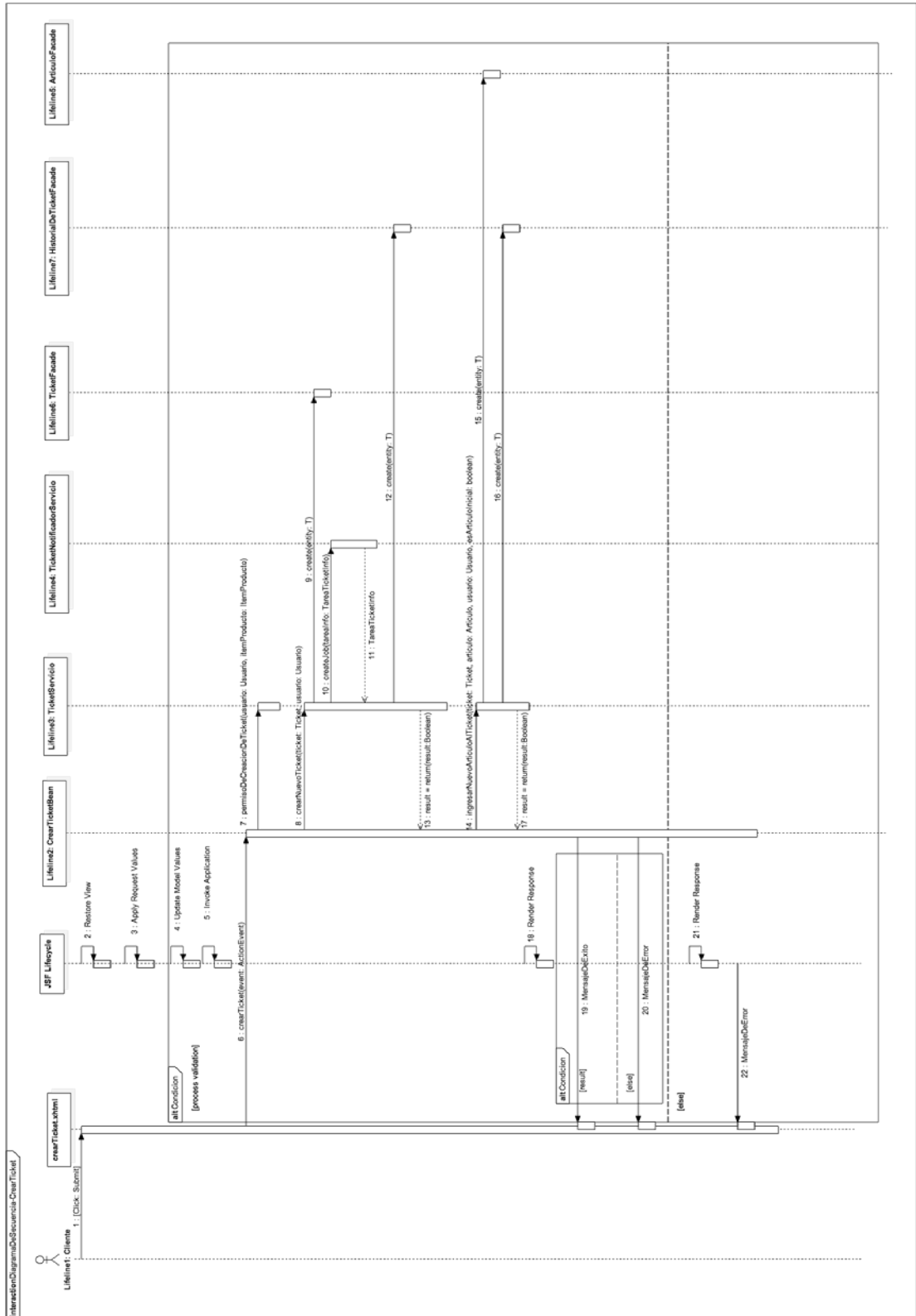


Figura 15: Diagrama de Secuencia - Crear.

## Ingreso de Inventario

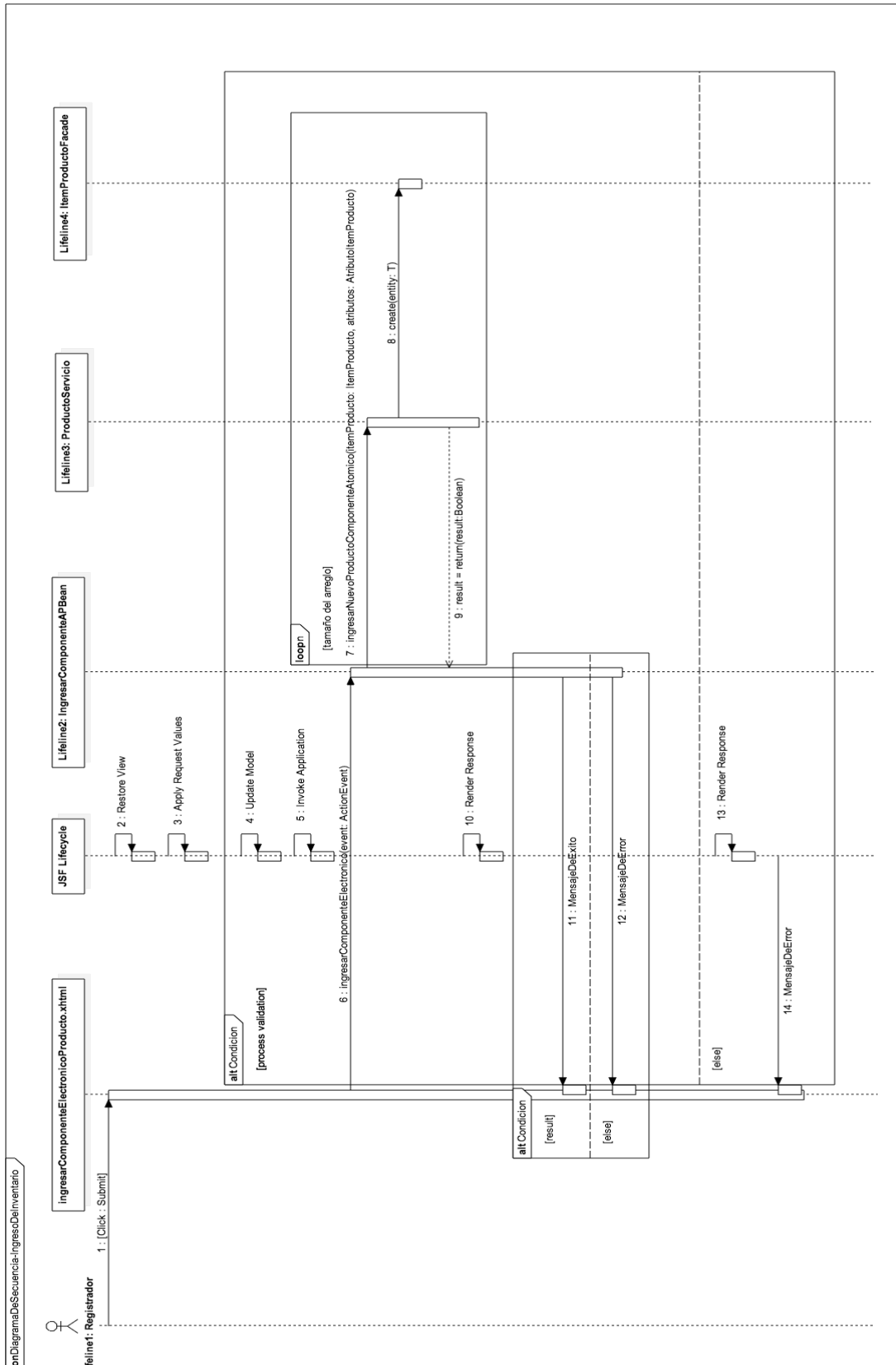


Figura 16: Diagrama de Secuencia - Ingreso Componente Electrónico.

### 3.3.4. Diagrama de Despliegue y Componentes

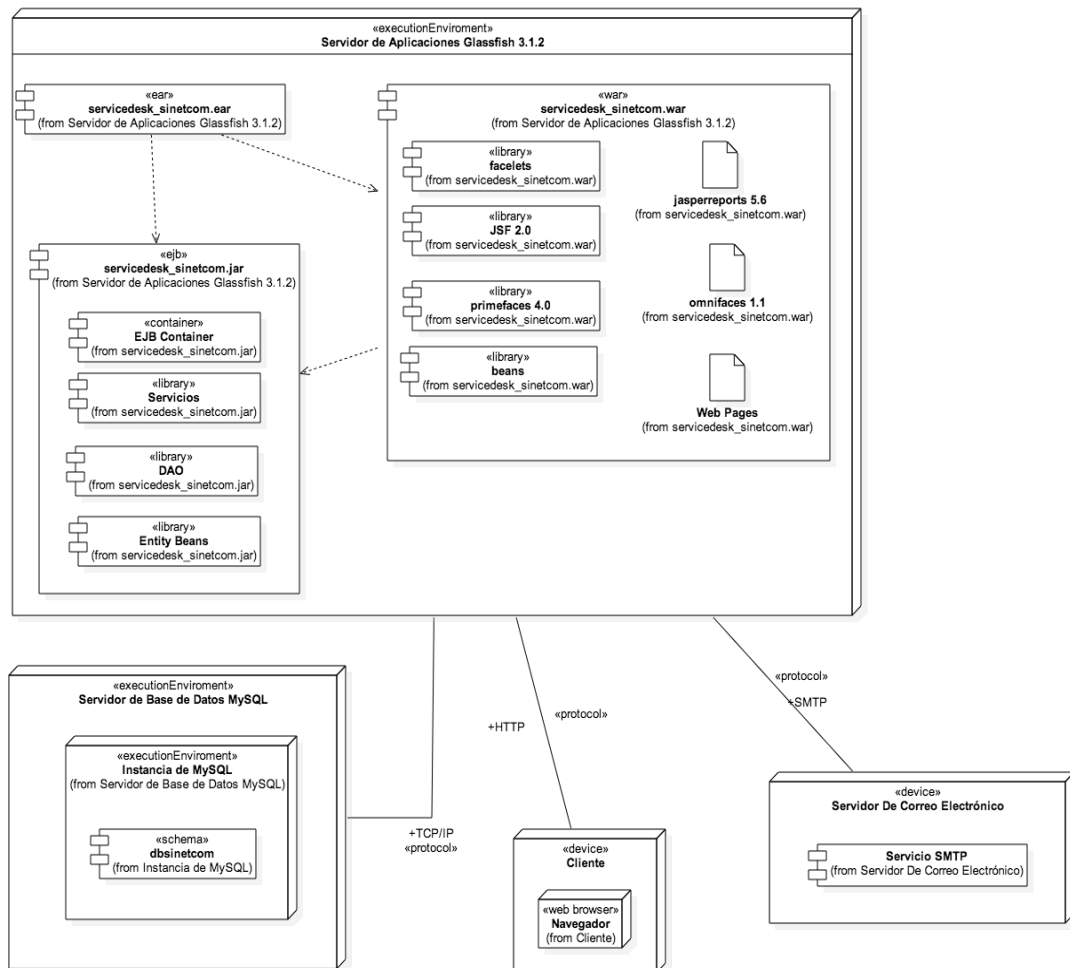


Figura 17: Diagrama de Despliegue y Componentes.

### 3.3.5. Diagrama Conceptual

El diagrama conceptual es un diagrama de clases normal, donde se pueden observar de mejor forma las relaciones de agregación y composición que existen entre las diferentes clases de modelo dentro del sistema.

En la (Figura 18: Diagrama Conceptual) del anexo B se puede apreciar lo mencionado previamente.

### 3.3.6. Diagrama de Navegación

Este diagrama representa precisamente la navegación de la aplicación a nivel macro. Se arranca desde el menú principal y se desglosa sobre las clases más importantes del sistema, aquí se representan las acciones y búsquedas que se pueden ejecutar sobre estas clases. En la (*Figura 19: Diagrama Navegación*) del anexo B se observa las diferentes opciones de navegación que tiene el sistema sobre las clases de modelo más importantes del sistema.

### 3.3.7. Diagrama de Presentación

El modelo de presentación representa la estructura visual del sistema, donde se incluyen todos los componentes visuales de un formulario web como lo son las cajas de texto, listas de selección, botones, etc. También se puede visualizar los contenedores, tipo de plantilla y menú principal que tiene el sistema. Esto lo podemos observar en la (*Figura 20: Diagrama de Presentación*) del anexo B.

### 3.3.8. Modelo Físico de Base de Datos

Este diagrama representa la estructura de la base de datos como claves primarias y foráneas así como la multiplicidad en las relaciones de las diferentes entidades o tablas de la base de datos. Esto se observa en la (*Figura 21: Modelo Físico de Base Datos*) del anexo B.

## CAPÍTULO 4: IMPLEMENTACIÓN

### 4.1. Construcción Base de Datos

La construcción de la base de datos se puede visualizar de forma detallada en el Script que se encuentra en el Anexo C.

### 4.2. Arquitectura de la aplicación

Esté desarrollo se realizó con una arquitectura de N Capas, basado en cliente – servidor. En el diagrama de despliegue (Figura 17) se observa que existe un capa de lógica de negocio ejecutándose sobre un servidor de aplicaciones Web y por otro lado se encuentra la capa de persistencia con un servidor de base de datos MySQL, la comunicación entre estos dos se realiza vía TCP/IP y con el cliente se utiliza un protocolo http sobre una red Ethernet.

En la Figura 22, se observa la arquitectura de la aplicación de forma secuencial.



*Figura 22: Arquitectura Sinetcom App.*

- **Estándar de Nomenclatura**

Para la elaboración del modelo de datos no se estableció ningún estándar de nomenclatura más que la singularidad de los sustantivos que se utilizan

para representar una entidad, no se emplean prefijos o sufijos en las entidades.

En este proyecto se utilizó la técnica de ORM (Object Relational Mapping) con el que se ha logrado hacer un mapeo de las entidades de la Base de Datos a las clases de modelo. Por lo que los sustantivos utilizados en las clases son iguales a los de la base datos.

- **Herramientas de desarrollo**

Para el desarrollo de este sistema se utilizó el IDE de desarrollo Open Source Netbeans. Se eligió este IDE por ser una herramienta libre y por la facilidad de uso que tiene frente a otros IDEs, como lo es Eclipse, también posee una gran cantidad de plugins que ayudan a que el desarrollo de un sistema sea más rápido y flexible. Este IDE es muy estable y posee una versión optimizada para el desarrollo de aplicaciones JEE.

- **Motor de Base de Datos**

Se usó MySQL Server como motor de base de datos del sistema desarrollado. Debido a la cantidad de carga, número de usuarios y tipo de aplicación no se optó por otro tipo de motor de base de datos más robusto como lo es ORACLE o PostgreSQL.

- **Servidor de Aplicaciones WEB**

Se usó el servidor Glassfish 3.1.2 que es una versión probada y muy estable mantenido por ORACLE. Se eligió este servidor de aplicaciones por la completa integración que tiene con el IDE de desarrollo utilizado y por sus varias ventajas como un óptimo consumo de CPU y de memoria según



estudios realizados. Glassfish también tiene una consola de administración WEB muy amigable por lo que es de fácil configuración y soporta completamente JEE.

### 4.3. Codificación

En el desarrollo de este sistema se utilizó el lenguaje de programación JAVA juntamente con la tecnología JSF 2.0 sobre la plataforma JEE.

En la capa de persistencia se utilizó JPA para el manejo de las transacciones y consultas de la base de datos. En la capa de negocio se utilizaron los Session Beans del contenedor EJB para codificar el núcleo del sistema. En la parte visual del sistema se utilizó el Framework PrimeFaces 4.0 juntamente con JSF.

En la Figura 23, se muestra unas pocas líneas de código de la implementación de un EJB.

```

21  @Stateful
22  @StatefulTimeout(unit = TimeUnit.MINUTES, value = 10)
23  @LocalBean
24  public class AutenticacionServicio {
25
26      @EJB
27      private UsuarioFacade usuarioFacade;
28
29      private Usuario usuario;
30
31      @PostConstruct
32      public void inicializarServicio() {
33          usuario = new Usuario();
34      }
35
36      /**
37       * Permite validar la existencia del usuario
38       * @param nombreUsuario
39       * @param password
40       * @return
41       */
42      public Usuario validarUsuario(String nombreUsuario, String password){
43          UtilidadDeEncriptacion utilidadDeEncriptacion = new UtilidadDeEncriptacion();
44          String passwordEncriptado = utilidadDeEncriptacion.encriptar(password);
45          this.usuario = this.usuarioFacade.verificarExistenciaDeUsuario(nombreUsuario, passwordEncriptado);
46          if(this.usuario != null){
47              return this.usuario;
48          }else{
49              return null;
50          }
51      }
52  }
53  // Add business logic below. (Right-click in editor and choose
54  // "Insert Code > Add Business Method")
55
56  }
57

```

**Figura 23: Implementación de EJB.**

### 4.3.1. Instalación de la Aplicación

La instalación se llevó a cabo en un servidor virtual con sistema operativo UBUNTU Server 12.04 LTS, ejecutándose sobre un servidor de Rack DELL PowerEdge R710 (Servidor de producción Con VMWare ESXi 5.1). Este servidor contiene también el servidor de correo electrónico Zimbra que maneja la empresa, por lo que la carga se distribuye únicamente sobre estos 2 servidores virtuales.

Los requisitos de software a ser instalados en el servidor son los siguientes:

- Glassfish 3.1.2
- JDK 6/7 se recomienda la última actualización en cualquiera de las versiones.
- Base de Datos MySQL 5.4 o superior

Las características de hardware del servidor donde se instaló el sistema son:

- Procesador Intel Xeon de 2,393 GHz con 4 núcleos.
- 32 GB de memoria RAM
- 1,8 TB de disco interno, con posibilidad de aumentar en capacidad a través de una SAN
- 2 Interfaces de red Ethernet 10/100/1000 Mbps – Full Duplex

Características del servidor virtual:

- Sistema Operativo Ubuntu Server 12.04 LTS

- 400GB de disco
- 1 Procesador 4 núcleos
- 1 Interfaz de Red (PortGroup0 ; vmnic1)
- 8 GB de memoria RAM

Dentro del servidor Ubuntu virtual se instaló el servidor de aplicaciones Glassfish 3.1.2 con un frontend Apache Web Server para la encriptación del tráfico http y para proxy del puerto 80 al puerto 8080 donde se ejecuta el sistema.

En este mismo servidor virtual se ha colocado el servidor de base de datos MySQL Server 5.4 con la base de datos de producción.

Para la creación de la base de datos se copió el script de creación en la raíz del usuario root y se ejecutó el comando de creación según se muestra en la Figura 24.

```

Last login: Fri Mar 13 09:41:31 2015 from 192.168.150.91
sinetcom@app:~$
sinetcom@app:~$ ls
UnlimitedJCEPolicy                mysql-5.6.16-debian6.0-x86_64.deb
UnlimitedJCEPolicyJDK7.zip        mysql-apt-config_0.3.2-1ubuntu12.04_all.deb
file.php?id=454497               mysql-connector-java-5.1.22-bin.jar
glassfish-3.1.2-unix-el.sh       sinetcomDBFull.sql
sinetcom@app:~$ history
 1  ls
 2  history
sinetcom@app:~$ mysql -u root -pXXXXX < sinetcomDBFull.sql

```

**Figura 24: Creación de Base de Datos.**

Como se ingresa a la consola de administración web de Glassfish, en este caso es <https://192.168.150.52:4848>, al desplegar la página web se debe ingresar los datos de autenticación y como siguiente paso en la Figura 25, se muestra los datos de configuración de la conexión de la base de datos con el servidor de aplicaciones Glassfish que se debe realizar.

Propiedades Adicionales (7)	
Nombre	Valor
<input type="checkbox"/> URL	jdbc:mysql://192.168.150.52:3306/dbsinetcom
<input type="checkbox"/> driverClass	com.mysql.jdbc.Driver
<input type="checkbox"/> Password	
<input type="checkbox"/> portNumber	3306
<input type="checkbox"/> databaseName	dbsinetcom
<input type="checkbox"/> User	root
<input type="checkbox"/> serverName	192.168.150.52

**Figura 25: Configuración de MySQL en Glassfish.**

Para desplegar la aplicación empresarial JEE del Maven se utiliza la consola de administración de igual forma. En la Figura 26, se observa la forma de hacerlo.

**Desplegar Aplicaciones o Módulos**  
Especifique la ubicación de la aplicación o del módulo que desea desplegar. Una aplicación puede estar en un archivo empaquetado o se puede especificar como directorio.

Ubicación:  Archivo Empaquetado que Cargar en el Servidor  
 Archivo empaquetado local o directorio accesible desde GlassFish Server

Tipo: \*

Nombre de Aplicación: \*

Servidores Virtuales:

Estado:  Activada

Java Web Start:  Activada

Precompilar JSP:

Ejecutar Verificador:

Compatibilidad:

Forzar Nuevo Despliegue:

Mantener Estado:

**Figura 26: Despliegue de aplicación .ear en Glassfish.**

Los usuarios que deseen entrar al sistema residente en el Datacenter de Sinetcom S.A., deben hacerlo a través de la única IP Pública que se posee por el momento, a través del puerto 4443 con el nombre de dominio support.sinetcom.com.ec registrado en el proveedor de internet de la empresa y sobre el protocolo https.

La aplicación se encuentra actualmente probada y en producción.

#### **4.3.1.1. Hosting**

La aplicación web se encuentra alojada en un servidor propietario de la compañía dentro de su Datacenter principal. El sistema es accesible tanto desde la intranet como del internet por la única IP pública que posee la compañía. Esta se encuentra accesible a través de la página web de la empresa en la pestaña de soporte.

#### **4.4. Pruebas**

Se usaron las pruebas de Caja Blanca y Caja Negra para verificar la funcionalidad correcta del sistema.

##### **4.4.1. Caja Blanca**

En las pruebas de caja blanca se utilizó el método de “Cobertura de Código” cuyo objetivo es el de elegir casos de prueba de una determinada propiedad o característica de código y dar la mayor cantidad de valores posibles.

A continuación se muestran 4 casos de prueba del sistema:

- Verificar Existencia de Usuario (Si)

- Verificar Existencia de Usuario (No)
- Ingresar Contrato
- Buscar Contrato

```

56
57
58 /**
59  * Test of validarUsuario method, of class AutenticacionServicio.
60  */
61 @Test
62 public void testValidarUsuarioNoExiste() throws Exception {
63     System.out.println("validarUsuario");
64     String nombreUsuario = "diego.flores@sinetcom.com.ec";
65     String password = "Sinetcom2014";
66     AutenticacionServicio instance = (AutenticacionServicio) ctx.lookup("java:global/classes/AutenticacionServicio");
67     Usuario expectedResult = null;
68     Usuario result = instance.validarUsuario(nombreUsuario, password);
69     assertEquals(expectedResult, result);
70 }
71
72 /**
73  * Test of validarUsuario method, of class AutenticacionServicio.
74  */
75 @Test
76 public void testValidarUsuarioSiExiste() throws Exception {
77     System.out.println("validarUsuario");
78     String nombreUsuario = "diego.flores@sinetcom.com.ec";
79     String password = "Sinetcom2013";
80     AutenticacionServicio instance = (AutenticacionServicio) ctx.lookup("java:global/classes/AutenticacionServicio");
81     UsuarioServicio instance2 = (UsuarioServicio) ctx.lookup("java:global/classes/UsuarioServicio");
82     Usuario expectedResult = instance2.obtenerUsuarioPorId(1);
83     Usuario result = instance.validarUsuario(nombreUsuario, password);
84     assertEquals(expectedResult, result);
85 }
86 }

```

Test Results

ec.com.sinetcom.servicedesksinetcom-ejb:ejb:1 x

100,00 %

Both tests passed.(29,856 s)

>>

**Figura 27: Pruebas de Caja Blanca - Verificar Usuario.**

En la Tabla 2, se observan los resultados esperados y obtenidos de estos dos casos de prueba.

**Tabla 2: Prueba de Caja Blanca - Verificar la existencia de un usuario**

Caso de Prueba	Resultado Esperado	Resultado Obtenido
<b>Existe Usuario</b>	Usuario	Usuario
<b>No Existe Usuario</b>	null	null

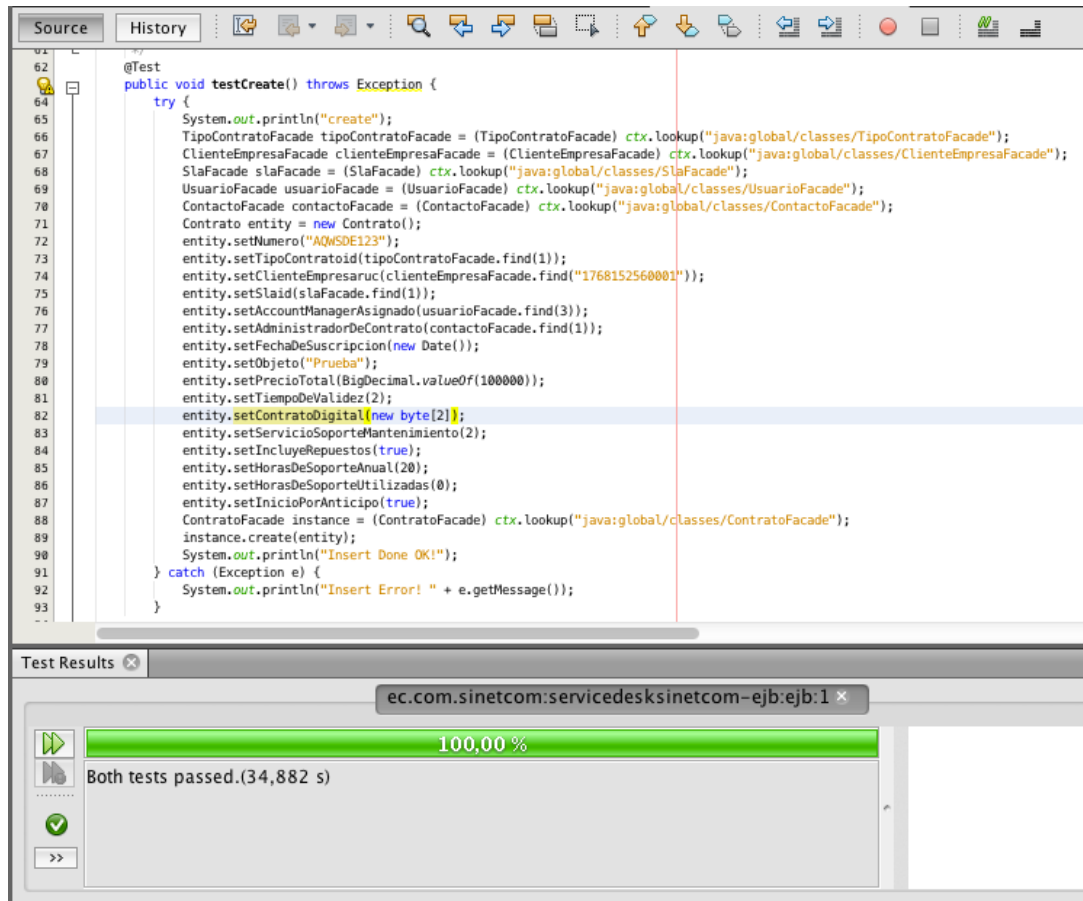


Figura 28: Pruebas de Caja Blanca - Insertar Contrato.

En la Figura 29, se observa el resultado en la base de datos MySQL Server después de ejecutar la prueba de caja blanca.

The screenshot shows a MySQL query result for the `Contrato` table. The query is `SELECT * FROM dbsinetcom.Contrato;`. The result set contains three rows of contract data, including a test entry with ID 'AQWSDE123' and a price of 100,000.00.

numero	TipoContrato_id	ClienteEmpresa...	Sla_id	accountManage...	administradorD...	fechaDeSuscrip...	objeto	precioTotal
059-2013-RE...	3	1768152560001	1	3	1	2015-01-29...	Venta de equi...	350000.00
A0001-9992	3	1768152560001	1	3	1	2015-01-14...	Contrato de la...	200000.00
AQWSDE123	1	1768152560001	1	3	1	2015-04-02...	Prueba	100000.00
		NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 29: Resultado de consulta de Contratos después de Prueba.

```

100  @Test
101  public void testFind() throws Exception {
102      System.out.println("find");
103      Object id = "059-2013-RE-CNT";
104      ContratoFacade instance = (ContratoFacade) ctx.lookup("java:global/classes/ContratoFacade");
105      Contrato expectedResult = null;
106      Contrato result = instance.find(id);
107      //assertEquals(expectedResult, result);
108      assertNotNull(result);
109  }
110
111  }

```

**Figura 30: Pruebas de Caja Blanca - Buscar Contrato.**

Finalmente después del ingreso del contrato a la base de datos se procede a realizar la búsqueda de uno de los contratos existentes en la base de datos, como se observa en la Tabla 3.

**Tabla 3: Resultados de Casos de Prueba - Contratos**

Caso de Prueba	Resultado Esperado	Resultado Obtenido
<b>Insertar Contrato</b>	Commit successfull	Commit successfull
<b>Buscar Contrato</b>	Contrato <> null	Contrato <> null

#### 4.4.2. Caja Negra

1) Para este tipo de prueba se utilizó el ingreso de datos para evaluar las salidas esperadas del sistema juntamente con la presencia del usuario final.

A continuación se observan los resultados obtenidos en resumen de las pruebas de caja negra.



**Tabla 4: Pruebas de Caja Negra**

Nombre de la clase	Nombre prueba caja negra	Descripción	Resultado
<b>Login</b>	Buscar	Verifica que los datos ingresados por el usuario correspondan al usuario y password registrados en el sistema.	✓
<b>Logout</b>	Eliminar	Elimina una sesión de usuario existente	✓
<b>Ticket</b>	Crear	Crea un nuevo ticket de soporte	✓
	Buscar Ticket Por estados, prioridades y colas	Busca un ticket por estados, prioridades y colas tanto los propietarios, asignados o todos los existentes	✓
	Asignar Ticket A Ingeniero	Asigna la atención de un ticket a un ingeniero determinado	✓
	Subir Hoja De Servicio	Carga una hoja de servicio en formato PDF en el sistema asociado a la atención de ese ticket	✓
	Poner Ticket En Pendiente	Coloca un ticket en estado pendiente	✓
	Re-abrir un Ticket De Soporte	Abre nuevamente un ticket de soporte que estaba en estado pendiente previamente	✓
	Cerrar Ticket	Cierra un ticket abierto para culminar con el soporte	✓
	Ingresar Hora De Primer Contacto	Registra la hora de primer contacto con el cliente para asuntos de cumplimiento de SLA.	✓
	Reportes	Muestra una lista de reportes útiles de tickets de soporte	✓
<b>Artículo</b>	Agregar Artículo a Ticket	Agrega un nuevo artículo al ticket de soporte, lo que permite un seguimiento más completo de la atención del ticket	✓
	Visualizar Historial de Artículos	Despliega una lista de artículos que se agregaron al ticket	✓
<b>ItemProducto</b>	Crear	Crea un nuevo ítem de inventario correspondiente a un producto existente en el sistema	✓
	Buscar	Lista todo el inventario disponible dentro de la empresa con opciones de búsqueda filtrada	✓

Continúa 

	Reporte	Despliega los reportes más importantes de la parte de inventarios tecnológicos, como lo son partes que están por acabarse en el inventario	✓
<b>Registro De Movimiento De Inventario</b>	Crear	Crea un nuevo registro de movimiento de inventario ya sea por préstamo, garantía o soporte.	✓
	Actualizar	Actualiza un registro de movimiento con las partes que ingresan como dañadas o si es devolución con las mismas que salieron en un principio	✓
	Buscar	Despliega los registros completos existentes en forma de PDF	✓
<b>Componente Electrónico Atómico</b>	Crear	Crea un nuevo componente electrónico con atributos varios.	✓
	Eliminar	Elimina un componente electrónico seleccionado	✓
<b>Parámetro de Producto</b>	Crear	Crea un nuevo parámetro u atributo de producto	✓
<b>Unidad de Medida</b>	Crear	Crea una nueva unidad de medida	✓
<b>Línea de Producto</b>	Crear	Crea una nueva línea de producto de un fabricante y categoría existente	✓
	Eliminar	Elimina una línea de producto seleccionada	✓
<b>Modelo de Línea de Producto</b>	Crear	Crea un nuevo modelo de una línea de producto existente	✓
	Eliminar	Elimina un modelo de una línea de producto existente	✓
<b>Fabricante</b>	Crear	Crea un nuevo fabricante o proveedor de hardware	✓
	Asociar Categorías	Asocia una categoría de producto a un fabricante puntual	✓
<b>Categorías</b>	Crear	Crea una categoría de producto en el sistema	✓
<b>Contrato</b>	Crear	Crea un nuevo registro de un contrato de bienes y/o servicios	✓
	Actualizar	Actualiza cierta información de un contrato, aquella que se puede modificar o cambiar en el tiempo	✓


 Continúa

	Reporte	Muestra varios reporte importantes del módulo de contratos	✓
<b>Cliente Empresa</b>	Crear	Crea una nueva empresa cliente de Sinetcom S.A.	✓
	Actualizar	Actualiza cierta información del cliente	✓
	Eliminar	Elimina un cliente del sistema	✓
<b>Contacto</b>	Crear	Crea un nuevo contacto de alguna empresa registrada en el sistema	✓
	Actualizar	Actualiza los datos de cierta información de un contactos de determinada empresa	✓
	Eliminar	Elimina el contacto de una determinada empresa	✓
	Consultar	Consulta los contactos de acuerdo a ciertos criterios de búsqueda	✓
<b>SLA</b>	Crear	Crea un nuevo SLA para asociarlo a un contrato	✓
	Eliminar	Elimina un SLA del sistema	✓
<b>Tipo Empresa</b>	Crear	Crea un tipo de empresa en el sistema	✓
	Eliminar	Elimina un tipo de empresa del sistema	✓
<b>Tipo Contrato</b>	Crear	Crea un tipo de contrato de la empresa	✓
	Eliminar	Elimina un tipo de contrato de la empresa	✓
<b>Tipo Disponibilidad SLA</b>	Crear	Crea un tipo de disponibilidad de SLA	✓
	Eliminar	Elimina un tipo de disponibilidad de SLA	✓
<b>Dirección</b>	Crear	Crea un dirección de un cliente determinado por ciudad	✓
	Eliminar	Elimina una dirección de un cliente por ciudad	✓
<b>Ciudad</b>	Crear	Crea una nueva ciudad donde la empresa brinde sus servicios	✓


 Continúa

<b>Provincia</b>	Crear	Crea una nueva provincia donde la empresa brinda sus servicios	✓
<b>Usuario</b>	Crear	Crea un nuevo usuario del sistema con un rol determinado	✓
	Actualizar	Actualiza cierta información de un usuario determinado	✓
	Cambiar Password	Cambia el password de un usuario seleccionado	✓
	Eliminar	Elimina un usuario del sistema	✓
<b>Información Sinetcom</b>	Actualizar	Actualiza la información de Sinetcom que se utiliza en el envío de notificaciones y alertas vía correo electrónico	✓

2) A continuación se muestra el caso de prueba de algunos requerimientos funcionales descritos en la ERS.

**Tabla 5: Caso de Prueba RF07**

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>Código de caso de prueba: PRU 1</b>	<b>Código de caso de Uso: (RF07)</b>
<b>Descripción de la Prueba:</b>	
La prueba consiste en crear una línea de producto.	
<b>Condiciones de Ejecución:</b>	
El usuario debe ingresar al sistema autenticado como Administrador	
<b>Entradas / Pasos de Ejecución</b>	
El usuario elige la opción "Productos" – Crear Línea de Producto	
Se selecciona el fabricante, categoría, nombre de la línea de producto, número de referencia y una descripción.	
Se presiona el botón "Crear Línea de Producto"	
Se despliega un mensaje de éxito o fallo, si es exitosa la transacción, se observa cómo se añade dentro de la tabla de líneas de producto.	
<b>Resultado Esperado</b>	
La línea de producto es insertada en la base de datos y se observan los cambios.	
<b>Evaluación de la Prueba</b>	
Los resultados esperados han sido correctos de acuerdo a los requerimientos del usuario	

**Tabla 6: Caso de Prueba RF08**

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>Código de caso de prueba: PRU 1</b>	<b>Código de caso de Uso: (RF08)</b>
<b>Descripción de la Prueba:</b>	
La prueba consiste en eliminar una línea de producto.	
<b>Condiciones de Ejecución:</b>	
El usuario debe ingresar al sistema autenticado como Administrador	
<b>Entradas / Pasos de Ejecución</b>	
El usuario elige la opción "Productos" – Crear Línea de Producto	
Ubica el registro que se desea eliminar y se presiona el botón con el icono de borrar.	
Se despliega un mensaje de éxito o fallo, si es exitosa la transacción, se observa cómo se elimina el registro de la tabla de línea de producto.	
<b>Resultado Esperado</b>	
La línea de productos es eliminada del sistema si es que no ha sido usada ya en el inventario tecnológico.	
<b>Evaluación de la Prueba</b>	
Los resultados esperados han sido correctos de acuerdo a los requerimientos del usuario	

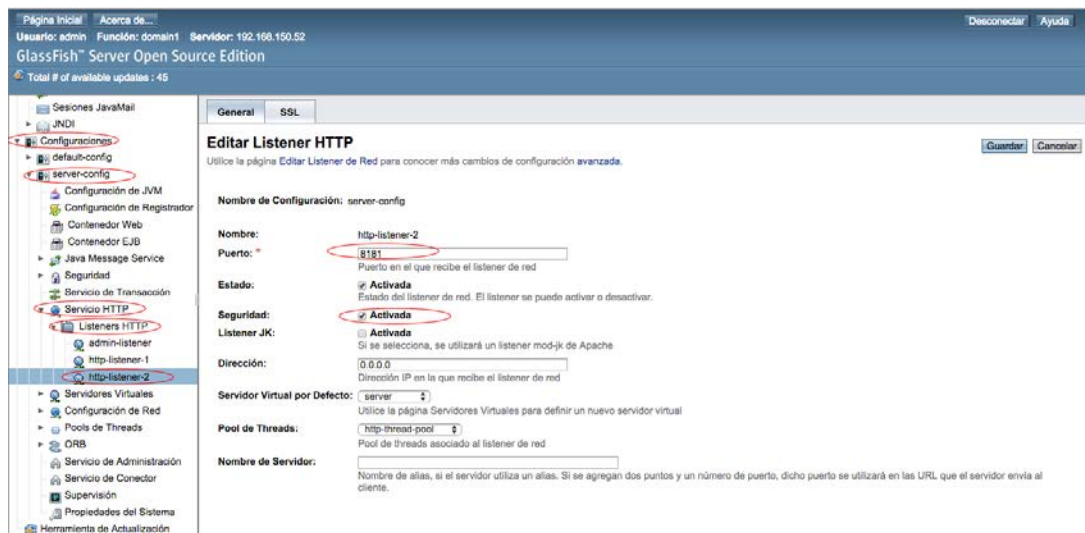
#### 4.5. Seguridad de la Aplicación

Se ha utilizado la emisión de certificados digitales auto firmados para poder encriptar el canal sobre el cuál se ejecuta el sistema, se hará el uso de SSL con el protocolo https.

El servidor de aplicaciones Glassfish hace el uso de algunos puertos por defecto para ejecutar la aplicación sobre protocolo http y https, en este caso el servidor Glassfish utiliza http sobre el puerto 8080 y https sobre el puerto 8181.

En la Figura 31, se observa la configuración por defecto del http-listener2

Para poder redireccionar todas las peticiones https que entran por el puerto 443, hacia el puerto 8181, se ha configurado los Iptables dentro del servidor UBUNTU 12.04 LTS. Esto se logra de la siguiente forma:



**Figura 31: Configuración de Listener SSL Glassfish.**

- Abrir una consola.
- Ingresar los siguientes comandos para redireccionar las peticiones.

```
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 8181
```

- Para que los cambios se guarden y sean permanentes.

```
iptables-save -c > /etc/iptables.rules
iptables-restore < /etc/iptables.rules
```

- Verificar que los cambios se hayan realizado con éxito.

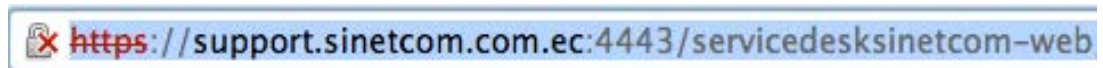
```
cat /etc/iptables.rules
```

- Adicionalmente verificar que el redireccionamiento esté funcionando.

- Si se desea que todo el tráfico que entra por el puerto 80 sea redireccionado al puerto 443, se puede configurar un virtualhost de apache2 para realizar este cometido. En las siguientes líneas se observa la forma de configurarlo en el archivo apache2.conf.

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName support.sinetcom.com.ec
    ServerAlias support.sinetcom.com.ec
    Redirect permanent / https://support.sinetcom.com.ec
</VirtualHost>
```

- Para finalizar el resultado obtenido es el siguiente:



*Figura 32: Protocolo HTTPS activo*

Finalmente, para brindar mayor seguridad al sistema se ha considerado el encriptar la contraseña de un usuario del sistema utilizando un algoritmo MD5 con un codificador base 64, como se observa en la Figura 33.

```

/**
 *
 * @author diego flores
 */
public class UtilidadDeEncriptacion {

    final String iv = "1243568790213465";
    final String secretKey = "SIN3TCOM@2014DFG";

    public UtilidadDeEncriptacion(){
        //Nada se inicializa
    }

    private static String md5(final String input) throws NoSuchAlgorithmException {
        final MessageDigest md = MessageDigest.getInstance("MD5");
        final byte[] messageDigest = md.digest(input.getBytes());
        final BigInteger number = new BigInteger(1, messageDigest);
        return String.format("%032x", number);
    }

    private Cipher initCipher(final int mode, final String initialVectorString, final String secretKey)
        throws NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException, InvalidAlgorithmParameterException

    public String encriptar(final String dataToEncrypt) {
        String encryptedData = null;
        try {
            // Initialize the cipher
            final Cipher cipher = initCipher(Cipher.ENCRYPT_MODE, this.iv, this.secretKey);
            // Encrypt the data
            final byte[] encryptedByteArray = cipher.doFinal(dataToEncrypt.getBytes());
            // Encode using Base64
            encryptedData = (new BASE64Encoder()).encode(encryptedByteArray);
        } catch (Exception e) {
            System.err.println("Problem encrypting the data");
            e.printStackTrace();
        }
        return encryptedData;
    }

    public String desencriptar(final String encryptedData) {
        String decryptedData = null;
        try {
            // Initialize the cipher
            final Cipher cipher = initCipher(Cipher.DECRYPT_MODE, this.iv, this.secretKey);
            // Decode using Base64
            final byte[] encryptedByteArray = (new BASE64Decoder()).decodeBuffer(encryptedData);
            // Decrypt the data
            final byte[] decryptedByteArray = cipher.doFinal(encryptedByteArray);
            decryptedData = new String(decryptedByteArray, "UTF8");
        } catch (Exception e) {
            System.err.println("Problem decrypting the data");
            e.printStackTrace();
        }
        return decryptedData;
    }
}

```

Figura 33: Utilidad de encriptación.



## CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

- Se cumplieron los objetivos del presente proyecto; se realizó el estudio y análisis de requerimientos de software de los procesos solicitados por la empresa auspiciante, dando como resultado un producto software de calidad.
- Con el desarrollo de la aplicación WEB se solucionó el principal problema que tenía la empresa, en el manejo de contratos, seguimiento de casos de soporte y manejo de inventarios tecnológicos.
- La metodología UWE permite realizar cambios en el diseño de forma efectiva y rápida, es una metodología ágil y encaja perfectamente en desarrollos de aplicaciones WEB. Es una metodología que involucra la retroalimentación del usuario en todo momento lo que hace que el ciclo de desarrollo de software sea más eficaz.
- UWE agiliza el proceso de pruebas, permitiendo que sean recurrentes y efectivas gracias a la retroalimentación directa del usuario, por lo que los cambios y correcciones se realizan a la brevedad.
- El uso del Framework visual PrimeFaces potenció la funcionalidad del sistema ya que según varios estudios se ha comprobado que el rendimiento para aplicaciones WEB es superior al de sus competidores como IceFaces y RichFaces, posee una mayor cantidad de componentes visuales e incorpora mayor compatibilidad con AJAX.
- Con el uso de la plataforma JEE, la aplicación WEB desarrollada heredó las características de seguridad y confiabilidad de aplicaciones empresariales de calidad.

## 5.2. Recomendaciones

- Se recomienda el uso de metodologías ágiles en un desarrollo de ambientes WEB, ya que los cambios a lo largo del ciclo de desarrollo son fácilmente adaptables sin necesidad de documentos formales, lo que permite acelerar el desarrollo y van de la mano con las pruebas de software.
- Se recomienda involucrar a la mayor cantidad de usuarios finales del sistema en las pruebas de caja negra y obtener la mayor cantidad de retroalimentación por parte de ellos, para que la aplicación sea más amigable al uso y brinde mayor valor agregado al usuario en cuestión.
- Se recomienda preparar un plan de capacitación formal y agrupar a los oyentes según roles o grupos existentes en el sistema para una mejor instrucción e inducción sobre el sistema.
- Se recomienda actualizar los diagramas de presentación de UWE al mismo tiempo que se realizan las correcciones en la interfaz gráfica del sistema, es decir todo lo que es HTML y validaciones hechas en el cliente.
- Se recomienda el uso de componentes nativos de la plataforma de desarrollo JEE al momento de implementar la característica de seguridad en una aplicación WEB desarrollada en JAVA y JSF.
- Para el desarrollo de aplicaciones web se recomienda el uso de la metodología UWE junto con la tecnología JSF, debido a que existen herramientas de modelado y generación de código que adaptan ambos conceptos.
- Se recomienda el uso de APIs como PrimeFaces para los desarrollos de aplicaciones WEB ricas en funcionalidades visuales y demandantes en el procesamiento parcial de datos.
- Se recomienda el uso de IDEs como Netbeans ya que estos permiten optimizar el proceso de desarrollo y van de la mano con los estándares internacionales en la construcción de software de calidad.

- Es muy recomendable automatizar los procesos manuales de una empresa, ya que se puede optimizar tiempo, dinero y recursos. Logrando una empresa más productiva y eficiente en sus procesos.

## Bibliografía

- Böck, H. (2011). *The Definitive Guide to Netbeans Platform 7*. Apress.
- Burns, E., Schalk, C., & Griffin, N. (2010). *JavaServer Faces 2.0: The Complete Reference*. New York: McGraw-Hill Companies .
- Caivano, R. M., & Villoria, L. N. (2009). *APLICACIONES WEB 2.0 - Google docs* (1ra ed.). Villa María: Eduvim.
- Corporation, O. (01 de 01 de 2014). *Netbeans IDE*. (Oracle, Productor)  
Recuperado el 01 de 01 de 2014, de Sitio Oficial de ORACLE:  
<http://www.oracle.com/technetwork/developer-tools/netbeans/overview/index.html>
- Departamento Ciencia de la Computación e IA. (s.f.). *El ciclo de vida de JSF*.  
Obtenido de <http://expertojava.ua.es/j2ee/publico/jsf-2012-13/sesion03-apuntes.html>
- Dept. Ciencia de la Computación e IA. (26 de 06 de 2014). *El ciclo de vida de JSF*. Recuperado el 12 de 09 de 2014, de <http://expertojava.ua.es/http://expertojava.ua.es/j2ee/publico/jsf-2012-13/sesion03-apuntes.html>
- Enríquez Toledo, A., Maldonado Ayala, J., Nakamura Ortega, Y., & Nogueron Toledo, G. (s.f.). *MySQL*. Recuperado el 10 de 10 de 2014, de <http://www.gridmorelos.uaem.mx/http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/MySQL.pdf>
- Gómez Adán, V. (14 de 08 de 2012). *Pruebas de Caja Negra*. Recuperado el 10 de 08 de 2014, de <http://www.globetesting.com/http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>
- Goncalves, A. (2010). *Beginning Java EE 6 with GlassFish 3*. Apress.
- ITIL Resources. (01 de 07 de 2008). *Service Desk Objectives in ITIL Foundation*. Recuperado el 03 de 07 de 2014, de

<http://www.itilfoundation.org/>: [http://www.itilfoundation.org/Service-Desk-Objectives-in-ITIL-Foundation\\_43.html](http://www.itilfoundation.org/Service-Desk-Objectives-in-ITIL-Foundation_43.html)

KOCH, N., & KRAUS, A. (2002). *The Expressive Power of UML-based Web Engineering*. München, Alemania.

Leahy, P. (2012). *What Is Netbeans?* Recuperado el 10 de 09 de 2014, de <http://java.about.com/>:  
<http://java.about.com/od/netbeans/p/whatisnetbeans.htm>

LMU – Ludwig-Maximilians-Universität München. (01 de 02 de 2014). *UWE Overview*. Recuperado el 02 de 07 de 2014, de <http://uwe.pst.ifi.lmu.de>: <http://uwe.pst.ifi.lmu.de>

Loor, J. M. (15 de 01 de 2014). *JSF Java Server Faces*. Recuperado el 05 de 08 de 2014, de <http://www.ndeveloper.com/>:  
[http://www.ndeveloper.com/ndeveloperDocuments/documents/nDeveloper\\_JavaServerFaces.pdf](http://www.ndeveloper.com/ndeveloperDocuments/documents/nDeveloper_JavaServerFaces.pdf)

López Orrala, J. C. (01 de 09 de 2011). *Tesis Digital*. Recuperado el 05 de 10 de 2014, de <http://dspace.ups.edu.ec/>:  
<http://dspace.ups.edu.ec/bitstream/123456789/1570/15/UPS-GT000241.pdf>

Luján Mora, S. (2002). *Programación de aplicaciones web*. Editorial Club Universitario.

Martinig, M. (01 de 01 de 2011). *StarUML - Open Source UML Tool*. Obtenido de <http://www.methodsandtools.com/>:  
<http://www.methodsandtools.com/tools/staruml.php>

Matango, R., & Arevalo, K. (2014). *Análisis, diseño e implementación de un sistema Web de gestión de la calidad para los programas de posgrados de la Universidad de las Fuerzas Armadas ESPE, mediante la utilización de la plataforma JAVA Enterprise Edition JEE6 Web aplicando la metodología UWE*. Sangolquí: ESPE.

Mora Pérez, J. J. (2012). *Capacity Planning IT*.

MySQL. (s.f.). *MySQL - The world's popular open source database*.

Obtenido de <http://www.mysql.com/products/>

Oracle Corporation. (2001). *Java EE Overview - Java EE at a Glance*.

Recuperado el 20 de 11 de 2014, de <http://www.oracle.com>:

<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

Oracle Corporation. (21 de 10 de 2014). *GlassFish - World's first Java EE 7*

*Application Server*. Recuperado el 01 de 11 de 2014, de

<https://glassfish.java.net>: <https://glassfish.java.net/es/>

Pichiyá Umul, L. A. (01 de 11 de 2009). *SISTEMA PARA EL CONTROL DE ACTAS DE LOS CURSOS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA*. Recuperado

el 21 de 09 de 2014, de <http://biblioteca.usac.edu.gt/>:

[http://biblioteca.usac.edu.gt/tesis/08/08\\_0455\\_CS.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0455_CS.pdf)

Ramos, A. (2011). *Aplicaciones Web (Novedades 2011)*. Editorial Paraninfo.

Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. (2007). *Web Engineering - Modelling and Implementing Web Applications*. (S. S. Media, Ed.)

Gustavo Rossi.

Srinivasan, K. (19 de 05 de 2007). *Introduction to Java Server Faces (JSF)*.

Recuperado el 15 de 10 de 2014, de <http://www.javabeat.net>:

<http://www.javabeat.net/introduction-to-java-server-faces/>

StarUML Official Site. (01 de 01 de 2014). *StarUML 2*. Obtenido de

<http://staruml.io/>: <http://staruml.io/>

Tamayo Barriga, D. P. (01 de 01 de 2014). *Tesis Digital*. Recuperado el 10 de 10 de 2014, de <http://dspace.esPOCH.edu.ec/>:

<http://dspace.esPOCH.edu.ec/bitstream/123456789/3328/1/18T00551.pdf>

The University of Texas At Austin. (01 de 01 de 2001). *MagicDraw Overview*.

Recuperado el 10 de 12 de 2014, de <http://www.utexas.edu/>:  
<http://www.utexas.edu/its/products/magicdraw/>

Tigris.org. (2001). *Welcome to ArgoUML*. Obtenido de

<http://argouml.tigris.org/>: <http://argouml.tigris.org/>

Venners, B. (2000). *Inside the Java 2 Virtual Machine* (2ª edición ed.).

McGraw-Hill Companies.

Vidal, G. (01 de 01 de 2007). *Ingeniería de Requerimientos*. Recuperado el

20 de 10 de 2014, de <http://users.dsic.upv.es/>:  
[http://users.dsic.upv.es/~jsilva/fin/idr/IDR\\_practica1.pdf](http://users.dsic.upv.es/~jsilva/fin/idr/IDR_practica1.pdf)

Wikipedia. (01 de 01 de 2013). *Cliente-servidor*. Recuperado el 01 de 03 de

2015, de <http://es.wikipedia.org/>: <http://es.wikipedia.org/wiki/Cliente-servidor>

Wikipedia. (10 de 08 de 2014). *Java (lenguaje de programación)*.

Recuperado el 11 de 09 de 2014, de <http://es.wikipedia.org/>:  
[http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

Wikipedia.org. (2013). *MySQL*. Recuperado el 10 de 10 de 2014, de

<http://es.wikipedia.org/>: <http://es.wikipedia.org/wiki/MySQL>

Wikipedia.org. (2014). *Java Platform, Enterprise Edition*. Recuperado el 12

de 11 de 2014, de <http://en.wikipedia.org/>:  
[http://en.wikipedia.org/wiki/Java\\_Platform,\\_Enterprise\\_Edition](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)

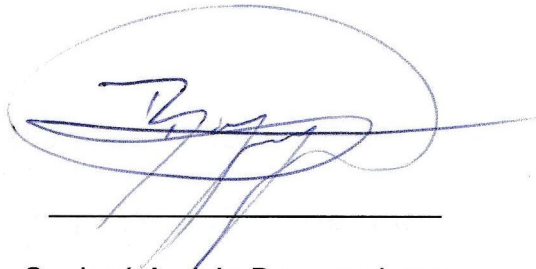
**HOJA DE LEGALIZACIÓN DE FIRMAS**

**ELABORADO POR**



Sr. Diego Antonio Flores Granja

**ELABORADO POR**



Sr. José Andrés Bozzano Larrea

**DIRECTOR DE LA CARRERA**



Sr. Ing. Mauricio Campaña

Sangolquí, Abril de 2015



# ANEXOS

- ***ANEXO A: Especificación de Requerimientos de Software***
- ***ANEXO B: Diagramas del sistema***
- ***ANEXO C: Script de la Base de Datos***
- ***ANEXO D: Manual de Usuario***